Department of Computing Science
Umeå University
SE-901 87, Umeå, Sweden

Spring 2007

PhD Thesis

Ghosts and Machines: Regularized Variational Methods for Interactive Simulations of Multibodies with Dry Frictional Contacts

Claude Lacoursière

June 2007

As authorized by the Vice-Chancellor of Umeå University, will be publicly defended on Wednesday June 13, at 10:00, in lecture hall MA 121, MIT-building, for the degree of Doctor of Philosophy.

Thesis advisor:
Professor Bo Kågström, Department of Computing Science, Umeå University
Faculty opponent:
Professor Anders Klarbring, Division of Mechanics, Department of Mechanical Engineering, Linköping University

HPC2N/VRlab and
Department of Computing Science
Umeå University
SE-901 87, Umeå, Sweden
claude@hpc2n.umu.se
Cover design by: Aron Hidman
Cover picture: Orange Organics, Aron Hidman ©2007
ISBN-13 978-91-7264-333-8
ISSN 0348-0542
UMINF 07.06
Printed by Arkitektkopia AB, V. Norrlandsgatan 10 A, 903 27 Umeå, Sweden.

Abstract

A time-discrete formulation of the variational principle of mechanics is used to provide a consistent theoretical framework for the construction and analysis of low order integration methods. These are applied to mechanical systems subject to mixed constraints and dry frictional contacts and impacts—machines. The framework includes physics motivated constraint regularization and stabilization schemes. This is done by adding potential energy and Rayleigh dissipation terms in the Lagrangian formulation used throughout. These terms explicitly depend on the value of the Lagrange multipliers enforcing constraints. Having finite energy, the multipliers are thus massless *ghost* particles. The main numerical stepping method produced with the framework is called SPOOK.

Variational integrators preserve physical invariants globally, exactly in some cases, approximately but within fixed global bounds for others. This allows to product realistic physical trajectories even with the low order methods. These are needed in the solution of nonsmooth problems such as dry frictional contacts and in addition, they are computationally inexpensive. The combination of strong stability, low order, and the global preservation of invariants allows for large integration time steps, but without loosing accuracy on the important and visible physical quantities. SPOOK is thus well-suited for interactive simulations, such as those commonly used in virtual environment applications, because it is fast, stable, and faithful to the physics.

New results include a stable discretization of highly oscillatory terms of constraint regularization; a linearly stable constraint stabilization scheme based on ghost potential and Rayleigh dissipation terms; a single-step, strictly dissipative, approximate impact model; a quasi-linear complementarity formulation of dry friction that is isotropic and solvable for any nonnegative value of friction coefficients; an analysis of a splitting scheme to solve frictional contact complementarity problems; a stable, quaternion-based rigid body stepping scheme and a stable linear approximation thereof. SPOOK includes all these elements. It is linearly implicit and linearly stable, it requires the solution of either one linear system of equations of one mixed linear complementarity problem per regular time step, and two of the same when an impact condition is detected. The changes in energy caused by constraints, impacts, and dry friction, are all shown to be strictly dissipative in comparison with the free system. Since all regularization and stabilization parameters are introduced in the physics, they map directly onto physical properties and thus allow modeling of a variety of phenomena, such as constraint compliance, for instance.

Tutorial material is included for continuous and discrete-time analytic mechanics, quaternion algebra, complementarity problems, rigid body dynamics,

Abstract

constraint kinematics, and special topics in numerical linear algebra needed in the solution of the stepping equations of SPOOK.

The qualitative and quantitative aspects of SPOOK are demonstrated by comparison with a variety of standard techniques on well known test cases which are analyzed in details. SPOOK compares favorably for all these examples. In particular, it handles ill-posed and degenerate problems seamlessly and systematically. An implementation suitable for large scale performance and accuracy testing is left for future work.

Keywords

Discrete mechanics, Variational methods, Least action principle, Multibody Systems, Numerical regularization, Constraint realization, Constraint stabilization, Differential Algebraic Equations, Nonsmooth problems, Linear Complementarity, Dry friction, Quaternion algebra, Numerical linear algebra, Physics modeling, Interactive simulation, Numerical stability, Rigid body dynamics, Dissipative systems, Contact problems, Impacts, Saddle point problems, Lagrange Multipliers, Constrained systems

Contents

Preface

Some years ago while working at a physics simulation company based in Montréal, I visited Prof. Michael M. Kostreva in Clemson where he explained an algorithm [82] for solving linear complementarity problems. This was needed to compute frictional contact forces in a real-time physics engine I was developing. The method processed a difficult problem, P_0, say, by solving a sequence easier problems, P_{ϵ_m}, differing from the original one by a small perturbation $0 < \epsilon_m \in \mathbb{R}, \epsilon_{m+1} < \epsilon_m, m = 1, 2, \ldots$. The solution produced at stage $m - 1$ was then used to warm start the solver for stage m and this saved considerable work. One would then check whether the solution at stage m differed only by a small amount from that of stage $m - 1$. This often happened after two or three stages and one could then construct the solution of P_0 without worrying about degeneracy or singularity.

My implementation of this numerical method yielded nice results but was too slow. A colleague of mine who had a knack for economy of means decided that it was not worth going through the sequence and fixed a value of $\epsilon > 0$ in his own implementation. Using the solution of P_ϵ instead of P_0 removed all degeneracy—which was well understood—but also introduced contact compliance, i.e., small amplitude, low frequency damped oscillations in the contact physics. It turned out that this was a *desirable* feature since these oscillations were stable. Indeed, we could now tune ϵ to model tires of wheeled vehicles or the ground compliance in our simulations. It was quickly realized that this perturbation corresponded to replacing hard contacts with springs of stiffness $k \approx O(1/\epsilon)$ though we could not quite understand the exact relationship yet. But the curiosity was that the oscillations were *stable*, despite the potentially high frequencies. We could often manage to use a very small value for ϵ, about 10^{-6} or less, while using large time steps for integration, approximately $h \approx 1/60$ s. By contrast, other techniques based on strong penalty forces which we had tried before were always numerically unstable for this set of parameters. Not only did this perturbation allow to approximately solve the original problem, it provided a useful physical model. This was a free lunch with beer included!

This curious result got me to think long and hard about the physical significance of this ϵ parameter and the connection to the integration methods. Would it be possible that given a physical system X_0 that is hard to solve, a perturbed *physical* system of the form $X_\epsilon = X_0 + \epsilon Y$ would be solvable quickly, efficiently, and reliably? If so, then, the first question is how to solve X_ϵ numerically and then, how to reconstruct the solution to X_0 if necessary. The second set of ques-

tions to address concern the nature of the perturbation problem Y. What form is sufficient to guarantee easy solution of the perturbed problem? What range of physical phenomena can be modeled with $X_0 + \epsilon Y$?

Alternately, it is common in numerical analysis to consider the finite precision solution $\bar{x}$ to problem P as computed by algorithm $\mathcal{A}$ to be the *exact* solution of a nearby problem P_ϵ. It is well known that the Cholesky algorithm (see Ref. [107], chapter 4 for instance) to compute the lower triangular factors G of a symmetric, positive definite matrix $A = GG^T$, produces the factors $\bar{G}$ of a nearby symmetric and positive definite matrix $A + E$ where matrix E is small, as long as A is positive definite *enough*(see [126] Chapter 10 for instance). By contrast, Gaussian elimination applied to matrix B which is not positive definite does not enjoy such stability as the Cholesky algorithm, which is also faster and simpler to code. With this in mind, given a physics problem X_0, it might be a good idea to look for a perturbation of the form $X_\epsilon = X_0 + \epsilon Y$ so that X_ϵ can be attacked with stable methods, such as the Cholesky algorithm, since it is better to make small errors on the perturbed problems than potentially large errors on the exact problem. Not only that but if the method is well chosen, the perturbed numerical solution is the exact solution of a slightly different *physical* problem. If efficiency and speed are also gained in the process, this is marvelous!

This points to a cunning paradigm. Since it is not possible to solve any problem exactly, given the finite precision of the MACHINE, one might as well introduce perturbations in the physical model itself if they allow the use of stable numerical methods which are both accurate and efficient. Thus we get better performance and better stability by simulating *more* physics, as long as we keep away from the singular cases. Instead of formulating the idealized physical problem, X_0, say, and passively expecting a numerical method to produce accurate results, an explicitly physical, small perturbation is added first making it possible to compute solutions of the nearby problem, $X_0 + \epsilon Y$, say, quickly, reliably, and with known error bounds. As will be shown in the various examples presented throughout the thesis, mathematical idealizations of physical problems often present numerical difficulties, and these are removed by relaxing the idealizations. For instance, the simple pendulum is a difficult problem when it is assumed that the rod holding it is *perfectly* rigid. But making it *only* as rigid as diamond makes the numerical difficulties go away! Since there is nothing that is much harder than diamond, solving the *exact* problem is pointless.

Prior to this, working at the same company and trying to construct numerical methods to simulate Coulomb friction [232, 34], I was struck by the difficulty of the problem. I could not get around the fact that Coulomb friction was essentially a min max principle, namely, that it maximized dissipation when bodies were sliding, and that it corresponded to the standard minimum constraint principle otherwise. Being fresh out of graduate school at McGill University where I had studied physics, I was convinced that some discretized Lagrangian could be constructed on which one could apply Hamilton's principle of least action, the theory I loved best. My reflection then went as follows. Some classical integration formulae are constructed starting from the assumption that the function being

sought, $x(t)$, satisfying the differential equation $\dot{x} = f(x)$, say, is approximated locally polynomial in t, say $\bar{x}(t) = \sum_{n=0}^{p} a_n t^n$. Using this fact, one identifies the best fit for $x(t)$ by expanding $f(x(t + hk_n))$ as a Taylor polynomial for some coefficients k_n and matching the result with the presumed form $\bar{x}(t)$ so as to minimize the error (see Ref. [113] for a more precise definition of this statement). I figured that one could perform the same trick to approximately evaluate the action integral $S = \int_0^h ds\mathcal{L}(q(s), \dot{q}(s))$, where L is the Lagrangian of the physical system and $q(t), \dot{q}(t)$ are generalized coordinates and velocities, respectively, and then, choose the coefficients a_n so that S is minimized, satisfying Hamilton's principle. I thought then that the discrete trajectory computed this way was bound to reproduce the physics as closely as the approximation allowed and since the method would consist of solving minimization problems, it would certainly be stable and efficient.

But I did not know how to formulate dissipative terms properly then and, due to commercial obligations and lack of resources, it was not possible to investigate this further. Little did I know that Moser and Veselov [208], Gillian and Wilson [100] as well as Wendlandt and Marsden [277] had developed this idea just a few years before it had crossed my mind. It was with some trepidation that I read the extensive literature by Marsden and colleagues [196, 225], with some of the papers even covering Coulomb friction applications. This was the missing piece of the puzzle.

The present work is a synthesis of these two ideas. All told, it took almost ten years split between commercial work at the company and studies at the university to build satisfactory answers to these questions. In a sense, the form of the new numerical methods I constructed are anticlimactic since it was clear from the start that a discrete variational principle coupled with physically regularized numerical methods would simply work. All that was wanting was my ability to stitch the argument together properly to build the desired techniques. On the other hand though, the results are better than I ever dreamed. Indeed, I have implemented and tested a large number of numerical methods over the years hoping that I would finally resolve one long standing issue or another, only to be profoundly disappointed in realizing that in most cases, a given method only performs well for the specific test problems discussed in the paper in which it was presented, i.e., not on those of relevance to my work. Watching the new methods pass one test after another was tremendously exciting. The voice of sweet reason had been heard and there she was, in person. I hope now that I made the presentation clear enough for others to reap the benefits of these new techniques in their work.

Acknowledgments

Producing the present thesis would not have been possible without the help of my friend Kenneth Bodin who invited me to VRlab and gave me carte blanche to pursue my work the way I saw fit.

Professor Bo Kågström was gracious in recommending my candidacy to the faculty and then taking me on as a doctorate student, knowing that I wanted to work independently. He contributed his scientific advice and editorial skills profusely and at the right time. He greatly helped shaping the text through his insistence that it should be understandable by the uninitiated and clear to all, and his coaching to make it so.

Kenneth and Bosse also helped considerably by making me feel welcome in Sweden, both at the academic and social levels, and by providing persistent enthusiastic encouragements as well as material support equally when progress was good, when it appeared that I was way out in left field, or when I got far out on a limb without a prayer and all looked grim indeed. They provided the right mix of freedom, support, and timely advice.

Professor Michael M. Kostreva of Clemson University provided coaching on complementarity problems, helpful advice and encouragements prior to my enrollment at Umeå University, and sponsored the work presented in Chapter 17.

Martin Servin had the courage to take the half baked ideas and use them for challenging applications, providing much insight. He also led the effort in publishing the papers we co-authored. Likewise, Tobias Hellman, Mats Dalgård Niklas Melin, Axel Seugling, and Martin Rölin, all performed Master's thesis work on applications, using early and incomplete forms of the theory, and their results were of great assistance in the completion of my work.

Anders Backman's consistent drive and enthusiasm for trying things out and putting things together with the software, despite "my physics" making his virtual objects race with the comets of the sky or behave generally erratically, provided essential motivation for and feedback.

I would have never thought of writing software to simulate physics in real-time if my colleagues from the McGill University physics department had not hired me at Lateral Logic back in 1996. I would have little to write about without the business development efforts continuing with MathEngine and CMLabs during my tenure, as these presented me with the right problem sets to chew on. Things went sour at the company but there were good times and great people to whom I am indebted, be they nameless here.

Both Umeå University in general and the department of Computing Science in

particular provided for a great work environment. From bibliographic resources to computer systems, including even highly functional office furniture and beautifully decorated and well tended common spaces, all that was needed was available all the time. Staff members were resourceful, serviceable and friendly at all levels of the administration, and were patient with my rudimentary Swedish as well. Computer systems worked flawlessly and support was outstanding, both at the department of Computing Science and at HPC2N. For prompt and effective assistance with a variety of details with my personal configuration details, thanks to Nikke, Torkel, Thomas, and Markus.

My colleagues at the department of Computing Science made me feel welcome in this foreign country. They helped me adapt to the customs and culture of the land and find my way around. They were patient answering the essential practical questions, generous with hospitality and friendship, stimulating in their discussions, generally supportive and encouraging, and plain fun to be around with. More specifically, Helena Lindgren helped me finding my way through the PhD defense process while she was doing the same thing, and Pedher Johansson contributed his experience from last fall on same, as well as some help with LaTeX formatting issues.

Warm thanks to Daniel Kressner for thorough last minute proof reading of the manuscript bringing, providing useful comments and a spotting a large number of typos.

Thanks to Derny Häggström for mandating Maria Johansson to teach me enough Swedish language and customs to get around, and thanks to Maria for her patience and dedication to the task. This made a world of difference.

Jonas Båtstrand helped me walk out of the musical closet in style by inviting me to play duets in front of a real audience. That renewed my interest in playing the classical guitar and in turn, this has proved a source of inspiration for my scientific work.

This research was conducted using the resources of High Performance Computing Center North (HPC2N), and supported in part by the "Objective 1 Norra Norrlands" EU grant VISTA awarded to VRlab/HPC2N at Umeå University, and and by the *Swedish Foundation for Strategic Research* (SSF) under the frame program grant SSF-A3 02:128, and by Vinnova/SSF grant VINST#247.

Notation

The main elements of the notation used throughout the thesis are described below. Though attempt was made at being systematic, there are clashes between different meanings of the same symbol in a few places. However, the text surrounding a given equation does contain brief description of each symbol used and its current meaning to avoid confusion.

Sets, groups, and algebras	
$\mathcal{A}, \mathcal{B}, \mathcal{C}, \ldots$	Point sets.
$\dim(\mathcal{A})$	The dimensionality of set $\mathcal{A}$.
$\mathbb{R}$	The real numbers.
$\mathbb{R}_+$	The non-negative reals, $x \in \mathbb{R}_+, x \geq 0$.
$\mathbb{R}_{++}$	The positive reals, $x \in \mathbb{R}_+, x > 0$.
$\bar{\mathbb{R}}$	The extended reals, $\mathbb{R} \cup \{\pm\infty\}$.
$\mathbb{C}$	The complex numbers.
$\mathbb{H}$	Hamilton's quaternion algebra.
Q	The configuration space of a mechanical systems, with dimension $n = \dim(Q)$.
TQ	The tangent bundle of Q.
$\mathbb{N}$	The natural integers $1, 2, \ldots$.
$\mathbb{Z}$	The integers $\ldots, -2, -1, 0, 1, 2, \ldots$.
$\mathbb{Z}_+$	The nonzero integers, $\mathbb{Z}_+ = \mathbb{Z} \setminus \{0\}$.
$SO(n)$	The special orthogonal group in n dimensions, i.e., the set of orthonormal $n \times n$ matrices R with $R^T R = I_n$, $\det(R) = +1$.
G	A Lie group. See also the use of $G(q)$ as the Jacobian of the function $g : \mathbb{R}^m \mapsto \mathbb{R}^n$.
$\mathfrak{g}$	A Lie algebra. Given a Lie group G, the Lie algebra of the infinitesimal generators is written $\mathfrak{g}$, and has elements $\xi_i, i = 1, 2, \ldots, n$, where $n = \dim(\mathfrak{g})$.
ξ_i	The basis of the Lie algebra $\mathfrak{g}$ of a Lie group G, for $i = 1, 2, \ldots, \dim(G)$. The Lie algebra $\mathfrak{g}$ is defined by the structure constraints, $[\xi_i, \xi_j] = \sum_k C_{ijk}\xi_k$, where $[a, b]$ is the anticommuting Lie bracket for $\mathfrak{g}$.
$\mathrm{co}(\mathcal{A})$	The convex hull of elements in set $\mathcal{A}$.
$\bar{\mathcal{A}}$	The closure of set $\mathcal{A}$.
$\emptyset$	The empty set.

$\mathcal{D}$	An integration domain.
$\partial\mathcal{A}$	The boundary of set $\mathcal{A}$.
$\mathcal{C}^n$	The class of n-fold differentiable functions.
$\mathcal{M}$	A smooth, differentiable manifold.
$GL_n(\mathbb{R})$	The general linear group over $\mathbb{R}^n$, the set of all $n \times n$ real matrices.
$N_A(x)$	The normal cone to the set A at the point $x \in A$.
$T_A(x)$	The tangent cone to the set A at the point $x \in A$.
α, β	Index sets, $\alpha, \beta \subseteq \{1, 2, \ldots, n\}$.

Vectors

x	General n-dimensional real column vector.
x^T	General n-dimensional real row vector.
x^H	The n-dimensional real row vector whose entries are the complex conjugates of those in x, i.e., $(x^H)_i = x_i^\dagger$.
x_i	The ith elements of vector x. As an exception to this rule, see the definition for x_k below.
$x^{(i)}$	The ith block component of an n-dimensional vector partitioned into m blocks of dimensions n_i with $\sum_{i=1}^m n_i = n$. If $n_i = 1$ for $i = 1, 2, \ldots, n$, this is equivalent to x_i.
x_k	For a time dependent n-dimensional vector $x(t)$, x_k is the value of the vector at discrete time k, $x_k = x(t_k) = x(hk)$, for fixed time step h. Index k is used specifically and systematically for discrete time.
$x_k^{(i)}$	For a time dependent n-dimensional vector $x : \mathbb{R} \mapsto \mathbb{R}^n$, the value of the ith block $x_k^{(i)} = x^{(i)}(t_k) = x^{(i)}(kh)$, for fixed time step $h > 0$.
x_α	A subvector of an n-dimensional vector x corresponding to the index set $\alpha \subseteq \{1, 2, \ldots, n\}$ with entries x_{i_j}, $i_j \in \alpha$ for $j = 1, 2, \ldots, \dim(\alpha)$.
$q = (q_s, q_v^T)^T$	The partitioning of the vector representation of a quaternion $q \in \mathbb{H}$ into the scalar part $q_s \in \mathbb{R}$, and a vector part $q_v \in \mathbb{R}^3$.
$q^\dagger$	For a quaternion $q \in \mathbb{H}$, $q^\dagger$ is the complex conjugate quaternion, $q^\dagger = (q_s - q_v^T)^T$.
$\mathrm{h}, \mathrm{i}, \mathrm{j}, \mathrm{k}$	The four basic elements of the quaternion algebra $\mathbb{H}$.

Special vectors

$\alpha, \beta, \zeta, \lambda, \nu, \rho, \sigma$ — n-dimensional column vector coordinates for ghost variables, i.e., the Lagrange multipliers for corresponding constraints.

ω, ω' — The angular velocity vector of a time dependent rotation matrix $R(t)$ in the inertial or body frame, respectively, $\dot{R}(t) = \widehat{\omega} R = R\widehat{\omega}'$. See definition below for $\widehat{\omega}$.

Matrices

$A = (a_{ij})$ — General $m \times n$ real matrix with elements

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}.$$

$A^T = (a_{ji})$ — The transpose of a general $m \times n$ real matrix with elements $[A^T]_{ij} = a_{ji}$.

$[A]_{ij} = a_{ij}$ — The element a_{ij} corresponding to the entry at row i and column j in matrix A.

$A_{\bullet j}$ — The jth column of $m \times n$ matrix A, i.e., the column vector with entries $a_{ij}, i = 1, 2, \dots, m$.

$A_{i\bullet}$ — The ith row of $m \times n$ matrix A, i.e., the row vector with entries $a_{ij}, j = 1, 2, \dots, n$.

$A_{\alpha\alpha}$ — Principal submatrix of the $n \times n$ matrix A for index set $\alpha \subseteq \{1, 2, \dots, n\}$, so that $A_{\alpha\alpha}$ is the $\dim(\alpha) \times \dim(\alpha)$ square matrix with elements $a_{i_k j_l}, i_k, j_l \in \alpha$, and $k, l = 1, 2, \dots, \dim(\alpha)$.

$A_{\alpha\beta}$ — Submatrix of the $n \times n$ matrix A for index sets $\alpha, \beta \subseteq \{1, 2, \dots, n\}$, so that $A_{\alpha\beta}$ is the $\dim(\alpha) \times \dim(\beta)$ matrix with elements $a_{i_k j_l}$, where $i_k \in \alpha$, for $k = 1, 2 \dots, \dim(\alpha)$, and $j_l \in \beta$, for $l = 1, 2 \dots, \dim(\beta)$.

$A = (A_{ij})$ — General block matrix A with matrix elements

$$A = \begin{bmatrix} A_{11} & A_{12} & \dots & A_{1N} \\ A_{21} & A_{22} & \dots & A_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ A_{M1} & A_{M2} & \dots & A_{MN} \end{bmatrix}$$ where the A_{ij} are sub-blocks or matrix entries.

$\det(A)$ — For a square $n \times n$ matrix A, the determinant of A, also written as $\det(A) = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix}$.

Special matrices

0 — An arbitrary size matrix with all zero elements, assumed to conform in size with respect to its context.

I_n — The $n \times n$ identity matrix.

$\widehat{a}$ — For any vector $a \in \mathbb{R}^3$, with elements $a = (a_1, a_2, a_3)^T$, the 3×3 antisymmetric matrix $\widehat{a}$ is defined as $\widehat{a} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$. For a vector $b \in \mathbb{R}^3$, the result $c = \widehat{a}b = -\widehat{b}a$ is the cross product, also written as $c = a \times b = -b \times a$.

$\mathcal{E}(q)$ — For a quaternion q, $\mathcal{E}(q)$ is the 3×4 matrix $\mathcal{E}(q) = \begin{bmatrix} -q_v & q_s I_3 + \widehat{q}_v \end{bmatrix}$.

$\mathcal{G}(q)$ — For a quaternion q, $\mathcal{G}(q)$ is the 3×4 matrix $\mathcal{G}(q) = \begin{bmatrix} -q_v & q_s I_3 - \widehat{q}_v \end{bmatrix}$.

$\mathcal{Q}(q)$ — For a quaternion $q \in \mathbb{H}$, $\mathcal{Q}(q)$ is the 4×4 matrix $\mathcal{Q}(q) = \begin{bmatrix} q_s & -q_v^T \\ q_v & q_s I_3 + \widehat{q}_v \end{bmatrix}$.

$\mathcal{P}(q)$ — For a quaternion $q \in \mathbb{H}$, $\mathcal{P}(q)$ is the 4×4 matrix $\mathcal{P}(q) = \begin{bmatrix} q_s & -q_v^T \\ q_v & q_s I_3 - \widehat{q}_v \end{bmatrix}$.

$R(q)$ — For a quaternion $q \in \mathbb{H}$, $R(q)$ is the 3×3 special orthogonal matrix, $R(q) \in SO(3)$, with

$$R(q) = \mathcal{E}(q)\mathcal{G}^T(q) = I_3 + 2q_s\widehat{q}_v + 2\widehat{q}_v\widehat{q}_v.$$

$\mathcal{I}, \mathcal{I}_0$ — The 3×3 inertia tensor of a body in the inertial and body frames, respectively, with $\mathcal{I} = R(q)\mathcal{I}_0 R^T(q)$, where $R(q)$ is the rotation matrix transforming from the body fixed frame to the inertial frame for a given orientation quaternion $q \in \mathbb{H}$.

$\mathbb{I}_0$ — The extended 4×4 inertia tensor of a body in body frame $\mathbb{I}0 = \begin{bmatrix} 0 & 0 \\ 0 & \mathcal{I}_0 \end{bmatrix}$, where the zeros are block matrices of appropriate sizes.

Functions, derivatives, and operators

$\phi : \mathcal{A} \mapsto \mathcal{B}$	A general map between sets $\mathcal{A}$ and $\mathcal{B}$. Given an element $a \in \mathcal{A}$, $\phi(a)$ is the (unique) element of $\mathcal{B}$ associated to a by ϕ.
1-1	A bijective mapping $\phi : \mathcal{A} \mapsto \mathcal{B}$.
$\phi^{-1} : B \mapsto A$	For a 1-1 map $\phi : A \mapsto B$, ϕ^{-1} is the inverse map $\phi^{-1} : B \mapsto A$ such that $\phi^{-1}(\phi(a)) = a$.
$\psi \circ \phi$	Given sets A, B, C and the maps $\phi : A \mapsto B$ and $\psi : B \mapsto C$, the map $\psi \circ \phi : A \mapsto C$ associates $c = \psi(\phi(a))$ to each element $a \in A$.
$g(x)$	A general multivariate function, $g : \mathbb{R}^n \mapsto \mathbb{R}^m$, which is the column vector with elements $g_i(x)$, $g_i : \mathbb{R}^n \mapsto \mathbb{R}$.
d	The standard derivative operator.
$\mathbf{d}$	The external derivative.
δ	The variation operator.
$\dot{x} = \frac{\mathrm{d}x}{\mathrm{d}t}$	The total time derivative of the mapping $x : \mathbb{R} \to A$, where A is any set. The form $\dot{x}$ is preferred. Chain rule always applies. See below for the special definition of θ'.
$\frac{\partial \theta(x)}{\partial x}$	The gradient of the scalar function $\theta : \mathbb{R}^n \mapsto \mathbb{R}$, evaluated at x. This is a row vector with entries $\partial\theta(x)/\partial x_i$ so that $\mathrm{d}\theta(x) = (\partial\theta(x)/\partial x)\,\mathrm{d}x$ is a scalar as well for any given differential column vector $\mathrm{d}x \in \mathbb{R}^n$, following the standard matrix multiplication rule. It is more convenient to reserve $\nabla\theta(x)$ to represent the column vector equivalent. See $\nabla\theta(x)$ below.
$\theta'(\tau) = \frac{\mathrm{d}\theta(\tau)}{\mathrm{d}\tau}$	For the scalar function of scalar argument, $\theta : \mathbb{R} \mapsto R$, $\theta'(\tau)$ is the derivative evaluated at τ. For this specific notation, the chain rule never applies. Note that $(\cdot)'$ has other meanings defined below.
$\frac{\partial \theta(x)}{\partial x^T} = \nabla\theta(x)$	For a scalar function $\theta : \mathbb{R}^n \mapsto \mathbb{R}$, the transpose of the gradient, i.e., the column vector with entries $\partial\theta(x)/\partial x_i$.
$G = \frac{\partial g}{\partial x}$	For the multivariate function, $g : \mathbb{R}^n \mapsto \mathbb{R}^m$ and the argument column vector $x \in \mathbb{R}^n$, the Jacobian matrix G has the entries $[G]_{ij} = \partial g_i/\partial x_j$.
$D_i\theta(x^{(1)}, \ldots, x^{(m)})$	For a function of m n_i-dimensional arguments, $\theta : \prod_{i=1}^{n} \times \mathbb{R}^{n_i} \mapsto \mathbb{R}$, the gradient with respect to the ith argument, namely, the row vector $D_i\theta(x^{(1)}, x^{(2)}, \ldots, x^{(m)}) = \partial\theta(x^{(1)}, x^{(2)}, \ldots, x^{(m)})/\partial x^{(i)}$.
$D_i^T\theta(x^{(1)}, \ldots, x^{(m)})$	The transpose of the row vector $D_i\theta(x^{(1)}, x^{(2)}, \ldots, x^{(m)})$, which is a column vector.

$\partial\theta(x)$	For a non-smooth function $\theta : \mathbb{R}^n \mapsto \mathbb{R}$, the generalized gradient at the point $x \in \mathbb{R}^n$.
$q \star p$	The quaternion product of two quaternions $p, q \in \mathbb{H}$.

Special functions

$\mathtt{cond}(A)$	The condition of n n matrix A, the ratio of largest to smallest singular values.
$\dot{\gamma}$	A continuous path in configuration space Q over the finite time interval $[t_0, t_1]$, so $\dot{\gamma} \in [t_0, t_1] \times TQ$.
$T(q, \dot{q})q$ or $T(q)q$	The kinetic energy function $T : TQ \mapsto \mathbb{R}$, for a given configuration space Q. The standard form is $T(q, \dot{q})q = (1/2)\dot{q}^T M(q)\dot{q}$ for a configuration dependent mass matrix. The simpler case of a configuration independent mass matrix, the standard form is written as $T(q)q = (1/2)\dot{q}^T M\dot{q}$.
$V(q)q$	Generic potential energy function $V : Q \mapsto \mathbb{R}$. Velocity dependent potentials are not considered.
$\mathcal{L}$	Generic Lagrangian of unspecified arguments.
$\mathcal{L}(q, \dot{q})q$	Generic time independent Lagrangian for variable $q : \mathbb{R} \mapsto Q$, and $(q, \dot{q}) \in TQ$.
$\mathcal{L}(q, \dot{q}, q)t$	Specifically time-dependent Lagrangian for variable $q : \mathbb{R} \mapsto Q$, $(q, \dot{q}) \in TQ$, and $t \in \mathbb{R}$.
$S[\dot{\gamma}]$	The action functional is the integral $S[\dot{\gamma}] = \int_{t_0}^{t_1} \mathtt{d}s\mathcal{L}(q(s), \dot{q}(s))$ for the path $\dot{\gamma} = \{(q(t), \dot{q}(t)) \mid t \in [t_0, t_1]\}$ for given configuration space Q, smooth trajectory function q so that $(q, \dot{q}) : [t_0, t_1] \mapsto TQ$.
$\mathfrak{R}(q, \dot{q}, t)$	A dissipation function, $\mathfrak{R} : TQ \times \mathbb{R} \mapsto \mathbb{R}$ producing the generalized force $f = -\partial\mathfrak{R}(q, \dot{q}, t)/\partial\dot{q}^T$.
$F_{\mathcal{L}}^t$	The flow map $F_{\mathcal{L}}^t : TQ \mapsto TQ$ associates the initial conditions $(q(t_0), \dot{q}(t_0)) \in TQ$ to the point $(q(t), \dot{q}(t)) \in TQ$ which is the result of integrating the Euler-Lagrange equations of the Lagrangian $\mathcal{L}(q, \dot{q})q$ from t_0 up to time t.
$\rho(A)$	The spectral radius of $n \times n$ matrix A (real or complex), $\rho(A) = \mathtt{max}_j \lvert\lambda_j\rvert = \lim_{k\to\infty} \lVert A^k \rVert^{1/k}$, where the scalars $\lambda_j \in \mathbb{C}, j = 1, 2, \ldots, n$, are the eigenvalues of matrix A.
$\theta_+(x)$	The ramp function, $\theta_+(x) = 0$ when $x < 0$, and $\theta_+(x) = x$ when $x \geq 0$.
$\partial\theta_+(x)$	The generalized derivative of the ramp function, $\partial\theta_+(x) = 0$ when $x < 0$, $\partial\theta_+(x) = 1$ when $x \geq 0$, and $\partial\theta_+(0) = [0, 1]$.
$\mathtt{sgn}(x)$	The signum function, $\mathtt{sgn}(x) = +1$ for $x > 0$, $\mathtt{sgn}(x) = -1$ for $x < 0$, and $\mathtt{sgn}(0) \in [-1, 1]$.

Special discrete functions

$u_d(q_0, q_1, s, h)$	The function $u_d : Q \times Q \times [0, h] \mapsto Q$ is the interpolation $q(s) = u_d(q_0, q_1, s, h)$ for $s \in [0, h]$, with $q_0 = u_d(q_0, q_1, 0, h)$, and $q_1 = u_d(q_0, q_1, h, h)$.
$\mathbb{L}_d(q_0, q_1, h)$	The discrete Lagrangian $\mathbb{L}_d(q_0, q_1, h) = \int_0^h \mathrm{d}s\mathcal{L}(q, \dot{q}, s)$ for a fixed time step h.
$\mathbb{L}_d(q_0, q_1, h_1)$	The discrete Lagrangian $\mathbb{L}_d(q_0, q_1, h_1) = \int_0^{h_1} \mathrm{d}s\mathcal{L}(q, \dot{q}, s)$ for a variable time step h_1.
$\mathbb{S}_d(q_0, \ldots, q_N, h)$	The discrete action $\mathbb{S}_d(q_0, \ldots, q_N, h) = \sum_{k=0}^{N-1} \mathbb{L}_d(L_k, L_{k+1}, h)$, for fixed time step $h > 0$. For variable time step, this is written as $\mathbb{S}_d(q_0, \ldots, q_N, h_1, \ldots, h_N) = \sum_{k=0}^{N-1} \mathbb{L}_d(q_k, q_{k+1}, h_{k+1})$, with time steps $h_k > 0, k = 1, 2, \ldots, N$.
$f_d^{(\pm)}(q_0, q_1, h)$	Discrete external forces not derived from a potential.
$\Phi_{\mathbb{L}_d}(q_0, q_1)$	The discrete flow map $\Phi_{\mathbb{L}_d} : Q \times Q \mapsto Q \times Q$ is the correspondence $(q_1, q_2) = \Phi_{\mathbb{L}_d}(q_0, q_1)$ under the action of the discrete Lagrangian $\mathbb{L}_d(q_0, q_1, h)$. The kth power $\Phi^k_{\mathbb{L}_d}(q_0, q_1)$ denotes self composition k-times so that $(q_k, q_{k+1}) = \Phi^k_{\mathbb{L}_d}(q_0, q_1)(q_0, q_1) = \overbrace{\Phi_{\mathbb{L}_d} \circ \Phi_{\mathbb{L}_d} \circ \cdots \circ \Phi_{\mathbb{L}_d}}^{k \text{ times}}(q_0, q_1).$

Scalars, constants

$(-)^n$	The nth power of -1.
t	Global time.
s	Natural or scaled time, or dummy time integration variable.
h	Fixed time step.
h_k	Variable time step.
k	Time step index.
t_k	Discrete global time, $t_k = kh$ for fixed step, or $t_k = \sum_{j<k} h_j$ for variable time step.

Differential Forms

ω	A general differential form.

ω^p	A differential form of degree p.
$\omega \wedge \eta$	The antisymmetric or wedge product of two differential forms ω, η.
$\phi^\star$	The pullback of a map $\phi : \mathcal{M} \mapsto \mathcal{N}$.
$\Theta_{\mathcal{L}}$	The momentum one-form associated with a given Lagrangian, $\Theta_L = \frac{\partial \mathcal{L}}{\partial \dot{q}} \mathbf{d}q$.
$\Omega_{\mathcal{L}}$	The canonical two-form of a given Lagrangian $\Omega_{\mathcal{L}} = \mathbf{d}\Theta_{\mathcal{L}}$.
$\Theta^{(+)}_{\mathbb{L}_d}, \Theta^{(-)}_{\mathbb{L}_d}$	The discrete momentum one-forms of a given discrete Lagrangian.
$\Omega_{\mathbb{L}_d}$	The discrete canonical two-form of a given Lagrangian, $\Omega_{\mathbb{L}_d} = \mathbf{d}\Theta^{(+)}_{\mathbb{L}_d} = \mathbf{d}\Theta^{(-)}_{\mathbb{L}_d}$.

Norm, modulus

$\|x\|$	The Euclidean norm of vector $x \in \mathbb{R}^n$ $x \in \mathbb{R}^n, \|x\| = \sqrt{\sum_i x_i^2}$. Norms of matrices are not used.
$\|x\|$	The modulus of a complex number, $x \in \mathbb{C}$, and the absolute value for a real number $x \in \mathbb{R}$.
$0 \leq x \perp y \geq 0$	Component-wise complementarity between vectors $x, y \in \mathbb{R}^n$ so that $x_i, y_i > 0$ and $x_i y_i = 0$ for $i = 1, 2, \ldots, n$.
$O(h^n)$	An infinitesimal remainder of order h^n as $h \to 0$, so that if $x(h) = O(h^n)$, then $\lim_{h\to 0} h^{-n} x(h)$ is a finite constant.
$O(n^m)$	The polynomial complexity order of an algorithm when applied to a problem of size n.
$\langle x(t) \rangle$	The time average of a variable $x(t)$.
$x^{(\pm)}$	Two alternatives of an expression, containing + and − signs, respectively.
$\pm a \mp b$	Two alternatives of an expression in which a + sign choice on the first term implies a − sign choice on the second, and *vice vesa*.
$\pm a \pm' b$	Four alternatives of an expression, containing two independent choices of + and − signs.
$\mathrm{tr}(A)$	The trace of an $n \times n$ matrix A, the sum of its diagonal elements, $\mathrm{tr}(A) = \sum_i a_{ii}$.
$x > y$ and $x \geq y$	Component-wise ordering between vectors $x, y \in \mathbb{R}^n$ so that $x_i > y_i$ (resp. $x_i \geq y_i$) for $i = 1, 2, \ldots, n$.
xy^T	The outer product of an m-dimensional column vector x and n-real column vectors y, which is an $m \times n$ matrix of rank 1 with entries $x_i y_j$.

$x \geqq y$	Component-wise, mixed relations, each of which holds either as an inequality, $x_i \geq y_i$, or as an equality $x_i = y_i$, for each $i = 1, 2, \ldots, n$.
$x \succeq y$ and $x \preceq y$	Lexicographic ordering of real vectors.
$F(\phi \mid \sin^2(\alpha))$	See $F(\phi \setminus \alpha)$.
$F(\phi \setminus \alpha)$	The incomplete elliptic integral of the first kind in parametric or modular form, respectively, so $$F(\phi \setminus \alpha) = F(\phi \mid \sin^2(\alpha)) = \int_0^\phi \mathrm{d}\theta \frac{1}{\sqrt{1 - (\sin\theta \sin\alpha)^2}}.$$
$x^T y$ or $x \cdot y$	The inner product of two n-dimensional real column vectors. Also known as "dot" or "scalar" product.
$\langle x, y \rangle$	The inner product of two elements $x, y \in H$ of a Hilbert space H. This reduces to $x^T y$ when H is $\mathbb{R}^n$.
$\mathrm{mid}(\alpha, \beta, \delta)$	The mid function for three scalars $\alpha, \beta, \delta \in \mathbb{R}$. The same notation is used to denote the component-wise operation on vectors of compatible dimensions.
$\mathrm{SOL}(M, q)$	The solution set of a given linear complementarity problem (LCP).
$x \perp y$	Component-wise perpendicularity between vectors $x, y \in \mathbb{R}^n$ so that $x_i y_i = 0$ for all $i = 1, 2, \ldots, n$, implying that $x^T y = 0$.
$x \times y = \widehat{x} y = -\widehat{y} x$	The vector or cross product of two 3D vectors, $x, y \in \mathbb{R}^3$. See the definition of $\widehat{a}$ also.

1 Introduction

1.1 Context

Interactive physics is the combination of a numerical simulation with multimodal input devices driving multisensory output devices with short response time. This allows to see, hear, touch and steer a numerical simulation as it executes. Interactive applications parallel the development of off-line computational physics applications in harnessing the monumental increase in computer hardware and software capabilities over the last four decades but with different goals, and usually at different scales. Whereas off-line computations solve increasingly large and complex problems at the very limit of available hardware, interactive physics applications address the needs of users, helping make sense of complex data, gaining deeper intuitive understanding of physical phenomena, or providing for hands-on experiences, be they virtual. In doing this, the requirements on hardware systems are balanced between computational, sensor, sensory, and communication resources.

The range of applications covers virtual classroom experiments [124], virtual prototyping of novel engineering designs [164], steering of large scale computations [279] to locate interesting parameter sets or identify problem cases, interactive experimentation on computational physics models, as well as the poster child in this field, namely, virtual environment (VE) training systems which have proved effective for both airline and military pilots [239], laparoscopic surgeons [257], heavy machinery operators [131] and car drivers [164]. Spectacular applications are produced in entertainment industries as well, particularly in computer graphics movies and computer games, though much artistic license with regards to physical motion is the norm here.

Numerical time integration of the equations of motion of a given physical system is the heart of interactive physics. Due to psychological factors, this task must execute at high speed and fixed rates, and is subject to different accuracy and efficiency requirements than in the off-line counterpart. The software implementing physics models and the time integrator is called a "physics engine".

An interactive physics application is typically a distributed, *soft real-time* system. Multimodal inputs from a user or a script are polled at various frequencies and communicated to the physics engine. The physics engine integrates the simulated system forward in time by a fixed amount and the new state is then used to generate output signals through a multisensory system. Soft real-time means that the flow of time in the simulation is nearly identical to the flow of wall clock time, and that response to inputs is quick and even, though not subject to hard guaranteed bounds as in the case of *hard real-time* systems used in control

engineering. Input devices can include tracking systems, microphones, switches, pedals and joysticks, pointing devices, keyboards and other such. Output devices generally include 3D graphics rendering hardware driving screens or head mounted displays, sound generation modules driving speakers, as well as haptic devices, motion platforms, and more. The soft real-time requirement stems from the fact that humans are intolerant of dynamic response lags, especially variations thereof. In particular, screen refresh rates above 20 Hz do provide the impression of continuous motion and visual comfort is achieved near 60 Hz which is the standard for game consoles in particular. Overall lags up to 100 ms or so in visual or motile response can be acceptable if constant, but smaller lags of the order of 10–20 ms make an application easier to use. The prevalent use of 3D computer graphics rendering to control all visual signals also means that 3D graphics models must be used to define the geometry of the physical objects simulated. The usual 3D graphics models are generally nonsmooth and nonconvex, being defined as arbitrary collections of triangles and thus having sharp edges and corners.

The demand for interactive physics engines is increasing in multiple domains. In surgery applications, it is necessary to model soft tissue, fluids, and tools, interactions including contacts, cutting, puncturing and stitching, as well as flows within soft enclosing membranes or with open boundaries. Flight simulators work well with lookup tables listing the drag force given attitude and speed for free flight though landing and takeoff does require models of contacting *multibodies*—collections of geometrically constrained physical bodies—as well as tires, power plant, and even full fledged hydrodynamics in some cases where one is interested in turbulence or precise drag and lift computations. In heavy machinery operator training, the need is for modeling of rigid multibody systems with contacts and friction, cables and flexible beams, power plants, drive train, tracks and tires, as well as soil, mud, dust and water. Simulation of cranes, forest machines, lifters and stackers of various kinds, as well as earth movers and other construction machines are quickly being pushed on the market and are met with high demand. Virtual prototyping requires the simulation of various types of robots and ground vehicles and their control systems, which is another application of contacting multibody systems. For all application domains mentioned above, there is a need for *validation* of the models used to avoid false training of either operators and intelligent control systems, which presents interesting challenges.

Given the variety of physical problems which could be useful in any given interactive physics application, it first seems difficult to decide where to start. What is common between the simulation of water and a complicated forest machine? Fortunately, reductionism, Occam's razor, and serendipity come to help.

Reductionism suggests starting with the simplest case of systems of particles interacting *via* central forces, following the deep footsteps of Newton. But 3D computer graphics is specialized to render rigid models so this is the place to start. By considering such graphics models as rigid bodies, connecting those with mechanical joints, introducing joint drivers to control these, and imposing nonpenetration constraints on the geometries, one is lead to the dynamics of

generic rigid multibody systems. To add more realism, one must then introduce *constraint compliance*, namely, the allowance of small amounts of constraint violations, which is a form of *regularization* easing the numerical solution process, as well as joint limits, limits on driver forces, nonpenetration contact constraints, and dry friction. The result is then a versatile and expressive physical model. The present thesis considers each of these elements in isolation first, and then conjointly and simultaneously in a unified framework.

The present thesis is a partial response to the demand for increasingly versatile interactive physics engines, providing the theoretical ground work for and the construction of original numerical methods which hopefully, suitably balance the various requirements which are described in details in Section 1.2. The *reference problem* is that of *regularized*, *stabilized*, contacting rigid multibodies subject to dry friction in *descriptor form*, terms which are described shortly. The discrete time-stepping schemes developed for this problem follow the principle of *discrete mechanical variational principle*, or *discrete mechanics* for short [196], which is described at greater length in Section 1.3 and Chapter 3.

Time integrating the equations of motion in descriptor form requires simultaneous solution of both velocity changes as well as the constraint forces—the Lagrange multipliers enforcing the constraints—which, given suitable optimized sparse matrix libraries, is generally not more expensive than the solution of the reduced problem, though much more versatile. The saddle point problems arising in constrained multibody dynamics have been well-studied and good direct codes are available for them [78, 71], as well as good sparse algorithms [74]. Indeed, direct computation of constraint forces allows to introduce constitutive laws to restrict their magnitudes, damp high oscillations, or to model physical phenomena such as dry friction.

In the analytic formulation of mechanics, the components of constraint force vectors can be interpreted as the coordinates of *ghost* particles, namely, massless, classical point particles. Ghost particles appear prominently in the chosen formulation since constitutive laws applied to the constraint forces thus appear as potential energy and dissipative forcing terms acting on the ghosts, and these then enforce constraints through linear couplings with the real physical bodies. Ghost fields are well established in quantum field theory where they play the equivalent role as constraint enforcing fields, justifying the terminology, though the term is rarely used in the classical mechanical context.

Regularization is the process of adding small perturbations to numerical problems to make them better behaved (well-posed) and easier to solve, though without altering the solution significantly. Numerical regularization allows for much savings in computational efforts as well as the robust construction of good approximate solutions to ill-posed or degenerate problems. In the present context, such perturbations are added directly to the physical models in the form of *constraint compliance* (or *constraint relaxation*), allowing small violations of geometric constraints imposed on the system.

Because regularization terms are added directly in the physical model, where they take the form of ghost potential energy, they correspond to physical phe-

nomena and thus augment the modeling expressiveness of the framework. In addition, the mechanical energy corresponding to regularization terms is fully accounted for and the dynamics of the corresponding constraint violations follow the laws of physics. Physics based regularization of constraints is similar to the use of penalty forces, a well-known bad idea. However, as shown in Chapter 3 and Chapter 4, discrete mechanics allows the construction of stable, linearly implicit, discrete stepping schemes by considering only the *average* constraint enforcing force over the time step, suppressing all high frequency components inherent of penalty forces.

Oscillations of constraint violations, arising either from the explicit introduction of regularization or from other sources of numerical errors, are further stabilized with the introduction of ghost dissipative forces. Yes, forces acting on forces. Again, discrete mechanics allows the construction of a stable discretization of such forces. When regularization is combined with stabilization, the resulting stepping schemes are linearly implicit and strongly stable at least in the case of linear constraints.

Various types of ghost dissipative forces can then be constructed to impose constraints on constraint forces, to model stiff joint actuators with finite force for instance, as well as dry friction.

Now, the descriptor form of the equations of motion of the generic multibody system just described contains the essential ingredients of a very large class of physical problems, differing only in details of the sparsity patterns of various block matrices entering the definitions. Occam's razor is the statement that "entities should not be multiplied beyond necessity". According to this principle, one should first attempt to co-opt an established entity into a new role, and given the similarity of the discretized form of various physical problems to that of the rigid multibody problem, at least some generalizations are possible. Indeed, the physical models and numerical methods described in this thesis have been reused to simulate rigid multibodies describing heavy ground based vehicles and machinery, virtual humans and robots, as well as cables, cloth, and flexible beams. An application of the techniques described in this thesis to the simulation of hydrodynamics using smoothed particles is also in development and looks promising.

Thus, the careful analysis of suitable numerical methods for contacting rigid multibodies subject to dry friction is the cornerstone of interactive physics. Whatever methods can be developed for this case can apply *mutatis mutandis* to contacting multiparticles, N-body problems subject to strong forces (treated as compliant constraints), and several instances of deformable multibodies or even fluids subject to incompressibility and boundary conditions. The one significant difference when treating new application domains is the design of an optimized equation solver which must be tailored carefully for each specific instance. Such changes however do not affect the basic stability and other qualitative properties of the integration model developed henceforth.

The following Section 1.2 describes the requirements imposed on an interactive physics simulation engine. Section 1.3 describes some of the feature of the

techniques described at length in the rest of thesis. Section 1.5 presents a broad overview of previous work in this field and Section 1.4 provides the thesis outline. Finally, Section 1.7 provides a synthesis of the ideas presented in this chapter and some general remarks.

1.2 Requirements for interactive physics simulation

The numerical integration of the differential equations of motion of classical physics is far from new. Indeed, Issac Newton himself described a simple but remarkably well-behaved numerical method in Book I, Theorem I, of *Principia* [215], published first in 1687, as observed in Hairer [112]. Of course, the numerical solution of systems of ordinary differential equations (ODE) enjoys an enormous literature and a sound theoretical backbone, brilliantly presented in the monographs of Hairer, Wanner, and Nørsett [113, 114]. There are efficient methods which can solve practically any special form of ODE to any desired level of accuracy. However, the equations of motion of constrained mechanical systems are differential algebraic equations (DAE) of index 3 (this is explained further in Section 3.14 and described extensively in [114]). Numerical solution methods for DAEs do not have the same maturity status as ODEs, as discussed later in this chapter, and are still subject to much inquiry, reformulations, and elementary experimentation [95]. Collisions, contacts and dry friction, are discontinuous or *nonsmooth* phenomena, involving step discontinuities in velocities or in the derivatives of forces at the length and time scales of interest in interactive physics. Despite the existence of some sound numerical integration methods [258, 225] for such nonsmooth problems, the literature is practically in its infancy and there is also much elementary experimenting still going on. Yes, almost every individual aspect of the numerical simulation of physical systems has been addressed previously, with varying degree of success, and with a few blind spots. One might conclude that building an interactive physics engine is the simple process of selecting the best among well-known methods and implementing them efficiently in software, devising simple special techniques for the tricky parts.

However, this is not the case since the requirements for interactive physics differ radically from those of standard numerical integration methods. Interactive physics is a soft real-time, dynamic, nonsmooth, hybrid problem, which is often ill-posed and degenerate as well. In this context also, overall stability and speed have priority over local accuracy and even, paradoxically perhaps, efficiency. It is thus necessary to build numerical integration methods fitting these specific requirements, which are now described with illustrative examples and generic observations on how to meet them. Similar arguments could presumably be provided for any specific dynamical system, be it in molecular dynamics, chemical kinetics, structural dynamics or other fields, where some overall symmetries take precedence over the order of local accuracy guaranteed by generic integration methods.

Real-time Because of the interactive context, given psychological factors, time

must flow uniformly and updates must be predictable and fast. Typical refresh rates for 3D graphics animations are 60 Hz and above and this means that 5–10 ms is the available time for integrating the simulated system by 16.667 ms. To be predictable, the integrator should only require the solution of a few linear systems per step but not require solutions of nonlinear systems to prescribed accuracy, or use adaptive time stepping methods, except perhaps for small systems.

Speed The real-time context of 3D graphics imposes an update frequency of at least 60 Hz for fresh data, leaving at most 10 ms of wall clock time for computation, irrespective of what is going on in the simulation. Haptics rendering imposes 1000 Hz update frequency, though interpolation can be used here to some degree. The numerical analysis benchmark for integrators is *efficiency*, the amount of computational work needed to achieve a given accuracy. In turn, accuracy is a quantitative measure the reliability of the data produced. Efficiency is not a good measure of performance here since a method might be efficient but still too slow, and, conversely, a method might be fast but inefficient with regards to achieving prescribed accuracy goals. The exact amount of time taken in the worst case is what matters here. Raw speed does come at the cost of decreased accuracy in general. Since the scale is strongly tilted towards low order methods, raw speed also comes at the cost of decreased overall efficiency. The final choice then depends on a balanced error analysis so as to meet both speed and validation requirements. Maximum speed is reached for fully explicit first order integration but this being hopelessly unstable, best practical speed is achieved for linearly implicit, strongly stable integrators of first and second order which require only one linear system solve per step.

Stability The total integration time interval is usually undetermined and can be of several hours, implying more than 200,000 integration steps per hour. An unstable integrator will usually force premature termination of the application, after sending objects flying to the Moon and beyond, and this is not acceptable. Stability is therefore fundamental to interactive physics. Since dealing with physical system, total energy is the natural measure of stability. Integrators should be biased so that numerical energy is monotonically non-increasing, or oscillatory within global bounds when the system is conservative. When dissipative and nonsmooth forces are added to a conservative system, the net change in numerical energy computed by the integrator should be nonpositive, at least in comparison to the same system not subjected to the additional forces.

Dynamic reconfiguration User inputs are neither known in advance nor bounded in variations in any way, and can include arbitrary changes to the physical problem under consideration at any time.

For instance, a forest machine simulation might include a command to saw a virtual log in two parts. This discrete event deletes one body and

creates two new ones while the simulation is running. Likewise, a bad driver can crash a simulated vehicle into another one at any point in time, thus dynamically adding several nonpenetration conditions to the simulated system.

The engine must thus be nearly stateless and designed to handle addition and deletion of bodies, constraints, and forces, or arbitrary changes in physical parameters, at any point in the simulation, without requiring too much computational effort. Some persistence and state information can be useful in reducing average computational costs but this information should be easy to reconstruct when it must be deleted, and not be essential for the computation of the next configuration. This means in particular that high order multistep integration methods such as the Adams family is not suitable.

Nonsmooth geometry and nonpenetration conditions To match the observations of the real world, solids must not be allowed to interpenetrate and this restriction must be imposed at the geometric level, on the visible 3D models.

But the geometric models of 3D graphics are made of arbitrary collections of triangles, edges and vertices, and these are generally not convex. Also because of the resulting sharp edges, kinks, and corners, the surface normals of these models are not continuous functions. Thus, nonsmooth analysis [66] is required to correctly determine contact regions and normals.

Impacts, contacts, friction Nonpenetration conditions lead to impacts—very large and rapid changes in the velocities. These are instantaneous at time and spatial resolutions used in interactive applications. The physics engine must correctly process localized step discontinuities in velocities.

For instance, the collision between two billiard balls is an instantaneous event at the time scales considered. During the actual impact, which might last just a few microseconds, the velocities change abruptly but the positions of the balls are not altered significantly. Likewise, objects in contact subject to dry friction are subject to discontinuous transitions between static and kinetic friction. These two cases exemplify the fundamentally nonsmooth nature of real-life physics. Indeed, dry friction and impacts are ubiquitous and fundamental to every day life. It would be impossible to walk or even sit without dry friction.

Such nonsmooth phenomena must necessarily be reproduced in interactive physics and therefore, discontinuous velocities are expected and must be processed correctly. A corollary to this is that accelerations are not well-defined everywhere, since a finite instantaneous change in velocities implies an infinite acceleration and thus, infinite forces as well. The physics engine should therefore be based on discrete time-stepping involving positions, velocities and *impulses*—time integrals of forces—but not directly on the differential equations relating accelerations and forces.

Ill-posedness and degeneracy Given the limited computational budget and the real-time requirement, errors can be relatively large leading to slightly inconsistent or ill-posed problems. For instance, numerical errors in computing contact forces between two solids can lead to penetration, and when this happens, the determination of contact normals is an ill-posed problem, and contact forces computations become degenerate. Exact arithmetic and large computational budgets could remove penetration but the contact configuration could still be ill-poised and degenerate, though consistent, as described for the case of a simple cube resting on a plane in the next paragraph.

The connection between visible geometry and physics also produces ill-posedness in itself. Consider for instance the determination of the contact geometry between a simple cubic box initially at rest on an ideal flat plane. Requiring the contacting face of the cube to be strictly coplanar with the flat plane, it becomes impossible to raise the cube on one of its edges. Enforcing nonpenetration conditions on each of the four corners of the contacting face instead, the configuration is degenerate since mathematically, it is sufficient to require that (any) three of the corners be nonpenetrating. When an external force not collinear with the plane normal acts on the cube, the physics engine either selects which three nonpenetrating conditions to activate, or computes reasonable contact forces at all four (redundant) contact points to impose the desired constraint.

The physics engine must process ill-posed and degenerate problems routinely, applying systematic numerical regularization if necessary.

Accuracy Humans do not usually go around bearing stopwatches and meter sticks to judge whether a given stone fell to the ground like any other previously witnessed stone falling to the ground. No. Yet something fundamental and easily appreciated qualitatively in the motion allows humans to perceive whether a stone fell as expected, or whether its natural motion was altered in some way, irrespective of the exact millimetric detail of the trajectory. Not granting license for deceit or disregard toward validation, these observations imply that efforts toward accuracy should be concentrated on that which makes a visible difference. Of course, exactly which quantitative validation criteria must be met to produce natural looking motion is for the psychologists to determine [220] but an educated guess that physical invariants should be preserved as much as possible, and that geometric constraints should exhibit minimal violation is a good place to start.

Modularity The simulation of a mobile crane, for instance, involves simultaneously solving for the dynamics of various subsystems, including power plant, drive train, flexible booms and hoisting cables, as well arbitrary sets of frictional contacts and collisions generated dynamically by a geometric computation module. The physics engine should be designed to allow cou-

pling between all different modules implementing each and every aspect of the physical problem considered.

No tuning As much as possible, correct simulation should not depend on a user or modeler spending hours finding suitable parameters such as masses, time step, force strength, and so on. This either implies a detailed automated local stability analysis, or a robust and strongly stable integrator capable of solving badly scaled problems. This is especially in the case where the physics engine is packaged as a software library to be used by modelers who do not know about the details of the numerical methods, and may not even know much about physics either.

Good tuning and reasonable scaling always improve performance and care should be taken in the definition of the physical models of a given application. Still, the engine should handle badly configured cases, perhaps with decreased performance and accuracy.

Better numerics are needed to meet these requirements, especially to cover more complex physical models. Several possible solutions have been proposed described in the review in Section 1.5. However, based on the findings described in the rest of the thesis, my personal view is that discrete variational physics, precisely defined in Chapter 3 and explained in broad terms in the next Section 1.3, fits the bill of requirements to a T.

1.3 Discrete variational physics

A narrow view of classical mechanics starts and ends with Newton's three laws of motion of point particles which are second order nonlinear ODEs. It is true that every single mechanical system can be first reduced to a system of point masses and then summed up appropriately to yield the correct equations of motion, as is done in textbooks for rigid or elastic bodies, fluids, gases, and so on. The equations of motion of a variety of mechanical systems have been derived long time ago and are well established starting point for computational physics. Since numerical analysts have been busy constructing integration methods for practically all types of differential equations occuring in physics, this is a good place to start.

But there is an alternative view. Notwithstanding the theological motivations at the origin of this quest, irrespective of the fact that it took more than 200 years and more than a dozen talented physicists and mathematicians to arrive at the correct formulation, the laws of mechanical motion can be *derived* from a variational principle, stating that a certain time integral defined over the physical trajectories—the action—is stationary with respect to infinitesimal variations of the trajectories respecting all imposed constraints. In short, physical motion satisfies the least action principle. The notion that Nature was parsimonious struck the imagination of a long series of physicists and mathematicians who

had to conclude in the end that the suitably defined "action" was not necessarily minimized on the physical trajectory but merely stationary.

There are several reasons to regard the variational principle as more fundamental than Newton's three laws of motion. First, the variational formulation leads to canonical equations of motions, the form of which is invariant under any continuous change of coordinates. Second, the variational principle not only reproduces the correct equations, but also gives a prescription to produce functions of the coordinates which are constant along the physical trajectory, corresponding one-to-one with symmetries of the system—the content of Emmy Noether's famous theorem. For instance, Galilean relativity states that the laws of physics are invariant under a change of origin, or a fixed global rotation. Noether's theorem correctly identifies the linear and angular momentum vectors as the corresponding constants of the motion. In Newtonian physics, conservation of linear momentum must be added explicitly as the third law of motion stating that action equals reaction. One must also define angular momentum in a slightly ad-hoc way and then prove that it is conserved when only central forces are present. For free, Noether's theorem also provides the fact that central forces only are compatible with Galilean relativity, which may not be so easy to deduce from Newton's laws. Likewise, it is in fact possible to construct the correct equations of motion of simple constrained systems using Newton's three laws but the variational principle produces the correct equations of motion for arbitrary geometric constraints, in arbitrary coordinate systems, using a systematic procedure. In addition, the variational formulation allows the clear identification of redundant degrees of freedom and a detailed prescription for how to remove them from the problem, correctly preserving the dynamics of the surviving, relevant variables, the content of reduction theory.

The variational formulation also allows the analysis of nonconservative systems and though much less can be said about these in general, their geometric and symmetry properties can be well understood within the variational framework. This is especially true of the Lagrangian formulation and the Hamiltonian form of d'Alembert's principle of vanishing virtual work, the variational principle which is used throughout this thesis.

True, the mathematical apparatus required to use the variational formulation is more complicated than the simple vector analysis needed in analyzing Newton's three laws. The prevalence of differential geometry, differential forms, Lie groups, Lie algebras, Lie derivatives, and other tools of global analysis can be daunting, especially when several results can be derived with more pedestrian mathematical techniques, though with devilish cleverness (see the first few chapters of [22] for example). But there is a unity to the variational method which is not apparent in the Newtonian formulation, and several results which would presumably be out of reach without the benefit of global analysis. Of course, working backward from the answers provided by variational methods and comparing with previously known methods can provide considerable enlightenment and understanding as to what went wrong, and perhaps ideas of how to fix things without necessarily using all the results.

Of course, at the numerical level, this can appear to be akin to armchair philosophy, or worse, the use of a sledgehammer to kill a fly. Recent developments however have proved this view wrong, very wrong.

It is a common textbook example to demonstrate that in certain cases, the evaluation of limits is a non-commuting process. A sum of limits is not necessarily equal to the limit of the sum. For the case at hand, one can first apply the variational principle to an analytic formulation, obtain the differential equations of motions and time-discretize these to compute discrete trajectories. Conversely, one can first time-discretize the trajectories, use these approximations to evaluate the discrete action, and only then apply the variational principle, imposing stationary conditions to recover discrete time-stepping equations of motion. These discrete stepping equations of motion are called variational mechanical integrators.

Now, Noether's theorem, the canonical form of the equations of motion under changes of coordinates, and variable reduction, are all direct consequences of the stationary conditions on the action. It follows that the same theorems and techniques apply *mutatis mutandis* to discrete trajectories if one requires that they make a discrete action stationary. By contrast, the direct discretization of the equations of motion may or may not preserve any of the symmetries of a given system. Though it is possible to explicitly impose symmetries when constructing a given integrator, as is done in the energy-momentum preserving and symplectic Runge-Kutta methods, or in the geometric integrators for instance, the construction of a mechanical integrator achieves this same result in a far simpler way, without any *a priori* knowledge of the symmetries of the problem. One merely has to construct approximations of *time integrals* over the trajectories as functions of the yet to be computed discrete points. The stationary conditions of discrete mechanics then prescribe a stepping scheme to compute these discrete points, guaranteeing the preservation of all motion invariants, exactly for some, approximately but within global bounds for others, as described in Chapter 3. In addition, extensions of a basic integration formula to include constraints and exotic forces is straight forward, requiring only the approximations of other time integrals.

Because the action is a time integral of a scalar function of the trajectory, the discrete action is well defined even in the case where the trajectories are discontinuous, and though some care is needed in constructing the stepping equations, this extension is also straight forward.

Assuming that invariants are the quantities corresponding to qualitative properties of physical motion on which human intuition is based, numerical methods which conserve the invariants, exactly or on average, are the ones needed for interactive physics. It is therefore possible to achieve raw speed using a low order method without giving up entirely on accuracy. In addition, the preservation of invariants often provides global stability, at least for the case where the level sets of the invariants are closed and bounded, which is usually the case for energy at least. For fixed step integrators, as shown in Chapter 3 and in the examples of Chapter 2, the energy oscillates around the correct analytic value between

bounds of second order in the time step. This is not quite as good as strict energy conservation but still leads to a closed and bounded collection of level sets and thus, global stability, though not entirely unconditional.

The Sun has been shining in the sky for a while and many of the variational integration methods have been in fact well known for some time, derived via other means, and appreciated for their qualities. For instance, the famous methods of Verlet for molecular dynamics [271] and Newmark for structural dynamics [214], among many others, have been proved to be variational [196, 112]. Marsden and West in [196] further demonstrate that symplectic Runge-Kutta methods [112] are indeed variational. But far from being limited to providing moot post rationalizations, the discrete variational formulation allows the tailoring of an integrator to handle various special types of dissipative forces, regularization, and stabilization terms. This is the feature exploited in the rest of the thesis to construct physical models of numerical regularization and their corresponding variational time-stepping equations.

Only the simplest possible discrete time-stepping equations of motion are derived for any given case, and this can appear to be a limitation of the method. But the objective was to construct stable physics based constraint regularization and stabilization schemes, and to develop a sound model of impacts and Coulomb dry friction as well. The fact that these goals are achieved with low order methods is actually a demonstration of the strength of the techniques. In addition, a stable, one step, low order method is very useful in a variety of contexts where raw speed is needed. The popularity of the Verlet method [271] in molecular dynamics (see [178] for more context about this) is a case in point. But even for these low order methods, one can achieve reasonable accuracy by solving the nonlinear problems with more precision and decreasing the time step. Brutal and inefficient perhaps, but a sound strategy nevertheless. This allows the use of these new methods in scientific application domains demanding high accuracy but otherwise sharing several of the requirements stated in Section 1.2, or the modeling flexibility and numerical robustness of physical regularization and stabilization techniques developed herein.

This said, higher order methods can indeed be constructed within the same framework. In cases where there is enough computational time available, as is true in off-line applications, and when there are not too many discontinuities, this would be more efficient. Higher order methods will be addressed in future work.

1.4 Thesis outline

The present thesis provides novel numerical methods to compute trajectories of classical mechanical systems of constrained multibodies subject to contacts, dry friction, and a variety of constraints, kinematic or otherwise. The text of what follows is divided into large chapters with theoretical contents and original results, as well as a sequence of Bagatelles, which are smaller and illustrative chapters,

either self-contained or based on the theoretical results of preceding chapters. Examples in the Bagatelles are very small and are typically restricted to one or two dimensions but this was done for clarity of exposition and to demonstrate how the new methods perform on known difficult problems in low dimension. This is not a limitation of the theoretical results though. An implementation of the new methods as a generic library designed specifically for 3D applications, as well as performance analysis on large and more difficult problems is part of future work.

Each chapter starts with a short abstracts explaining the main argument, the logical organization, and the content listing of each section. All chapters also end with an "End notes" section collecting additional references to the literature, general remarks and comments based on personal experience, and indications of future work. A glossary with definitions of most of the technical terms used in the thesis is also provided at the end, and a notational index—perhaps less thorough—is found in the front matter. With these, the book is nearly self-contained.

A first simple example of numerical discretization of a physical system is provided in Chapter 2 where the simple harmonic oscillator is discretized in four different ways and stability is analyzed.

In Chapter 3, the fundamental principles of both classical and *discrete classical* mechanics are exposed in Lagrangian form. This provides some explanations for the results demonstrated in Chapter 2 but goes much further in constructing good integration methods for mechanical systems subject to conservative and non-conservative forces, as well as a variety of constraints. Methods of discrete classical mechanics are used to construct low order fixed step integration formulae for all the systems considered. The same chapter exposes the theory behind the conservation laws of classical mechanics and demonstrates the discrete mechanical counterparts. The only new element introduced here is an analytic representation of nonholonomic constraints using a special form of Rayleigh dissipation which is not found in the literature.

The content of Chapter 4 is entirely novel in providing stable numerical integration schemes for regularized constrained systems, or in other words, stable discretization of strong penalty forces implementing constraints. This is the first physics based constraint regularization and stabilization scheme that is provably linearly stable.

Following Chapter 4, five Bagatelles illustrate several aspects of the theory. First in Chapter 5, the stability properties of explicit Runge-Kutta methods up to order four applied to the simple harmonic oscillator are investigated using standard techniques. For this specific case, the explicit Runge-Kutta methods are not performing as well as they do on arbitrary systems including dissipation. In fact, only methods of order 4 and above are stable for the harmonic oscillator, the second simplest physical system. This strengthens the case that for simulating physical systems at least, a low order physically motivated method is much better than a high order general purpose one. The next Bagatelle in Chapter 6 investigates the motion of the simple pendulum to illustrate the differences

between various strategies to integrate constrained mechanical systems. Once again, it is found that the stepping methods derived in Chapter 3 are performing very well, and are very efficient when including the modifications presented in Chapter 4.

The Bagatelle of Chapter 7 contains investigations of the motion of a planar slider crank mechanism. This simple system exhibits an isolated kinematic singularity, a point in configuration space where the constraint Jacobians do not have full row rank. Simulations are performed using a variety of methods and the point is made that using the regularized methods of Chapter 4 *removes* this singularity at no extra computational cost. Continuing with Chapter 8, the main stability result of Chapter 4 is illustrated in the simplest possible context, namely, a planar point particle constrained to move along a line but subjected to a force acting to veer it off course. This example illustrates how typical penalty methods normally introduce highly oscillatory forces. This is contrasted with the regularized method of Chapter 4 which, when stabilized in a simple fashion, damps these oscillations.

The following Bagatelle in Chapter 9, a simple spring and damper model of one-dimensional contacts is analyzed to illustrate that a suitably damped high frequency oscillator can be used to model instantaneous impacts. Though this is well known, the example serves to illustrate that impact dynamics can be decoupled from the rest of the smooth motion by separating time scales, which is then analyzed in depth in the next chapter.

In Chapter 10, a number of nonsmooth problems are investigated including collisions between nonpenetrating solids, and dry frictional contacts. After presenting an exact discrete collision model which preserves energy due to Fetecau and Marsden [87], a novel two-step approximation is constructed which is strictly dissipative but mildly so. This can be implemented at very little cost in the general framework built in Chapter 4 by requiring one additional solution of the linear problem with different data but the same matrix. Proceeding from there, a novel physical model based on non-smooth Rayleigh dissipation functions is developed to construct nonideal constraints imposing hard bounds on the velocities as well as on the constraint forces themselves. This is then used as a building block to construct a regularized model of Coulomb friction. In turn, after discretization, this model produces a novel, solvable, isotropic nonlinear complementarity problem formulation of dry friction with a genuine stiction mode, in sharp contrast to several previously published smoothed models such as [33] and [108]. In addition, though this is not pursued here, the new dry friction model is extensible to higher order stepping schemes *via* the variational framework.

Next Bagatelle in Chapter 11 expands further on the famous Painlevé paradox which illustrates a fundamental problem in the analytic formulation of Coulomb dry friction: there are configurations where the contact forces are infinite and others where they are not uniquely defined. It is shown by numerical example that the discretization strategy of Chapter 4 and the discrete dry friction model of Chapter 10, resolve the paradox satisfactorily. Multiple solutions are still

possible in zero regularization, but that less important than the existence of a true stiction mode and the absence of infinite forces.

The following four chapters cover various aspects of rigid body dynamics. Starting in Chapter 12, the rigid body is defined and its motion is resolved into translational and rotational components, as is well known. The connection between rigid body motion and the group of rigid rotations in three dimensions, $SO(3)$, is established and this serves as motivation for the next chapter. A short section providing an analysis of two-dimensional rigid bodies is also provided for reference for Bagatelle V in Chapter 7 and Bagatelle VIII in Chapter 11.

Before continuing with rigid body mechanics *per se*, Chapter 13 develops a concrete representation of the quaternion algebra $\mathbb{H}$ in terms of 4×4 real matrices and four dimensional real vectors. A large number of theorems, lemmata, corollaries and identities related to the various special matrices which appear in the representation are provided, some of which might very well be new though without great consequence, except for their usefulness in what follows. The material of this chapter is seldom found collected in one place, or with the details necessary for applications to rigid body kinematics.

Armed with the results of Chapter 13, Chapter 14 proceeds to define a number of kinematic rotational constraints between pairs of rigid bodies using quaternion algebra. This strategy removes global degeneracies encountered in vector algebra based definitions of hinge and prismatic joints for instance, in which anti and collinear conditions are indistinguishable. The quaternion definition disambiguates anti and collinear conditions. Of the constraints considered, the definition of the Hooke joint in terms of quaternion algebra is novel. For the others, the representation provided is more general than what was published previously [263].

The results of Chapter 13 and Chapter 14 are then combined in Chapter 15 to investigate the dynamics of rigid bodies, and especially the numerical discretization thereof. A Lagrangian formulation in terms of the quaternion algebra is constructed and the quaternionic equations of motion are derived. This derivation was not found in the literature. After observing that the discretized, nonlinear, quaternionic equations of motion do not have the correct form to be processed together with the equations of motion for the center of mass, a novel approximation method is constructed. This approximation captures much of the nonlinear dynamics of rigid body rotations and yet, it allows the processing of all the rigid body variables in a unified way in the framework developed in Chapter 4. This is similar in spirit to the solution of Anitescu [18], but the new approximation preserves energy better and correctly integrates the free rigid body case with very small errors. The approximation fails on gyroscopes though, because of the high rotational velocities involved. For this case, a specialized method such as [51] should be considered.

Since the approximation of the friction model of Chapter 10 requires the solution of linear complementarity problems (LCP), an introduction to this topic is presented in Chapter 16, including solution methods for these, straight forward extensions to cover mixed linear complementarity problems (MLCP), as well

as simple performance comparisons between the best known methods on random problems. This histogram-based performance analysis is novel. It greatly helps making a rational decision in choosing a LCP solver. The presentation of the LCP algorithms uses block matrices instead of the traditional pivotal algebra and basis representation common in the linear and quadratic programming literature. This allows the use of the high performance basic linear algebra subroutines (BLAS) level-3 set of operations [107] and the general matrix matrix multiplication (GEMM)-based approach [149, 148], instead of the lower performance BLAS level-2 set to which pivotal operations are restricted. In particular, the block form allows the use of sparse linear algebra packaged supplemented with factorization update and down-date routines.

In Chapter 17, a well-known form of splitting which can be applied to the dry friction model of Chapter 10 to make the problem easier to solve is analyzed in detail. An original proof is constructed to characterize the geometric and convergence properties of the sequence of the iterates. A novel form of splitting, which also simplifies the computation and makes it parallelizable, is also presented but in this case, only numerical results are provided.

Finally, in Chapter 18, the various numerical linear algebra problems encountered in building a software library implementing the stepping scheme of Chapter 4, including the nonlinear complementarity conditions arising from the impact, contact and dry friction models of Chapter 10, are discussed. A number of data layout issues are first discussed and detailed solutions, known to work in practice, are provided. This is followed with a presentation of a known sparse factorization algorithm for saddle point matrices that is then used as a preconditioner for a modification of a conjugate gradient algorithm which can process the special types of saddle point matrices appearing in the regularized stepping equations of Chapter 4. Gauss-Seidel type iterative methods are then analyzed with respect to their data movement requirements. Some "poor man" computational techniques are also presented to allow the implementation of factorization updates and down-dates discussed in Chapter 16, using third party matrix factorization libraries, to provide for the case where such operations are not provided natively.

A synthesis is then provided in Chapter 19 to collect the various sub-problems analyzed through the thesis, to provide an overall view of a physics engine, and collect closing remarks as well as description of future work.

1.5 Survey of related work

Interactive physics draws on several independent branches of investigations. The first strictly concerns the coupling of interactive 3D graphics with some sort of physics simulation, best illustrated with the development of the commercial flight simulators, in which all aspects of interactivity come first and foremost. Then comes a development in computational physics and engineering addressing the dynamics of constrained multibodies, considering first the simple constraints

found in robot design, and eventually the more complicated dry frictional contact constraints. Also of interest is the development in numerical analysis of methods for handling DAEs, which include methods applicable to robots and machines, and similarly in molecular dynamics, a set of special, fast methods with various symmetry properties suitable for multibody systems. Each of these branches contributes something of significance to interactive physics though none provide a satisfactory answer meeting all the requirements described in Section 1.2. The review provided below is not exhaustive, mostly concentrating on articles and books related to contacting multibodies subject to dry friction, but it serves to illustrate the range of techniques which have been developed so far.

Indisputably, the early developments of flight simulator technology not only set the course for interactive 3D computer graphics [92] but that of physics driven VEs. The simulation of a single rigid body subject to drag and lift forces, computed by interpolating from experimental tables given the difficulty of a full fledged hydrodynamic flow simulation of a complex body moving at high speed, might not appear terribly difficult or exciting today. However, the realization of flight simulators established the possibility of creating immersive virtual worlds using computer systems. In creating the link between computational physics and computer graphics, flight simulator development must be credited with the invention of interactive physics, as well as providing inspiration for all that came after.

Following the pioneering efforts in flight simulation, 3D computer graphics became a field of research in its own right but it did not take so long for the proceedings of the yearly meetings of the Association for Computing Machinery (ACM) special interest group on computer graphics and interactive techniques (SIGGRAPH) to start including computational physics papers describing techniques for adding realism via physics simulations. The trend generalized in all the 3D computer graphics journals.

Perhaps the first few instances including both physics simulation and graphics are credited to Terzopoulos [264] who started right away with deformable bodies, with impressive effect but with forgettable techniques, and then Barzel and Barr [45] who introduced the important concept of kinematic constraints for controlling virtual objects, and as a necessary modeling element of a physics engine. The treatment of constraints was through index 1 reduction and Baumgarte stabilization [47], a technique described in Chapter 4 which, unfortunately, is both in widespread use and unstable or, at least, notoriously difficult to tune properly. Barzel proceeded later to publish a "howto" book to add simple physics models in simulations [44], and then introduced the notion of "plausible animations" [46], arguing that many details of a given physical phenomenon are beyond observation or fundamentally noisy, and that instead of unachievable goals towards accuracy, a balanced view should be adopted, even including some provisions for artistic license. The basis of the Barzel's argument [46] is that variability is inherent in nature and he reflected carefully about how and where to include this variability explicitly to save on useless computations and thus produce simulations which were closer to common experience without requiring so much accuracy. In

simulating a pool (billiard) break for instance, explicitly adding randomness can produce realistic results, whilst an exact contact formulation might not. This argument is not followed here as, strictly speaking, it is orthogonal to the construction of good time stepping schemes. This said, Barzel's careful and sound argument is unfortunately often grossly misquoted to grant license for blatant disregard for accuracy, without any checks on whether the simulated motion matches observations, at least in some average sense.

The case of dry friction between contacting rigid bodies in two dimensions was pioneered by Lötstedt [189, 190, 191] during this PhD thesis work concerned with the collapse of buildings in two dimensions. His techniques are discussed further below. This work found its way in the SIGGRAPH proceedings in a series of papers by Baraff who built on Lötstedt ideas and presented techniques for two-dimensional rigid bodies subject to dry friction based on the Lemke algorithm [179] in [31, 33, 34] and on the Cottle-Dantzig algorithm in [35], also providing some analysis of smooth contacting geometries in [32], and addressing flexible bodies with frictional contacts in [38]. Finally, he generalized this to three dimensions in [36] and suggested a sparse factorization technique to speedup some of the computations in [37]. The integration method used throughout this work is the common Runge-Kutta method of fourth order, RK4a [113], which requires four stages, i.e., four derivative evaluation per step. Constraints are treated by reducing the problem to an index 1 DAE and stabilized using Baumgarte's method [47]. The impact of these papers on the graphics and computer game development communities was tremendous. I started my own work by implementing these methods, and so was everyone else I was meeting who was working on these problems. However, the shortcomings of the dry friction model and its suggested solution scheme [35], as well as the integration and constraint stabilization scheme made these techniques frustrating to use. The resolution of these problems came later from the engineering and mathematics literature as described further below.

In parallel to this, Mirtich chose to treat all collisions and contacts using strictly pairwise interactions, computed via a simple binary impulse model [204, 205, 203]. This involves conservative prediction of time impact between each pairs, sorting these in a hierarchical table, integrating the system to the next predicted impact, resolving it, updating the estimates, and continuing. This is quite easy to implement but suffers from non-uniform time flow and Zeno points where the time step cannot be advanced significantly because of dense clusters of nearly simultaneous impacts. This is especially true when approaching resting contact and in this case, the method amounts essentially to a Gauss-Seidel process since time no longer moves forward, until some thresholds are met. To some extent, this problem can be addressed via parallelism as in [203], or using clever sorting strategies. Such techniques become quickly intractable when many simultaneously contacting bodies are considered. At that point, one must relax the conservative impact estimates and allow some interpenetration to recover performance. A method allowing multiple simultaneous resting contacts is generally more efficient in that regime.

More papers appeared in the graphics literature after Baraff and Mirtich, chiefly concerned with large unstructured piles of rigid bodies which became a benchmark. First with Milenkovic [202] who used a standard quadratic program solver to approximate dry friction in two steps, then with Guendelman, Bridson, and Fedkiw [109], and similarly in the work of Kaufman, Edmunds and Pai [155], an iterative process based on pairwise interactions—akin to the Mirtich strategy described above—was used to resolve pairwise collisions, which was then extended to process more general constraints as well in [275]. It is impressive to see that an iterative solver can quickly approximate the contact forces of thousands of stacked bodies but the size of the errors and the slow convergence of such methods make them of little use when validation is required. Iterative techniques are not fundamentally bad and even Gauss-Seidel is routinely used in high accuracy scientific applications. However, I strongly believe that the choice of using an iterative method should be taken *after* a sound overall time-stepping technique has been selected, and the problem to solve is clearly identified, but not be built into a procedural description of the integrator, leaving no room for mathematical analysis. Such procedural methods are usually presented with the claim that they are "plausible" but one is left to wonder whether this implies the overall credibility and variability described by Barzel [46] when he introduced the term, or intentional deceit, as implied by the dictionary definition of "plausibility", motivated by an aversion for quantitative evaluation of statistics, stability, and accuracy. Because of the fundamentally procedural nature of the recent rigid multibody papers presented in the graphics literature, impervious to any form of mathematical analysis, and given the current trend of not providing any form of quantitative information about the size of errors or the time evolution of energy, such methods are not considered any further.

The engineering literature contains several threads addressing the simulation of robots and machines. On the robotic front, much work was done to construct methods for simulating tree structured mechanisms with ideal joints in linear time, which culminated in the work of Featherstone [86]. This type of topological or recursive technique, as it is called in robotics because it is based on traversing the connectivity graph of the multibody, allows the reduction of the dynamics of a multibody system to ODEs, but only in the case where there are no closed kinematic loops. This problem can be addressed to some extent as in [127, 128], and more recently, using a technique called natural orthogonal complements [12, 13, 243, 233]. Topological methods are just one form of coordinate reduction technique, others being based on analytic reduction, or numerical state space reductions based on orthogonal factorization (QR) of constraint Jacobians.

All topological methods of robotics eliminate the constrained degrees of freedom and though they might be more efficient in some cases, the descriptor method which I favor allows the introduction of joint compliance which is useful in modeling, allowing the simulation of elastic bodies, for instance. It is part of some future work however to consider hybrid formulations based on the recursive formulations within the variational framework, in order to improve performance and for the ability to provide compliance free joints, since this can be useful in

some cases.

A large and broad research effort covering all aspects of computer simulations of multibody systems was sponsored by the German Research council (DFG) from 1987 to 1993 and the papers describing several results are collected in [246], and generated such off springs as the Mobile library of Kecskeméthy [156] which is a recursive model allowing loop closure, and the MBSSIM library of von Schwerin [272] which is a hybrid between a descriptor form and a recursive method using integrators from the Adam Bashforth family [113], both software packages reducing the DAEs of motion to ODEs using differentiation. Interesting aspects of the DFG multibody dynamics research effort are the overall software environments which were built to produce complete work flow pipelines from system design to interactive simulations, described in [221, 130]. It is part of future work to adapt such design ideas to produce a complete work flow around the numerical methods developed herein.

Perhaps an illustration of the difficulty of multibody simulation is the fact that the collection of papers in [246] contains several incompatible numerical models, each with its own set of restrictions. These divisions remain to this day and no one formulation of the problem or numerical integration strategy dominates.

Also in the engineering literature, but with a closer relation to interactive physics, is the work of Haug and his group which led to both the development of the now apparently defunct DADS software package for Design Analysis for Dynamic Systems, an early version of which is described to some extent in [118], and the development of the Iowa Driving Simulator [93], eventually superseded by the National Advanced Driving Simulator [119]. Haug and his group investigated both descriptor and reduced coordinate formulations, with a particular emphasis on hybrids and state space projection methods. The early development is recorded in [282, 283, 230] and more recently in [251], with some efforts to solve dry friction problems with explicit addition and deletion of constraints [123, 280, 281]. The development of real-time techniques is documented in [267, 268, 65], and the development of stiff integration methods based on local state space reduction in [122, 252, 212, 121, 120, 211, 213, 244, 274]. There is also much work from this group on covering several other aspects of modeling and validation which is outside of the scope of the present thesis. Any form of state-space reduction technique is of course independent of the integrator and it is possible they will be used in the future in trying to optimize performance when addressing larger problems. Nevertheless, state-space reduction and coordinate partitioning is generally slower. There are cases when a global reduction or or a complete set of realizable reductions can be precomputed, leaving only small problems to be solved at run-time. This hardly applies to frictional contact problems, however. Such strategies are not covered further.

In numerical analysis, the problem of integrating the DAEs of motion of constrained mechanical systems attracted some attention. Brenan, Campbell and Petzold [54] produced the DASSL library based on backward difference formula (BDF) methods, which does work well on index 1 and some index 2 problems but not on index 3 problems. The limitations of DASSL can be addressed to

some extent by index reduction techniques of Gear Gupta and Leimkuhler [96]. For certain forms of linear time dependent coefficient DAEs, BDF methods can be applied successfully for arbitrarily high index [43], and one can also add the first k time derivatives of constraints to the system, for a DAE of index k, and solving everything as a least squares problem as in [41, 94]. The least squares strategy of Barrlund [41] can help preserving physical invariants by adding them as constraints, though this introduces dense Jacobian blocks since the invariants typically depend on all the coordinates of the system. The GELDA package by Kunkel, Mehrmann, Rath, and Weickert [165], improves vastly on DASSL, using robust linear algebra methods to compute the local invariants, again, at the cost of performing singular value decomposition (SVD) or QR at each (sub) step. A number of strategies for solving the DAEs of motion using least squares strategies and projection methods are discussed by Führer and Leimkuhler [94]. In addition, Arévalo Führer and Söderlind [20] have developed multistep methods [113] for the specific case of the DAEs of motion of mechanical system with some success.

However, in general, the BDF methods are strongly dissipative and destroy physical symmetries. Given the typical computational cost of these methods, especially when invariant subspaces have to be computed, they are not appealing for interactive physics.

Hairer and Brasey [53] developed Runge-Kutta methods for mechanical systems subject to holonomic constraints which they called PHEM5, and Hairer observed in [114] that the RADAU5 code can process DAEs of index 3, though at the cost of having to use SVD extensively, and augmenting the formulation to include positions, velocities, *and* accelerations in the same nonlinear system along with constraints and their derivatives. Also related is the extrapolation method of Lubich, Engstler and Nowak [192], which led to the MEXAX code, which would be interesting to use as benchmark, as it offers very high local accuracy and good stability. This said, the PHEM5 method is probably the most interesting to consider as a starting point for future work on higher order integration, at least for smooth systems not subject to friction, because of the connection between higher order variational integrators and symplectic Runge-Kutta methods as described by Marsden and West in [196]. It would be interesting to see the difference between PHEM5 and a variational formulation of the same order.

Ascher and his colleagues investigated index reduction methods in which a DAE is replaced formally by an ODE with an invariant [24], as did Barrlund [42] as well. The problem is then to stabilize the invariant for all times and this can be done reliably as shown in Chin's PhD Thesis [61]. A subsequent original idea from this group is the sequential regularization of Ascher and Lin [25, 26, 27], which was also applied by Lin [185] to the problem of incompressible fluid flow, in which constraints are replaced by stiff penalties which are decoupled from the integrator. This works by repeatedly reintegrating the system subject to a penalty force evaluated on the previous trajectory estimate, making the strong force independent of the currently integrated variables. A higher order integration method is required to integrate the resulting ODE though, and the

regularization process is not based on physics. One could presumably integrate similar ideas in the variational formulation to construct a multirate integrator though this avenue is not pursued for now.

However, in all high order numerical methods for solving DAEs of index 3, it is far from clear how to introduce impacts, contacts and dry friction. Since a high order integrator must be restarted at each discontinuity, processing a problem with dry friction would often result in no better than first order accuracy. Of course, the first order BDF method is nothing else than first order implicit Euler and is strongly stable but also strongly dissipative, which is bad for physics problems. It is thus not clear that any of the other strategies just mentioned would work satisfactorily on nonsmooth problems. This is the main reason why they have not been considered. Ideally, it should be possible to achieve the precision and stability of PHEM5 for certain problems when there are no impacts or other discontinuities due to dry friction and a need for accuracy and this will be addressed in future work.

As mentioned previously, Lötstedt [191, 190, 189] performed simulations of collapsing 2D rigid body structures subject to Coulomb friction and it appears that this is the start of the LCP formulation of the Coulomb problem, which was continued by Pfeiffer and Glocker [232]. However, this formulation involves a second order DAE and therefore, as explained in Chapter 11, there are configurations which are not solvable, a fact known as the Painlevé paradox [223]. Several issues related to the problem formulation where investigated by Pang, Trinkle, Sudarsky, and Lo [229, 228, 266], for rigid body contacts, and existence conditions were clarified also [227]. In engineering, problem formulation and solvability was addressed for deformable bodies in quasistatic contacts in several articles by Klarbring, Andersson, Pang, Christensen, and Strömberg [8, 64, 63, 62, 160, 161].

Much of this work is collected and reviewed in an imposing monograph with over 1000 references by Brogliato [56], which also includes the first few *solvable* dry friction formulations discussed shortly. More recently, Tzitzouris [269] applied a variant of the friction formulation of Trinkle, Pang, Sudarsky, and Loalcitetrinkle:1997:odm, and a modification of Newton's method to solve the resulting complementarity problems due to Pang [226] to construct a high order integration method for contacting rigid bodies. However, this requires locating each impact discontinuity, including the time location of any transition between static and kinetic friction, and restarting the integrator at each such event. This is not a strategy that can process large collections of contacting bodies efficiently.

A recent development of friction modeling is due to Glowinski and his colleagues [101, 102, 255, 103], which bears some interest and provides a number of simple examples suitable for benchmarking. However, the formulation is not a standard complementarity problem and the question of whether it is solvable in general is open. In addition, the graphs included in these papers illustrate the presence of undesirable high frequency noise, and the time steps needed to remove this are much smaller than required in interactive physics.

As for existence and uniqueness of solutions of dry friction problems, Pang and Stewart did obtain an existence proof given a number of general assump-

tions [227], providing some indications of what fails in the case of a rigid rod contacting a plane—the example that leads to the Painlevé paradox. Klarbring and Andersson [7, 8] provide results for the static and quasistatic deformable body case but the conclusions are limited to small friction coefficients. In connection to this work, Hassani [117] provides sufficient conditions for non-uniqueness, and Włodarczyk [278] characterized the multiple solutions, which always appear in odd numbers.

However, most formulations of dry friction mentioned in the previous paragraphs are based directly on Newton's second law and involve the acceleration of physical bodies. As observed previously, the accelerations are not well defined at discontinuous points, which include transitions between static and kinetic friction. According to the existence theorem of Pang and Stewart [227], this implies that the friction problem may not always be solvable in the acceleration form as they all exhibit the Painlevé paradox. But there are solvable formulations, and the first few of these are also included in Brogliato's monograph [56], and these are in fact the methods of interest which we now turn to.

Stewart observed that Coulomb friction was a non-smooth phenomena, even when restricted to one dimension, and that one should regard the equations of motion as differential inclusions (DI). He then constructed high accuracy numerical methods for these [258]. Later, in a series of papers by Stewart, Trinkle, Anitescu and Potra [261, 259, 17, 19], a solvable LCP model of Coulomb friction was developed part of a first order time-stepping scheme. Resolution of velocity discontinuities is achieved by formally integrating the equations of motion over one time step, yielding a finite difference scheme, which can either be implicit or explicit, bearing similarities to the variational discretization described herein. The resulting stepping scheme is essentially the first consistent mathematical representation of Coulomb friction which satisfactorily resolves the Painlevé paradox [259]. Adding to this, Anitescu looked at stabilization of the DAEs of motion [15], addition of stiff forces using linearly implicit stepping, and stabilization of the gyroscopic terms [18]. This set of papers has been influential in the graphics literature as well but it has turned out that solutions of the friction model take too long to compute when considering 3D applications. More details of this model and its limitations are provided in Chapter 10. Though the literature just mentioned was a great source of inspiration for the present work, limitations of the allowed time integration methods, which are strongly connected to the solvability of the friction model, became frustrating and this ultimately led to choosing the variational approach of the present effort. The final form of the Coulomb friction model developed in Chapter 10 is rooted more in the variational formulation though it reduces to the solvable model of Stewart, Trinkle, Anitescu, and Potra mentioned above in the appropriate limit, meaning that it is both a derivation from fundamental physical principles, and an augmentation of previously known solvable models.

In collaboration with Kane and Marsden, whose work in connection with variational methods is described further below, Pandolfi, Kane, Marsten, and Ortiz [225], present a variational method for frictional contact problems of de-

formable bodies which is consistent but relies on solving nonsmooth, non-convex, nonlinear programs. There is no indication how fast this is or whether solutions always exist, and it appears that the framework cannot be applied to rigid bodies because the rotational physics cannot be formulated as a minimization problem, as shown in Chapter 15. This approach has nevertheless been inspiring for the development of the regularized Coulomb friction model presented in Section 10.11, especially with regards to the definition of nonsmooth Rayleigh dissipation functions.

In all previously mentioned contact and dry friction formulations, one must solve special types of complementarity problems in which the coupling between normal contact and tangential friction forces is asymmetric. This means in particular that the resulting LCP or nonlinear complementarity problem (NCP) cannot be reduced to a quadratic programming (QP) problem. Several numerical solution methods for the LCPs of friction have been proposed to address this, based either on sequential QPs, solving alternatively for normal and tangential forces as in [191, 273, 234, 235, 236, 237, 238, 143, 76, 77, 2, 143, 142] or directly using smoothed Newton methods [62, 183, 226], succeeding to various degree. However, the proposed solution methods do not always converge as shown in Chapter 17 and the construction of a robust solver for dry frictional contact problems, even for the provably solvable ones, is in fact an open issue which will be addressed in future work.

The problem of finding integrators which preserve geometric properties of given physical systems has stimulated much work in the field molecular dynamics where the motion of molecules is simulated for long periods of time [178]. The low order method of Verlet [271] has been a long time favorite due to sheer speed and good physical properties. Symplectic and energy-momentum preserving Runge-Kutta methods were actively developed in the 1990s [112], and these are definitely good choices for conservative systems as reviewed briefly in [247]. They are also connected to the variational formulation in an interesting way [196]. Another similar effort is the development of geometric integrators, and, particularly, splitting schemes for these [199]. Here, a complicated dynamics problem is decomposed into simpler components which are then integrated individually using exact propagation formula within an interleaved scheme. There is no indication in the literature whether such splitting techniques can be extended to multibody problems or dissipative systems, let alone dry friction.

Finally, we come to review some of the literature on discrete mechanics, the theoretical basis for the present thesis. The idea to discretize trajectories before applying the principle of least action in Lagrangian mechanics appeared almost simultaneously by Moser and Veselov [208] who first studied the motion of a free rigid body with the intent to construct a discrete equivalent of an integrable system—a dynamical system which had as many invariants of motion as variables—and by Gillilan and Wilson [100] in the molecular dynamics literature, concentrating on computation of closed orbits. Whereas Gillilan and Wilson simply recovered the known Verlet algorithm for point particles [271], the stepping method constructed by Moser and Veselov for the free rigid body was new. These

ideas were pursued in depth by Marsden and his colleagues and graduate students, first with Master's theses extending the basic idea [194, 277, 276], then constructing an extensive theory of discrete mechanics [196]. Cortés, de Diego, and Martínez [68, 67], generalized to the case of nonholonomic constraints, whilst Kane and colleagues, in collaboration with Marsden, covered nonsmooth contacts [152], as well as Coulomb friction for contacting elastic bodies [225], and clarified the well-known good properties of certain Newmark integrators [151], revealing these to be variation methods in fact. Application to nonsmooth problems including impacts was done by Fetecau [87]. There are also developments to continuum systems [180, 181], as well as application of classical reduction theory [51]. But these and other current developments of discrete variational integrators fall outside the scope of the present thesis.

One aspect that is missing in the aforementioned literature on discrete mechanics is the development of concrete numerical algorithms which can reliably and efficiently process the stepping equations. This is the topic specifically addressed in the present thesis with the prominent treatment of constraint regularization and stabilization, as well as several approximation techniques.

1.6 Previous contributions

Most of the new results presented in this monograph were not published previously. The monograph format allows for the inclusion of tutorial material and provides enough space for detailed explanations. This was deemed necessary because variational stepping methods are radically different from existing techniques and relatively new.

Nevertheless, some aspect of this work was presented at conferences and published in peer-reviewed proceedings. Quantitative analysis of the performance of LCP solver on frictional contact problems using various models and splitting strategies [167] was presented at the Swedish chapter of SIGGRAPH, SIGRAD, at the annual conference on November 20–21 2003, Umeå, Sweden.

A time stepping technique for multibody systems with constraint regularization and stabilization, not based on the variational principle [168], was then presented at the International Workshop on PDE Methods in Computer Graphics March 31–April 1, 2005 University of Copenhagen, Denmark, along with an approximation scheme for stabilizing the gyroscopic forces of rigid bodies [169]. Both of these papers are to appear in a special issue of Electronic Letters on Computer Vision and Image Analysis. However, the publication of the special issue has been delayed twice which is why both papers [168, 169] appear as departmental scientific reports at the time of writing.

A parallel splitting method for solving frictional contact problems on multicore CPUs[170] was presented at the Workshop on State-of-the-art in Scientific and Parallel Computing, Umeå, Sweden, June 18-21, 2006. This article has been reviewed and accepted for publication in the Springer Verlag series "Lecture Notes in Computer Science" (LNCS). The LNCS volume is to be printed in

2007.

Mats Dalgård did Master's thesis work [70] applying the regularized stepper [168] to a rigid multibody model for the real-time simulation of cloth and defended his work successfully in September 2005. The model of cloth for this consists of spherical rigid bodies connected by regularized fixed distance constraints. This allows for arbitrary interconnection topology. The constraint forces can be computed quickly using the preconditioned conjugate gradient method presented in Section 18.6, using the sparse factorization of Section 18.4 as preconditioner. Dalgård performed the software implementation based strictly on my theoretical work.

Dalgård's work built on Tobias Hellman's Master's thesis project which used a point particle model for cloth with the regularized stepper. Regrettably, Hellman did not defend his thesis for personal reasons, even though the report was complete and the results were good. Both Dalgård's and Hellman's simulations, based on my work, delivered far superior performance, better stability and better modeling that the commonly used technique of Baraff and Witkin [39], based on a linearized implicit Euler integration, which is good for damping everything as shown in Chapter 6 for instance. A scientific paper reporting on that work is in preparation.

An application of the regularized method of [168] to elastic bodies subject to dry frictional contacts was developed in the Master's thesis of Niklas Melin [201] which was successfully defended in January 2006, and a paper was then published at the SIGRAD'06—Computer Games, conference in Skövde, Sweden, November 22–23, 2006.

In collaboration with Martin Servin, the same regularized stepping scheme was used to model massless cables [253] using relaxed kinematic constraints of special type, allowing to model cable torsion and bending resistance. This paper has been accepted for publication in Computer Graphics Forum and will appear in 2007. Also in collaboration with Servin, a rigid body model for cables with non-zero mass is in preparation and to be submitted for publication in June 2007.

These applications of the regularized stepping scheme [168] helped isolate some issues related to energy fluctuations. The resolution of these came with the application of the variational techniques discussed herein, and the same also produced sound strategies to handle impacts, dry friction, as well as the gyroscopic forces of rigid bodies.

The new results of the thesis will soon be submitted for publication in scientific journals.

1.7 End notes

Interactive physics provides for both interesting applications of standard computational physics techniques as well as for challenging problems due to special requirements.

Despite ample literature on numerical integration of general differential equa-

tions as well as the specific differential equations of mechanical problems, existing methods are not entirely satisfactory, warranting new developments to fully meet the requirements of interactive physics demands, both in terms of better models for nonsmooth, constrained mechanical systems, and in terms of overall balance between stability, speed and accuracy.

Recent advances in discrete variational mechanics are very promising. Indeed, discrete mechanics offers a systematic framework to discretizing all forms of physical phenomena, starting directly from the Lagrangian formulation. The resulting stepping methods automatically enjoy a number of invariance properties which closely approximate the natural equivalents observable in both the continuous formulation of mechanics, and in the real world as well.

The present thesis covers the basics of discrete mechanics techniques with a particular emphasis on general constrained systems, and special attention to the numerical implementation. This justifies the development of special force models, physics based constraint regularization and stabilization methods, a solvable isotropic model of dry friction, and approximation techniques to handle nonsmooth problems, dry friction, and the gyroscopic forces of rigid bodies. Also because of the emphasis on numerical methods, some chapters are devoted to the presentation of algorithms for LCPs and numerical linear algebra, as well as an explicit representation of the quaternion algebra in terms of matrices. For concreteness, some chapters contain the description of standard mechanical constraints and robust implementation thereof, and others are devoted to the analysis of self-contained numerical examples used to test the new methods and illustrate differences with previously known ones.

The main theme is the analysis of the interplays between ideal physical models, discretizations thereof, and numerical implementations. The result is the construction of a fast, stable, regularized, single fixed time-step method, requiring only one linear system or LCP solve operation per step. Being based on discrete mechanics, this method exhibits the symmetries of the original physical problem at the numerical level and thus achieves a reasonable degree of accuracy for the given computational effort, and offers good potential for eventual validation.

2 Bagatelle I: The Discrete Simple Harmonic Oscillator

2.1 Background

Much of the present thesis is aimed at the construction of low order, fixed step, time integration methods, suitable for the equations of motion of classical mechanical systems. The present chapter offers a brief and self-contained illustration of the range of behavior of such methods by analytically computing the discrete trajectories they produce for the simplest possible physical problem, the simple harmonic oscillator. The results illustrate the point that local accuracy is far from telling the complete story of the qualitative behavior. A systematic framework for constructing such well behaved methods is the content of Chapter 3 and 4.1.

The problem is defined in Section 2.2 and discretized to first order in parametric form in Section 2.3. Discrete trajectories are then computed analytically for the entire family in Section 2.4 and stability is investigated for four specific choices of parameters corresponding to standard, low order integration techniques. The results of numerical experiments are presented in Section 2.5 and a summary follows in Section 2.6.

2.2 Problem definition

Consider a unit mass point particle in one dimension with time-dependent coordinate $q : \mathbb{R} \mapsto \mathbb{R}$, attached with an ideal spring to the origin so the force on it is $f = -\omega^2 q$. The equation of motion for this system is known to be the second order linear ordinary differential equation (ODE)

$$\ddot{q} = -\omega^2 q, \tag{2.1}$$

and the solution is the trigonometric expression

$$q(t) = q(0)\cos(\omega t) + \frac{\dot{q}(0)}{\omega}\sin(\omega t), \tag{2.2}$$

where $q(0), \dot{q}(0)$ are the initial position and velocity, respectively. The energy for the system is known to be

$$E = \frac{1}{2}\dot{q}^2 + \frac{\omega^2}{2}q^2 \tag{2.3}$$

and is a constant of the motion, as is readily verified by substituting the expression for the trajectory (2.2). The definition (2.3) is recognized as an ellipse and this means that the *phase plot*—the graph $(x(t), \dot{x}(t))$—has finite area, which also means that trajectories are bounded and periodic: they move around the constant energy curve. The expression for the energy also tells us that this system is bounded in time and, in particular, $|q| \leq \omega\sqrt{E/2}$, $|\dot{q}| \leq \sqrt{E/2}$. Finally, because an ellipse is a closed curve, the system is periodic.

Introduce the natural time variable $s = \omega t$ and the natural coordinate $x(s) = q(t) = q(s/\omega)$. In addition, set the time reference so that the initial conditions are $x(0) = 1$, and $\dot{x}(0) = 0$. The equation of motion now reads $\ddot{x} + x = 0$, after writing $\mathrm{d}x/\,\mathrm{d}s = \dot{x}$, and this second order ODE is now expanded to two coupled first order ODEs and written as the following system

$$\begin{aligned} \dot{x} &= v \\ \dot{v} &= -x. \end{aligned} \tag{2.4}$$

With these definition, the geometry of the trajectory $(x, \dot{x})$ is easily recognized to be the circle $\dot{x}^2 + x^2 = 2E$, where E is the constant energy of the system. Given the simplified initial conditions, the solution of (2.4) is

$$x(s) = \cos s. \tag{2.5}$$

2.3 Discretization

Introduce the parametrization $x_k = x(kh), 0 < h \in \mathbb{R}, k \in \mathbb{N}$ and the parametrized discretization

$$\begin{aligned} \frac{x_{k+1} - x_k}{h} &= \alpha v_k + (1-\alpha) v_{k+1} \\ \frac{v_{k+1} - v_k}{h} &= -\left[\beta x_k + (1-\beta) x_{k+1}\right], \end{aligned} \tag{2.6}$$

with $\alpha, \beta \in [0, 1]$. The method is fully explicit when $\alpha = \beta = 1$, and fully implicit when $\alpha = \beta = 0$. The stepping equations (2.6) correspond to the stationary recurrence

$$\begin{bmatrix} 1 & -h(1-\alpha) \\ h(1-\beta) & 1 \end{bmatrix} \begin{bmatrix} x_{k+1} \\ v_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & h\alpha \\ -h\beta & 1 \end{bmatrix} \begin{bmatrix} x_k \\ v_k \end{bmatrix} \tag{2.7}$$

Define the two-dimensional vector $z = (x, v)^T$, so the recurrence (2.7) can now be written as $Bz_{k+1} = Cz_k$ where B and C are the 2×2 matrices appearing on the left and right hand side of (2.7), respectively. Equivalently, we have

$$z_{k+1} = Az_k, \text{ where } A = B^{-1}C. \tag{2.8}$$

Now, matrix B has the explicit inverse

$$B^{-1} = \gamma^{-1} \begin{bmatrix} 1 & h(1-\alpha) \\ -h(1-\beta) & 1 \end{bmatrix}, \text{ where } \gamma = 1 + h^2(1-\alpha)(1-\beta), \tag{2.9}$$

and this is well defined for the given range $\alpha, \beta \in [0, 1]$. From this, we get an explicit form for matrix $A = B^{-1}C$:

$$A = \gamma^{-1} \begin{bmatrix} 1 - h^2\beta(1-\alpha) & h \\ -h & 1 - h^2\alpha(1-\beta) \end{bmatrix} = \gamma^{-1} D, \tag{2.10}$$

and matrix D has the form

$$D = \begin{bmatrix} u & h \\ -h & v \end{bmatrix}, \tag{2.11}$$

with $u = 1 - h^2\beta(1-\alpha)$, and $v = 1 - h^2\alpha(1-\beta)$. Given the simple form of matrix D in (2.11), it is easy to compute everything there is to know about this family of steppers, and that is done next.

2.4 Discrete trajectories and stability

With this notation, the trajectory of the system is given by $z_k = Az_{k-1} = \ldots = A^k z_0$. An analytic expression for z_k is constructed below. First note that the iterates z_k are bounded as $k \to \infty$ if and only if the spectral radius of matrix A is unity or less:

$$\rho(A) = \max_j |\lambda_j| = \lim_{k\to\infty} \|A^k\|^{1/k}, \tag{2.12}$$

where λ_j are the eigenvalues of matrix A, $|\cdot|$ is the modulus operator over the complex numbers, $\mathbb{C}$, and $\|\cdot\|$ is any compatible matrix norm (see [107], section 2.3 for definitions).

To compute the spectral radius $\rho(A)$, let $\lambda_\pm$ and $\mu_\pm$ be the eigenvalues of matrices A and D, respectively, so that $\lambda_\pm = \gamma^{-1}\mu_\pm$. Concentrating first on matrix D, the characteristic polynomial is

$$\det(D - \mu I) = \begin{vmatrix} u-\mu & h \\ -h & v-\mu \end{vmatrix} = \mu^2 - (u+v)\mu + h^2 + uv = 0, \tag{2.13}$$

and the eigenvalues are

$$\mu_\pm = \frac{u+v}{2} \pm \sqrt{\frac{(u-v)^2}{4} - h^2}. \tag{2.14}$$

We can now reconstruct the eigenvalues of the original problem with straightforward algebraic substitutions to get

$$\lambda_\pm = \gamma^{-1}\left(1 - \frac{h^2}{2}(\alpha+\beta-2\alpha\beta) \pm ih\sqrt{1 - \frac{h^2}{4}(\alpha-\beta)^2}\right), \tag{2.15}$$

where $\gamma = 1 + h^2(1-\alpha)(1-\beta)$ as before in (2.9), and $i = \sqrt{-1}$ is the imaginary unit. It is interesting to note already that for all cases with $\alpha = \beta$, the two roots have the imaginary components $\pm ih$ for any value of h. This imaginary part is what produces oscillatory behavior as we show shortly.

But we can do better than just finding the stability range for the different methods. Recall from the Cayley-Hamilton theorem that any square matrix H satisfies its own characteristic polynomial. Given a 2×2 real matrix H with characteristic polynomial $m(\nu) = \nu^2 + a\nu + b = 0$, having roots $\nu_\pm$, we have

$$H^k = p_k H + q_k I_2, \tag{2.16}$$

where p_k and q_k are polynomials of degree k in the coefficients a and b, and I_2 is the 2×2 identity matrix. Using induction to get the recurrence, and noting that it holds for $k = 0, 1$, and 2, by inspection, and observing that

$$H^{k+1} = p_k H^2 + q_k H = p_k(-aH - bI) + q_k H = (q_k - ap_k)H - bp_k I. \tag{2.17}$$

This defines the polynomials:

$$p_{k+1} = -ap_k - bp_{k-1}, \text{ and } q_k = -bp_{k-1}, \tag{2.18}$$

and so q_k can be recovered easily from p_k and is therefore not considered further. The polynomials p_k now satisfy a two-term linear recurrence relation and the general solution for this is easily found to be

$$p_k = \psi_+ \nu_+^k + \psi_- \nu_-^k, \tag{2.19}$$

where $\psi_\pm$ are scalar coefficients. These are computed by taking account of the initial values $p_0 = 0, p_1 = 1, p_2 = -a$, yielding the formula

$$p_k = \frac{\nu_+^k - \nu_-^k}{\nu_+ - \nu_-}. \tag{2.20}$$

The trajectory of $z_k = H^k z_0$ is thus explicitly given by

$$z_k = \frac{\nu_+^k - \nu_-^k}{\nu_+ - \nu_-} H z_0 - b\frac{\nu_+^{k-1} - \nu_-^{k-1}}{\nu_+ - \nu_-} z_0. \tag{2.21}$$

When the eigenvalues of H are complex, they can be written as $\nu_\pm = re^{\pm i\phi}$ where $r, \phi \in \mathbb{R}, r \geq 0$ leading to the trigonometric expression

$$z_k = r^{k-1}\frac{\sin(k\phi)}{\sin\phi} H z_0 - br^{k-2}\frac{\sin([k-1]\phi)}{\sin\phi} z_0. \tag{2.22}$$

The sinusoidal nature of the dynamics is now clearly visible. As a matter of curiosity, one can recognize the trigonometric form of the Chebyschev polynomials of the second kind as found in [21] for instance, defining $x = \cos\phi$ and $U_k(\cos\phi) = \sin([k+1]\phi)/\sin\phi$.

Applying this result to the iteration matrix A defined in (2.12), we have:

$$A^k = \begin{bmatrix} \gamma^{-1}p_k u - \frac{h^2+uv}{\gamma^2}p_{k-1} & \gamma^{-1}hp_k \\ -\gamma^{-1}hp_k & \gamma^{-1}p_k v - \frac{h^2+uv}{\gamma^2}p_{k-1} \end{bmatrix}, \tag{2.23}$$

where γ is defined as in (2.9), u, v defined as in (2.11), p_k defined as in (2.20) as a function of the eigenvalues of matrix A defined in (2.10). We can then construct the following four cases.

2.4.1 Explicit Euler integration: $\alpha = \beta = 1$

Setting $\alpha = \beta = 1$ in the definition of the recurrence matrix $A = \gamma^{-1}D$ of (2.10) and (2.11), we have $u = v = \gamma = 1$, $(h^2 + uv)/\gamma^2 = h^2$, and $\lambda_\pm = 1 \pm ih$. Define $r = \sqrt{1+h^2} \geq 1$, and $\lambda_\pm = re^{\pm i\phi}$, so that $\cos\phi = (\lambda_+ + \lambda_-)/(2r) = 1/\sqrt{1+h^2}$, and $\sin\phi = (\lambda_+ - \lambda_-)/(2ir) = h/\sqrt{1+h^2}$. Approximating these, we have

$$\cos\phi = 1 - \frac{h^2}{2} + O(h^4), \quad \text{and} \quad \sin\phi = h + O(h^3). \tag{2.24}$$

Notice that the exact solution $\cos s$ given in (2.5) implies the discretized trajectory $x_k = \cos(kh)$. We thus expect that $\phi \approx h$. Now, as is well known, $\cos\phi = 1 - \phi^2/2 + O(\phi^4)$ for small ϕ. Comparing with (2.24) yields $\phi = h + O(h^2)$, as required.

We can then explicitly compute $p_k = r^{k-1}\sin(k\phi)/\sin\phi = (r^k/h)\sin\phi$. Using the trigonometric angle addition formula to evaluate p_{k-1}, we find

$$A^k = (1+h^2)^{k/2}\begin{bmatrix} \cos(k\phi) & \sin(k\phi) \\ -\sin(k\phi) & \cos(k\phi) \end{bmatrix}, \tag{2.25}$$

which implies that $z_k^T z_k = (1+h^2)^k z_0^T z_0 \to \infty$ as $k \to \infty$. This scheme is fundamentally unstable for this system, even though it can process systems of the form $\dot{x} = -\rho x$ in a stable way when $h\rho < 1$ (see [114] for definitions of stability and analysis of the Dahlquist test equation $\dot{x} = -\rho x$).

Explicit Euler is never advertised as a particularly stable scheme but the fact that it is *unconditionally* unstable for the simple harmonic oscillator is seldom reported. In the game physics literature and community, this problem is avoided by introducing a damping force of the form $-\sigma\dot{x}$, for a positive scalar $\sigma > 0$. Further analysis shows that this only works for a relatively small range of σ. In other words, by increasing the damping coefficient enough, the solution becomes unstable again! Explicit Euler is of no use whatsoever in the context of physics simulation since it cannot even process the simplest possible problem.

2.4.2 Implicit Euler integration: $\alpha = \beta = 0$

Setting $\alpha = \beta = 0$ in the definition of the recurrence matrix $A = \gamma^{-1}D$ of (2.10) and (2.11), we have: $u = v = \gamma = 1 + h^2$, $(h^2 + uv)/\gamma^2 = 1/(1+h^2)$, and $\lambda_\pm = \gamma^{-1}(1 \pm ih)$. Define $r = 1/\sqrt{1+h^2}$, and $\lambda_\pm = re^{\pm i\phi}$, so that $\cos\phi = (\lambda_+ + \lambda_-)/(2r) = 1/\sqrt{1+h^2} = r$, and $\sin\phi = (\lambda_+ - \lambda_-)/(2ir) = h/\sqrt{1+h^2} = hr$.

We therefore recover the same dynamics as defined in (2.25) but with the definition $r = 1/\sqrt{1+h^2} < 1$. This implies that $z_n^T z_k = (1+h^2)^{-k} z_0^T z_0 \to 0$ as $k \to \infty$. This scheme is unconditionally stable but it damps oscillations artificially, destroying the physical properties of the system.

Interestingly, the observed oscillation frequency, $\tilde{\omega} = \phi/h$, is identical to that found in the fully explicit case above, namely, $\tilde{\omega} \to 1$ as $h \to 0$. However, as h increases, we reach the limit $\tilde{\omega} \to 0$. Put differently, the limit $\phi = \tan^{-1}(h) \to$

$\pi/2$ as $h \to \infty$ means that the solution oscillates with a period of four steps for large step size, which has nothing to do with the statement of the problem!.

Therefore, not only does the solution decay to zero amplitude exponentially fast, the observed frequency quickly becomes very wrong.

The dead fast stability of the implicit Euler method is old news, both in the numerical analysis and the game physics communities. However, the anomalies just observed are far from well known or widely reported. Strong stability, as desirable as it is, should not come at the cost of destroying the qualitative physical behavior and thus, the implicit Euler method, just like its explicit older brother, is *persona non grata* in physics simulation.

2.4.3 Implicit midpoint rule: $\alpha = \beta = 1/2$

Setting $\alpha = \beta = 1/2$ in the definition of the recurrence matrix $A = \gamma^{-1} D$ of (2.10) and (2.11), we have: $u = v = 1 - h^2/4$, $\gamma = 1 + h^2/4$, and $(h^2 + uv)/\gamma^2 = 1$. The eigenvalues become

$$\lambda_\pm = \frac{1}{1 + h^2/4} \left(1 - \frac{h^2}{4} \pm ih \right), \ |\lambda_\pm| = 1. \tag{2.26}$$

Define $\cos\phi = (\lambda_+ + \lambda_-)/2 = \frac{1-h^2/4}{1+h^2/4}$, and $\sin\phi = (\lambda_+ - \lambda_-)/(2i) = h/(1+h^2/4)$. After some trigonometric manipulations, we find

$$A^k = \begin{bmatrix} \cos(k\phi) & \sin(k\phi) \\ -\sin(k\phi) & \cos(k\phi) \end{bmatrix}. \tag{2.27}$$

This is easily recognized as a planar orthonormal rotation by the angle $k\phi$, so that ${z^T}_k z_k = 1, k = 1, 2, \ldots$. Therefore, the trajectory is bounded and moreover, the energy of the discrete system is constant and equal to that of the continuous system.

For the simple harmonic oscillator which is a linear system, the cost of the implicit midpoint is identical to that of the fully implicit Euler. But for the same money, we get to hear the music as it was intended.

In general, the cost of using the implicit midpoint rule is nearly identical to that of using the implicit Euler methods and the benefits just observed persist. For physical systems, the implicit midpoint rule, which is known at least since Gauss [112], is a godsend and it is indeed very popular in molecular dynamics. Curiously though, the midpoint rule is practically never used in the game physics community despite being equally easy (or hard) to implement as the implicit Euler method. The striking improvements gained form using implicit midpoint instead of implicit Euler was reported in the context of a cloth simulator in [218], though it was not credited with any of its given names. One can only hope that the good example will be followed by the practitioners.

2.4.4 Symplectic Euler: $\alpha = 1, \beta = 0$

Finally, setting $\alpha = 1$ and $\beta = 0$ in the definition of the recurrence matrix $A = \gamma^{-1}D$ of (2.10) and (2.11), we have: $u = 1$, $v = 1 - h^2$, $\gamma = 1$, and $(h^2 + uv)/\gamma^2 = 1$. The eigenvalues are now

$$\lambda_{\pm} = \begin{cases} 1 - \frac{h^2}{2} \pm ih\sqrt{1 - \frac{h^2}{4}}, & |\lambda_{\pm}| = 1, \text{ when } h \leq 2. \\ 1 - \frac{h^2}{2} \pm h\sqrt{\frac{h^2}{4} - 1}, & \text{when } h > 2. \end{cases} \tag{2.28}$$

Therefore, when $h > 2$, $|\lambda_+| > 1 + h^2/2 > 1$ so the spectral radius $\rho(A) > 1$ and the iterates diverge. For the case where $h \leq 2$, define $\cos\phi = (\lambda_+ + \lambda_-)/2 = 1 - h^2/2$ and $\sin\phi = (\lambda_+ - \lambda_-)/(2i) = h\sqrt{1 - h^2/4}$. After some trigonometric manipulations the iteration matrix is found to be

$$A^k = \frac{1}{\cos\psi} \begin{bmatrix} \cos(k\phi + \psi) & \sin(k\phi) \\ -\sin(k\phi) & \cos(k\phi - \psi) \end{bmatrix}, \tag{2.29}$$

where $\cos\psi = \sqrt{1 - h^2/4}$, and $\sin(\psi) = h/2$. Obviously, matrix A^k is *not* a rotation matrix so the norm of z_k is not preserved. However, introducing the rotation matrix

$$R(k\phi) = \begin{bmatrix} \cos(k\phi) & \sin(k\phi) \\ -\sin(k\phi) & \cos(k\phi) \end{bmatrix}, \tag{2.30}$$

and defining $F_k = \tan(\psi)\sin(k\phi)\,\mathrm{diag}(-1, 1)$, we have $A^k = \frac{1}{\cos\psi}R(k\phi) + F_k$ and therefore,

$$\|z_k - F_k z_0\| = \|(A^k - F_k)z_0\| = \|\frac{1}{\cos\psi}R(k\phi)z_0\| = \frac{1}{\cos\psi}\|z_0\| = const. \tag{2.31}$$

This defines a quadratic invariant on the trajectory that is preserved by the iteration which is close to the energy of the system. In fact, the difference is of the order of $\tan(\psi) = O(h^2)$.

This is where one should breath deeply and take a step back. For a general system, this symplectic Euler method (there are two actually, the second one having $\alpha = 0$ and $\beta = 1$) is no more expensive computationally than explicit Euler, and much cheaper than implicit Euler or midpoint, and yet it captures some essential features of the physical problem. Besides, as long as the forces on the system are velocity independent, it has second order accuracy like its cousin the midpoint rule of the previous section.

2.5 Numerical experiments

The most instructive way to look at the output signal for the simple harmonic oscillator is to plot the phase portrait, i.e., build a graph of velocity versus position. Recall that in natural coordinates,

$$v^2 + x^2 = 2E, \tag{2.32}$$

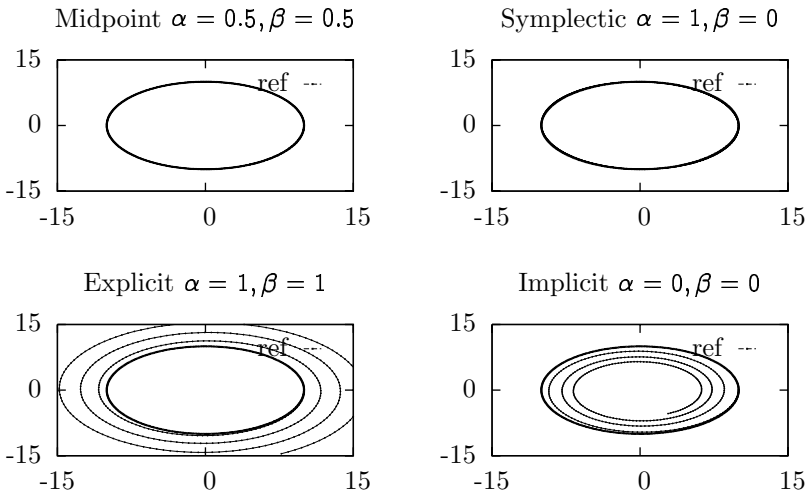

Figure 2.1: Phase portraits of the four different methods for small time step $h\omega = 1/20$. Both symplectic Euler and implicit midpoint methods produce closed ellipsoids but explicit Euler spirals outward and implicit Euler spirals inward, slowly though.

where v is the natural velocity, x is the position, and E is the energy. This is the equation of a circle when natural time is used but is an ellipsoid otherwise. The stationary recurrence $z_{k+1} = A(\alpha, \beta) z_k$ of (2.8) was implemented and started with $z_0 = (10, 0)^T$, i.e., an initial position of $x(0) = 10$ and an initial natural velocity of $\dot{x}(0) = 0$. The results are plotted in Figure 2.1 for small time step $h\omega = 1/20$, in Figure 2.2 for the moderate time step $h\omega = 1/5$, and in Figure 2.3 for the large time step $h\omega = 1/2$. Note the outward spiral profile of the explicit method with $\alpha = \beta = 1$ and the inward spiral profile of the implicit method $\alpha = \beta = 0$. Both the midpoint method with $\alpha = \beta = 1/2$ and the symplectic method with $\alpha = 1, \beta = 0$, produce the correct phase portrait as easily seen from the graphs.

Note that for large time step $h\omega = 0.5$, the explicit method quickly escapes the range $[-15, 15]$ allowed on the plot of Figure 2.3. Given that the cost of this method is identical to that of the symplectic method, even for nonlinear problems, this clearly shows that explicit Euler should never be let near any second order differential equation describing a physical problem. For implicit Euler, the problem is not stability but total destruction of the qualitative aspects of the system—a numerical realization of the great deceiver himself!

2.6 End notes

A detailed analysis of the four schemes considered is found in Ref. [112], Chapter 1, for instance. It is easy to demonstrate that the implicit midpoint method

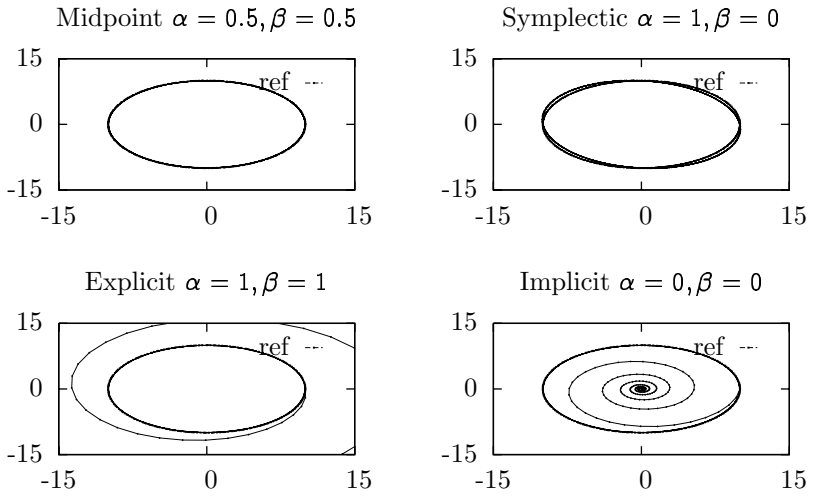

Figure 2.2: Phase portraits of the four different methods for moderate time step: $h\omega = 0.2$. The symplectic Euler and implicit midpoint methods keep producing closed ellipsoids but explicit Euler spirals outward and implicit Euler spirals inward at sizable speed.

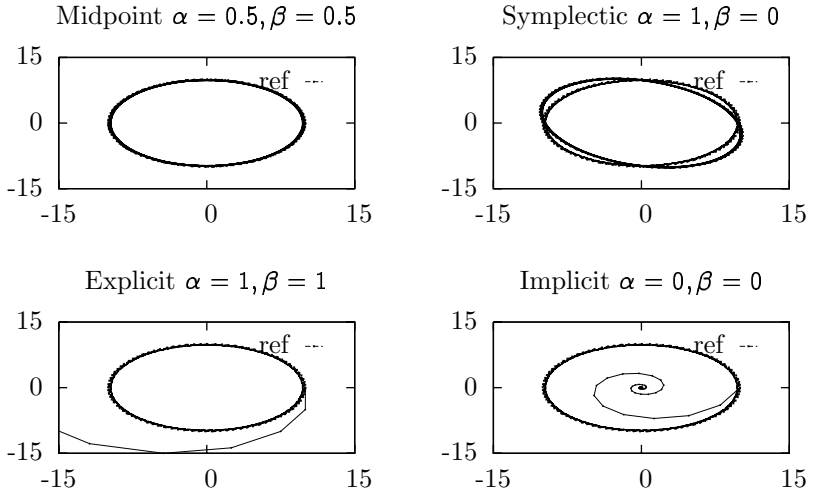

Figure 2.3: Phase portraits of the four different methods for large time step: $h\omega = 0.2$. Whilst the implicit midpoint method keeps computing the correct energy surface, the symplectic Euler method is now producing a skewed ellipsoid, but ellipsoid nevertheless. The other two methods cannot survive 10 steps.

is of second order but that the other three schemes are first order only in general, though symplectic Euler has second order accuracy for velocity independent forces. Nevertheless, only the symplectic and implicit midpoint methods are of any use since only these two schemes reproduce any of the basic physics. In addition, the symplectic scheme approximates the energy of the system within $O(h^2)$ *for all times* and exactly preserves a quadratic invariant which is close to the energy of the system! This is precisely the sort of integration scheme that is needed to meet the requirements described in Section 1.2.

There is a fundamental reason why both the symplectic Euler method and the midpoint rule perform well on physical problems which is investigated in depth in Chapter 3. As will be shown, these methods are good because they can be constructed using the principle of least action applied directly on the discrete trajectory. Also to be shown in Chapter 3, the variational principle is the culprit enforcing preservation of symmetries of both continuous and discrete mechanical systems, and is precisely the daemon needed if one wants good physical trajectories. It is because of these exact invariants of the discrete motion that low order methods are usable at all. By contrast, the implicit and explicit Euler methods cannot be constructed from the variational principle. Though they appear equally valid discretizations of the equations of motion, they do not preserve any qualitative aspect of the physical trajectory over sizable time intervals.

The striking difference between the qualitative behavior of explicit or implicit first order Euler integration versus the implicit midpoint or first order symplectic method is a traceable to the non-commutation of two limits. One could first construct the equations of motion from the principle of least action and then discretize these to achieve consistency at a given order. Conversely, one could start with discretizing the trajectories and then apply the least action principle on the discrete samples to construct stepping formulae. The latter sequence is best for constructing integrators of mechanical systems because it preserves all the natural symmetries of the motion, as even hardened numerical analysts agree [112].

3 Analytic and Discrete Mechanics

This chapter introduces the necessary concepts of classical mechanics—the study of how and why physical bodies move—along with time discretization techniques based on the principle of least action, the cornerstone of analytic mechanics. The presentation unifies the well known results of continuous analytic mechanics with the more recent developments in discrete mechanics. The latter allow the construction of discrete time-stepping methods, the *discrete mechanical integrators*. The emphasis weighs on the global conservation properties of discrete mechanical integrators on the one hand, and the discretization of non-conservative forces and constraints on the other. The latter two are essential elements for the construction of physics motivated regularization and constraint stabilization techniques presented in Chapter 4, as well as the dry friction model presented in Chapter 10.

After defining the scope of application, elementary notions of mechanics, and the contrast between Newtonian and analytic mechanics in Section 3.1, especially with regards to their respective time discretizations, Newton laws of motion are detailed in Section 3.3, followed by definitions of the fundamental of *work* and *energy* in Section 3.4, and the introduction of both continuous and discrete forms of the variational principle of mechanics in Section 3.5. After illustrative examples of the time discretization strategy are presented in Section 3.7, continuous and discrete spatial symmetries and conservation laws of physical systems are investigated in Section 3.8, followed by the analysis of symmetries of time and connections to energy in Section 3.10. A summary of geometric concepts and an analysis of the symplectic nature of the flow of both continuous and discrete trajectories are presented in Section 3.11, accompanied with illustrative examples. The basic model of Section 3.5 is extended to include non-conservative forces and their variational discretization in Section 3.12, and a wide array of constraints in Section 3.14, in which the notion of *constraint regularization* is introduced, as well as a novel analytic representation of nonholonomic constraints using a special type of Rayleigh dissipation function. The minimization structure of the stepping equations is analyzed in Section 3.16 which is followed by a survey of other work and a short summary in Section 3.17.

The present chapter is self-contained and tutorial in nature.

3.1 Introduction

Classical mechanics is the study of motion of *physical bodies*, tangible objects with finite extent and mass. It provides answers both to how and why these objects move. The "how" is answered by *kinematics*: the study of motion without regards

to what caused it. The "why" is answered by *dynamics*: the study of the causes of motion, namely, the description of forces. It is limited in scope to bodies which are not so small as to be subject to the laws of quantum mechanics, too big as to be the subject the laws of general relativity, or moving at speeds close to the speed of light so as to be subject to the laws of special relativity. Classical mechanics also neglects all phenomena related to heat and thermodynamics as well as chemical reactions. Specifically, the description of motion produced by classical mechanics applies to physical bodies ranging from molecules up to the smaller and slower celestial bodies.

The application context of the present thesis fits comfortably within the range of validity of classical mechanics. Given the focus on real-time interactive simulation of typical everyday situations, the time scales range from milliseconds to days, and the length scales range from millimeters to kilometers, and the masses range from milligrams to tonnes. The archetype multibody systems to keep in mind range from robots to ground vehicles, including virtual humans and so on.

In addition to the restrictions just stated, the analysis is focused on systems with finite degrees of freedom. Fluids, gases and deformable solids are explicitly excluded in what follows in order to concentrate on systems of point particles for the most part, and including rigid bodies in some sections. Extensions to certain types of spatially discretized continuous systems are straight forward and have been successfully realized. These will only be mentioned in the literature notes to limit the scope. Likewise, some of the methods developed below could also be extended beyond the range restrictions mentioned above to cover celestial mechanics and perhaps molecular dynamics as well but these avenues will not be pursued further.

Within the stated range of mass, length and time scales, the only fundamental physical interactions of any relevance are the electromagnetic and gravitational forces. At the length and mass scales of interest for interactive physics, only uniform gravitational fields are relevant. Also, since most macroscopic objects are electrically neutral, electromagnetic forces are restricted to contact physics and drivers and these are usually modeled using macroscopic constitutive laws instead of molecular models. This justifies the extensive use of *constraints* in what follows.

The purpose of the present analysis is the construction of numerical methods which can produce discrete time approximations of the motion of certain types of physical bodies, point particles and rigid bodies, as well as aggregates thereof, in particular. Specifically, the focus is the development of fast computational methods which meet the requirements stated in Section 1.2. The strategy chosen relies on *discrete mechanics* instead of pure numerical analysis. This allows establishing a direct correspondence between physical models and the numerically computed motion, which is important for understanding the numerical results. The presentation thus begins at the heart of classical mechanics and moves quickly to *discrete mechanical integrators*.

Dynamics establishes the relation between the time rate of change of the kinematics variable and the interactions between the different parts, which are called

the *forces*. To do this, one first needs to introduce the notion of *momentum* or *quantity of motion*. For a point particle with Cartesian coordinates $x^{(i)}$, the momentum is $p^{(i)} = m\dot{x}^{(i)}$, where the *mass* m is a positive scalar. For generalized coordinates, the momentum is a more complicated expression which is derived later in this chapter. The fundamental principle of dynamics, which was established by experimentation, is that the rate of change of momentum is directly proportional to the applied force, the content of Newton's second law which is explained in more details in Section 3.3.

3.2 Essential kinematics

The kinematics analysis of classical mechanics starts with point particles labeled with indices $i = 1, 2, \ldots, N$, with time-dependent positions described as Cartesian coordinate vectors $x^{(i)}(t) \in \mathbb{R}^n$, and velocity vector $\dot{x}^{(i)}(t) \in \mathbb{R}^n$, in either one, two or three dimensions. This can be extended to rigid bodies as described later in Chapter 12. Coordinates $x^{(i)}(t)$ are agglomerated into a *configuration vector*, $x(t)$ by concatenation

$$x(t) = \begin{bmatrix} x^{(1)}(t) \\ x^{(2)}(t) \\ \vdots \\ x^{(N)}(t) \end{bmatrix}, \tag{3.1}$$

and similarly for the velocities.

One may impose restrictions on this motion via constraints such as a sufficiently smooth function $g : \mathbb{R}^n \mapsto \mathbb{R}^m$, with $g(x) = 0$, say. When $m = n$, this amounts to a general change of coordinates. Defining the *smooth manifold* $Q = g^{-1}(0)$, the configuration of the system is then given by the *generalized coordinates* $q(t) \in Q$ where the manifold Q is the *configuration space*. Since the generalized coordinates $q(t)$ are restricted to the configuration space Q, the corresponding generalized velocities $\dot{q}(t)$ must be tangent to Q at the point $q(t)$. The full kinematic description is then provided by the time history of generalized coordinates and their corresponding generalized velocities which is written as $(q(t), \dot{q}(t)) \in TQ$, where TQ is the *tangent bundle* of the configuration manifold Q, i.e., the disjoint unions of tangent spaces T_qQ for each point $q \in Q$. The tangent bundle TQ is often called *phase space* as well, especially in the physics literature. Though we shall not delve in the depths of differential geometry, this abstract notation is used throughout for the sake of consistency and precision. It is a fundamental result of classical mechanics—which we explain further below—that kinematics analysis stops with the description of phase space, since the equations of motion prescribe the relation of the generalized acceleration. Thus, the kinematic variables consist solely of the phase space vector $(q(t), \dot{q}(t))$.

A computer simulation necessarily implies a discretization of kinematics, and in particular, a discretization of time. In other words, a simulation should produce *trajectories* at discrete times $t_1 < t_2 < \ldots < t_k$, consisting of discrete generalized

coordinates $q_k = q(t_k)$ and velocities $\dot{q}_k = q(t_k)$. The computation of q_k and $\dot{q}_k$ for a given set of forces and constraints is the heart of a physics engine. Since the changes in kinematics are governed by the laws of dynamics, this is what we now turn to.

3.3 Newton's laws of motion

The starting point in classical mechanics is Newton's three laws of motion which summarize experimental observations made by Newton himself but also by his predecessors, notably including Galileo.

The starting point is Galileo's principle of relativity which states, (as quoted in [22]): "There exist coordinate systems (called inertial) possessing the following two properties:

1. All the laws of nature at all moments of time are the same in all inertial coordinate systems.
2. All coordinate systems in uniform rectilinear motion with respect to an inertial one are themselves inertial."

In addition to this, time is assumed to be universally defined, having the same rate of flow in any reference frame, a definition which was modified eventually in Einstein's theory of special relativity [188].

Newton defines the concept of a "physical body" which is a point particle with time dependent position $x : \mathbb{R} \mapsto \mathbb{R}^3$, with scalar mass $m > 0$, defined as the product of volume and density. A physical body has a "quantity of motion"—called momentum today—which is the product of mass and velocity: $p(t) = m\dot{x}(t)$, where $\dot{x}(t)$ is the total time derivative of the position. Newton then describes forces as "that which causes changes in momentum", and this leads to the three laws which we quote directly from [215]:

- An object at rest will remain at rest unless acted upon by an external and unbalanced force. An object in motion will remain in motion unless acted upon by an external and unbalanced force;
- The rate of change of momentum of a body is proportional to the resultant force acting on the body and is in the same direction;
- All forces occur in pairs, and these two forces are equal in magnitude and opposite in direction.

From this, we get the mathematical description:

Momentum $p(t) = m\dot{x}(t)$;

Newton's first and second laws: time rate of change of motion, $\dot{p}(t)$ is proportional to net force $f(t)$:

$$\dot{p}(t) = f(t); \tag{3.2}$$

Newton's third law: action equals reaction so that if $f^{(i,j)} \in \mathbb{R}^3$ is the force acting on body i due to body j, and conversely $f^{(j,i)} \in \mathbb{R}^3$ is the force acting on body j due to body i, then, $f^{(i.j)} = -f^{(j,i)}$, i.e., the action and reaction forces are equal and opposite.

This formulation is very general. It can be extended to cover continuous bodies with finite extents as well as fluids and gasses by dividing these objects into collection of N point particles, $x^{(i)}(t), p^{(i)}(t) = m^{(i)}\dot{x}^{(i)}(t)$, $i = 1, 2, \ldots, N$, and estimating the forces $f^{(i,j)}$ these particles exert on each other. Aggregating the system vectors of positions $x(t)$ and momentum $p(t) = M\dot{x}(t)$, as done in (3.1), and introducing the block diagonal mass matrix

$$M = \mathrm{diag}(m^{(1)}I_n, m^{(2)}I_n, \ldots, m^{(N)}I_n), \tag{3.3}$$

where n is the dimension of the kinematic space considered, as well as the aggregated *net force*, with $f^{(i)} = \sum_j f^{(i,j)}$, maintains the form of (3.2). Thus, Newton's second law can be seen as a formulation of the dynamics of systems of point particles. Using elementary calculus, it is also possible to change coordinates and define aggregate entities as is done in Chapter 12 for the case of the rigid body.

When considering systems of point particles, several symmetries of the problem can be discovered directly from this formulation as is done in [22], chapters 1 and 2. However, as we shall see below, the Lagrangian formulation offers a more systematic way to discover symmetries and to correlate these to *integrals of motion*, i.e., scalar constants which can be used to reduce the number of variables in the problem.

Also, since the equation of motion in (3.2) is a second order ODE, it can be integrated using standard methods. This is good enough in several cases in fact. But when it comes to numerical preservation of the underlying symmetries lurking in the three laws and the Galilean principle of relativity, standard integration methods are found wanting and much effort has been invested in the last decade to construct numerical integration formulae preserving well known physical invariants, such as energy and momentum.

For these reasons, Newton's formulation of dynamics is abandoned here in favor of the more satisfactory *analytic* formulation.

3.4 Work and energy

Before introducing the principle of least action central to analytic mechanics, we need to introduce the concepts of *work* and *energy*. When a force $f : \mathbb{R} \mapsto \mathbb{R}^3$ acts on a point particle with mass m and coordinates $x : \mathbb{R} \mapsto \mathbb{R}^3$, it eventually alters its kinematic state. To measure this, consider the correlation between the force and the change in position along a physical trajectory $C = \{x(t) \mid t \in [t_0, t_1]\}$ and thus define *mechanical work* done by a force as the following line integral

$$W = \int_C f^T \, \mathrm{d}x. \tag{3.4}$$

The first observation is that given Newton's second law (3.2), and assuming constant mass m, this integral reduces to

$$\begin{aligned} W &= m \int_C \ddot{x} \, \mathrm{d}x = m \int_{t_0}^{t_1} \mathrm{d}s \ddot{x}^T(s)\dot{x}(s) = m\frac{1}{2} \int_{t_0}^{t_1} \mathrm{d}s \frac{\mathrm{d}}{\mathrm{d}s} \left(\dot{x}^T(s)\dot{x}(s) \right) \\ &= \frac{1}{2} m \|\dot{x}\|^2 \Big|_{t_0}^{t_1}. \end{aligned} \tag{3.5}$$

This introduces an important quantity, namely, the *kinetic energy* of a point particle as

$$T(\dot{x}) = \frac{1}{2} m \|\dot{x}\|^2. \tag{3.6}$$

The action of a force f over a given trajectory thus yields a change in kinetic energy in proportion of the work done.

Next, consider the line integral (3.4) defining work again. For an arbitrary force $f(t)$, the work done depends on the details of the path C. However, the line integral (3.4) is notably *independent* of the path C for the case where

$$f = -\frac{\partial V(x)}{\partial x^T} = -\nabla V(x), \tag{3.7}$$

where $V : \mathbb{R}^3 \mapsto \mathbb{R}$ is called the *potential energy* or the *potential function*, and the negative sign convention will become clear presently. For this important special case, we have

$$W = -V(x(t_1)) + V(x(t_2)) = T(\dot{x}(t_1)) - T(\dot{x}(t_2)). \tag{3.8}$$

Therefore, the sum of kinetic and potential energy is conserved so that we can define the *total energy* as

$$E = T(\dot{x}) + V(x). \tag{3.9}$$

These observations generalize to the n-dimensional case where $x : \mathbb{R} \mapsto \mathbb{R}^n$ by replacing the scalar mass m with a constant, $n \times n$, symmetric and positive definite mass matrix M, and introducing the n-dimensional force vector $f = -\nabla V(x)$. Clearly, the function $V(x)$ can also be replaced by a sum $\sum_i V^{(i)}(x)$, each producing a force $f^{(i)} = -\nabla V^{(i)}$

We thus define *conservative* mechanical systems as those for which all forces can be derived from potential functions. In [173], among other references, the forces which can be derived from potential functions are called *monogenic* to distinguish them from *polygenic* forces, the latter being generally non-conservative. Some special types of dissipative forces can be represented with *pseudo-potentials*—the Rayleigh functions introduced in Section 3.12.

Note that under a curvilinear coordinate transformation, the kinetic energy generally becomes dependent on both $\dot{q}$ *and* q. Indeed, for a two-dimensional case with $x : \mathbb{R} \mapsto \mathbb{R}^2$ and $m > 0$, switching to polar coordinates (r, θ) with $x_1 = r\cos\theta$, and $x_2 = r\sin\theta$, the velocity becomes

$$\dot{x} = \begin{bmatrix} \dot{r}\cos\theta - \dot{\theta} r \sin\theta \\ \dot{r}\sin\theta + \dot{\theta} r \cos\theta \end{bmatrix}, \tag{3.10}$$

and thus $T = (1/2)m\|\dot{x}\|^2 = (1/2)m(\dot{r}^2 + \dot{\theta}^2 r)$ clearly depends on r as well as on $\dot{r}$ and $\dot{\theta}$.

It is also possible to define potential functions which depend on $\dot{x}$ as well, and this is the case for the magnetic vector potential in electrodynamics [136]. However, magnetic phenomena are not considered in the present thesis and we thus concentrate on potential functions of the restricted form $V(q)$. Potential energy cannot depend on time since then, the line integral (3.4) would be path dependent again.

3.5 Basic variational principle

The most general statement of the variational principle is due to d'Alembert who postulated that, given a physical trajectory and infinitesimal displacements thereof at each point in time, and compatible with all imposed constraints, the work done by the virtual displacements along the physical trajectory vanishes when the motion is not bounded, and is non-positive otherwise (as quoted in [173]). This last statement being known as the Fourier inequality and which will be useful in the study of contacts. This statement is stated mathematically for both the continuous and discrete cases.

To give flesh to this principle, consider a mechanical system with generalized coordinates $q \in Q$, where Q is the configuration space. In general, Q is a manifold but for what follows, it is sufficient to think of a coordinate chart on Q so that, at least locally, $q \in \mathbb{R}^n$ (see [248, 104] for essential notions of differential geometry). The velocity of this point is denoted by $\dot{q}$ and the configuration space is the tangent bundle TQ. The structure of TQ is the set of all pairs, $(q, \dot{q})$, such that $\dot{q}$ is locally tangent to the manifold Q at the point q.

A Lagrangian is a scalar function $\mathcal{L} : TQ \mapsto \mathbb{R}$. A case of special interest is

$$\mathcal{L}(q, \dot{q}) = T(q, \dot{q}) - V(q), \tag{3.11}$$

where $T : TQ \mapsto \mathbb{R}$ is the kinetic energy and $V : Q \mapsto \mathbb{R}$, is the potential energy as defined in Section 3.4.

Systems with Lagrange function of the form 3.11 are conservative as described in Section 3.4, a fact which will be proven with more mathematical rigor in Section 3.9 below.

The kinetic energy term $T(q, \dot{q})$ is a quadratic form. It is either defined as $T(q, \dot{q}) = \frac{1}{2}\dot{q}M\dot{q}$, where the $n \times n$ matrix M is symmetric, positive definite, and constant, or, for the rigid body case in particular, as explained in details in Chapter 12, $T(q, \dot{q}) = \frac{1}{2}\dot{q}^T M(q)\dot{q}$, and the square $n \times n$ matrix function $M(q)$ is block diagonal, symmetric, positive definite, smooth matrix function $M : Q \mapsto \mathbb{R}^{n \times n}$. To simplify both notation and the exposition, the rigid body case will be analyzed in separate sections. Therefore, in what follows, the mass matrix is assumed constant except when explicitly stated otherwise. Constant mass matrices cover the cases of systems of point particles in three dimensions, rigid bodies in two dimensions, and dynamically homogeneous rigid bodies (those

which have identical principal inertiae) in three dimensions, among others, as long as they are expressed in Cartesian coordinates.

Collecting these observations, our basic model Lagrangian has the explicit form

$$\mathcal{L}(q, \dot{q}) = \frac{1}{2}\dot{q}^T M \dot{q} - V(q), \tag{3.12}$$

where the constant $n \times n$ real matrix M is symmetric and positive definite, and the rigid body Lagrangian has the explicit form

$$\mathcal{L}(q, \dot{q}) = \frac{1}{2}\dot{q}^T M(q) \dot{q} - V(q), \tag{3.13}$$

where the $n \times n$ real matrix $M(q)$ is a symmetric and positive definite smooth function of the coordinates q. In addition to this explicit form, the potential function $V(q)$ is assumed to contain uniform gravitational potential terms which have the form

$$V_g(q) = a_g u^T M q \tag{3.14}$$

where a_g is a positive constant—the acceleration due to uniform gravity—and u^T is a $1 \times n$ projection matrix. For a single point particle with mass m and Cartesian coordinates x, this term is just $a_g m z^T x$, where z is a normal vector pointing upward.

One of the main theme of the present thesis is to extend simple Lagrangian as defined above to include terms which are necessary for stabilizing and regularizing the numerical methods in a systematic, physical way. These terms will be carefully chosen so as not to strictly preserve the overall variational formulation exposed henceforth.

Introduce the action integral as the following functional

$$S[q] = \delta \int_{t_0}^{t_1} \mathrm{d}s \mathcal{L}(q(s), \dot{q}(s)), \tag{3.15}$$

where $q : [t_0, t_1] \mapsto Q$ is a smooth, known function of time. For the general case, $q(t)$ must be of class C^2 but this can be relaxed at the cost of a more complicated analysis to include nonsmooth phenomena [87]. Now, Hamilton's principle of least action states that the physical trajectory of this system is the path which minimizes the value of the functional S over all smooth paths which go through given endpoints. This is not only an interesting postulate but an experimentally verified fact. In fact, as we show below, Hamilton's principle of least action does reproduce Newton's laws of motion.

Concretely, assume that $(q(t), \dot{q}(t))$ is the trajectory of the physical system. Following the original notation of Lagrange and using δ to denote the *variational operator*, let $\delta q(t)$ be an infinitesimal, time dependent perturbation of the trajectory so that $q + \delta q \in Q$ be a small perturbation of the motion. Then, using calculus of variation [284, 173], the resulting variation δS of the action functional

is computed to

$$\begin{aligned}\delta S[q] &= \delta \int_{t_0}^{t_1} \mathrm{d}s \mathcal{L}(q(s), \dot{q}(s)) \\ &= \int_{t_0}^{t_1} \mathrm{d}s \left(-\frac{\mathrm{d}}{\mathrm{d}s} \left(\frac{\partial \mathcal{L}(q,\dot{q})}{\partial \dot{q}} \right) + \frac{\partial \mathcal{L}(q,\dot{q})}{\partial q} \right) \delta q + \left[\frac{\partial \mathcal{L}(q,\dot{q})}{\partial \dot{q}} \delta q \right]_{t_0}^{t_1}.\end{aligned} \tag{3.16}$$

Assuming that S is an extremal or, at least, stationary on the physical trajectory, the variation δS should vanish at least to first order in δq. The boundary term can be neglected if we restrict the variations to satisfy: $\delta q(t_0) = \delta q(t_1) = 0$. A necessary and sufficient condition for δS to vanish is that the trajectory satisfies the Euler-Lagrange equations

$$\frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial \mathcal{L}}{\partial \dot{q}^T} - \frac{\partial \mathcal{L}}{\partial q^T} = 0. \tag{3.17}$$

This is a set of non-linear, second order ODEs in semi-implicit form. For the case of the constant mass matrix simple Lagrangian of (3.12), we have

$$\frac{\mathrm{d}}{\mathrm{d}t} (M\dot{q}) = M\ddot{q} = -\frac{\partial V}{\partial \dot{q}^T} = -\nabla V(q) = f(q), \tag{3.18}$$

which is Newton's second law (3.2), writing $f(q) = -\nabla V$ for the vector of forces acting on the system. Assuming the mass matrix M is invertible, the equations are explicit, non-linear and second order ODEs which can be integrated with standard methods. But we can do much better by reversing the order of discretization.

The process of evaluating the Lagrange function $\mathcal{L}(q, \dot{q})$ for a given system and computing the resulting Euler-Lagrange equations of motion for it is called the variational method. The method extends to cases where the system is non-conservative by using d'Alembert's principle as described in Section 3.12 but in that case, the equations of motion are no longer strictly the Euler-Lagrange equations.

3.6 Discrete variational principle

Introduce the discretized Lagrangian as the action integral over a short interval of time, $h > 0$

$$\mathbb{L}_d(q_0, q_1, h) = \int_0^h \mathrm{d}s \mathcal{L}(q(s), \dot{q}(s)), \tag{3.19}$$

where $q_0 = q(0)$ and $q_1 = q(h)$. The integral can be approximated in a number of ways. For instance, if h is small enough, we have $\dot{q}(s) \approx h^{-1}(q_1 - q_0)$, and $q(s) \approx (1/2)(q_1 + q_0)$. We describe some alternative choices in Section 3.7.

Define the discretized action as follows

$$\mathbb{S}_d(q_0, \ldots, q_N, h) = \sum_{k=0}^{N-1} \mathbb{L}_d(q_k, q_{k+1}, h). \tag{3.20}$$

Now, the condition that $\mathbb{S}_d(q_0, \ldots, q_N, h)$ should be optimal over the trajectory is that all partial derivatives should vanish namely

$$\frac{\partial \mathbb{S}_d(q_0, \ldots, q_N, h)}{\partial q_k} = \frac{\partial \mathbb{L}_d(q_k, q_{k+1}, h)}{\partial q_k} + \frac{\partial \mathbb{L}_d(q_{k-1}, q_k, h)}{\partial q_k} = 0. \tag{3.21}$$

These stationarity conditions yield the fundamental equations of discrete mechanics, namely, the discretized Euler-Lagrange equations

$$D_1^T \mathbb{L}_d(q_k, q_{k+1}, h) + D_2^T \mathbb{L}_d(q_{k-1}, q_k, h) = 0, \tag{3.22}$$

which are the transpose of (3.21), and where D_1, D_2 are the partial derivatives operators with respect to the first or second argument, respectively. To enunciate further, given a scalar function of two vector arguments, $f(x, y)$, we write

$$\begin{aligned} D_1 f &= \frac{\partial f(x,y)}{\partial x}, \qquad D_2 f = \frac{\partial f(x,y)}{\partial y}, \text{ and also,} \\ D_1^T f &= \frac{\partial f(x,y)}{\partial x^T}, \qquad D_2^T f = \frac{\partial f(x,y)}{\partial y^T}. \end{aligned} \tag{3.23}$$

The discrete Euler-Lagrange equations (3.22) is a set of simultaneous nonlinear equations which can be solved for q_{k+1} given q_k and q_{k-1}. This defines the map: $\Phi : \mathcal{Q} \times \mathcal{Q} \mapsto \mathcal{Q} \times \mathcal{Q}$ with $\Phi(q_k, q_{k-1}) = (q_{k+1}, q_k)$, which is the discrete time integrator. Solutions to the nonlinear equations (3.22) exist in general at least for the case where the matrix $D_2 D_1^T \mathbb{L}_d(q_k, q_{k+1}, h)$ is invertible. For the simple case considered here, this matrix is precisely the mass matrix M which is assumed symmetric and positive definite, hence invertible. Extensions covering applications where the matrix $D_2 D_1^T \mathbb{L}_d(q_k, q_{k+1}, h)$ is singular are discussed in [240]. We will return to this later on when discussing the discretization of *ghost* variables, since these have no mass.

The process of computing the discrete Lagrangian and evaluating the discrete Euler-Lagrange equations for a given system is the discrete variational method or variational method for short, when the context is clear.

It might appear that this discretization strategy can only yield first or second order methods. However, section 2.6 of [196] contains constructions of higher order methods which include higher order symplectic partitioned Runge-Kutta as well as Galerkin methods. It is also possible to generate higher order methods using composition rules, also presented in [196], though this is not necessarily the most efficient strategy. Other constructions include some of the Newmark integration methods as well [151]. For the purpose of this thesis however, the emphasis is on low order methods and we do not pursue this topic further.

3.7 Examples of discrete mechanical integrators

To construct approximations of (3.19), we start with:

$$\dot{q}(s) \approx \frac{q_1 - q_0}{h}, \qquad q(s) \approx q_0. \tag{3.24}$$

Substituting these choices in the reference case (3.12) produces

$$\mathbb{L}_d(q_0, q_1, h) = \frac{1}{2h}(q_1 - q_0)^T M(q_1 - q_0) - hV\left(q_0\right), \tag{3.25}$$

and applying the discrete Euler-Lagrange equations (3.22) to this yields

$$-\frac{1}{h}M\left(q_{k+1} - 2q_k + q_{k-1}\right) - h\nabla V(q_k) = 0. \tag{3.26}$$

After rearrangement, this becomes

$$q_{k+1} = 2q_k - q_{k-1} - h^2 M^{-1}\nabla V(q_k), \tag{3.27}$$

which is recognized as the Verlet [271] formula, and also known as Leapfrog or Störmer-Verlet [112].

Despite its simple appearance, this formula is in fact widely used in molecular dynamics simulation [178] for instance. For the simple harmonic oscillator, it corresponds to the Symplectic Euler method analyzed in Chapter 2.

It is interesting to note that any approximation of the form

$$\mathbb{L}_d^{(\beta)}(q_0, q_1, h) = \frac{1}{2h}(q_1 - q_0)^T M(q_1 - q_0) - \beta hV\left(q_0\right) - (1-\beta)hV\left(q_1\right), \tag{3.28}$$

leads to the same discrete stepping equations (3.27). This symmetry is analyzed in [196] to demonstrate that the innocuous looking stepping scheme (3.27) is in fact *second order* accurate.

Using $q(s) \approx \frac{1}{2}(q_0 + q_1)$ and $\dot{q}(s) \approx h^{-1}(q_1 - q_0)$, we get the discrete Lagrangian

$$\mathbb{L}_d^{(1/2)}(q_0, q_1, h) = \frac{1}{2h}(q_1 - q_0)^T M(q_1 - q_0) - hV\left(\frac{q_0 + q_1}{2}\right), \tag{3.29}$$

and applying the discrete Euler-Lagrange equations (3.22) to that yields

$$q_{k+1} = 2q_k - q_{k-1} - \frac{h^2}{2}M^{-1}\left[\nabla V\left(\frac{q_k + q_{k-1}}{2}\right) + \nabla V\left(\frac{q_{k+1} + q_k}{2}\right)\right], \tag{3.30}$$

which is the implicit mid-point rule. This was also studied in Chapter 2 and was one of the two good methods, in fact.

Conspicuously absent from the set of first and second order stepping formulae one can find using the variational formulation are the implicit and explicit first order Euler methods described in Chapter 2. As shown in Section 3.11, all methods derived from the discrete variational principle are *symplectic* and this means that their stepping matrix–matrix A of (2.8) for the simple harmonic oscillator—should have unit determinant. Neither explicit nor implicit first order Euler have that property as shown in Section 2.4. At the very least, the variational principle can guide the selection amongst known methods, restricting candidates to those which can be constructed from the discrete Euler-Lagrange equations (3.22). Indeed, as demonstrated in the next few sections, where the differential equations of motion of mechanical systems are concerned, variational integrators are provably superior to general methods in every aspect, quantitative as well as qualitative.

Further numerical investigations of the methods described above are provided in Section 3.11 below.

3.8 Symmetries and conservation laws

We now investigate some conservation laws which are implied by the variational formulation of classical mechanics in Lagrangian form. To fix the ideas, consider a system made of two unit point masses with coordinates $q^{(1)}(t), q^{(2)}(t) \in \mathbb{R}^3$, which are subject to a *central force* potential, i.e., $V(q) = V(\|q^{(1)} - q^{(2)}\|)$, so the Lagrangian for this system is simply

$$\mathcal{L}(q, \dot{q}) = \frac{1}{2}\|\dot{q}^{(1)}\| + \frac{1}{2}\|\dot{q}^{(2)}\| - V(\|q^{(1)} - q^{(2)}\|). \tag{3.31}$$

It is clear that shifting the origin of the coordinate system by a constant vector leaves the Lagrangian unchanged since neither the velocities nor the relative distance are affected by such a shift. Consider then the infinitesimal variation: $\delta q = \epsilon q_0$ where $q_0 \in \mathbb{R}^6$ is constant. Since this leaves the Lagrangian unchanged, we have $\mathcal{L}(q + \delta q, \dot{q}) = \mathcal{L}(q, \dot{q})$ and this means in turn that $\delta\mathcal{L}(q, \dot{q}) = \mathcal{L}(q + \delta q, \dot{q}) - \mathcal{L}(q, \dot{q}) = 0$. The variation of the action thus vanishes identically over any given time interval and so we find

$$0 = \delta S[q] = \epsilon \int_{t_0}^{t_1} \mathrm{d}s \left\{ -\frac{\mathrm{d}}{\mathrm{d}s}\left(\frac{\partial\mathcal{L}(q,\dot{q})}{\partial\dot{q}}\right) + \frac{\partial\mathcal{L}(q,\dot{q})}{\partial q} \right\} q_0 + \epsilon \left[\frac{\partial\mathcal{L}(q,\dot{q})}{\partial\dot{q}} q_0 \right]_{t_0}^{t_1}. \tag{3.32}$$

Since the integrand vanishes identically along the physical trajectory by virtue of the Euler-Lagrange equations (3.17), we are left with the boundary term which reveals the symmetry

$$\left.\frac{\partial\mathcal{L}(q,\dot{q})}{\partial\dot{q}} q_0\right|_{t_0} = \left.\frac{\partial\mathcal{L}(q,\dot{q})}{\partial\dot{q}} q_0\right|_{t_1}, \tag{3.33}$$

which means that the six components of the one-form $p = \partial\mathcal{L}/\partial\dot{q}$ are *constant*, since there are 6 independent choices for q_0, all of which yielding the same symmetry.

As a second illustrative example of the same phenomenon, consider the infinitesimal generators of orthogonal transformations in $\mathbb{R}^3$. By this we mean 3×3 matrices ξ_i such that for a vector $x \in \mathbb{R}^3$, the transformation

$$y = (I_3 + \epsilon\xi_i)x, \tag{3.34}$$

is orthonormal up to order $O(\epsilon^2)$. Simple algebra reveals that the generators ξ_i must be antisymmetric, yielding the basis

$$\xi_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}, \quad \xi_2 = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad \xi_3 = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{3.35}$$

For those versed in Lie groups (introductions can be found in any of [193, 248, 104]), we are now dealing with the special orthogonal group $SO(3)$ which is the group of orthonormal transformations on $\mathbb{R}^3$, which can be represented by the

group of 3×3 orthonormal matrices with unit determinants. This is an example of a continuous group of symmetries as the elements of $SO(3)$ actually form a differentiable manifold of dimension 3. As is well known [137], the infinitesimal transformations of a given Lie group can be finitely generated by linearly independent elements $\xi_i, i = 1, 2, \ldots$, which form a Lie algebra $\mathfrak{g}$. We need not be concerned here with the exact details, but only that to each generator ξ_i there corresponds a conserved scalar.

Now, the simple Lagrangian $\mathcal{L}(q, \dot{q}) = (1/2)\|\dot{q}\| - V(\|q\|)$ is invariant under the action of $SO(3)$ since these define isometries and thus preserve the norm of any vector. Thus, computing the variations due to an infinitesimal rotation $I_3 + \epsilon\xi_i$ for each of the generators ξ_i and performing the same computation as in (3.32) yields the symmetry

$$\frac{\partial \mathcal{L}(q, \dot{q})}{\partial \dot{q}} \xi_i(q(t)) \bigg|_{t_0} = \frac{\partial \mathcal{L}(q, \dot{q})}{\partial \dot{q}} \xi_i(q(t)) \bigg|_{t_1}, \tag{3.36}$$

and after evaluating this for each index $i = 1, 2, 3$, we recover the conserved vector

$$\begin{bmatrix} J_1 \\ J_2 \\ J_3 \end{bmatrix} = \begin{bmatrix} \dot{q}_2 q_3 - \dot{q}_3 q_2 \\ -\dot{q}_1 q_3 + \dot{q}_3 q_1 \\ \dot{q}_1 q_2 - \dot{q}_2 q_1 \end{bmatrix}, \tag{3.37}$$

which is the angular momentum about the origin: $l = q \times \dot{q}$, where the operator $\times$ is the *cross product* in three dimensions.

These two cases cover the essential content of Noether's theorem which we now enunciate. This is stated in an inordinately complicated field theoretic framework in [105], but with great lucidity in [22], and rigorous precision in [196]. The presentation here follows that found in the latter.

Consider a Lie group G acting on the configuration manifold Q, with associated Lie algebra $\mathfrak{g}$ and generators $\xi_i \in \mathfrak{g}$, so that $\xi_i : Q \mapsto Q$. Assuming that the action of the group on the elements of Q leave a given Lagrange function, $\mathcal{L} : TQ \mapsto R$, strictly invariant, then, to each generator ξ_i there will come a scalar which is called the *conserved current*, by following the procedure of (3.32) yielding boundary terms of the form (3.33) or (3.36) in general. We now state this fact formally.

Theorem 3.1. *[Noether's theorem] Given a configuration manifold Q, a Lagrangian function $\mathcal{L} : TQ \mapsto \mathbb{R}$, a Lie group G and its Lie algebra $\mathfrak{g}$ with generators $\xi_i \in \mathfrak{g}$, where $\xi_i : Q \mapsto Q$. If the Lagrangian $\mathcal{L}(q, \dot{q})$ is invariant under the (left or right) action of the elements of the group G, then, along any the trajectory $(q(t), \dot{q}(t))$ satisfying the Euler-Lagrange equations of motion, the following differential forms are constant*

$$J_i = \frac{\partial \mathcal{L}}{\partial \dot{q}} \xi_i(q). \tag{3.38}$$

The forms J_i are called the conserved currents.

Proof. It suffices to put $\delta q = \epsilon \xi_i(q)$ in (3.16). Since the $\xi_i(q)$ are the generators of G, $\tilde{q} = q + \delta q$ is a transformation which belongs to G and which leaves $\mathcal{L}$ invariant to $O(\epsilon)$ and therefore, $\delta S = O(\epsilon^2)$ for this choice of δq. Thus, when computing the variation of the action, the integrand vanishes identically on the interval $[t_0, t_1]$ which leaves only the boundary terms. Since these must also vanish given that the total variation is indeed null, equation (3.38) is recovered after canceling out the scalar ϵ and the result is proven. □

The discrete case follows similar logic. Consider first a Lie group G with action $\Phi : G \mapsto Q$, parametrized as Φ_g for each $g \in G$. The associated Lie algebra is $\mathfrak{g}$ and this has generators $\xi_i, i = 1, 2, \ldots$. If our discrete Lagrangian is invariant under the action of G so that $\mathbb{L}_d(\Phi_g(q_0), \Phi_g(q_1), h) = \mathbb{L}_d(q_0, q_1, h)$ for all elements $g \in G$, then, considering an infinitesimal element of the form $g_\epsilon = \mathsf{id} + \epsilon \xi_i$, where id denotes the identity map, the change in the coordinates is then: $\mathbf{d}q_k = \epsilon \xi_i$ and the change in the discrete action vanishes to first order in ϵ so

$$
\begin{aligned}
0 = \mathbf{d}\mathbb{S}_d(q_0, \ldots, q_N, h) = \mathbf{d} \sum_{k=0}^{N-1} \mathbb{L}_d(q_k, q_{k+1}, h) = \\
\sum_{k=1}^{N-2} \left[D_1 \mathbb{L}_d(q_k, q_{k+1}, h) + D_2 \mathbb{L}_d(q_{k-1}, q_k, h) \right] \mathbf{d}q_k \\
+ D_2 \mathbb{L}_d(q_{N-1}, q_N, h)\, \mathbf{d}q_N + D_1 \mathbb{L}_d(q_0, q_1, h)\, \mathbf{d}q_0 \\
= \epsilon \left[D_2 \mathbb{L}_d(q_{N-1}, q_N, h) + \epsilon D_1 \mathbb{L}_d(q_0, q_1, h) \right] \xi_i = 0, \quad (3.39)
\end{aligned}
$$

where the summand on the second line vanishes by virtue of the discrete Euler-Lagrange equations (3.22). This identity is not as simple to interpret as the conserved currents J_i of Theorem 3.1. However, following [196] and introducing the definitions

$$
\begin{aligned}
J^+_{\mathbb{L}_d}(q_0, q_1) &= D_2 \mathbb{L}_d(q_0, q_1, h) \xi_i(q_1), \\
J^-_{\mathbb{L}_d}(q_0, q_1) &= -D_1 \mathbb{L}_d(q_0, q_1, h) \xi_i(q_0), \\
J^\Delta_{\mathbb{L}_d}(q_0, q_1) &= J^+_{\mathbb{L}_d}(q_0, q_1) - J^-_{\mathbb{L}_d}(q_0, q_1),
\end{aligned} \quad (3.40)
$$

the following evolution laws are readily deduced from (3.39)

$$
\begin{aligned}
J^+_{\mathbb{L}_d}(q_k, q_{k+1}) &= J^-_{\mathbb{L}_d}(q_{k-1}, q_k), \\
J^+_{\mathbb{L}_d}(q_k, q_{k+1}) &= J^+_{\mathbb{L}_d}(q_{k-1}, q_k) - J^\Delta_{\mathbb{L}_d}(q_{k-1}, q_k), \\
J^-_{\mathbb{L}_d}(q_k, q_{k+1}) &= J^-_{\mathbb{L}_d}(q_{k-1}, q_k) + J^\Delta_{\mathbb{L}_d}(q_{k-1}, q_k).
\end{aligned} \quad (3.41)
$$

Therefore, given an invariant Lagrangian, we have $J^\Delta_{\mathbb{L}_d} = 0$ and thus, the two currents $J^+_{\mathbb{L}_d}, J^-_{\mathbb{L}_d}$ are conserved and are equal. This result is formally summarized in the following theorem which is also found in [196] up to some changes in notation.

Theorem 3.2 (The discrete Noether Theorem). *Given a configuration manifold Q and a Lie group G acting on Q with (left or right) action $\Phi : G \times Q \mapsto$*

Q so that $q \mapsto \Phi_g(q)$ for $g \in G$. Let the associated Lie algebra $\mathfrak{g}$ have generators ξ_i with infinitesimal action $\xi_i : q \mapsto q + \epsilon\xi_i(q), \epsilon \in \mathbb{R}$. If the discrete Lagrangian $\mathbb{L}_d(q_0, q_1, h)$ is invariant under the action of Φ, then, the following currents are preserved along the flow $q_k, k = 0, 1, \ldots$, satisfying the discrete Euler-Lagrange equation (3.22)

$$\begin{aligned} J^+_{\mathbb{L}_d}(q_k, q_{k+1}) &= D_2\mathbb{L}_d(q_k, q_{k+1}, h)\xi_i(q_{k+1}), \\ J^-_{\mathbb{L}_d}(q_k, q_{k+1}) &= -D_1\mathbb{L}_d(q_k, q_{k+1}, h)\xi_i(q_k), \end{aligned} \tag{3.42}$$

and in addition

$$J^+_{\mathbb{L}_d}(q_k, q_{k+1}) = J^-_{\mathbb{L}_d}(q_k, q_{k+1}). \tag{3.43}$$

To get a better understanding of this, consider a two body problem so that the configuration is $q = (q^{(1)}, q^{(2)})$, where $q^{(i)} \in \mathbb{R}^3$. Define a central force potential and discrete Lagrangian as

$$\begin{aligned} V(q) = U(r) &= U(q^{(1)} - q^{(2)}) \\ \mathbb{L}_d(q_0, q_1, h) &= \frac{1}{2h}(q_1 - q_0)^T M (q_1 - q_0) - hV(q_0), \end{aligned} \tag{3.44}$$

with constant 6×6 diagonal mass matrix M and time step $h > 0$. This is invariant under translations $q^{(j)} \mapsto q^{(j)} + \epsilon\xi_i, j = 1, 2$ and $i = 1, 2, 3$, where the generator of translation are defined as

$$\xi_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad \xi_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad \xi_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}. \tag{3.45}$$

From the definition of the separation vector r and the fact that $\partial r/\partial q^{(1)} = -\partial r/\partial q^{(2)}$, we have

$$\nabla_q V(q) = \begin{bmatrix} \nabla_r U(r) \\ -\nabla_r U(r) \end{bmatrix}, \tag{3.46}$$

so that $\nabla_q V(q) \cdot \xi_i = 0, i = 1, 2, 3$. Concatenating the three currents in a vector, we find

$$J^+ = J^- = h^{-1}M^{(1)}(q_1^{(1)} - q_0^{(1)}) + h^{-1}M^{(2)}(q_1^{(2)} - q_0^{(2)}), \tag{3.47}$$

which is the standard definition of linear momentum.

3.9 Extended variational principle

The computation of first order variations stated in d'Alembert's principle and considered in Section 3.5 above neglected the time re-parametrization. As shown

in Section 3.10, the variation of the time variable plays an important role in analyzing the energy of a system. This warrants the present computation of a more general form of infinitesimal transformations and corresponding first order variations.

Consider the real variable $t \in \mathbb{R}$, the real function $f : \mathbb{R} \mapsto \mathbb{R}$ and the definite integral $\mathcal{I}(t_0, t_1) = \int_{t_0}^{t_1} \mathrm{d}t f(t)$. The change of variable theorem states that for a function $\phi : \mathbb{R} \mapsto \mathbb{R}$ which is 1-1 on the interval $[\phi(s_0), \phi(s_1)] \mapsto [y_0, y_1]$, then

$$\mathcal{I}(t_0, t_1) = \int_{t_0}^{t_1} \mathrm{d}y f(t) = \int_{\phi^{-1}(t_0)}^{\phi^{-1}(t_1)} \mathrm{d}s \dot{\phi}(s) f(\phi(s)), \tag{3.48}$$

where $\phi^{-1}(t)$ is the inverse function and $\dot{\phi}(s) = \mathrm{d}\phi(s)/\,\mathrm{d}s$. Now, if we apply this to a composition function of the form $f(t) = \mathcal{L}(q(t), \dot{q}(t), t)$, where $q : \mathbb{R} \mapsto Q$ is $\mathcal{C}^1$ continuous and where $\dot{q}(t) = \mathrm{d}q(t)/\,\mathrm{d}t$, then, we must first transform the functions $q, \dot{q}$ as follows

$$\begin{aligned} r(s) &= q(\phi(s)) = q(t), \\ \dot{q}(t) &= \frac{\mathrm{d}q(\phi(s))}{\mathrm{d}s}\frac{\mathrm{d}s}{\mathrm{d}t} = \dot{r}(s)\frac{1}{\dot{\phi}(s)}, \end{aligned} \tag{3.49}$$

where the implicit function theorem was used to substitute $\mathrm{d}s/\,\mathrm{d}t = 1/\dot{\phi}(s)$. Therefore, for the case at hand, the change of variables theorem reads:

$$\int_{\phi^{-1}(t_0)}^{\phi^{-1}(t_1)} \mathrm{d}s \dot{\phi}(s) \mathcal{L}\left(r(s), \frac{\dot{r}(s)}{\dot{\phi}(s)}, \phi(s)\right) = \int_{t_0}^{t_1} \mathrm{d}t \mathcal{L}(q(t), \dot{q}(t), t). \tag{3.50}$$

Define the generalized variation as follows

$$\begin{aligned} t = \phi(s) &= s + \epsilon\theta(s), \\ \tilde{r}(s) = \tilde{q}(t) &= r(s) + \epsilon\eta(s) = q(t) + \epsilon\eta(\phi^{-1}(t)), \\ \dot{\tilde{q}}(t) &= \frac{1}{\dot{\phi}(s)}\left(\dot{r}(s) + \epsilon\dot{\eta}(s)\right), \end{aligned} \tag{3.51}$$

where we assume that η and θ are uniformly bounded $\mathcal{C}^1$ functions of time $s \in \mathbb{R}$, that $\theta(s_0) = \theta(s_1) = 0$, and finally that $\eta(s_0) = \eta(s_1) = 0$. Assuming that $\epsilon \in \mathbb{R}$ is small and expanding the integrand to first order yields

$$\begin{aligned} \mathcal{L}(q, \dot{q})\left(\tilde{r}(s), \frac{\dot{\tilde{r}}(s)}{\dot{\phi}(s)}, \phi(s)\right) &= \mathcal{L}(q, \dot{q})(\tilde{r}(s), \dot{\tilde{r}}(s), s) \\ &\quad - \epsilon\frac{\partial\mathcal{L}(q(s), \dot{q}(s), s)}{\partial\dot{q}(s)}\dot{q}(s)\dot{\theta}(s) + \epsilon\frac{\partial\mathcal{L}(q(s), \dot{q}(s), s)}{\partial s}\theta(s), \end{aligned} \tag{3.52}$$

and finally

$$\begin{aligned} &\mathcal{L}(\tilde{r}(s), \dot{\tilde{r}}(s), s) = \\ &\qquad \mathcal{L}(q(s), \dot{q}(s), s) + \epsilon\frac{\partial\mathcal{L}(q(s), \dot{q}(s), s)}{\partial q(s)}\eta + \epsilon\frac{\partial\mathcal{L}(q(s), \dot{q}(s), s)}{\partial\dot{q}(s)}\dot{\eta}. \end{aligned} \tag{3.53}$$

Noting now from the first line of (3.51) that $\mathrm{d}s\dot{\phi}(s) = \mathrm{d}s(1+\epsilon\dot{\theta}(s))$, multiplying this with the terms found in (3.52) and (3.53), discarding all terms of order $O(\epsilon^2)$ and higher, and integrating the term containing θ' and η' by parts, the following simplification is found

$$\begin{aligned}\int_{t_0}^{t_1} \mathrm{d}s\dot{\phi}(s)\mathcal{L}(\tilde{r}(s),\dot{\tilde{r}}(s),s) &= \int_{t_0}^{t_1} \mathrm{d}s\mathcal{L}(q(s),\dot{q}(s),s) \\ &+ \epsilon\int_{t_0}^{t_1} \mathrm{d}s\left(\frac{\partial\mathcal{L}(q,\dot{q},s)}{\partial s} - \frac{\mathrm{d}}{\mathrm{d}s}\left\{\mathcal{L}(q,\dot{q},s) - \frac{\partial\mathcal{L}(q,\dot{q},s)}{\partial\dot{q}}\dot{q}\right\}\right)\theta(s) \\ &+ \epsilon\left\{\mathcal{L}(q,\dot{q},s) - \frac{\partial\mathcal{L}(q,\dot{q},s)}{\partial\dot{q}}\dot{q}\theta(s)\right\}\Bigg|_{t_0}^{t_1} \\ &+ \epsilon\int_{t_0}^{t_1} \mathrm{d}s\left\{\frac{\partial\mathcal{L}(q,\dot{q},s)}{\partial q} - \frac{\mathrm{d}}{\mathrm{d}s}\left(\frac{\partial\mathcal{L}(q,\dot{q},s)}{\partial\dot{q}}\right)\right\}\eta(s) \\ &+ \epsilon\left(\frac{\partial\mathcal{L}(q,\dot{q},s)}{\partial\dot{q}}\right)\eta(s)\Bigg|_{t_0}^{t_1} \\ &= \int_{t_0}^{t_1} \mathrm{d}t\mathcal{L}(q,\dot{q},t) + O(\epsilon). \quad (3.54)\end{aligned}$$

The last line of this equation is just a relabeling of the variable s and therefore, the integral is stationary if all the terms proportional to ϵ vanish. This reproduces the previously derived Euler-Lagrange equations of motion (3.17) in addition to a new equation, namely

$$\frac{\partial\mathcal{L}(q,\dot{q},t)}{\partial t} - \frac{\mathrm{d}}{\mathrm{d}t}\left\{\mathcal{L}(q,\dot{q},t) - \frac{\partial\mathcal{L}(q,\dot{q},t)}{\partial\dot{q}}\dot{q}\right\} = 0. \quad (3.55)$$

In the case where $\partial\mathcal{L}/\partial t = 0$, this last equation contains the definition of the energy of the system which is time invariant, namely

$$E(q,\dot{q},t) = \frac{\partial\mathcal{L}(q,\dot{q},t)}{\partial\dot{q}}\dot{q} - \mathcal{L}(q,\dot{q},t). \quad (3.56)$$

But in case $\partial\mathcal{L}/\partial t = 0$, the equation adds nothing new as is easily seen by expanding the total time derivative in (3.55) and applying the Euler-Lagrange equations of motion (3.17). Indeed, energy conservation is already implied in the Euler-Lagrange equations of motion. In the discrete case however, energy conservation is no longer implied by the basic discrete Euler-Lagrange equations of motion (3.22). The significance of this is investigated next in Section 3.10.

3.10 Variation of time and energy conservation.

So far, we have seen that spatial translation invariance leads to conservation of momentum and that rotational invariance leads to angular momentum. Since

the Lagrangians considered so far do not explicitly depend on time we should be able to exploit time translation invariance—labeling of the origin of time—to produce another symmetry, namely, energy.

This symmetry can be discovered by using the extended variations of Section 3.9 but a simpler approach is now used as is customary, (see in [173] and in [150] or [87]) for instance).

Indeed, consider applying an infinitesimal time translation to the generalized coordinates $q(t)$ with the variation $\delta q = \epsilon\eta(t)\dot{q}(t)$, where $\eta(t_0) = \eta(t_1) = 0, d\eta/dt > 0$, and $\epsilon > 0$ is small. Variations in the function $\eta(t)$ thus create local distortion of this small time shift. Using this form of variation δ leads to the following form for the variation of the action

$$\delta S = \epsilon \int_{t_0}^{t_1} \mathrm{d}s\eta \left\{ -\frac{\mathrm{d}}{\mathrm{d}s}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}}\right) + \frac{\partial \mathcal{L}}{\partial q} \right\} \dot{q} = 0, \tag{3.57}$$

and the integrand is easily verified to be the negative of the total time derivative of the energy $E = (\partial\mathcal{L}/\partial\dot{q})\dot{q} - \mathcal{L}(q, \dot{q})$, introduced earlier in (3.56). This is constant along the physically realized trajectory $(q, \dot{q})$ as long as the Lagrangian does not depend explicitly on time, as demonstrated previously in Section 3.9.

For cases where the kinetic energy $T(q, \dot{q})$ is a homogeneous function of $\dot{q}$ with degree k and the potential energy $V(q)$ is independent of $\dot{q}$, then, $\partial\mathcal{L}/\partial\dot{q} = kT(q, \dot{q})$ and so $E = (k-1)T(q, \dot{q}) + V(q)$. For the most common case where $T(q, \dot{q}) = \frac{1}{2}\dot{q}^T M(q)\dot{q}$, we have $k = 2$ and in this case, $E = T(q, \dot{q}) + V(q)$. This is not too surprising since we started the presentation by introducing $T(q, \dot{q})$ and $V(q)$ as the kinetic and potential energy, respectively. We now find that the total energy of the system is simple the sum of these two terms. For the discretized case, we start by generalizing the definition of the discretized Lagrangian to include time

$$\mathbb{L}_d(q_0, t_0, q_1, t_1) = \int_{t_0}^{t_1} \mathrm{d}s\mathcal{L}(q(s), \dot{q}(s), s), \tag{3.58}$$

and this now leads to the generalized action which now depends on the choices of times $t_0 < t_1 < t_2 < \ldots < t_N$ as

$$\mathbb{S}_d(q_0, t_0, \ldots, q_N, t_N) = \sum_{k=0}^{N-1} \mathbb{L}_d(q_k, t_k, q_{k+1}, t_{k+1}). \tag{3.59}$$

The minimum of the action over the full set of available parameters leads to the two sets of conditions

$$\begin{aligned} \frac{\partial \mathbb{S}_d(q_0, t_0, \ldots, q_N, t_N)}{\partial q_k} &= D_3\mathbb{L}_d(q_{k-1}, t_{k-1}, q_k, t_k) + D_1\mathbb{L}_d(q_k, t_k, q_{k+1}, t_{k+1}) \\ &= 0 \\ \frac{\partial \mathbb{S}_d(q_0, t_0, \ldots, q_N, t_N)}{\partial t_k} &= D_4\mathbb{L}_d(q_{k-1}, t_{k-1}, q_k, t_k) + D_2\mathbb{L}_d(q_k, t_k, q_{k+1}, t_{k+1}) \\ &= 0, \end{aligned} \tag{3.60}$$

where the differential operators D_j simply mean "differentiation with respect to the jth argument," generalizing the definition provided in (3.23).

Note that the second of these equations—the conservation of energy—is not a direct consequence of the first as was the case in the continuous case but is genuinely a new equation. If we write $h_k = t_k - t_{k-1}$, we find the definition of the discrete energy as

$$\mathbb{E}_d(q_{k-1}, q_k, h_k) = -\frac{\partial \mathbb{L}_d(q_{k-1}, q_k, h_k)}{\partial h_k}. \tag{3.61}$$

This leads to the observation that an energy conserving discrete integrator must use an adaptive time step [150, 206]. However, given that energy fluctuates only within bounds of $O(h^2)$ when ignoring the additional equations (3.60), and given that the main focus in the present thesis is on dissipative systems, this is not considered further.

3.11 Continuous and discrete symplectic flows.

We have already demonstrated in Section 3.8 how several conservation laws can be deduced from the invariance of the Lagrange function under the action of symmetry groups. There is also a fundamental symmetry of the flow, $F_{\mathcal{L}}^t : TQ \times \mathbb{R} \mapsto TQ$, which maps the initial conditions $(q(0), \dot{q}(0))$ to the configuration at time t, $(q(t), \dot{q}(t))$. There is in fact a *differential form* $\Omega_{\mathcal{L}}$ associated with the Lagrange function $\mathcal{L}(q, \dot{q})$ which is preserved by the flow map. Without going deeply into the theory of differential forms or differential geometry, suffice to say that this implies the invariance of certain scalars—integrals of $\Omega_{\mathcal{L}}$—along the trajectory $(q(t), \dot{q}(t)) = F^t_{\mathcal{L}(q(0),\dot{q}(0))}$. We first summarize some basic facts regarding differential forms and proceed to demonstrate that the flow $F_{\mathcal{L}}^t$ defined by the Lagrange functions $\mathcal{L} : TQ \mapsto \mathbb{R}$ preserves a differential two-form and that this is also true for the discrete flow defined by the discrete variational principle. In fact, as we shall see below, the preservation of this two-form along the flow is directly connected to the variational structure defining the flow $F_{\mathcal{L}}^t$.

Following the brief but lucid introductions found in Flanders [90] and in [22], we define exterior differential forms as the integrands of oriented integrals over n-dimensional domains. Simple examples in $\mathbb{R}^3$ include the one-form $\omega^1 = a\,\mathrm{d}x + b\,\mathrm{d}y + c\,\mathrm{d}z$ leading to the line integral $\lambda = \int \omega^1$, the two-form $\omega^2 = p\,\mathrm{d}y\,\mathrm{d}z + q\,\mathrm{d}z\,\mathrm{d}x + r\,\mathrm{d}x\,\mathrm{d}y$ leading to the oriented surface integral $\sigma = \iint \omega^2$, and the three-form $\omega^3 = h\,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z$ leading to the volume integral: $\zeta = \iiint \omega^3$. The scalar functions a, b, c, p, q, r, and h, are all assumed to be smooth integrable mappings of the form $\mathbb{R}^3 \mapsto \mathbb{R}$.

Because orientation is a fundamental aspect of integral forms, the calculus of p-forms involves the *wedge product* $q^{(i)} \wedge q^{(j)}$ which is linear, associative, and *antisymmetric*. Using this, general differential $p-$forms are defined in $\mathbb{R}^n, n \geq p$ as:

$$\omega^p = \frac{1}{p!} \sum a_{i_1, i_2, \ldots, i_p}\, \mathbf{d}q^{(i_1)} \wedge \mathbf{d}q^{(i_2)} \wedge \cdots \wedge \mathbf{d}q^{(i_p)}, \tag{3.62}$$

and $p = \deg \omega^p$ is known as the *degree* of the p-form. Thus defined ω^p is a multilinear mapping: $\omega^p : \wedge^p \mathbb{R}^n \mapsto \mathbb{R}$.

The wedge calculus can be used to define $p+q$-forms by composition $\omega^{p+q} = \omega^p \wedge \omega^q$ given a p-form ω^p and a q-form ω^q.

General p-forms can be differentiated using the exterior derivative operator, $\mathbf{d}$, which is defined via the following axioms (quoted from Flanders [90])

1. linearity: $\mathbf{d}(\omega + \eta) = \mathbf{d}\omega + \mathbf{d}\eta$,
2. distributivity: $\mathbf{d}(\omega \wedge \eta) = \mathbf{d}\omega \wedge \eta + (-)^{\deg \eta} \omega \wedge \mathbf{d}\eta$,
3. the Poincaré lemma (closure): for each differential form ω, such that the coefficients $a_{i_1,i_2,\ldots,i_p}$ of definition (3.62) are $\mathcal{C}^2$ functions, then:
$$\mathbf{d}(\mathbf{d}\omega) = 0, \tag{3.63}$$
4. For each function $f : \mathbb{R}^n \mapsto \mathbb{R}$:
$$\mathbf{d}f = \sum_i \frac{\partial f}{\partial q^{(i)}} \mathbf{d}q^{(i)}. \tag{3.64}$$

Differential forms are useful chiefly because of two fundamental facts which distinguish them sharply from general tensors. First, consider a map $\phi : \mathbb{R}^m \mapsto \mathbb{R}^n$ (in general, between two manifolds $\mathcal{M} \mapsto \mathcal{N}$). Given a p-form $\omega^p : \mathbb{R}^n \mapsto \mathbb{R}$, the composition of these two mappings is: $\omega^p \circ \phi : \mathbb{R}^m \mapsto \mathbb{R}^n$. This composition is called the *naturally induced p-form* and is denoted $\phi^* \omega^p$. The map $\phi^\star$ is called the *pullback* since it maps p-forms ω^p operating on elements of $\mathcal{N}$ back to p-forms $\phi^\star \omega^p$ operating on $\mathcal{M}$, i.e., in the opposite direction of ϕ. The situation is explained in Figure 3.1. For the simple case where manifolds $\mathcal{M}$ and $\mathcal{N}$ have the same dimension and the map ϕ is 1–1, the pullback $\phi^\star$ is the inverse transpose of the Jacobian matrix of ϕ. The Jacobian of the transform corresponds to the *pushforward* map $\phi_\star$ which, in this case, would map p-forms operating on $\mathcal{M}$ to p-forms operating on $\mathcal{N}$. A simple way to understand this picture is to consider the case of a line integral which involves only 1-forms and vectors. This leads to evaluating integrals of scalar products of the form $\sum a_i(x)\, \mathrm{d}x_i$ where the $a_i(x)$ are the components of a 1-form. For these products to be invariant, the transformation of the row vector $a^T = (a_1, a_2, \ldots, a_n)$ must cancel out that of the vector $\mathrm{d}x$. Thus if the differentials transform as $\mathrm{d}\tilde{x} = J\, \mathrm{d}x$, the 1-forms must transform as $\tilde{a} = J^{-T} a$, locally, and this corresponds to the pushforward $\phi_\star$ and the pullback $\phi_\star$, respectively. In physics, this is known as co-variant and contra-variant transformation rules in the context of tensor algebra, especially when dealing with general relativity [249, 248].

When applied to differential forms, mappings ϕ and their naturally induced differential forms have the following important properties:

1. linearity: $\phi^*(\omega + \eta) = \phi^* \omega + \phi^* \eta$,
2. distributivity: $\phi^*(\omega \wedge \eta) = (\omega^* \phi) \wedge (\omega^* \eta)$,

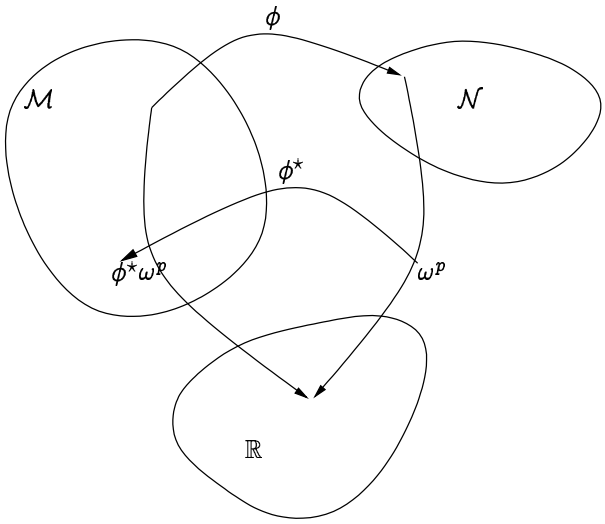

Figure 3.1: Illustration of the naturally induced p-form by a mapping ϕ, the pullback $\phi^\star$.

3. invariance under differentiation: $\mathbf{d}(\phi^*\omega) = \phi^*(\mathbf{d}\omega)$,

4. composition: if $\phi : \mathbb{R}^m \mapsto \mathbb{R}^n$ and $\psi : \mathbb{R}^n \mapsto \mathbb{R}^k$, then, $(\psi \circ \phi)^* = \phi^* \circ \psi^*$ (the dressing undressing principle).

To see the usefulness of this, consider the integral of the differential p-form $\omega^p : \mathbb{R}^n \mapsto \mathbb{R}$ over the domain $D \in \mathbb{R}^n$ and the smooth map $\phi : \mathbb{R}^n \mapsto \mathbb{R}^n$, with preimage ϕ^{-1}. With this notation, the generalization of the change of variables theorem reads

$$\int_D \omega^p = \int_{\phi^{-1}(D)} \phi^*\omega^p. \tag{3.65}$$

Now armed with this arsenal, we look at the definition of the action in (3.15) and the functional derivative in (3.16) to rewrite this as

$$\delta S = \mathbf{d}S\delta q. \tag{3.66}$$

Define the mapping $F^t_{\mathcal{L}} : TQ \mapsto TQ$ so that $(q(0), \dot{q}(0))$ is mapped to the point $(q(t), \dot{q}(t))$ *on the physical trajectory* defined by the Lagrangian $\mathcal{L}(q, \dot{q})$ and the initial conditions. Therefore, the evaluation of the differential one-form $\mathbf{d}S$ on the trajectory $w(t) = (q(t), \dot{q}(t))$ is

$$\mathbf{d}Sw(t) = (\Theta_{\mathcal{L}} \circ F^t_{\mathcal{L}})w(0) - \Theta_{\mathcal{L}}w(0), = \left((F^t_{\mathcal{L}})^*\Theta_{\mathcal{L}}\right)w(0) - \Theta_{\mathcal{L}}w(0), \tag{3.67}$$

where the time interval $[t_0, t_1]$ was relabeled as $[0, t]$, and $\Theta_{\mathcal{L}}$ is defined as the differential one-form

$$\Theta_{\mathcal{L}} = \frac{\partial \mathcal{L}}{\partial \dot{q}}\,\mathbf{d}q. \tag{3.68}$$

Indeed, since the Euler-Lagrange equations are identically satisfied along the physical trajectory $F_{\mathcal{L}}^t(w(0))$, the integrand in (3.16) does not contribute.

Now, by the Poincaré Lemma (3.63), $\mathbf{d}^2 S = 0$ and using the previously stated fact that $\mathbf{d}(\phi^* \omega) = \phi^*(\mathbf{d}\omega)$, we differentiate (3.67) to find

$$(F_{\mathcal{L}}^t)^* \, \mathbf{d}\Theta_{\mathcal{L}} = \mathbf{d}\Theta_{\mathcal{L}}. \tag{3.69}$$

Therefore, defining the differential two-form $\Omega_{\mathcal{L}} = \mathbf{d}\Theta_{\mathcal{L}}$, the following two-form is invariant under the mapping $F_{\mathcal{L}}^t$

$$\Omega_{\mathcal{L}} = \mathbf{d}\Theta_{\mathcal{L}} = \frac{\partial^2 \mathcal{L}}{\partial q^{(j)} \partial \dot{q}^{(i)}} \, \mathbf{d}q^{(j)} \wedge \mathbf{d}q^{(i)} + \frac{\partial^2 \mathcal{L}}{\partial \dot{q}^{(j)} \partial \dot{q}^{(i)}} \, \mathbf{d}\dot{q}^{(j)} \wedge \mathbf{d}q^{(i)}. \tag{3.70}$$

Now, if we pick a domain $D \in TQ$ consisting of a set of initial coordinates, $(q(0), \dot{q}(0)) \in TQ$. Assume that each point in D moves in time according to the same flow map $F_{\mathcal{L}}^t$. The fact that the two-form $\Omega_{\mathcal{L}}$ is invariant under the action of $F_{\mathcal{L}}^t$ means we have the following scalar invariant of the flow

$$\int_{(F_{\mathcal{L}}^t)^{-1}(D)} \Omega_{\mathcal{L}} = \int_D \Omega_{\mathcal{L}} = A \in \mathbb{R}. \tag{3.71}$$

One needs the theory of symplectic manifolds and Hamiltonian flows to analyze the meaning of this further. It suffices to say here that symplectic invariance of the flow implies the invariance of the Hamiltonian function $H(p, q)$ which is a Legendre transform of the Lagrangian. For conservative systems, the Hamiltonian is energy of the system expressed in the Legendre coordinates (p, q), with $p = \partial \mathcal{L} / \partial \dot{q}^T$, and the symplecticity of the flow is equivalent to the conservation of energy.

The discretized version of this result proceeds in the same way. We first define the mapping

$$\begin{aligned} \Phi_{\mathbb{L}_d} : Q \times Q &\mapsto Q \times Q, \\ \Phi_{\mathbb{L}_d}(q_0, q_1) &= (q_1, q_2), \end{aligned} \tag{3.72}$$

which is implicitly defined by the discrete Euler-Lagrange equations established from (3.22). Now, evaluating the one form $\mathbf{d}\mathbb{S}_d(q_0, \ldots, q_N, h)$ on a trajectory $w_k = (q_k, q_{k+1})$, we have

$$\begin{aligned} \mathbf{d}\mathbb{S}_d(q_0, \ldots, q_N, h) = & \\ & \sum_{k=1}^{N-2} \left[D_1 \mathbb{L}_d(q_k, q_{k+1}, h) + D_2 \mathbb{L}_d(q_k, q_{k+1}, h) \right] \mathbf{d}q_k \\ & + \frac{\partial \mathbb{L}_d(q_0, q_1, h)}{\partial q_0} \, \mathbf{d}q_0 + \frac{\partial \mathbb{L}_d(q_{N-1}, q_N, h)}{\partial q_N} \, \mathbf{d}q_N \\ & = \left(\Theta_{\mathbb{L}_d}^{(+)} \circ \Phi_{\mathbb{L}_d}^{N-1} \right) \cdot w_0 - \Theta_{\mathbb{L}_d}^{(-)} \cdot w_0, \end{aligned} \tag{3.73}$$

with the definitions

$$\begin{aligned}\Theta^{(+)}_{\mathbb{L}_d}(q_0,q_1) &= D_2\mathbb{L}_d(q_0,q_1,h)\,\mathbf{d}q_1 = \frac{\partial \mathbb{L}_d(q_0,q_1,h)}{\partial {q_1}^{(i)}}\,\mathbf{d}{q_1}^{(i)},\\ \Theta^{(-)}_{\mathbb{L}_d}(q_0,q_1) &= -D_1\mathbb{L}_d(q_0,q_1,h)\,\mathbf{d}q_0 = \frac{\partial \mathbb{L}_d(q_0,q_1,h)}{\partial {q_0}^{(i)}}\,\mathbf{d}{q_0}^{(i)},\end{aligned} \tag{3.74}$$

and after observing that the summand in (3.73) vanishes on the trajectory. Note that the discrete Euler-Lagrange equations imply that $\Theta^{(+)}_{\mathbb{L}_d}(q_{k-1},q_k) = \Theta^{(-)}_{\mathbb{L}_d}(q_k,q_{k+1})$. Also, since $\mathbf{d}\mathbb{L}_d = \Theta^{(+)}_{\mathbb{L}_d} - \Theta^{(-)}_{\mathbb{L}_d}$ and since $\mathbf{d}^2\mathbb{L}_d = 0$, we have

$$\begin{aligned}\mathbf{d}\Theta^{(+)}_{\mathbb{L}_d}(q_0,q_1) &= \mathbf{d}\Theta^{(-)}_{\mathbb{L}_d}(q_0,q_1) = \Omega_{\mathbb{L}_d}(q_0,q_1)\\ &= \frac{\partial^2 \mathbb{L}_d(q_0,q_1,h)}{\partial {q_0}^{(i)}\partial {q_1}^{(j)}}\,\mathbf{d}{q_0}^{(i)}\wedge\mathbf{d}{q_1}^{(j)}.\end{aligned} \tag{3.75}$$

Using the Poincaré Lemma (3.63) on (3.73), we find that

$$(\Phi^{\star}_{\mathbb{L}_d})^{N-1}\,\mathbf{d}\Theta^{(+)}_{\mathbb{L}_d} = \mathbf{d}\Theta^{(-)}_{\mathbb{L}_d} = \Omega_{\mathbb{L}_d}. \tag{3.76}$$

In other words, the time sequences produced by solving the discrete Euler-Lagrange equations (3.22) preserve the two-form $\Omega_{\mathbb{L}_d}$, just as in the continuous case. And this is true for *absolutely any* choice of discretization of the Lagrangian, irrespective of the order of the approximation of $\mathbb{L}_d$. As long as we solve the stepping equations (3.22) accurately enough, the two-form $\Omega_{\mathbb{L}_d}$ is preserved by the flow.

To grasp the meaning of this identity, remember that the initial conditions for the discrete variational stepper are given by a pair of points, (q_0,q_1). Consider a domain of initial conditions $D_0 \in Q\times Q$. Choose D_0 to have small area so that all points in it are somewhat close to each other initially. Then, each point $(q_0,q_1)\in D$ moves according to the map $\Phi^k_{\mathbb{L}_d}(q_0,q_1)$ to the set D_k, and the invariance theorem says that D_k has the same surface area as D_0, i.e., $A(D_k) = A(D_0)$. Thus, a cluster of initial conditions cannot spread too far from each other, unless of course there is a collapse along some of the dimensions, allowing escape towards infinity along the others. Barring this pathological case, this observation means that small local errors do not accumulate in variational integration methods. In particular, making a small error at one stage does not lead to an exponential divergence in the long run, unless similar errors are made at each step.

This can be illustrated in two dimension using the simple harmonic oscillator introduced in Chapter 2. The domain of initial conditions is chosen as the contour of the face of a stylized cat. The choice of the face of a cat to illustrate symplecticity goes back to Arnol'd [22]. Points (q_0,q_1) lying on the outline of the face of the cat at time $t=0$ are used as initial conditions for the simple harmonic oscillator which is simulated using a numerical integration method such as the four described in Chapter 2 or any other one. Each point on the contour is then propagated in time and should at least approximately follow the ellipse $E = (1/2)\dot{q}^2 + (1/2)\omega^2 q^2$ in phase space as shown in Figures 2.1 2.2, and 2.3 in Chapter 2

The outline at time $t = kh$ consists of points with coordinates $(q_k, h^{-1}(q_{k+1} - q_k))$ which are the image under the a given integration method. All told, each picture involves several hundreds of integrations over a given time interval, each starting at different initial conditions $(q_0, h^{-1}(q_1 - q_0))$ on the outline.

In Figures 3.2, 3.3, and 3.4, and 3.5, we present data from the Verlet, implicit midpoint, second order Runge Kutta (explicit midpoint) and implicit first order Euler, respectively, for low values of frequency, namely, using $h\omega = 0.2$, where h is the time step, and $\omega = \sqrt{k}$ is the natural frequency, k being the spring constant. The scales on each sub-figure are identical but varies from figure to figure. The plot key denotes the time of each snapshot.

These pictures illustrate that the surface area of the cat is preserved for the variational methods, namely, the Verlet and implicit midpoint methods as seen from Figure 3.2 and Figure 3.3. However, the surface area of the face of the cat area obviously increases for the explicit midpoint method in Figure 3.4, and quickly shrinks for the Euler implicit method in Figure 3.5.

A similar series of figures illustrates what happens when the frequency $h\omega$ is higher. Both variational methods still preserve the surface area of the cat as seen in Figures 3.6 and 3.7. However, the explicit Runge-Kutta method of second order explodes the area faster in Figure 3.8, whilst the implicit Euler method shrinks it faster in Figure 3.9.

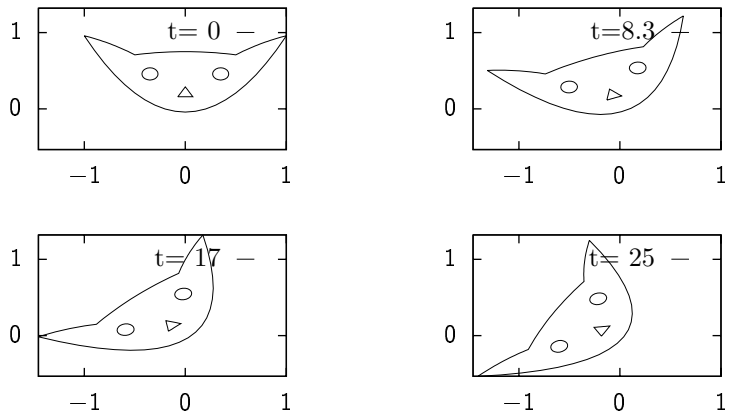

Figure 3.2: Phase portrait of simple harmonic oscillator with frequency $h\omega = 1/5$ using Verlet integration.

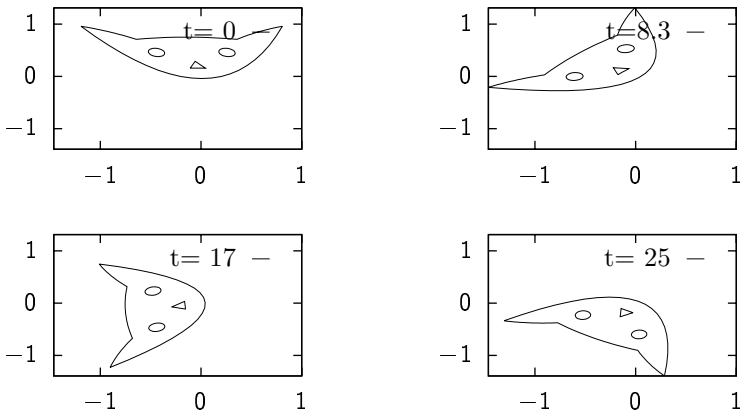

Figure 3.3: Phase portrait of simple harmonic oscillator with frequency $h\omega = 1/5$ using implicit midpoint integration.

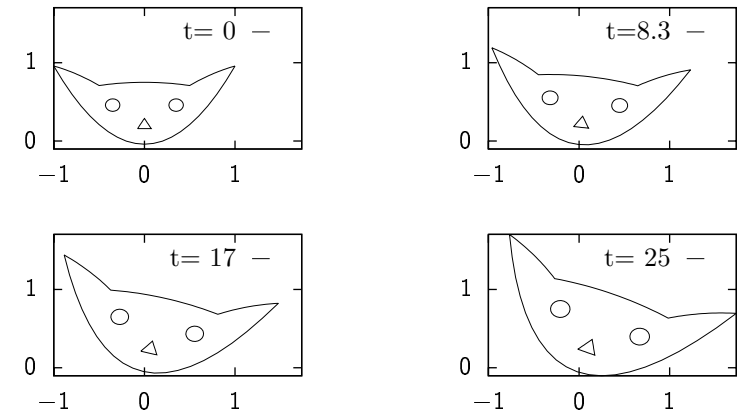

Figure 3.4: Phase portrait of simple harmonic oscillator with frequency $h\omega = 1/5$ using explicit midpoint method integration (Runge-Kutta second order).

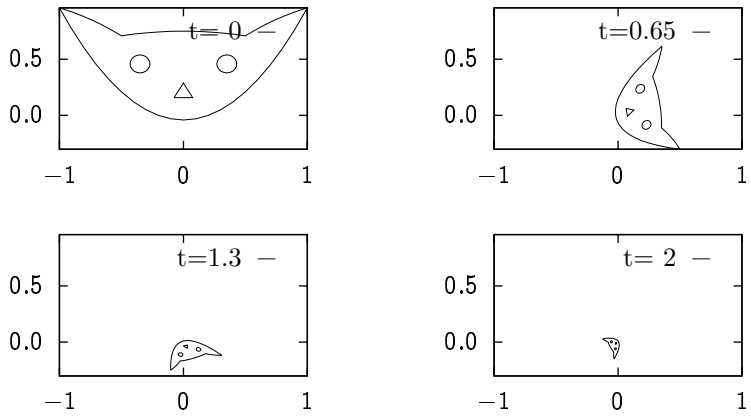

Figure 3.5: Phase portrait of implicit first order Euler integration applied to the simple harmonic oscillator with frequency $h\omega = 1/5$.

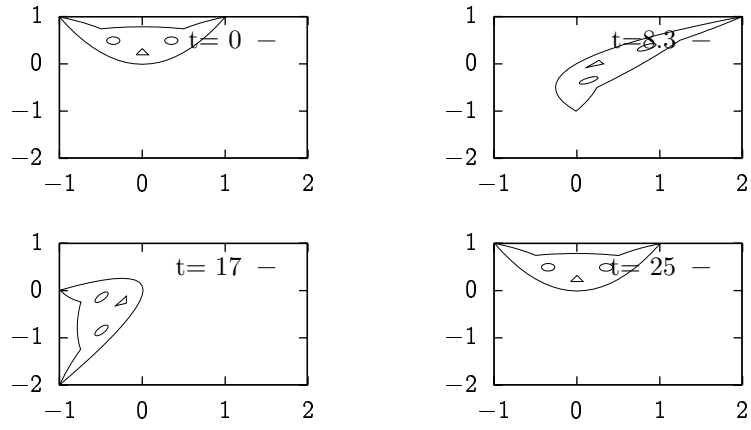

Figure 3.6: Phase portrait of Verlet integration applied to the simple harmonic oscillator with frequency $h\omega = 1$.

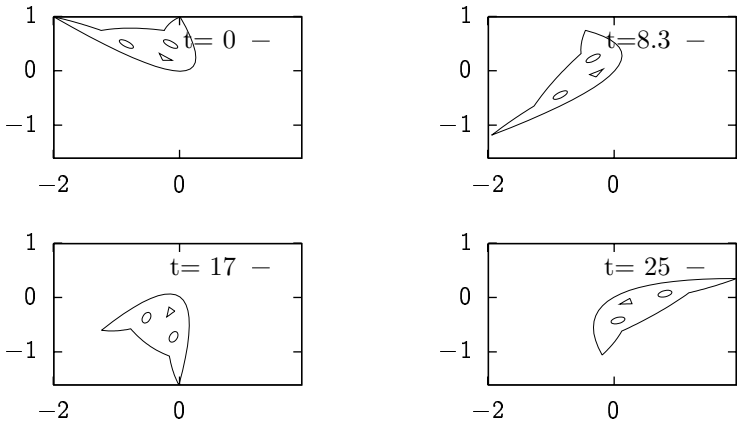

Figure 3.7: History of the phase portrait of the Arnol'd cat initial conditions for the implicit midpoint integrator applied to the simple harmonic oscillator with frequency $h\omega = 1$.

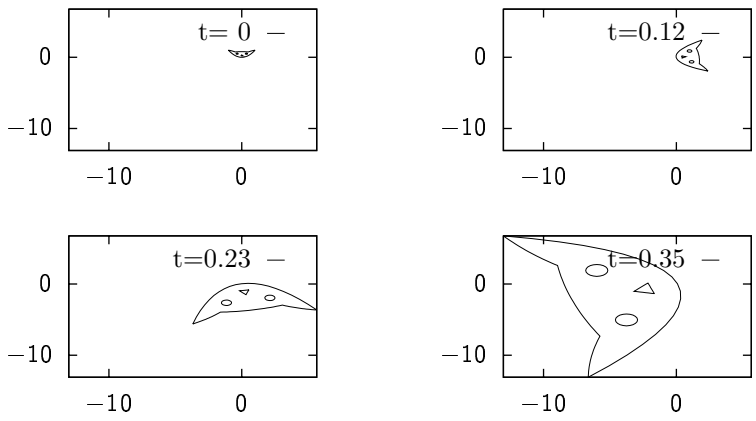

Figure 3.8: History of the phase portrait of the Arnol'd cat initial conditions for the explicit midpoint integrator (Runge Kutta second order) applied to the simple harmonic oscillator with frequency $h\omega = 1$.

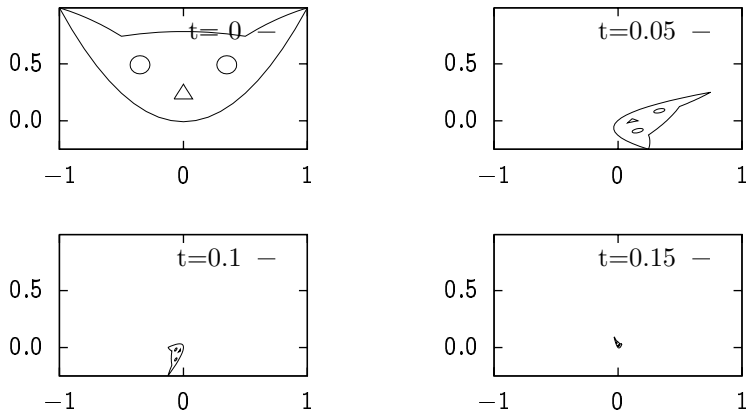

Figure 3.9: History of the phase portrait of the Arnol'd cat initial conditions for the implicit Euler first order integrator applied to the simple harmonic oscillator with frequency $\hbar\omega = 1$.

3.12 Forced and dissipative systems

Mechanical systems in which all forces can be derived from potential energy function, $V(q) : Q \mapsto \mathbb{R}$, are called *conservative* as they preserve energy on the trajectory, as shown in Section 3.9 and Section 3.10. We now extend the analysis to cover forced and dissipative systems as well. These are the systems for which the forces are not the gradient of potential functions and, thus, they do not necessarily conserve energy.

Assume that a mechanical system is subjected to forcing functions $f(q, \dot{q})$ which cannot be derived from a potential function $V(q)$ (or even $V(q, \dot{q})$ for the general case). Such forces are also known as *polygenic* forces in contrast with *monogenic* ones which can be derived from a potential function. D'Alembert's principle [173] states that any variation δq of the physical trajectory satisfying the boundary conditions $\delta q(t_0) = \delta q(t_1) = 0$ should produce zero net variation of the following functional

$$\delta \int_{t_0}^{t_1} \mathrm{d}s \mathcal{L}(q(s), \dot{q}(s)) + \int_{t_0}^{t_1} \mathrm{d}s f^T \delta q = 0, \tag{3.77}$$

where the combination $x^T y$ is the ordinary inner or dot product of any two n-dimensional vectors x, y.

The Euler-Lagrange equations of motion for this system are readily recovered noting that δq is arbitrary, yielding the following system after transposition

$$\frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial \mathcal{L}}{\partial \dot{q}^T} - \frac{\partial \mathcal{L}}{\partial q^T} = f(q, \dot{q}). \tag{3.78}$$

To see that this is consistent with the case of conservative forces case introduced in (3.11), consider $f(q,\dot{q}) = -\partial U(q)\partial/q^T$ and therefore, the second integrand reads $f^T\delta q = -(\partial U/\partial q)\delta q = -\delta U(q)$ so terms can be collected as $\delta[\mathcal{L} - U]$ in the first integral. After updating the definition of the potential function, $V(q) + U(q)$, we recover the conservative problem defined previously in (3.11).

Since δq is arbitrary, the condition above must hold over each segment of the time integration and in particular, we must have $\int_0^h \mathrm{d}s f(q,\dot{q})^T\delta q = 0$. The value of $q(s)$ in the integrand is approximated by the smooth function $q(s) = u_d(q_0,q_1,s,h)$ on the interval $[0,h]$, and therefore, the relation between $\delta q(s)$ and the discretized coordinates q_0, q_1 is expressed as

$$\delta q(s) = \frac{\partial u_d(q_0,q_1,s,h)}{\partial q_0}\delta q_0 + \frac{\partial u_d(q_0,q_1,s,h)}{\partial q_1}\delta q_1. \tag{3.79}$$

This leads to two terms to be evaluated, namely

$$\begin{aligned}\int_0^h \mathrm{d}s f(q,\dot{q})^T\delta q &= \int_0^h \mathrm{d}s f(q,\dot{q})^T\frac{\partial u_d}{\partial q_0}\delta q_0 + \int_0^h \mathrm{d}s f(q,\dot{q})^T\frac{\partial u_d}{\partial q_1}\delta q_1 \\ &= f_d^{(+)}(q_0,q_1,h)^T\delta q_0 + f_d^{(-)}(q_0,q_1,h)^T\delta q_1,\end{aligned} \tag{3.80}$$

with definitions

$$\begin{aligned}f_d^{(-)}(q_0,q_1,h)(q_0,q_1) &= \int_0^h \mathrm{d}s\frac{\partial u_d}{\partial q_0^T}f(q,\dot{q}), \\ f_d^{(+)}(q_0,q_1,h) &= \int_0^h \mathrm{d}s\frac{\partial u_d}{\partial q_1^T}f(q,\dot{q}).\end{aligned} \tag{3.81}$$

Collecting terms with δq_n, we find the stepping equations

$$\begin{aligned}D_2^T\mathbb{L}_d(q_{k-1},q_k,h) + D_1^T\mathbb{L}_d(q_k,q_{k+1},h) \\ + f_d^{(-)}(q_k,q_{k+1},h) + f_d^{(+)}(q_k,q_{k+1},h) = 0.\end{aligned} \tag{3.82}$$

A simple general choice for the function $u_d(q_0,q_1,s,h)$ is the linear map $u_d(q_0,q_1,s,h) = \tau q_0 + (1-\tau)q_1 = q_{1-\tau}$, and the velocity definition $v_1 = (1/h)(q_1 - q_0)$, leading to the evaluations

$$\begin{aligned}f_d^{\tau(-)}(q_0,q_1,h) &= \tau h f(q_{1-\tau},v_1), \\ f_d^{\tau(+)}(q_0,q_1,h) &= (1-\tau)h f(q_{1-\tau},v_1),\end{aligned} \tag{3.83}$$

and summing the contributions, we have

$$\begin{aligned}f_d^{\tau(-)}(q_k,q_{k+1},h) + f_d^{\tau(+)}(q_{k-1},q_k,h) = \\ \tau h f(q_{k+1-\tau},v_{k+1}) + (1-\tau)h f(q_{k-\tau},v_{k+1}).\end{aligned} \tag{3.84}$$

The limiting cases $\tau = 0$ and $\tau = 1$ lead to the stepping equations

$$\begin{aligned}D_2^T\mathbb{L}_d(q_{k-1},q_k,h) + D_1^T\mathbb{L}_d(q_k,q_{k+1},h) &= -hf(q_k,v_k), \text{ for } \tau = 0, \\ D_2^T\mathbb{L}_d(q_{k-1},q_k,h) + D_1^T\mathbb{L}_d(q_k,q_{k+1},h) &= -hf(q_k,v_{k+1}), \text{ for } \tau = 1.\end{aligned} \tag{3.85}$$

Also of interest is the symmetric version of the forces defined as

$$\begin{aligned} f_d^{\text{sym }\tau(-)}(q_0, q_1, h) &= \frac{h}{2}\left[\tau f(q_{1-\tau}, v_1) + (1-\tau) f(q_\tau, v_1)\right], \\ f_d^{\text{sym }\tau(+)}(q_0, q_1, h) &= \frac{h}{2}\left[(1-\tau) f(q_{1-\tau}, v_1) + \tau f(q_\tau, v_1)\right]. \end{aligned} \tag{3.86}$$

These different discretizations will be used alternately in what follows, depending on the stability requirements and the form of the function $f(q, \dot{q})$.

3.13 Rayleigh dissipation functions

A special type of forcing involves *Rayleigh dissipation functions* which are scalar functions of the form $\mathfrak{R}(q, \dot{q})$ and which generate forces

$$f_{\mathfrak{R}}(q, \dot{q}) = -\frac{\partial \mathfrak{R}(q, \dot{q})}{\partial \dot{q}^T}, \tag{3.87}$$

which are reminiscent from the potential forces defined previously in (3.18).

A simple example of this is the function $\mathfrak{R}(q, \dot{q}) = (1/2)\gamma\|\dot{q}\|$ which generates a viscous drag $f_{\mathfrak{R}} = -\gamma\dot{q}$.

The significance of such a scalar formulation is seen from the computation of the time rate of change of energy. Indeed, for a Lagrangian without explicit time dependence, the rate of change of energy due to the introduction of a Rayleigh dissipation function is

$$\begin{aligned} \frac{dE}{dt} &= \frac{\mathrm{d}}{\mathrm{d}t}\left\{\frac{\partial \mathcal{L}}{\partial \dot{q}}\dot{q} - \mathcal{L}\right\} = \left[\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathcal{L}}{\partial \dot{q}^T} - \frac{\partial \mathcal{L}}{\partial q^T}\right]\dot{q} = f^T(q, \dot{q})\dot{q} \\ &= -\frac{\partial \mathfrak{R}}{\partial \dot{q}}\dot{q}. \end{aligned} \tag{3.88}$$

The most common form for a Rayleigh function is homogeneous of degree 2 in $\dot{q}$. This yields $dE/dt = -2\mathfrak{R}(q, \dot{q})$. Therefore, the value of Rayleigh function is often associated with the energy dissipation rate.

Since the energy of a system subjected to the force produced by a Rayleigh function decreases at a rate proportional to $\mathfrak{R}(q)$, the dynamics will settle when $\mathfrak{R}(q) = 0$. If $\mathfrak{R}(q, \dot{q})$ is bounded below by 0, it is minimized by the dynamics but it also *constrains* the system on the surface $\mathfrak{R}(q, \dot{q}) = 0$. This is in fact the definition of a kinematic constraint. Since there are is no real restriction to the definition of a Rayleigh function, there offer the most general definition of a constraint on the system.

3.14 Constraints

From a reductionist perspective, it should be possible to model everything under the sun as systems of point particles interacting through pairwise interactions. In fact, for the range of time, length and mass scales mentioned previously, it should

be sufficient to use only the forces of classical universal gravitation and classical electromagnetism. But this argument is deceptive because it does not distinguish data from noise in both time and space domains. Indeed, a tennis ball can be viewed as a composite of several thousands of point particles, including millions of pairwise forces. But in fact, these point particles do not move much relative to each other and the most important data here is the location and velocity of the center of mass, the attitude and angular velocity about the center of mass, followed by the bulk deformations. In fact, these deformations are related to the velocity of sound in a solid and this typically in the range of 10^3 to 10^4, and propagation through a solid of size 1m for instance takes 10^{-3} to 10^{-4} seconds, far, far below the time resolution of interactive physics. The point here is that at the time resolution scale of 10ms of interactive physics, most bodies are rigid.

If we choose to ignore the microscopic physics, we must allow for geometric restrictions on the motion of our objects which are in fact the net result of the underlying microscopic dynamics. For instance, if a spherical marble of radius r is put on a flat plane described as the surface $z = 0$, a macroscopic model of this should impose the restriction that the center of mass of the bead satisfy the inequality $q_z - r \geq 0$, instead of computing the magnetic dipole interactions between the surface molecules of the marble and the ground surface. On the time scales we are interested in, any impact between the marble and the ground plane can be considered as instantaneous and we can therefore safely ignore all the microscopic physics and concentrate on enforcing the macroscopic constraint: $q_z - r \geq 0$.

In general, constraints are restrictions on the variables describing the system, namely, the coordinates q, their velocities $\dot{q}$, the forces $f(q, \dot{q})$, and time t. Any such restriction can be written as $c(q, \dot{q}, f^{(c)}, t) \geq 0$. Of course, for the restriction $c \geq 0$ to have an influence on the system, it must produce some sort of net force $f^{(c)}$, which will be deduced from d'Alembert's principle. This is where the strength of the variational formulation is truly palpable.

In what follows, we first present different categories of restrictions and then proceed to derive the constraint forces they generate using the fundamental principles we have exposed so far. The connection between constraints and strong forces with fast dynamics will be explored further below.

3.14.1 Kinematic constraints nomenclature

First consider restrictions which are purely kinematic, i.e., independent of the forces applied on the system so they can be written as algebraic inequalities $c(q, \dot{q}, t) \geq 0$. The following categories are distinguished.

scleronomic: without explicit dependence on time, $c(q, \dot{q}) \geq 0$;

rheonomic: explicitly time-dependent, $c(q, \dot{q}, t) \geq 0$, $\dfrac{\partial c}{\partial t} \neq 0$;

holonomic: independent of velocity, $c(q, t) \geq 0$;

nonholonomic: explicitly dependent on velocity so that $c(q, \dot{q}, t) \geq 0$ is a *non-integrable function*;

bilateral: the restriction is a strict equality $c(q, \dot{q}, t) = 0$;

unilateral: the restriction is an inequality $c(q, \dot{q}, t) \geq 0$;

ideal: the constraint forces are perpendicular to the trajectory so that the constraint does zero net virtual work on the system;

nonideal: the constraint forces produce nonzero net virtual work;

Unfortunately, this taxonomy is not hierarchical. To confuse things further, the physics literature often uses the term "holonomic constraints" as a shorthand for bilateral, scleronomic, holonomic, ideal constraints, and "nonholonomic constraints" for the rest. In what follows, an attempt is made to be as precise as possible in referring to different types of constraints.

The motivation for each item in the nomenclature differs. The split between *scleronomic* and *rheonomic* constraints is based on whether or not the is an explicit time dependence in the constraint definition. Rheonomic constraints, because they depend on time, can violate energy conservation.

Some constraints can be eliminated by a change of coordinates and this is said to make them *integrable*. For instance, if a point particle is constrained to stay at unit distance from the origin, with a light rigid rod for instance, one can introduce polar coordinates, set the radius to unity, and delete all time derivatives of the radius from the equations of motion. This is not possible in general, however, as discussed in Section 3.14.3. This motivates the distinction between *holonomic* constraints and *nonholonomic* ones, meaning that the former is integrable in principle, but the latter is generally not, as shown with a counter-example in Section 3.14.5.

The distinction between *ideal* and *nonideal* constraints is prone to generate confusion. Indeed, holonomic constraints are ideal as shown in Section 3.14.3. However, a rheonomic constraint can perform net work on the mechanical system, adding or removing energy.

In addition, the engineering literature [177] uses the term *effort constraint* as described in Section 3.14.2. The discretization technique of Section 3.12 suffices for these.

Special techniques are needed to discretize holonomic and non-holonomic constraints correctly, as discussed in Section 3.14.3 and Section 3.14.5, respectively.

3.14.2 Effort constraints

In the best of all circumstances, the forces applied on a system can be computed from the positions and velocities so that we have an explicit function of the configuration $f(q, \dot{q})$. But there are several cases where the best that can be done is the formulation of a constitutive relation $r(q, \dot{q}, f^{(r)}, t) \geq 0$, contributing $f^{(r)}$ to the generalized forces. In this case, the restriction generates additional

forces which may change the energy of the system according to the value of the integral $\int \mathtt{d}s f^T \dot{q}$, which is the work done done by a given force f on the system. The term *effort* comes from the systems engineering literature [154] [177] where it stands for generalized forces.

The best example of an effort constraint is the Coulomb friction model which describes the contact forces between two rigid objects. In this model, a normal force $f^{(\mathrm{nor})}$ enforces the non-penetration condition between the objects, and a tangential force $f^{(\mathrm{tan})}$ prevents them from moving relative to each other *provided* the magnitude of $f^{(\mathrm{tan})}$ does not exceed the product of the *friction coefficient* times the magnitude of the normal force. When the maximum magnitude is reached, the tangential contact velocity is opposed by the tangential friction force, producing dissipation. This is thus also an example of a nonideal constraint.

The only effort constraints considered in what follows are those which can be derived from Rayleigh functions, which is typical in the engineering system dynamics literature [177], where they are called dissipative efforts. The interesting symmetry here is that potential functions generate forces according to the negative generalized coordinates gradient whereas Rayleigh functions generate forces according to the negative of the generalized velocity gradient. As shown further below, these can be used to model Coulomb friction—provided appropriate non-smooth functions are used—as well as a number of constitutive models for drivers and motors. Since the forces generated by Rayleigh functions were already stated in (3.87) and since the discretization of non-conservative forces was already performed in Section 3.12, we do not expand further on effort constraints until special non-smooth forces are considered in Chapter 10.

3.14.3 Holonomic constraints

Consider an m-dimensional vector function of the configuration coordinates $g(q,t) : Q \times \mathbb{R} \mapsto \mathbb{R}^m$ with $m \times n$ Jacobian matrix $G = \partial g / \partial q$, where $n = \dim Q$, and, generally but not necessarily, $m \leq n$. The Jacobian maps the changes in coordinates q to changes in the function $g(q,t)$. Each row of G weights the the changes in each component of q in the global conspiracy to make the value of of g_i change. Matrix G can include conversion in units, scales, or general coordinate systems. The case where $m = n$ is precisely a change of coordinate systems, from Cartesian to polar or whatnot. When the restriction $g(q,t) = 0$ is imposed on the system, the net effect is to restrict the values of the coordinates q to live on a manifold $\mathcal{M} \in Q$. In fact, it should be possible to parametrize $\mathcal{M}$ directly and to reformulate the Lagrangian $\mathcal{L}(q,\dot{q}) : Q \mapsto \mathbb{R}$ as $\bar{\mathcal{L}}(y,\dot{y}) : \mathcal{M} \mapsto \mathbb{R}$. This is called a *reduced coordinates* approach but we choose instead an *augmented coordinate* strategy.

To understand the forces required to maintain the system on the submanifold $\mathcal{M}$, we go back to the variational principle and modify it by restricting the variations δq to be consistent with the constraints. Specifically, we require that

they be tangent to the manifold

$$\delta g(q,t) = \frac{\partial g}{\partial q}\delta q = 0. \tag{3.89}$$

Now, assuming that G has full row rank and that G can be partitioned as $G = \begin{bmatrix} A & B \end{bmatrix}$, where the square $m \times m$ matrix A is assumed invertible, and matrix B is $m \times n - m$. Given $x \in \mathbb{R}^n$ with partitioning $x = (y, z), y \in \mathbb{R}^m, z \in \mathbb{R}^{(n-m)}$, the solution to $Gx = 0$ is found to be

$$x = \begin{bmatrix} y \\ z \end{bmatrix} = \begin{bmatrix} -A^{-1}B \\ I_{n-m} \end{bmatrix} z = Hz, \tag{3.90}$$

where I_{n-m} is the $(n-m) \times (n-m)$ identity matrix, for any $z \in \mathbb{R}^{(n-m)}$. In other words, the columns of $n \times n - m$ Matrix H span the null space of matrix G. Conversely, consider a vector $u = G^T\lambda$ where $\lambda \in \mathbb{R}^m, u \in \mathbb{R}^n$, then, $u^T H = \lambda^T G H = 0$. Therefore, the left null space of H is mapped by the vectors $G^T\lambda, \lambda \in \mathbb{R}^m$. Thus we write the variations $\delta q(t) = \epsilon H\eta(t), \eta : \mathbb{R} \mapsto \mathbb{R}^{n-m}, \eta(t_0) = \eta(t_1) = 0$, where η is uniformly bounded on $[t_0, t_1]$ and at least $\mathcal{C}^2$ but otherwise, arbitrary. From this, we find that the functional derivative of the action (3.16) becomes

$$\delta S(t_0, t_1) = \int_{t_0}^{t_1} \mathrm{d}s \left\{ -\frac{\mathrm{d}}{\mathrm{d}s}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}}\right) + \frac{\partial \mathcal{L}}{\partial q} \right\} H\eta = 0. \tag{3.91}$$

For this to be true for arbitrary η, we need the term in the braces to be in the left null space of matrix H which means that it can be written as $\lambda^T G$ for some $\lambda \in \mathbb{R}^m$, and the value of λ is determined by enforcing the condition $g(q,t) = 0$. After transposing the vectors in (3.91), the Euler-Lagrange equations of motion now read

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathcal{L}}{\partial \dot{q}^T} - \frac{\partial \mathcal{L}}{\partial q^T} - G^T\lambda &= 0 \\ g(q,t) &= 0. \end{aligned} \tag{3.92}$$

This is now a system of differential algebraic equation (DAE)s of index 3. There are several definitions for the index of a DAE [114]). The simplest is the *differentiation* index. This is one plus the number of times the algebraic equation—the second line in (3.92)—has to be differentiated *twice* before the $\ddot{q}$ factors can be extracted and substituted in the differential equation. The index starts at 1 when there is a non-trivial algebraic part which must be inverted and substituted in the differential part, and increases by one for each derivative needed to perform this substitution and thus eliminate the algebraic variables, λ in the present case. Another way to see that something is missing is that the *ghost* variable λ appears in the first line of (3.92) but its time derivative $\dot{\lambda}$ does not appear anywhere in the system. Thus, if we view (3.92) as an implicit differential equation of the form $F(\ddot{q}, \dot{q}, q, \ddot{\lambda}, \dot{\lambda}, \lambda, t) = 0$, the function F has several zero blocks in its

Jacobian which is in fact rank deficient. Solving linear systems with rank deficient matrices requires using singular value decomposition (SVD), rank revealing orthogonal factorization (QR), or some form of least squares approximation.

Numerically integrating general DAEs of index higher than 1 is difficult and usually requires the SVD or rank revealing QR at least once per time step to solve a $2n + m$-dimensional system of nonlinear equations for $q, \dot{q}$ and λ several times. Example of Runge-Kutta methods are found in [111, 114] and backward difference formula (BDF) based methods are described in [54, 29] and in [165]. Some interesting connection to nonlinear least squares methods are described in [41, 43].

By comparison, the variational time stepping method offers a simpler and more robust alternative, at least for the case of constrained mechanical systems, as shown shortly. Before discretizing the Lagrangian, an alternative derivation of the constrained equations of motion is derived.

3.14.4 The ghosts enter

Consider the augmented Lagrangian defined as

$$\bar{\mathcal{L}}(q, \dot{q}, \lambda, \dot{\lambda}) = \mathcal{L}(q, \dot{q}) + \lambda^T g(q, t), \tag{3.93}$$

and consider the variable λ to be the coordinates of a new type of physical bodies, namely, m one-dimensional point particles, each having position $\lambda_j, j = 1, 2, \ldots, m$ and to be treated like any other kinematic variable in the system. These point particles have no mass and do not appear in the kinetic energy though they appear in the potential term $\lambda^T g(q)$. This justifies calling them *ghost* particles since they have no material realization though, as we shall see, they are of great importance in the construction of stable and regularized numerical methods. In the present context, only massless ghosts are considered but, as we show later, if the mass were to be nonzero, it would have to be negative since otherwise, the energy would quickly diverge. The usefulness of this conceptualization is that we recover precisely the standard variational structure where the variations on the λ and q variables are *unrestricted*. Note also that from (3.93), there is no energy associated with the ghost at all, since $g(q) = 0$ along the trajectory. Regularization will change that though.

The Euler-Lagrange equations of motion (3.17) for the Lagrangian (3.93) are easily computed to yield

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \bar{\mathcal{L}}}{\partial \dot{q}^T} - \frac{\partial \bar{\mathcal{L}}}{\partial q^T} &= \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathcal{L}}{\partial \dot{q}^T} - \frac{\partial \mathcal{L}}{\partial q^T} - G^T\lambda = 0 \\ \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \bar{\mathcal{L}}}{\partial \dot{\lambda}^T} - \frac{\partial \bar{\mathcal{L}}}{\partial \lambda^T} &= -g(q, t) = 0, \end{aligned} \tag{3.94}$$

which is identical to (3.92). The point being that with the ghost interpretation, we can proceed to discretize the Lagrangian (3.93) with the same techniques used for the unconstrained systems in Section 3.5 and 3.7, which is an alternative to what has previously been reported in the literature.

The standard technique to discretize a Lagrangian subject to holonomic constraints is to starting from the discretization defined in (3.20) and simply restrict the variables q_k to satisfy the condition $g(q_k, t_k) = 0$. Making the action stationary is now a constrained extremal problem in several dimensions (see [195], section 4.3 for instance), and the result is the constrained, discrete, Euler-Lagrange equations

$$\begin{aligned} \frac{\partial \mathbb{S}}{\partial q_k^T} = D_1^T \mathbb{L}(q_k, q_{k+1}) + D_2^T \mathbb{L}(q_{k-1}, q_k) = {G^T}_k \lambda \\ g(q_{k+1}, t_{k+1}) = 0. \end{aligned} \tag{3.95}$$

It is important to note that G_k appears in the first equation whereas the second equation calls for vanishing $g(q,t)$ at time $k+1$.

Discretizing the ghost variables in the augmented Lagrangian directly, the same result is obtained using

$$\int_0^h \mathrm{d}s \lambda^T g(q,t) \approx h \lambda_0^T g_0. \tag{3.96}$$

However, assuming for instance that λ and g are *uncorrelated*, we could use

$$\int_0^h \mathrm{d}s \lambda^T g(q,t) \approx h \left(\frac{\lambda_1 + \lambda_0}{2} \right)^T \left(\frac{g_1 + g_0}{2} \right), \tag{3.97}$$

to get a slightly different stepping equation

$$\begin{aligned} \frac{\partial \mathbb{S}}{\partial q_k^T} = D_1^T \mathbb{L}(q_k, q_{k+1}) + D_2^T \mathbb{L}(q_{k-1}, q_k) = G_k^T \bar{\lambda} \\ g(q_{k+1}, t_{k+1}) + 2g(q_k, t_k) + g(q_{k-1}, t_{k-1}) = 0, \end{aligned} \tag{3.98}$$

where $\bar{\lambda} = \frac{1}{4}(\lambda_{k+1} + 2\lambda_k + \lambda_{k-1})$. This formulation preserves the constraints on average even though it allows for $g_k \neq 0$, and this could produce oscillations at high frequencies. This will be analyzed in depth Chapter 4 when considering constraint regularization and stabilization.

3.14.5 Nonholonomic constraints

Nonholonomic constraints are kinematic restrictions of the form $a(q, \dot{q}, t) = 0$ which cannot be integrated back to a holonomic conditions of the form $\tilde{a}(q) = 0$. As such, they cannot be eliminated by a change of coordinates. There is still controversy regarding the correct equations of motion when nonholonomic constraints are concerned [182] [50].

The case a wheel contacting a plane which moves by rolling without slipping is the archetype of a nonholonomic constraints and offers a good illustration of what is meant by nonintegrable. Consider a vertical wheel of radius $\rho > 0$ which rolls without slipping on the plane $z = 0$. The center of rotation has coordinates $x, y, z = \rho$. The angle of rotation of the wheel about the center of rotation is

ϕ. The projected velocity of the wheel onto the plane is $\|v\| = \rho\dot{\phi}$. Now, let θ be the angle between the plane of rotation of the wheel and the x-axis of the plane. As the wheel rolls without slipping, its velocity vector is aligned with the vector $v = r(\dot{\phi})(\cos(\theta), \sin(\theta))^T$. This leads to the following pair of constraint equations:

$$\begin{aligned} \dot{x} - r\cos(\theta)\dot{\phi} &= 0 \\ \dot{y} - r\sin(\theta)\dot{\phi} &= 0. \end{aligned} \tag{3.99}$$

Now, it is tempting to assume that it is possible to express the motion entirely in terms of ϕ, θ and eliminate x, y but this turns out to be impossible. To show this, first relabel the coordinates x, y, ϕ, θ as q_1, q_2, q_3, q_4. Assume assume that (3.99) can be rewritten as $dh(q)/dt = 0$, by using an integrating factor, $f(q)$. If we write the first equation as $\sum_i g_i\dot{q}_i = 0$, with $g_1 = 1, g_2 = 0, g_3 = -r\cos(\theta), g_4 = 0$, then, we need to find $f(q)$ which satisfies equality of mixed partials

$$\frac{\partial(fg_i)}{\partial q_j} = \frac{\partial(fg_j)}{\partial q_i}. \tag{3.100}$$

But $g_4 f = 0$ identically and so we need $\dfrac{\partial(fg_3)}{\partial q_4} = 0$, and $\dfrac{\partial(fg_1)}{\partial q_4} = 0$. The first of these two equations implies that $f(q) = \tilde{f}(q_1, q_2, q_3)/\cos(\theta)$ but after substituting into the second equation, we get a contradiction. Therefore, there is no integrating factor and so it is not possible to reduce this constraint to an algebraic conditions on the generalized coordinates. This means that even though the constraint reduces the number of degrees of freedom locally, it does not reduce them globally. This is a fundamental feature of nonholonomic constraint.

To elucidate how a nonholonomic constraint affects the motion, start with a Pfaffian form (a vanishing differential form of degree one) so that $a(q, \dot{q}, t) = A(q)\dot{q} + w(t) = 0$, where A is an $m \times n$-dimensional matrix. In general, $m \leq n$, and $w(t)$ is a given m-dimensional time dependent vector. If we require that the virtual displacements satisfy the simplified constraint: $A(q)\delta q = 0$, then, using the same analysis as in Section 3.14.3, we find the Euler-Lagrange equations of motion to be

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathcal{L}}{\partial \dot{q}^T} - \frac{\partial \mathcal{L}}{\partial q^T} - A^T\alpha &= 0 \\ a(q, \dot{q}, t) &= 0. \end{aligned} \tag{3.101}$$

The motivation here is that the constraint force is $A^T\alpha$ and therefore, the virtual work is $\alpha^T A\delta q$ and since we imposed that $A\delta q = 0$, this formulation is indeed workless as desired.

Note that in this case, it is not possible to augment the Lagrangian with terms of the form $\beta^T a(q, \dot{q}, t)$ as this causes spurious terms proportional to $\dot{\alpha}$ which are not physical as verified in [182] [91], though this *vakonomic* description—a term coined by Arnold [23] to mean *variational and axiomatic*—has been used in the literature several times. As demonstrated in [91], the issue here is whether the

condition $a(q, \dot{q}, t) = 0$ is strictly enforced on the variational paths $q + \delta q$, or, instead, only the simpler, tangential condition $A\delta q = 0$. Only the latter form is correct.

However, consider the function $\mathfrak{R} = \dot{\alpha}^T a(q, \dot{q}, t)$ where α is the generalized coordinate of a ghost particle and use the augmented Lagrangian $\bar{\mathcal{L}}(q, \dot{q}, \alpha, \dot{\alpha}) = \mathcal{L}(q, \dot{q})$. The generalized force due to this Rayleigh function is

$$f_{\mathfrak{R}} = -\begin{bmatrix} \dfrac{\partial \mathfrak{R}}{\partial \dot{q}^T} \\ \dfrac{\partial \mathfrak{R}}{\partial \dot{\alpha}^T} \end{bmatrix} = -\begin{bmatrix} A^T \dot{\alpha} \\ a \end{bmatrix} \tag{3.102}$$

and therefore, if we impose the restriction that the virtual displacements generate no virtual work, we recover the same equations of motion as (3.101) with a redefinition of the α variables. On the other hand, we also recover the same equations (3.101) if we allow for free variations applying d'Alembert's principle:

$$\delta \int_{t_0}^{t_1} \mathrm{d}a \bar{\mathcal{L}}(q, \dot{q}, \alpha, \dot{\alpha}) + \int_{t_0}^{t_1} \mathrm{d}s \left(\frac{\partial \mathfrak{R}}{\partial \dot{q}} \delta q + \frac{\partial \mathfrak{R}}{\partial \dot{\sigma}} \delta \alpha \right) = 0. \tag{3.103}$$

It is curious that this simple fact is not mentioned in the literature and it is not entirely clear whether the correct equations of motion would be reproduced using this technique on nonholonomic constraints which do not have a Pfaffian form. Nevertheless, we consider the formulation $\mathfrak{R} = \dot{\alpha}^T a(q, \dot{q}, t)$ and the variational formulation in (3.103) as axiomatic since we will only use constraints which have Pfaffian form.

Nonholonomic constraints are now discretized using two alternative methods which produce the same result. First, following the analysis found in [68], the restrictions on the variations are imposed as: $A_k \delta q_k = 0$, and the constraints are now written as $a(q, \dot{q}, t) \approx A_k (q_k - q_{k-1})/h + w_k$. Note the choice of time index for these equations which is very important for consistency and stability of the resulting stepper, and is based on a careful interpretation and discretization of the classical d'Alembert's principle of virtual work and geometric analysis. Neglecting the boundary terms, i.e., setting $\delta q_0 = \delta q_N = 0$, the discrete variational principle is then expressed as

$$\begin{aligned} \delta \mathbb{S}_d(q_0, \ldots, q_N, h) &= \sum_{k=0}^{N-1} \delta \mathbb{L}_d(q_k, q_{k+1}, h) \\ &= \sum_{k=0}^{N-1} \left[D_1 \mathbb{L}_d(q_k, q_{k+1}, h) \delta q_k + D_2 \mathbb{L}_d(q_k, q_{k+1}, h) \delta q_{k+1} \right] \\ &+ \sum_{k=0}^{N-1} \left[D_1 \mathbb{L}_d(q_k, q_{k+1}, h) + D_2 \mathbb{L}_d(q_{k-1}, q_k, h) \right] \delta q_k = 0. \end{aligned} \tag{3.104}$$

If we write $\mathbb{U}_k = D_1 \mathbb{L}_d(q_k, q_{k+1}, h) + D_2 \mathbb{L}_d(q_{k-1}, q_k, h)$, then, the stationary condition implies that $\mathbb{U}_k = \alpha^T A_k$ for some vector α, and by simultaneously im-

posing the discretized constraints, we find the discrete Euler-Lagrange equations for nonholonomic constrained systems

$$\begin{aligned} D_1^T \mathbb{L}_d(q_k, q_{k+1}, h) + D_2^T \mathbb{L}_d(q_{k-1}, q_k, h) - A_k^T \alpha &= 0 \\ \frac{1}{h} A_{k+1}(q_{k+1} - q_k) + w_{k+1} &= 0. \end{aligned} \tag{3.105}$$

Note once more that the time index of the discretized constraint is shifted by one here because it includes the yet unknown variable q_{k+1}. This introduces a slight asymmetry between A_{k+1} in the second equation and A_k in the first.

But simply using the Rayleigh function $\mathfrak{R} = \dot{\alpha}^T a(q, \dot{q}, t)$ along with the discretization of forced system given in Section 3.12, in (3.82), and choosing the parameter $\tau = 0$ in (3.85), we recover the same stepping scheme as in (3.105) but with a relabeling of the dummy variables α, i.e.,

$$\begin{aligned} D_1^T \mathbb{L}_d(q_k, q_{k+1}, h) + D_2^T \mathbb{L}_d(q_{k-1}, q_k, h) - A_k^T (\alpha_k - \alpha_{k-1}) &= 0 \\ \frac{1}{h} A_{k+1}(q_{k+1} - q_k) + w_{k+1} &= 0. \end{aligned} \tag{3.106}$$

The equivalence between these two discretization strengthens the case that nonholonomic constraints are indeed equivalent to special Rayleigh functions. If this were the end of the story, it would be only a curiosity but in fact, this pony can do more than one trick. Indeed, careful design of Rayleigh functions—and careful discretization of the force terms—will allow to produce constraint stabilization forces as well as a variety of effort constraints, opening a continuum of possibilities apart from standard, pure holonomic constraints.

3.14.6 Inequality constraints

We now turn to unilateral kinematic constraints of the form $c(q, t) \geq 0$, where $c : \mathbb{R} \times \mathbb{R}^n \mapsto \mathbb{R}^m$, with $m \leq n$. The restriction on the virtual displacements is now $C\delta q \geq -c(q, t)$. Therefore, whenever $c(q, t) > 0$, we recover the free equations for small enough δq. When $c(q, t) = 0$, we recover the same equations of motion as in the constrained case with the constraint force $C^T \nu$. However, there is yet another restriction, namely, that $\nu \geq 0$. To see this, consider a time t such that $c(q, t) = 0$. Assume that at time t, the other forces on the system are summed up as $f(t)$. Considering the simple constant mass Lagrangian of (3.12) and the constrained equation of motion (3.92), the acceleration $\ddot{q}$ at time t must satisfy

$$\begin{aligned} M\ddot{q} - C^T \nu &= f \\ c(q, t) &= 0, \end{aligned} \tag{3.107}$$

which can be solved locally for $\ddot{q} = M^{-1}(f + C^T \nu)$, where matrix C is the Jacobian $C = \partial c / \partial q$ as previously.

Using a first order expansion around t with a small time step $h > 0$ and using the expression just derived for $\ddot{q}$, the constraints will still be satisfied at $t + h$ if

the following inequality is satisfied

$$\begin{aligned} 0 \leq c(q(t+h), t+h) &= hC\dot{q} + h\frac{\partial c}{\partial t} + \frac{h^2}{2}C\ddot{q} + \dots \\ &= h\left(C\dot{q} + c_t + \frac{h}{2}CM^{-1}f\right) + \frac{h^2}{2}CM^{-1}C^T\nu + O(h^3), \end{aligned} \tag{3.108}$$

where $c_t = \partial c/\partial t$. For the present analysis, all terms of order $O(h^3)$ and above are neglected. Rewriting the last line of (3.108) in matrix form as

$$\begin{aligned} &0 \leq q + H\nu, \\ &q = h\left(C\dot{q} + c_t + \frac{h}{2}CM^{-1}f\right), \qquad \text{and } H = CM^{-1}C^T, \end{aligned} \tag{3.109}$$

the question now is how to choose ν so the function c is free to become positive, but restricted from becoming negative. Observe first that the components of the vector q are the rates at which the constraint components are changing in the absence of constraint force. Restrict the analysis to $m = 1$ first so that $CM^{-1}C^T = \mu > 0$ is a positive scalar. Obviously, if $q > 0$, the inequality is satisfied for $\nu = 0$ which leads to $c(q(t+h), t+h) > 0$. Conversely, if $q < 0$, we need $\nu > 0$ to satisfy the inequality (3.108) and this will keep $c(q(t+h)t+h) = 0$. In both cases, the result is that $\nu^T(q + H\nu) = 0$, which can also be written as $0 \leq q + H\nu \perp \nu \geq 0$. This argument can be generalized to m dimensions as is done in [224] and the result is the unilaterally constrained Euler-Lagrange equations of motion

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathcal{L}}{\partial \dot{q}^T} - \frac{\partial \mathcal{L}}{\partial q^T} - C^T\nu &= 0 \\ 0 \leq c(q,t) \quad \perp \quad \nu &\geq 0, \end{aligned} \tag{3.110}$$

where the perpendicularity sign, $\perp$ is understood componentwise. This means that given two n-dimensional vectors, x, y, $0 \leq x \perp y \geq 0$ implies that for each index $i = 1, 2, \dots, n$, the following three conditions hold simultaneously

$$x_i \geq 0, \quad y_i \geq 0 \quad \text{, and } x_i y_i = 0. \tag{3.111}$$

With the conditions stated in (3.111), the familiar perpendicularity $x^T y = 0$ is recovered. The system of equations and conditions in (3.110) is a nonlinear complementarity system. The variable $\nu > 0$ is essentially a *slack variable* which enforces the Fourier inequality, namely, that virtual work is non-positive when the configuration space has a closed boundary [173] (remember that the Euler-Lagrange equations (3.17) are the negative of the integrand of the variation of the action in (3.16)).

The solutions $q(t)$ of this system are *nonsmooth* since there are jump discontinuities at impact points, i.e., instants t_i such that for at least one index j, $g_j(q(t_i)) = 0$ but $g_j(q(t_i - \epsilon)) > 0$ for a small $\epsilon > 0$.

The discretization of inequality constraints can be handled using standard techniques for constrained extrema from vector calculus. A naive application of

these produces the discrete, unilaterally constrained, Euler-Lagrange equations

$$\begin{aligned} \frac{\partial \mathbb{S}}{\partial q_k^T} = D_1^T \mathbb{L}(q_k, q_{k+1}) + D_2^T \mathbb{L}(q_{k-1}, q_k) = C_k^T \nu \\ 0 \leq c(q_{k+1}, t_{k+1}) \quad \perp \quad \nu \geq 0, \end{aligned} \tag{3.112}$$

which defines a nonlinear complementarity problem (NCP). However, this formulation is deceptive since we neglected to constrain the energy of the problem. A careful analysis of this issue is found in [87] in which a time stepping method is constructed by requiring that energy be strictly conserved at each impact. However, since locating all impacts is prohibitively expensive for large systems, we will instead construct an inelastic regularized time stepper which resolves impact conditions only at the discretized times $t = kh$. We will restore the incident momentum when an impact has been detected but this will be resolutely *post facto*. In order to build a stable method for this though, we need both constraint regularization and constraint stabilization techniques which are developed in Chapter 4.

3.15 Discrete momenta and velocities

The description of the stepping equations so far relies on three terms recurrence relations of the coordinates q_{k-1}, q_k, and q_{k+1}. For the archetypal case of the Verlet stepping formula, this leads to

$$q_{k+1} = 2q_k - q_{k-1} + h^2 M^{-1} \nabla V(q_k) \tag{3.113}$$

It is however often more convenient to work with two step recurrences based on velocities and positions only. For the simplest cases, using the approximation

$$v_k = \dot{q}_k \approx \frac{q_k - q_{k-1}}{h}, \tag{3.114}$$

as we did previously in the definition of $\mathbb{L}_d(q_0, q_1, h)$, we can rewrite the recurrence relation as

$$\begin{aligned} v_{k+1} &= v_k + hM^{-1} \nabla V(q_k) \\ q_{k+1} &= q_k + hv_{k+1}. \end{aligned} \tag{3.115}$$

But since a typical simulation does need both the values of positions q and velocities $v = \dot{q}$, an implementation should be realized in terms of these observable variables and not using the three term recurrence in q. It is expected that this evaluation is a slight numerical improvement. Specifically, it might be a good idea to use velocities to avoid the potentially problematic sum $2q_k - q_{k-1}$. Indeed, this looks much like the textbook example for catastrophic cancellation. Other than being practical though, this formulation adds no new information. For this reason, the theory exposition continues with variables q_k only.

Of course, this approximation of velocity is only good to first order and, in deed, a second order approximation could be obtained by putting

$$v_{k+1/2} = \frac{1}{h}(q_{k+1} - q_k). \tag{3.116}$$

This leads to the velocity Verlet scheme (see [196] for a variational derivation of this) which requires two force computation per step in the case where we have velocity dependent forces or constraints. This is not pursued further in order to concentrate on constraint regularization and stabilization, as well as the treatment of friction and impacts.

An alternative formulation which does slightly change the resulting algorithm formulation is to introduce the momentum variables defined as:

$$p_k = D_2^T \mathbb{L}_d(q_{k-1}, q_k, h). \tag{3.117}$$

As shown in [196], it is possible to construct *Hamiltonian* flows, $(q_k, p_k) \mapsto (q_{k+1}, p_{k+1})$ as long as we can indeed construct the discrete Hamiltonian function, $\mathbb{H}_d(q, p, h)$ which is the Legendre transform of the discrete Lagrangian $\mathbb{L}_d(q_k, q_{k+1}, h)$. The theoretical ramifications of a Hamiltonian representation are interesting but in the context of dissipative systems with nonholonomic constraints, many subtleties are avoided by concentrating on the Lagrangian formulation. We shall therefore not consider the Hamiltonian theory at all in the rest of this thesis. Nevertheless, the definition of discrete momentum will be used whenever useful. It is interesting to note here that for the simple conservative mechanical Lagrangian $\mathcal{L} = (1/2)\dot{q}^T M \dot{q} - V(q)$ of (3.12), the momentum p_k defined by (3.117) is *not* simply $M(q_k - q_{k-1})/h$ as one might first expect but can in fact contain terms proportional to ∇V as well.

3.16 Minimization structure

An interesting fact which was exploited at length in [151, 152, 225] is that in some cases, with appropriate discretization choice, the solution of the discrete Euler Lagrange equations (3.22) is in fact the solution of a *minimization* problem. For instance, the simple discretization of the prototype mechanical Lagrangian of (3.12) yields

$$\mathbb{L}_d(q_k, q_{k-1}, h) = \frac{1}{2h}\left(q_k - q_{k-1}\right)^T M \left(q_k - q_{k-1}\right) - hV(q_{k-1}), \tag{3.118}$$

leads to the objective function

$$\begin{aligned}&\phi(q_{k+1}, q_k, q_{k-1}) = \\ &\frac{1}{2}\left(q_{k+1} - q_k\right)^T M \left(q_{k+1} - q_k\right) + q_{k+1}^T M \left(q_k - q_{k-1}\right) - h^2 q_{k+1}^T \nabla V(q_{k-1})\end{aligned} \tag{3.119}$$

and the trajectory is found by solving the problems

$$\arg\min_{q_{k+1}} \phi(q_{k+1}, q_k, q_{k-1}). \tag{3.120}$$

This is a quadratic minimization problem which is easy to solve with a variety of methods suitable for unconstrained convex minimization. Numerical convex minimization is generally better behaved that general nonlinear equation solving and this can be a great advantage. It is shown in detail in [151] [225] how to modify the function ϕ to different integration methods, in particular, to a class of the Newmark integrators. Since we do not pursue this avenue, the details are not covered further here.

The true usefulness of this formulation lies in that we can introduce holonomic constraints directly to the minimization problem. This leads to a constrained quadratic minimization problem which can also be solved robustly by various numerical methods. But so far, this formulation cannot handle either external dissipative forces or nonholonomic constraints. The real interest of the minimization strategy is revealed when considering forces derived from Rayleigh functions, as we have done above in Section 3.12, 3.14.1, and 3.14.5. Rayleigh functions are *maximally* dissipative since the rate of energy decay is $-\dot{q}^T \partial \mathfrak{R} / \partial \dot{q}$. From this fact, it is inferred that q_{k+1} should be chosen so as to minimize the value of the discretized Rayleigh functions at the end of the time step. Discretizing the Rayleigh functions with

$$\mathfrak{R}_d(q_k, q_{k-1}, h) = h\mathfrak{R}(\frac{q_k - q_{k-1}}{h}, q_{k-1}), \tag{3.121}$$

the following algorithm leads to a maximum dissipation. First, solve

$$\arg\min_{\substack{\tilde{q} \\ \text{subj. to } g(\tilde{q})=0}} \phi(\tilde{q}, q_k, q_{k-1}), \tag{3.122}$$

respecting all constraints, and then, solve the minimum problem:

$$\arg\min_{\substack{q_{k+1} \\ \text{subj. to } g(q_{k+1})=0}} T_q(q_{k+1}, \tilde{q}) + \mathfrak{R}_d(q_{k+1}, q_k), \tag{3.123}$$

where T_d is the discrete kinetic energy defined here as:

$$T_d(q_1, q_0) = \frac{1}{2h} (q_1 - q_0)^T M (q_1 - q_0). \tag{3.124}$$

This strategy can even be extended for cases where the Rayleigh functions are non-smooth, which is the case for Coulomb friction as we explain further below. In particular, it is possible to handle contacts in this way as well, even non-smooth ones, by imposing the relevant constraints on the minimization problem as is done in [225], where a non-smooth formulation of Coulomb friction is also provided. The main issue here is that solving for non-smooth, non convex quadratic minimization problems subject to nonlinear constraints is not exactly simple or economical. This is why a different avenue will be explored when dealing with friction and when computing approximations to the constrained, forced discrete Euler Lagrange equations.

3.17 End notes

The historical development of the principle of least action and analytical mechanics in general is covered at length in the still very fresh and highly recommendable

monograph of Lanczos [173]. Lanczos explain to great length all facets and all formulations of the variational principle, including the thinking that went behind the development of the techniques as well as the important elements the historical debates that opposed the proponents of the variational methods and those who insisted on sticking to Newton's three laws.

Another historical account which concentrates on the inception of the principle and its application to conservative systems, including quantum mechanical and special relativistic ones, is found in the brief and direct monograph of Yourgrau and Mandelstam [284]. A more scholarly historical account is provided by Goldstine [106] who went to great length in tracing the original correspondence between the pioneers, namely, Fermat, Maupertuis, Euler, Lagrange, the Bernoulli brothers, Hamilton and Jacobi. The original work by Hamilton [115, 116], who is credited with the modern version of the variational principle in integral form, the Lagrange function itself, as well as the first investigations of the symplectic nature of the flow of conservative systems is still very readable and now broadly available since the Royal Society of London has digitized its archives and published them on-line. Also of interest is the original *magnum opus* of Lagrange [171, 172], especially because of the direct style in which it is written and the insistence on relying on the equations themselves, instead of geometric constructions.

When it comes to Newtonian mechanics, the original *Principia* [215] is striking for its direct writing style and, in my opinion, superior to many modern textbook presentations.

A systematic analytic formulation of non-conservative systems is found in the brief and lucid monograph of Layton [177], which is perhaps the first attempt to phrase what the engineers call "system dynamics" into a consistent Lagrangian form, including effort constraints. This served as inspiration to use Rayleigh dissipation functions—Layton's dissipative efforts—systematically.

For the discretization of the least action principle, the credit goes to Moser and Veselov [208] in the mathematical literature and to Gillilan and Wilson [100] in the chemical physics literature. A very interesting difference between these two seminal papers is that whereas Moser and Veselov were chasing after examples of discrete *integrable systems*—they used the freely rotating rigid body as their example—and derived the discrete Euler-Lagrange equations (3.22), Gillian and Wilson were considering periodic systems and thus used fixed endpoints on the discrete action and derived a special type of minimum principle for such cases. It is not clear whether Gillilan and Wilson were aware of the work of Moser and Veselov or if the idea was already common knowledge. Extensions to systems subject to holonomic constraints were presented by Wendlandt and Marsden in [277] and nonholonomic constraints were covered by Cortès and Martínez in [68]. A comprehensive account of the discrete variational technique which covers all aspects is found in Marsden and West [196] which is a must read. Additional extensions of the variational principle to non-smooth contact is presented in Pandolfi, Kane, Marsden, and Oritz [225, 152]. Energy properties of non-smooth contacts are presented in Kane, Marsden, and Oritz [150], and dissipative systems are

analyzed in Kane, Marsden, Oritz, and West [151].

The low order symplectic methods derived *via* the discrete variational principle were published by Verlet [271] for unconstrained molecular dynamical systems and this was extended to constrained systems in a technique called SHAKE by Ryckaert, Ciccotti, and Berendsen [242], a direct extension of the Verlet stepping scheme which works directly on the generalized coordinates, and later in an extension called RATTLE, Andersen [5] which computes both generalized coordinates and velocities. These three methods are bread and butter in molecular dynamics simulations [178] but are little known outside that field. These two latter methods which solve DAEs of index 3 and require only one or two solutions of systems of nonlinear equations per step, respectively, are seldom reported in the DAE literature. They are found in [114] but neither in [54] nor in [29], and that even though constrained mechanical systems are generally taken as an archetypal example of a higher index DAE. A comprehensive survey or geometric integration methods, of which variational methods are a subset, is found in the monograph of Hairer, Lubich, and Wanner [112], who provide numerous numerical examples. The same topic but with particular focus on conservative physical systems and applications to molecular dynamics is covered in the book by Leimkuhler and Reich [178]. Kane, Marsden, Oritz, and West [151], reported on the connection between some Newmark integrators, used widely in structural engineering, and variational formulation. They found that some Newmark integrators could be derived from the variational principle by redefining the potential energy term—precisely those members of the Newmark family which were known to produced better results. But in addition, they proved a "shadowing theorem", implying that trajectories generated by these Newmark methods intersect those produced by *any* variational integrator applied to the original problem, within each time step. In other words, the Newmark integrators do not generate identical trajectories as any variational integrator but for each time step k, there is a point $\alpha_k \in (0, h)$, where h is the fixed time step, such that $q_{k+\alpha_k}^{(\mathrm{New})} = q_{k+\alpha_k}^{(\mathrm{Var})}$, where $k + \alpha_k$ denotes the linear interpolation $q_{k+\alpha_k} = q_k + \alpha_k(q_{k+1} - q_k)$.

Standard techniques for integrating differential equations are found in the encyclopedic two volume monograph of Hairer Wanner and Nørsett [113, 114], for instance, and well covered in the literature as well. The difference between standard techniques and the variational ones which are the subject of this thesis is illustrated in Figure 3.10. Fundamentally, discretization of trajectories and application of the principle of least action are non-commuting operations. Applying discretization on the equations of motion—using a standard integration technique on the Euler-Lagrange equations (3.17) or the extension (3.94) for holonomic constraints and (3.101) for nonholonomic constraints—offers no guarantee on the preservation of invariants, unless the method is constructed specifically to account for these, as is done in symplectic Runge-Kutta methods for instance [112]. By contrast, *any* approximation of the time integral of the Lagrangian (3.19) yields a stepping scheme which preserves the symplectic nature of the flow and other symmetries as well, and the construction of the stepping scheme *via* the discrete Euler-Lagrange equations (3.22) is straight forward.

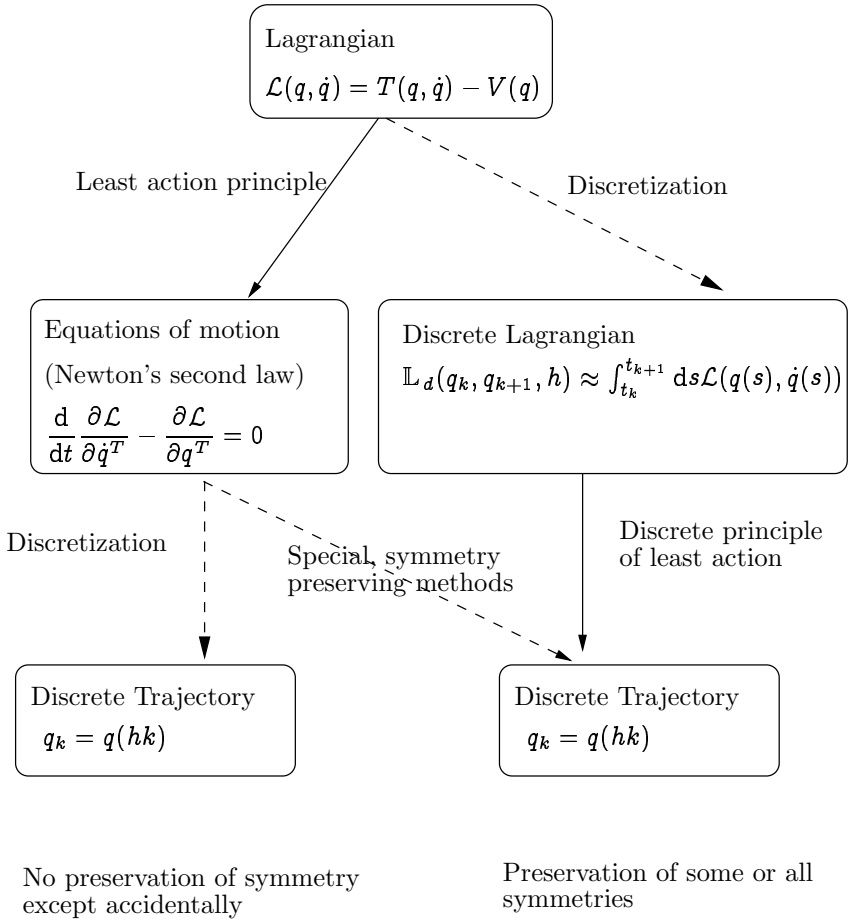

Figure 3.10: The non-commutativity of discretization and the principle of least action.

The fact is that the two operations, namely, applying Hamilton's principle of least action and discretizing the trajectories, do not commute. By discretizing the Lagrangian function first, the only possible stepping equations that can be produced are those which preserve a number of physical invariants down to the precision used in solving the nonlinear stepping equations. By contrast, if discretization is used after the least action principle is applied to the Lagrangian to produce the equations of motion, physical symmetries are preserved *only when they are explicitly built into the numerical method*. But of course, the numerical analysts have been industrious and have identified very many integration methods over the years and, therefore, usually, when producing discrete stepping equations using the principle of least action, we do recover a known method and this is why there is a cross link.

Nevertheless—and this is remarkable—, when it comes to integrating differential algebraic equations of index 3, one of the black beasts of numerical analysis, the discrete principle of least action produces easily implemented methods with extraordinary properties, whilst the standard methods for this problem, either BDF [54] or RADAU5 [114], take enormous amounts of computational power, fail to preserve the symmetries, and are extremely difficult to implement.

4 Regularized and Stabilized Analytic and Discrete Mechanics

The variational stepping methods introduced in Chapter 3 generally require the exact solution of nonlinear systems of equations. This is always required when nonlinear constraints are present and moreover, the well-posedness of the nonlinear equations is dependent on constraint Jacobians having full row rank. This is not compatible with the requirements listed in Section 1.2.

The present chapter addresses this issue by introducing *regularization*—a small perturbation parameter—directly in the physical formulation. This is done by adding small self-potential energy on the ghost variables and by perturbing the Rayleigh dissipation functions. Combining these two techniques makes the time-stepping well conditioned and provides constraint stabilization. In turn, this allows to linearize the stepping equations since resulting constraint errors can be strongly stabilized without diverting the numerical solution too far from that of the idealized unperturbed problem.

After introducing justifications for and problems related to constraint realization in Section 4.1, the case of holonomic constraints is covered in Section 4.2. A suitable discretization of the ghost variables is constructed after taking due consideration of the results of the classical theorem of Rubin and Ungar [241], which demonstrates that penalty forces converge only weakly and generally suffer from high oscillations. An analysis of the realization of nonholonomic constraints leading to a regularized stepping scheme is then presented in Section 4.3. The two results are combined in Section 4.4 to produce a physics based stabilization of holonomic constraints. This is followed by a linearization of stepping schemes for systems containing both types of constraints in Section 4.5, defining the SPOOK stepper. A proof that the constraint stabilization scheme is strongly stable for the case of linearly constrained systems is provided in Section 4.6. Relation to previous work is discussed in Section 4.7.

4.1 Introduction

It was mentioned in Section 3.14 that constraints essentially model the net effect of physical phenomena occuring over time and length scales much smaller than the ones considered in a given system. At the macroscopic level, the high frequency physics should be ignorable, allowing for a reduction of the problem without much loss.

The analysis which yields constraints from integrating over fast oscillatory modes is known as *constraint realization* in the literature and has received consid-

erable attention over the years from both theoretical and numerical perspectives. In the latter case, this is known as *penalty forces*. The purpose of this chapter is to introduce a constraint realization scheme which can be discretized reliably within the variational framework, and which can be interpreted in terms of physical parameters. At the numerical level, this is a *regularization* scheme whereby one solves a perturbed problem with a stable numerical scheme to produce a well behaved solution that is hopefully close to that of the original problem.

The need for regularization is threefold. For one, more realistic systems can be simulated if it is possible to "relax" the constraints, making them compliant or yielding a little. This is closer to the real world. Indeed, there is no such thing as a perfect hinge in nature. Hanging a heavy enough bag on a door knob will make the door scrape the floor, whilst pulling on that door hard enough will rip it off the hinges or off the frame. Forcing exact numerical satisfaction of the geometric constraints, as is common in robotics and mechanical engineering literature, is unnatural. Given that mechanical problems with exact constraint satisfaction are incredibly more difficult to solve efficiently and with decent error bounds, it seems a complete waste of computational power to pursue this route.

Second, the stepping equations (3.95) presented in Section 3.14.3 can only be solved accurately in the case when the Jacobian matrix G has full row rank. There are simple examples such as the planar slider crank mechanism, presented in Chapter 7, which exhibit degenerate configurations where the Jacobian matrix G is rank deficient. When the problem is close to rank deficiency, the numerical errors become unbounded and the solutions can be erratic. Regularization removes such degeneracies in a physical way by introducing compliance, and finite compliance makes the solution easier to compute reliably and efficiently. Finally, a numerical solution of the nonlinear system (3.95) will contain errors of the order of machine precision at best, but typically much larger in case of bad conditioning, ill-posedness, or lack of processing time to produce a high precision result. And that is discounting the fact that the problem to be solved in (3.95) is a non-symmetric saddle point problem, a very unusual animal. Regularization can help average out or bias such errors by keeping condition numbers low for instance, and help keep the overall balance in precision by demanding that only constraint violation averaged over a few steps be forced to vanish. With a good stabilization strategy, these averages can be kept small whilst larger errors can be forced out of existence within a few steps, all the while sticking to nicely behaved problems at the cost of increased modeling power. In the limit, this issue is a realization of the fable of the hare and the tortoise: keep a steady pace instead of rushing frantically.

In more serious terms, the aim with regularization is to first formulate a family of perturbed physical problems, R_ϵ say (R stands for "real"), which are closely related to a given ideal problem R_0 in the sense that the data of R_0 and R_ϵ differ by a term of $O(\epsilon)$ whilst the solutions $S(R_\epsilon)$ converge uniformly to the ideal solution $S(R_0)$ as the perturbation ϵ vanishes, i.e., $\lim_{\epsilon\to 0} S(R_\epsilon) = S(R_0)$. A discrete formulation P_ϵ of the perturbed problem R_ϵ is then constructed which can be solved by a backward stable numerical method M, yielding exact answers

to $P_{\epsilon'}$ so that $|\epsilon - \epsilon'|$ is small to machine precision. If the numerical solutions $N_\epsilon = M(P_\epsilon)$ converge as $\epsilon \to 0$, then, the limit is also a discretized solution of P_0 and a good approximation of R_0. Because every approximation step and every regularization step is nicely behaved, this roundabout way of computing an answer should never be too far off.

An additional consideration is that, typically, the physical problem R_ϵ is well posed as long as $\epsilon \geq 0$. However, the same problem is often nonsensical for $\epsilon < 0$, and has no finite solution then. For instance, a unit mass point particle can be attached to the origin with a harmonic oscillator of frequency $\omega = 1/\sqrt{\epsilon}$, corresponding to a spring constant of $1/\epsilon$. As $\epsilon \to 0$, the particle cannot move away from the origin. However, $\epsilon < 0$, describes an exponential escape trajectory with unbounded energy—no longer a conservative mechanical system.

This means that the solution $S(R_\epsilon)$ cannot be expected to be an analytic function of ϵ. Thus, a sensitivity or error analysis of R_ϵ is in fact expected to break down at R_0, likewise for the numerical problems P_ϵ and P_0. By contrast, if P_ϵ is well posed for $\epsilon > 0$, one expects to be able to analytically continue the solution to extract $S(R_0)$ from $M(P_\epsilon)$. A concrete example of this is the Cholesky factorization algorithm [107] which has good stability properties for symmetric, strictly positive definite matrices and can be extended to process symmetric matrices which are only positive semi-definite at the numerical level by perturbing them "just enough" [125, 126].

The distinction of the present approach to regularization is that we always relate the perturbed discrete problem P_ϵ both to a perturbed *physical problem* R_ϵ and to a backward stable numerical method M. In other words, the only perturbations allowed in P_ϵ are those which can be traced directly to a physical parameter and which improve the performance or stability of the numerical method by allowing the use of a backward stable algorithm. In particular, this scheme is used so that the main computation is the factorization of symmetric, positive definite matrices, for which there exists backward stable numerical methods.

In addition to regularization, we need constraint *stabilization* scheme so that errors made in solving for a constraint equation, $g(q_k) = 0$ at time step k, say, do not accumulate to produce large constraint error $\|g(q_k)\|$ over time, or cause instability. As discussed above for the case of regularization, the aim here is to construct stabilized trajectories corresponding to a nearby *physical* problem, as opposed to an *ad hoc* construction with spurious, non-physical dynamics.

In what follows, analytic formulations of constraint realization is presented for both holonomic and nonholonomic kinematic constraints. It is shown below that holonomic constraints are a limit case of strong potential forces—essentially harmonic oscillators. Nonholonomic constraints on the other hand are the limit of a strong *dissipative* force. In both cases, the limit of the trajectories is well behaved and uniformly convergent but in the case of holonomic constraints, the forces generated by the realizations are only *weakly* convergent, in the sense that time integrals involving the penalty forces do converge but the forces themselves do not. The cause of this is high frequency low amplitude oscillations which fail

to converge but which can easily be averaged away. In consequence, care must be taken in the discretization so that this noise is filtered out of the numerical solution.

The combination of the oscillatory terms generated by the numerical realization of holonomic constraints with the dissipative terms generated by the realization of nonholonomic constraints yield regularized, stabilized discrete stepping schemes. These are proven to be unconditionally stable for the case of linear constraints, though, regrettably, a nonlinear stability analysis is still missing.

As a final note, the weak convergence of highly oscillatory forces is of consequence in molecular dynamics. Indeed, if one replaces very strong molecular forces with exact constraints, the oscillation energy of these modes is lost and the overall thermodynamics properties of the system are altered, since the equipartition theorem states that each degree of freedom should receive equal share of the energy on average [132]. Therefore, using exact constraints in molecular dynamics is problematic though there are resolutions [89, 60].

4.2 Regularization of holonomic constraints

Classical mechanics text books often state as fact that kinematic constraints of the form $g(q) = 0$ can be understood as the limit of a potential function of the form [173, 22]

$$U_\epsilon(q) = \tfrac{1}{2\epsilon}\|g(q)\|^2. \tag{4.1}$$

However, the proof that the trajectories of a mechanical system described by a Lagrangian of the form

$$\mathcal{L}_\epsilon(q,\dot{q}) = \mathcal{L}(q,\dot{q}) - U_\epsilon(q), \tag{4.2}$$

with trajectories $(q_\epsilon(t), \dot{q}_\epsilon(t))$, converges to the trajectories of the *constrained* Lagrangian

$$\mathcal{L}^{(c)} = \mathcal{L} + \lambda^T g(q), \tag{4.3}$$

is rarely provided. In addition, the fact that the constraint forces generated by the potential U_ϵ of (4.1) oscillate with frequency proportional to $\epsilon^{-1/2}$, and converge only in the *weak* (time averaged) sense is seldom stated. As we show below, this is what plagues penalty techniques in which constraints are directly replaced by strong potentials. From the numerical point of view, these high frequencies in the forces are usually catastrophically unstable, unless they are filtered properly.

As was first done in [241], and with somewhat simpler methods in [166], consider a sequence of Lagrangians of the form (4.2) subject to potential functions U_{ϵ_k} of (4.1), where the sequence $\{\epsilon_k\}_{k=0}^{\infty}$ converges with $\lim_{k\to\infty} \epsilon_k = 0$. This implies that $\lim_{k\to\infty} U_{\epsilon_k} = \infty$, except where $g(q) = 0$. To simplify the notation, the Lagrange function $\mathcal{L}(q,\dot{q})$ of (4.2) is taken as $\mathcal{L}(q,\dot{q}) = (1/2)\|\dot{q}\|^2$. This particularly simple form does not affect the theoretical results.

Fixing the initial conditions $(q(0), \dot{q}(0))$ consistent with the constraint so that $g(q(0)) = 0$ and $G(q(0))\dot{q}(0) = 0$, the trajectories $(q_{\epsilon_k}(t), \dot{q}_{\epsilon_k}(t))$ are found to

converge uniformly to the limit trajectory $(q_0(t), \dot{q}_0(t))$, which is a solution of the constrained Lagrangian $\mathcal{L}^{(c)}$ of (4.3). However, the forces generated by the potentials U_{ϵ_k} of (4.1) only converge *weakly* to the constraint forces, $f^{(c)} = G^T\lambda$, computed from the constrained Lagrangian $\mathcal{L}^{(c)}$. The Rubin and Ungar [241] version of the theorem is now stated.

Theorem 4.1 (Rubin and Ungar [241]). *Assuming that conditions a–e below are fulfilled,*

a. *The potential $U(q) : Q \mapsto \mathbb{R}$ is $\mathcal{C}^1$ in some bounded domain $\Omega \in Q$;*

b. *The functions $g_i(q) : Q \mapsto \mathbb{R}, i = 1, 2, \ldots, m$, where $m < n$, are $\mathcal{C}^2$ over Ω such that $\mathcal{M} = \{q \in Q \mid g_i(q) = 0, i = 1, 2, \ldots, m\}$ is non empty and the Jacobian matrix $G_{ij} = \partial g_i / \partial q_j$ has full row-rank except at isolated points in $\Omega \in Q$;*

c. *There exists $q_0 \in Q$ and $(q_0, \dot{q}_0) \in TQ$ with $g(q_0) = 0$ and $G(q_0)\dot{q}_0 = 0$, i.e., the initial position q_0 lies on the manifold $\mathcal{M}$ and the initial velocity is tangent to $\mathcal{M}$;*

d. *The sequence $\{\epsilon_k\}_{k=1}^{\infty}$ satisfies $\epsilon_k \geq 0$ and $\lim_{k\to\infty} \epsilon_k = 0$;*

e. *The sequence of problems $\{A_k\}_{k=1}^{\infty}$ consists of the ordinary differential equations*

$$\ddot{q}_{\epsilon_k} + \nabla U(q_{\epsilon_k}) + \frac{1}{\epsilon_k} \nabla \sum_{i=1}^{m} \left(g_i(q_{\epsilon_k})\right)^2 = 0 \tag{4.4}$$

with initial conditions

$$q_{\epsilon_k}(0) = q_0, \qquad \dot{q}_{\epsilon_k}(0) = \dot{q}_0.$$

Then, the following statements A–G hold:

A. *There exists a positive number δ and a sequence of functions $q_k : [0, \delta] \mapsto Q, k = 1, 2, \ldots,$ such that $q_k(t)$ solves problem A_k;*

B. *The sequence $\{q_{\epsilon_k}(t)\}_{k=1}^{\infty}$ converges uniformly to a continuous function, $q : [0, \delta] \mapsto Q$;*

C. *The functions $g^{(i)}(q(t)) = 0, i = 1, 2, \ldots, m$, vanish identically in $t \in [0, \delta]$;*

D. *The function $q : [0, \delta] \mapsto Q$ is $\mathcal{C}^2$;*

E. *The sequence $\{(q_{\epsilon_k}(t), \dot{q}_{\epsilon_k}(t))\}_{k=1}^{\infty}$, $(q_{\epsilon_k}, \dot{q}_{\epsilon_k}) : [0, \delta] \mapsto TQ$ converges uniformly to $(q, \dot{q}) : [0, \delta] \mapsto TQ$ on $[0, \delta]$;*

F. *The initial conditions $q(0) = q_0$ and $\dot{q}(0) = q_0$ hold;*

G. *There exist continuous functions $\lambda^{(i)} : [0, \delta] \mapsto \mathbb{R}, i = 1, 2, \ldots, m$, such that*

$$\ddot{q} + \nabla U(q) + G^T\lambda = 0 \tag{4.5}$$

identically on $[0, \delta]$, i.e., so that the limit trajectory $(q(t), \dot{q}(t))$ satisfies the Euler-Lagrange equations of the constrained problem.

The proof is omitted for being far too long and technical. What is most interesting to note is that the constraint forces generated with this procedure are *weak*$-*$ convergent with respect to time integration so that

$$\begin{aligned} \lambda_{\epsilon_k}(t) &= -\frac{1}{\epsilon_k} g(q_{\epsilon_k}(t)), \text{ and} \\ \lim_{k\to\infty} \int_0^\delta \mathrm{d}s \lambda_{\epsilon_k}^T(s) h(s) &= \int_0^\delta \mathrm{d}s \lambda^T(s) h(s), \end{aligned} \tag{4.6}$$

for any integrable function $h(s) : [0, \delta] \mapsto \mathbb{R}^m$. Indeed, it is expected in general—see Chap. 8 for a concrete example—that λ_{ϵ_k} contains sinusoidal terms with frequencies proportional to $1/\sqrt{\epsilon_k}$ and fixed amplitude.

The analysis of Rubin and Ungar [241] shows that naively replacing a constraint $g(q) = 0$ with a strong potential of the form $U_\epsilon(q) = (2\epsilon)^{-1} g^T g$ generates high oscillations which must be damped by the integration methods if the results are to make any sense at all. Concrete examples demonstrating the failure of such numerical constraint realizations, even when using the stable implicit mid-point rule, are presented in [30].

The regularization starts from the analysis of the extended variables Lagrangian presented in Section 3.14.1, and particularly from the augmented Lagrangian (3.93), namely $\bar{\mathcal{L}}(q, \dot{q}, \lambda, \dot{\lambda}) = \mathcal{L}(q, \dot{q}) + \lambda^T g(q)$. As previously argued, the ghost variables λ are to be considered as full fledged dynamical variables. In some sense, each variable λ_i corresponds to the coordinate of a one-dimensional point particle with *zero mass*. Adding the self-potential term $(\epsilon/2)\lambda^T\lambda$ to the augmented Lagrangian (3.93), the resulting Euler-Lagrange equations of motion become

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathcal{L}}{\partial \dot{q}^T} - \frac{\partial \mathcal{L}}{\partial q^T} - G^T\lambda &= 0 \\ \epsilon\lambda + g(q) &= 0. \end{aligned} \tag{4.7}$$

Obviously, the second equation can be solved for $\lambda = -\epsilon^{-1} g(q)$ and this can be substituted back to yield

$$\mathcal{L}_\epsilon(q, \dot{q}) = \mathcal{L}(q, \dot{q}) - \frac{1}{2\epsilon} g^T(g) g(q), \tag{4.8}$$

which establishes a direct correspondence with Theorem 4.1. However, this form of the regularization is fundamentally different from the usual form (see for instance, in [114], Section VI.3), where the second equation of (4.7) is replaced with $\epsilon\dot{\lambda} + g(q) = 0$, thus adding spurious dynamics to the λ pseudo particles. Indeed, no term can be added to the Lagrangian to yield that equation of motion using the variational method.

Other formulations include the addition of a kinetic energy term of the form $-(\epsilon/2)\dot{\lambda}^T\dot{\lambda}$. Note note the minus sign in this expression which justifies the *ghost* terminology advocated here. Now, such a ghost kinetic energy changes the second line in the equations of motion (4.7) for $\epsilon\ddot{\lambda} + g(q) = 0$. This is used in [83], among many other instances. This form does not map to the framework of Theorem 4.1 and it is not clear that it produces the correct dynamics at all.

One must understand that the motivation to introduce either $\epsilon\dot{\lambda}$ or $\epsilon\ddot{\lambda}$ to the dynamics is so that the DAE of motion of the constrained systems becomes an ODE and thus, can be discretized correctly, as done in [114] for instance. The reason why we can do without derivatives is because the variational principle provides the discretization.

To discretize the regularized Lagrangian, let us first write the following

$$\begin{aligned} \mathcal{L}(q,\lambda,\dot{q},\dot{\lambda}) &= \mathcal{L}_0(q,\dot{q}) + \epsilon\mathcal{L}_1(\lambda,\dot{\lambda}) + \mathcal{L}_{01}(q,\lambda,\dot{q},\dot{\lambda}), \\ \mathcal{L}_1(\lambda,\dot{\lambda}) &= \frac{1}{2}\lambda^T\lambda, \\ \mathcal{L}_{01}(q,\lambda,\dot{q},\dot{\lambda}) &= \lambda^T g(q), \end{aligned} \tag{4.9}$$

dropping the ϵ subscripts, and proceed to evaluate discrete expressions for $\mathcal{L}_1$ and $\mathcal{L}_{01}$. In view of the high oscillations and weak limit of λ as $\epsilon \to 0$, we approximate the integrals using symmetric averages. To correctly approximate $\mathcal{L}_{01}$, we first recall that we are considering λ to be an *independent* variable at this stage of the process so that $\mathcal{L}_{01} \approx h < \lambda >^T < g >$ where $< \cdot >$ denotes the time average over the time interval h. This leads to the following approximations

$$\begin{aligned} \mathbb{L}_{d1} &= \int_0^h \mathrm{d}s \frac{1}{2}\|\lambda\|^2 \approx \frac{h}{8}\|\lambda_0 + \lambda_1\|^2, \\ \mathbb{L}_{d01} &= \int_0^h \mathrm{d}s \lambda^T g(q) \approx \frac{h}{8}(\lambda_0 + \lambda_1)^T (g_0 + g_1), \end{aligned} \tag{4.10}$$

from which the discrete time stepping Euler-Lagrange equations are extracted

$$\begin{aligned} D_1^T\mathbb{L}_0(q_k,q_{k+1}) + D_2^T\mathbb{L}_0(q_{k-1},q_k) + hG_k^T\bar{\lambda} &= 0 \\ \epsilon\bar{\lambda} + \frac{1}{4}(g_{k+1} + 2g_k + g_{k-1}) &= 0, \end{aligned} \tag{4.11}$$

where we relabeled the λ variables as

$$\bar{\lambda} = \frac{1}{4}(\lambda_{k+1} + 2\lambda_k + \lambda_{k-1}). \tag{4.12}$$

Averages are used since there is no need to keep track of the λ_k variables directly from step to step—except perhaps if one wants to keep track of the ghost energy in the system.

Observe also that we recover precisely the same stepping equations as in the standard constrained case (3.95) as $\epsilon \to 0$, as long as the initial conditions $g(q_0) = g(q_1) = 0$ are imposed. Indeed, once we set $\epsilon = 0$ and $g(q_0) = g(q_1) = 0$, the stepping equation (4.11) sequentially enforces $g(q_k) = 0$ for $k \geq 2$. There only remains a labeling difference between (4.11) and the original derivation (3.95).

In addition, the numerical solutions of (4.11) are expected to yield $\bar{\lambda}$ which differ by ϵ from the unperturbed problem. To see this, consider the case where the Jacobian $G_k = G$ is constant and the basic Lagrangian $\mathcal{L}$ has constant mass matrix as in (3.12). Solving (4.11) requires the solution of a linear system $S_\epsilon\lambda_\epsilon = \bar{b}$, with matrix $S_\epsilon = GM^{-1}G^T + \epsilon I$, and some vector $\bar{b}$, whereas solution

of the unperturbed problem (3.95) requires the solution of the problem $S_0\lambda = b$, with $S_0 = GM^{-1}G^T$ and for a vector b which differs from $\bar{b}$ by terms of second order in h.

Whenever G has full row rank, S_ϵ is symmetric and positive definite and so is the limit case S_0. Therefore, the solution vectors λ and λ_ϵ differ by order $O(\epsilon) + O(h^2)$.

Thus the regularized numerical solution is neither far from the exact solution of the regularized problem nor from the numerical solution of the exact problem.

In other words, the following diagram commutes.

$$\begin{array}{ccc} \bar{\mathcal{L}} & \xrightarrow{\text{regularize: } \epsilon > 0} & \mathcal{L}^{(\epsilon)} \\ \Big\downarrow \text{discretize: } h{>}0 & & \Big\downarrow \text{discretize: } h{>}0 \\ \bar{\mathbb{L}}_d & \xleftarrow{\text{realization: } \epsilon \to 0} & \mathbb{L}_d^{(\epsilon)} \end{array} \tag{4.13}$$

The simplest approximation of the second equation of (4.11) consists of expanding $g_{k\pm1}$ to first order in terms g_k. This is done as

$$\begin{aligned} g_{k\pm1} &= g(q_{k\pm1} = g(q_k + (q_{k+\pm1} - q_k)) \\ &= g(q_k) + G_k(q_{k+\pm1} - q_k) \\ &= g_k + G_k(q_{k+\pm1} - q_k). \end{aligned} \tag{4.14}$$

From this, the result is

$$\frac{1}{4}\left(g_{k+1} + g_k + g_{k-1}\right) \approx g_k + \frac{h}{4} G_k \left(q_{k+1} - 2q_k + q_{k-1}\right). \tag{4.15}$$

After introducing $\bar{\lambda} = h^2\tilde{\lambda}$, and specializing to the constant mass model Lagrangian (3.12), the stepping equation is defined by the linear system

$$\begin{bmatrix} M & -G_k^T \\ G_k & \frac{4\epsilon}{h^2} \end{bmatrix} \begin{bmatrix} q_{k+1} \\ \bar{\lambda} \end{bmatrix} = \begin{bmatrix} M\left[2q_k - q_{k-1}\right] - h^2 \frac{\partial V}{\partial q^T}_k \\ -4g_k - G_k\left[2q_k - q_{k-1}\right] \end{bmatrix}. \tag{4.16}$$

Observe now that the matrix involved in (4.16) is strictly positive definite, though non-symmetric, for any value of $\epsilon > 0$. A linear stability analysis of this scheme is presented in Section 4.6.

The usefulness of a regularized scheme without any stabilization is limited since errors made while solving the nonlinear stepping equations can quickly accumulate and cause wild oscillations. Regularization of nonholonomic constraints is now introduced so that a scheme for regularizing the implied velocity constraints $G\dot{q} = 0$ can be constructed. This will be used to impose both $g(q_k) \approx 0$ and $G_k(q_k - q_{k-1})/h \approx 0$ simultaneously.

4.3 Regularization of nonholonomic constraints

As was noted in the Section 4.2, the standard mechanics texts often mention the correspondence between holonomic constraints and strong forces. However,

the correspondence between nonholonomic constraints and the limit of strong Rayleigh dissipation functions is less well known and seldom quoted, except perhaps in more recent monographs [23].

An analog of the proof or Rubin and Ungar [241] for nonholonomic systems had to wait until the early 1980s for a demonstration [55, 153], using two different techniques, but with identical results, as quoted in Theorem 4.2.

Theorem 4.2 (Brendelev [55] and Karapetian [153]). *Given a function, $a : \mathbb{R} \times TQ \mapsto \mathbb{R}^m$, of the form: $a(q, \dot{q}, t) = A(q, t)\dot{q} + w(t)$, where A is an $m \times n$ real matrix valued function. Assuming that a is smooth and differentiable, there exists a sequence of numbers $\{\gamma_i\}_{i=1}^{\infty}$ with $\gamma_i > 0, \gamma_i \to 0$ as $i \to \infty$, such that the motion of a Lagrangian system with $\mathcal{L}(q, \dot{q}, t)$ subject to the Rayleigh dissipation function*

$$\mathfrak{R}_{\gamma_i} = \frac{1}{2\gamma_i} a^T a, \tag{4.17}$$

having the trajectories $(q_{\gamma_i}(t), \dot{q}_{\gamma_i}(t))$, converge uniformly over a time interval $[0, \delta]$ to the trajectory of the Lagrangian system $\mathcal{L}(q, \dot{q}, t)$ subject to the nonholonomic constraints $a(q, \dot{q}, t) = 0$.

The proof hinges around Tykhonov's theorem on separation of slow and fast variables (see [186] for instance). Note that, as per the analysis provided in Section 3.12, the rate of decrease of energy is $-(1/\gamma)\dot{q}^T A^T (A\dot{q} + w(t))$ and so, for the homogeneous case $w(t) = 0$, given that $A^T A$ is symmetric and positive semi-definite, the regularized system dissipates energy until it reaches the constraint surface $A(q, t)\dot{q} = 0$.

It is noted in [153] that the convergence is independent of the initial conditions so in fact, the theorem should state uniform convergence on the *open* interval $(0, \delta)$. This contrasts with Theorem 4.1 on realization of holonomic constraints where satisfaction of constraints at the initial time was essential. The constraint forces are not expected to exhibit highly oscillatory terms should thus converge directly, in contrast to the weak convergence of the holonomic constraint case.

Note also that Rayleigh dissipation functions of the form $\mathfrak{R} = 2\gamma^{-1}\dot{q}^T D\dot{q}$, for a square, symmetric, positive semi-definite $n \times n$ matrix D corresponds to a linear *viscous drag* force. When dry friction is considered in Chapter 10, the corresponding nonholonomic constraint reads $D\dot{q} = 0$ for a suitable matrix D, and regularization thus introduces a very small creep velocity of the order γ, which is a form of viscous friction. Though this is undesirable, this creep can be kept at the order of machine precision since γ only serves to guard against degeneracy and thus, the result is stable stable friction, within achievable accuracy. Essentially, the discretization strategy saves the flawed idea of replacing dry friction terms with viscous drags which has been reported numerous times in the literature but is also known to be a spectacular failure, mostly because the viscous drag coefficient, $1/\gamma$, cannot be made large enough to recover the dry friction mode without introducing stability problems.

To use this new theorem, introduce the auxiliary variable α and write

$$\mathfrak{R}_\gamma(q,\dot{q},\alpha,\dot{\alpha},t) = -\frac{\gamma}{2}\dot{\alpha}^T\dot{\alpha} - \dot{\alpha}^T a(q,\dot{q},t). \tag{4.18}$$

This leads to the equations of motion

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathcal{L}}{\partial \dot{q}^T} - \frac{\partial \mathcal{L}}{\partial q^T} - A^T\dot{\alpha} &= 0 \\ \gamma\dot{\alpha} + a(q,\dot{q},t) &= 0, \end{aligned} \tag{4.19}$$

which are are satisfied by the trajectories of the regularized system which is now denoted as the pair $(\mathcal{L}(q,\dot{q}),\mathfrak{R}_\gamma(q,\dot{q},t))$ or simply $(\mathcal{L},\mathfrak{R}_\gamma)$.

A simple substitution using the second equation shows that the new Rayleigh function is strictly equivalent to the original definition (4.17) in Theorem 4.2. As this form of constraint realization only introduces first order dynamics—exponential decay—high oscillations of the constraint forces are not expected.

Therefore, in the limit case where $\gamma = 0$, the regularized equations of motion (4.19) produce the trajectories of the *restricted* Lagrangian, written as $\mathcal{L}(q,\dot{q})|_{a(q,\dot{q},t)=0}$. The discretized version of the restricted Lagrangian is written $\mathbb{L}_d|_{a(q,\dot{q},t)=0}$ and the numerical trajectories are computed by the discrete stepping equations (3.105).

The regularized physical system (4.19) is now discretized using the framework introduced in Section 3.12. The discrete forces acting on the q and α variables are as follows as follows

$$\begin{aligned} f_{d,q}^{(+)}(q_0,q_1,\alpha_0,\alpha_1) &= \int_0^h \mathrm{d}t \frac{\partial \mathfrak{R}_\gamma}{\partial \dot{q}^T}\frac{\partial q(t)}{\partial q_1} = 0 \\ f_{d,q}^{(-)}(q_0,q_1,\alpha_0,\alpha_1) &= \int_0^h \mathrm{d}t \frac{\partial \mathfrak{R}_\gamma}{\partial \dot{q}^T}\frac{\partial q(t)}{\partial q_0} = hA_0\left(\frac{\alpha_1-\alpha_0}{h}\right) \\ f_{d,\alpha}^{(+)}(q_0,q_1,\alpha_0,\alpha_1) &= \int_0^h \mathrm{d}t \frac{\partial \mathfrak{R}_\gamma}{\partial \dot{\alpha}^T}\frac{\partial \alpha(t)}{\partial \alpha_1} = 0 \\ f_{d,\alpha}^{(-)}(q_0,q_1,\alpha_0,\alpha_1) &= \int_0^h \mathrm{d}t \frac{\partial \mathfrak{R}_\gamma}{\partial \dot{\alpha}^T}\frac{\partial \alpha(t)}{\partial \alpha_0} \\ &= h\left(\gamma\left(\frac{\alpha_1-\alpha_0}{h}\right) + a(\tilde{q},\frac{q_1-q_0}{h},0)\right). \end{aligned} \tag{4.20}$$

The choice of approximation for $q(t) = u_d(q_0,q_1,t,h)$ that is made to compute $f_{h,q}^{(\pm)}(q_0,q_1,\alpha_0,\alpha_1)$ does not affect the choice of the approximation of the last two integrals $f_{h,\alpha}^{(\pm)}(q_0,q_1,\alpha_0,\alpha_1)$. The choice of the value of $\tilde{q}$ in the last equation of (4.20) is set to q_1 so the known unregularized nonholonomic stepper of (3.105) is recovered in the limit where $\gamma \to 0$. This leads to the following discrete stepper by using the forced, discrete, Euler-Lagrange equations (3.82)

$$\begin{aligned} D_1\mathbb{L}_d(q_k,q_{k+1},h) + D_2\mathbb{L}_d(q_{k-1},q_k,h) + A_k^T(\alpha_{k+1}-\alpha_k) &= 0 \\ \gamma(\alpha_{k+1}-\alpha_k) + ha(q_{k+1},\frac{q_{k+1}-q_k}{h},hk) &= 0, \end{aligned} \tag{4.21}$$

and this is referred to as the pair $(\mathbb{L}_d, f_d^{(\pm)}(\mathfrak{R}_\gamma))$.

Now, the numerical trajectories of the pair $(\mathbb{L}_d, f_d^{(\pm)}(\mathfrak{R}_\gamma))$ computed by solving the stepping equations (4.21) are expected to differ by $O(\gamma)$ from those of the discrete restricted problem, $\mathbb{L}_d|_{a(q,\dot{q},t)=0}$ computed by (3.105), at least as long as the Jacobian matrices A_k have full row rank. Indeed, in that case at least, the nonlinear systems are well conditioned. In addition, the same regularized numerical solutions of (4.21) are expected to be close within $O(h^2)$ of the trajectories of the regularized system, $(\mathcal{L}, \mathfrak{R}_\gamma)$ for a fixed value of $\gamma > 0$. Finally, the result of Theorem 4.2 says that trajectories of $(\mathcal{L}, \mathfrak{R}_\gamma)$ are uniformly close within $O(\gamma)$ to the trajectories of the restriction $\mathcal{L}_{a(q,\dot{q},t)=0}$.

These facts indicate that the following diagram commutes, though some steps are still missing for a rigorous proof.

$$\begin{array}{ccc} \mathcal{L}|_{a(q,\dot{q},t)=0} & \xrightarrow{\text{regularize: } \gamma > 0} & (\mathcal{L}, \mathfrak{R}_\gamma) \\ \Big\downarrow \text{discretize: } h{>}0 & & \Big\downarrow \text{discretize: } h{>}0 \\ \mathbb{L}_d|_{a(q,\dot{q},t)=0} & \xleftarrow{\text{realization: } \gamma \to 0} & (\mathbb{L}_d, f_d^{(\pm)}(\mathfrak{R}_\gamma)) \end{array} \tag{4.22}$$

4.4 Physical stabilization of holonomic constraints

Armed with discretizations of both holonomic and nonholonomic regularized constraints constructed in Section 4.2 and Section 4.3, respectively, the problem of *stabilizing* holonomic constraints by combining the two aspects is now investigated. First observe that a constraint of the form $g : Q \mapsto \mathbb{R}^m, g(q) = 0$, *implies* that $\dot{g}(q) = G(q)\dot{q}$, where the $n \times n$ Jacobian matrix G is defined as $G = \partial g/\partial q$ at the analytic level. However, this relation does not survive discretization since even with $g(q_k) \approx 0, g(q_{k+1}) \approx 0$, then, using Taylor's theorem

$$\frac{1}{h}\left(g(q_{k+1}) - g(q_k)\right) = \frac{1}{h}G_k(q_{k+1} - q_k) + O(h), \tag{4.23}$$

so that even nailing $g(q_{k+1}) = g(q_k) = 0$ exactly, the approximation of the constraint velocity might only vanish to $O(h)$.

Strangely enough, the notion keeps appearing in the literature that constraint drift is due to *round-off error*, which is typically of the order of machine precision. This is utterly negligible in comparison to the *discretization* errors of the type just described. It should be clear also that if a local linear approximation is used to compute $g(q) \approx 0$, the errors made are $O(h)$ which makes the constraint velocity estimate $O(1)$ as per (4.23). Severe constraint drift is thus expected over time.

Secondly, it seems reasonable to relax the accuracy requirements on the solution of $g(q_k) \approx 0$ to balance it with the overall error made by the stepper which is expected to be $O(h^2)$ at best. In fact, it should be best to keep both $\|g(q)\| = O(h)$ *and* $\|G(q)\dot{q}\| = O(h)$ along the trajectory. This argument often

appears in the finite element literature [52] but seldom in the ODE or DAE literature. In fact, in [114], Section VI, where solution methods for DAEs are exposed in detail, the notion of violating the constraints $g(q) = 0$ is never considered and no effort is spared to satisfy $g(q) = 0$ to machine precision. This leads to solving *degenerate* systems of nonlinear equations and in turn, this requires using either SVD, or QR which are expensive computationally compared to regular factorization methods such as LU factorization (LU) or Cholesky [107].

The strategy adopted here is to consider $g(q) = 0$ and $G(q)\dot{q} = 0$ as *independent* constraints, to be enforced separately. To this end, the following regularization and coupling terms are introduced

$$\begin{aligned} \mathcal{L}_{01} = \lambda^T g(q) &\Longrightarrow \mathbb{L}_{d,01} = \frac{1}{8}(\lambda_0 + \lambda_1)^T (g_0 + g_1)\,, \\ \mathcal{L}_1 = \frac{\epsilon}{2}\|\lambda\|^2 &\Longrightarrow \mathbb{L}_{d,1} = \frac{\epsilon}{8}\|\lambda_0 + \lambda_1\|^2, \end{aligned} \tag{4.24}$$

and the Rayleigh functions and associated discrete forces

$$\begin{aligned} \mathfrak{R}(q,\dot{q},\lambda,\dot{\lambda}) = -\tau\left(\frac{\epsilon}{2}\dot{\lambda}^T\dot{\lambda} + \lambda^T G(q)\dot{q}\right) \Longrightarrow \\ f_{d,q}^{(+)} = 0, \qquad f_{d,q}^{(-)} = h\tau G_k \frac{\lambda_1 - \lambda_0}{h}, \\ f_{d,\lambda}^{(+)} = 0, \qquad f_{d,\lambda}^{(-)} = h\tau\left(\epsilon\frac{\lambda_1 - \lambda_0}{h} + G_1\frac{q_1 - q_0}{h}\right). \end{aligned} \tag{4.25}$$

With these choices, after relabeling with

$$\lambda = \frac{1}{4}\left[\lambda_{k+1} + 2\lambda_k + \lambda_k\right] + \frac{\tau}{h}(\lambda_{k+1} - \lambda_k), \tag{4.26}$$

the stepping equations for $\mathcal{L} + \mathcal{L}_1 + \mathcal{L}_{01}$ subject to the forces derived from $\mathfrak{R}(q,\dot{q},\lambda,\dot{\lambda})$, are found to be

$$\begin{aligned} D_1\mathbb{L}_d(q_k, q_{k+1}, h) + D_2\mathbb{L}_d(q_{k-1}, q_k, h) + hG_k{}^T\lambda = 0 \\ \epsilon\lambda + \frac{1}{4}\left(g_{k+1} + 2g_k + g_{k-1}\right) + \frac{\tau}{h}G_{k+1}(q_{k+1} - q_k) = 0. \end{aligned} \tag{4.27}$$

This provides for a natural parametrization of the dissipation rate parameter τ, namely, τ/h is the half life of the constraint violation decay.

4.5 Linearized mixed systems: spook

The stepping equations derived so far for various types of systems such as (4.27), are systems of nonlinear equations in the general case. It is very instructive to compute the linearized form of these equations since this is what is ultimately processed, whether only the first linear estimate is used, or whether this is refined using any form of Newton-Raphson iterations.

The local linear approximation is now constructed explicitly, in both position and velocity formulations, for a general system containing both holonomic and

nonholonomic constraints as well as potential functions. The analysis is restricted to constant mass matrices. The changes that must be made to this formulation to cover the rigid body cases where the mass matrix is configuration dependent are covered in Chapter 15.

The linearized version of the regularized and stabilized stepping scheme for mixed systems with both holonomic and nonholonomic constraint needs a short name for convenience. With blatant disregard for paganism and superstition, it is hereby christened SPOOK. The reason for the name is that previous instances of variational steppers for constrained systems were called RATTLE [242] for the simple extension of the Verlet stepping scheme [271] to constrained system, and SHAKE [5] for a velocity formulation of the same based on Hamiltonian mechanics. The implementation of these steppers in molecular dynamics involves using approximate solutions of the constraints and thus, the method developed here is much in line with that development. Since "rattle" and "shake" are both verbs and nouns, "spook" fits in the nomenclature scheme. Now, Lagrange multipliers are classical equivalents to the ghost particles of quantum field theory [265]. Regularization and stabilization provided them with real energy, making them manifest themselves. Since to spook is to appear as a ghost, the name SPOOK is appropriate here.

Consider a system moving configuration manifold Q of dimension n according to the constant-mass model Lagrangian function of (3.12), namely, $\mathcal{L}(q,\dot{q}) = (1/2)\dot{q}^T M\dot{q} - V(q)$, where the square $n\times n$ matrix M is constant and symmetric positive definite, and $V : Q \mapsto \mathbb{R}$ is a smoothly differentiable function of q. Assume the system is subject to nonholonomic constraints of the form $a : \mathbb{R}\times TQ \mapsto \mathbb{R}^{m_n}$, with $a(q,\dot{q},t) = A(q)\dot{q} + w(t) = 0$, where matrix A is of size $m_n \times n$, and the vector function $w : \mathbb{R} \mapsto \mathbb{R}^{m_n}$ is m_n-dimensional. In addition, assume holonomic constraints of the form $g : Q \mapsto \mathbb{R}^{n_h}$, with $g(q) = 0$, and Jacobian $m_h \times n$ matrix $G = \partial g/\partial q$.

For each nonholonomic constraint function $a_i(q,\dot{q},t) = 0, i = 1,2,\ldots,m_n$, define a regularization parameter $\gamma_i \geq 0$. Likewise, for each holonomic constraint function $g_i(q) = 0, i = 1,2,\ldots,m_h$, define the regularization and stabilization parameters, $\epsilon_i, \tau_i \geq 0, i = 1,2,\ldots,m_h$, respectively.

Next, the simplest discretized Lagrangian computed previously in (3.25) is used, yielding the following basic stepping equations as shown in Section 3.7,

$$D_1^T\mathbb{L}_d(q_k,q_{k+1},h) + D_2^T\mathbb{L}_d(q_{k-1},q_k,h) = \\ -\frac{1}{h}M(q_{k+1}-2q_k+q_{k-1}) - h\nabla V(q_k) = 0. \quad (4.28)$$

After linearizing all the constraint terms in the second line of (4.27), expanding all terms about time step k, and reorganizing the explicit terms of (4.28), the linearized stepping equations become

$$\begin{bmatrix} M & -A_k^T & -G_k^T \\ A_k & \Gamma & 0 \\ G_k & 0 & \Sigma \end{bmatrix} \begin{bmatrix} q_{k+1} \\ \alpha \\ \lambda \end{bmatrix} = \begin{bmatrix} 2Mq_k - Mq_{k-1} - h^2\nabla V_k \\ A_kq_k + hw_k \\ -4\Upsilon g_k + \frac{1}{2}(I+\Upsilon)G_kq_k - \Upsilon G_kq_{k-1} \end{bmatrix}, \quad (4.29)$$

where the following square diagonal matrices were introduced

$$\begin{aligned}
\Gamma &= \frac{1}{h}\,\mathrm{diag}(\gamma_1, \gamma_2, \ldots, \gamma_{m_n}),\\
\Sigma &= \frac{4}{h^2}\,\mathrm{diag}(\frac{\epsilon_1}{1+4\frac{\tau_1}{h}}, \frac{\epsilon_2}{1+4\frac{\tau_2}{h}}, \ldots, \frac{\epsilon_m}{1+4\frac{\tau_{m_n}}{h}}),\\
\Upsilon &= \mathrm{diag}(\frac{1}{1+4\frac{\tau_1}{h}}, \frac{1}{1+4\frac{\tau_2}{h}}, \ldots, \frac{1}{1+4\frac{\tau_{m_n}}{h}}),
\end{aligned} \tag{4.30}$$

and the Lagrange multipliers have been redefined to absorb a factor of h for α and h^2 for λ.

When using a velocity formulation, the linearized stepping equations become

$$\begin{bmatrix} M & -A_k^T & -G_k^T \\ A_k & \Gamma & 0 \\ G_k & 0 & \Sigma \end{bmatrix} \begin{bmatrix} v_{k+1} \\ \bar{\alpha} \\ \bar{\lambda} \end{bmatrix} = \begin{bmatrix} Mv_k + h\nabla V_k \\ w_k \\ -\frac{4}{h}\Upsilon g_k + \Upsilon G_k v_k \end{bmatrix}, \tag{4.31}$$

where the matrices Γ, Σ and Υ are defined as previously in (4.30). The difference here is that no factor of h is absorbed in the definition of $\bar{\alpha}$, whilst only one factor of h is absorbed in the new definition of λ. The specific formulation in (4.31) is the SPOOK stepping scheme.

Now define the matrix

$$H = \begin{bmatrix} M & -A_k^T & -G_k^T \\ A_k & \Gamma & 0 \\ G_k & 0 & \Sigma \end{bmatrix}, \tag{4.32}$$

which is common to both the position formulation of (4.29) or the velocity formulation of (4.31). This matrix H is not symmetric but it is strictly positive definite as long as all parameters γ_i and ϵ_i are strictly positive. Indeed, the symmetric part is positive definite for $\gamma_i > 0$. For the partitioned vector $w = (x^T, y^T, z^T)^T$ of appropriate dimension yields

$$w^T H w = x^T M x + y^T \Gamma y + z^T \Sigma z \geq 0, \tag{4.33}$$

and given the assumption that mass matrix M is symmetric and positive definite, the equality is satisfied only when all components $w_i = 0$.

If Newton-Raphson iterations are used, the iteration matrix is easily verified to change to

$$\begin{bmatrix} M & -A_k^T & -G_k^T \\ A_{k'} & \Gamma & 0 \\ G_{k'} & 0 & \Sigma \end{bmatrix}, \tag{4.34}$$

where $q_{k'}$ is the current estimate for the final position q_{k+1} and $A_{k'}$ and $G_{k'}$ are the Jacobian matrices evaluated at $q_{k'}$. Strict symmetry is lost but the symmetric structure is preserved.

The simple linearized form of the velocity formulation (4.31) has been used in practice. Future work will determine when it makes sense to compute a better approximation of the nonlinear equations.

4.6 Linear stability analysis

To understand the properties of the stepping equation (4.11), consider the case of linear homogeneous constraints $g(q) = Gq = 0$ with constant $m \times n$ matrix G. Define $\theta_j = \tau_j/h, j = 1, 2, \ldots, m$, and $\Theta = \mathsf{diag}(\theta_1, \theta_2, \ldots, \theta_m)$. Relabel the Lagrange multipliers of the constraints, λ, to absorb a factor of h as done in (4.31), and reuse the definition of the diagonal matrices with strictly positive entries Σ and Υ of (4.30). Starting with the constant mass model Lagrangian of (3.12), define the generalized forces as usual with $f_k = -\partial V/\partial q_k^T$ and thus, the SPOOK stepper of (4.27) expressed in terms of positions q_k and velocities $v_k = (1/h)(q_k - q_{k-1})$ is

$$\begin{aligned} q_{k+1} - hv_{k+1} &= q_k \\ Mv_{k+1} - Mv_k - G^T\lambda &= hf_k \\ \frac{h}{4}\Upsilon^{-1}\Sigma\lambda + \frac{1}{4}G\left(q_{k+1} + 2q_k + q_{k-1}\right) + h\Theta Gv_{k+1} &= 0. \end{aligned} \tag{4.35}$$

Multiplying the first equation with GM^{-1} and isolating the term Gv_{k+1} in both equations, the following identities are revealed

$$\begin{aligned} Gv_{k+1} &= Gv_k + GM^{-1}G^T\lambda + h^2GM^{-1}f_k, \\ Gv_{k+1} &= -\Sigma\lambda - \frac{4}{h}\Upsilon Gq_k - \Upsilon Gv_k. \end{aligned} \tag{4.36}$$

The following definitions will simplify what follows

$$S = GM^{-1}G^T, \tag{4.37}$$
$$S_\epsilon = S + \Sigma, \tag{4.38}$$
$$K_\epsilon = SS_\epsilon^{-1} = (S_\epsilon - \Sigma)S_\epsilon^{-1} = I_m - \Sigma S_\epsilon^{-1}, \tag{4.39}$$
$$x_k = Gq_k, \tag{4.40}$$
$$y_k = hGv_k, \tag{4.41}$$
$$l_k = h^2GM^{-1}f_k. \tag{4.42}$$

Using these and grouping the λ terms between the two lines in (4.36), one finds

$$S_\epsilon\lambda = -\frac{4}{h}\Upsilon x_k + \frac{1}{h}(\Upsilon - I_m)y_k - \frac{1}{h}l_k. \tag{4.43}$$

Given that S is symmetric and positive semi-definite at least, S_ϵ is strictly positive definite and invertible. Solving (4.43) for λ and substituting in the first line of (4.36), the stepping equation is now reduced to

$$\begin{aligned} x_{k+1} - y_{k+1} &= x_k \\ y_{k+1} &= (I_m - 4K_\epsilon\Upsilon\Theta)y_k - 4K_\epsilon\Upsilon x_k + (I_m - K_\epsilon)l_k, \end{aligned} \tag{4.44}$$

and the factor of $1/h$ of (4.43) was absorbed in the definition of $y_k = hGv_k$ as previously explained.

The force term l_k is dropped at this point to understand the stability of the homogeneous equation. First rewrite (4.44) without forces as the recurrence

$$\begin{aligned} Bz_{k+1} &= Az_k, \qquad \text{or} \\ z_{k+1} &= Hz_k, \qquad \text{with } z_k = \begin{bmatrix} x_k \\ y_k \end{bmatrix}, \end{aligned} \tag{4.45}$$

and with the definitions:

$$\begin{aligned} B &= \begin{bmatrix} I_m & -I_m \\ 0 & I_m \end{bmatrix}, \qquad B^{-1} = \begin{bmatrix} I_m & I_m \\ 0 & I_m \end{bmatrix}, \\ A &= \begin{bmatrix} I_m & 0 \\ -D & I_m - D\Theta \end{bmatrix}, \\ D &= 4K_\epsilon \Upsilon = 4(I_m - \Sigma S_\epsilon^{-1})\Upsilon, \qquad \text{and} \\ H &= B^{-1}A = \begin{bmatrix} I_m - D & I_m - D\Theta \\ -D & I_m - D\Theta \end{bmatrix}. \end{aligned} \tag{4.46}$$

Convergence is thus guaranteed if the spectral radius of H is within the unit circle, i.e., $\rho(H) < 1$. Consider an eigenvalue λ of H with eigenvector $z = (x^T, y^T)^T$. A short computation yields that $\lambda y = (1-\lambda)x$, and so, in terms of the vector x only, the eigenvalue λ must satisfy the following vector equation

$$\lambda^2 x - 2\lambda(I_m - \frac{1}{2}D(\Theta + I_m))x + (I_m - D\Theta)x = 0. \tag{4.47}$$

This is in fact a quadratic eigenvalue problem. Consider for a moment that matrix D is symmetric and positive definite and that Θ is merely a multiple of the identity. Decompose vector x along the eigenvectors v_i of matrix D so that $x = \sum_i \mu_i v_i$, where $\mu_i \in \mathbb{C}$. Take any eigenvector v_i of D and multiply (4.47) on the left with v_i^T. If the scalar $\mu_i v^T v \neq 0$ divide through to obtain a polynomial of second order with real coefficients in λ. Given estimates on the spectrum of D and the parameter θ, bounds can be placed on the magnitude of each given λ using all eigenvectors v_i of D. This is the content of Theorem 4.4 below.

To clarify this analysis, consider the one-dimensional case, $m = 1$ where $\Sigma = \epsilon$ and likewise for $\Theta = \theta$. Let the Schur complement matrix S of (4.37) (a nonnegative scalar here) have the eigenvalue λ_S so that $S_\epsilon^{-1} = 1/(\lambda_S + \epsilon)$ and $0 < K_\epsilon = \lambda_S/(\lambda_S + \epsilon) = \sigma \leq 1$. Equation (4.47) reduces to

$$\begin{aligned} &\lambda^2 - 2\lambda\beta + \alpha = 0, \text{ where} \\ &\qquad \beta = 1 - \frac{2(\theta+1)\lambda_S}{(1+4\theta)(\lambda_S + \epsilon)}, \\ &\qquad 0 < \beta < 1 \text{ when } \theta > 1/2, \text{ and } \epsilon > 0, \\ &\qquad 0 < \beta < 1/3 \text{ when } \theta > 2, \text{ and } \epsilon > 0, \\ &\qquad \alpha = 1 - \frac{4\theta\lambda_S}{(1+4\theta)(\lambda_S + \epsilon)}, \text{ and} \\ &\qquad 0 < \alpha < 1 \text{ when } \theta > 0, \text{ and } \epsilon > 0. \end{aligned} \tag{4.48}$$

and the roots are

$$\lambda_{\pm} = \beta \pm \sqrt{\beta^2 - \alpha}, \tag{4.49}$$

which have modulus $|\lambda_{\pm}| = |\alpha| < 1$ when $\beta^2 > \alpha$. A short computation reveals that

$$\beta^2 - \alpha = -\frac{4\lambda_S}{(1+4\theta)(\lambda_S+\epsilon)} + \frac{4(\theta+1)^2\lambda_S^2}{(1+4\theta)^2(\lambda_S+\epsilon)^2}, \tag{4.50}$$

which means that $\beta^2 - \alpha < 0$ when $\theta < 2$ and $\epsilon > 0$. Thus, for fixed $\epsilon > 0$, if $\theta = 0$, $\alpha = 1$ and $\beta = 1 - 2\sigma$ and so, $\beta^2 - \alpha = -4\sigma(1-\sigma) < 0$ and $|\lambda_{\pm}| = 1$. For $0 < \theta < 2$, $\beta^2 - \alpha < 0$ and $|\lambda_{\pm}| = |\alpha| = \alpha < 1$. Beyond $\theta > 2$, if $\beta^2 - \alpha < 0$, we have $\beta < 1/3$ and so $|\lambda_{\pm}| < 2/3$. Thus, $|\lambda_{\pm}| < 1$ is all cases where $\theta, \epsilon > 0$. This includes the case where $\lambda_S = 0$, in fact, which always leads to a unit eigenvalue. Note that for $\theta = 0$, $|\lambda_{\pm}| = 1$ and this leads to oscillatory dynamics which never decays, much in accordance with the results of Theorem 4.1 though any amount of dissipation provides some stability, as per Theorem 4.2.

This analysis can be repeated for dimensions $m > 1$ in the case where $\Theta = \theta I_m$ and $\Sigma = \epsilon I_m$ since then, matrices $(I_m - \frac{1}{2}D\Theta)$ and $(I_m - D(\Theta - I_m))$ in (4.47) commute and their eigenvectors are those of matrix H defined in (4.46). The bounds just derived for $\lambda_{\pm}$ still hold here.

A similar analysis holds for the case where Σ is not a multiple of the identity but $\Theta = \theta I_m$ since in that case, the decomposition of D into eigenvectors with positive eigenvalues is guaranteed by Lemma 4.3 below, a well-known result. However, these eigenvectors are no longer orthogonal which means that a convexity analysis is required when considering the bounds on the coefficients α and β of the quadratic equation (4.48).

Lemma 4.3. *For a real, $n \times n$ symmetric, positive definite matrix A and a strictly positive diagonal matrix $\Gamma = \mathrm{diag}(\gamma_1, \gamma_2, \ldots, \gamma_n), \gamma_i > 0$, the matrix $B = \Gamma A$ has the same real, positive eigenvalues as the symmetric, positive definite matrix $V = \Gamma^{1/2} A \Gamma^{1/2}$, and if v is an eigenvector of V with eigenvalue λ, then, $b = \Gamma^{1/2} v$ is an eigenvector of B with the same eigenvalue. Unless Γ is a scalar multiple of the identity, the eigenvectors of ΓA are not orthogonal.*

Proof. Since A is symmetric and positive definite, so is $V = \Gamma^{1/2} A \Gamma^{1/2}$. Consider an eigenvalue λ of matrix V. Since V is positive definite, λ is real and positive. Now, $Vv = \lambda v = \Gamma^{1/2} A \Gamma^{1/2} v$ and thus, writing $b = \Gamma^{1/2} v$, assuming that all elements of Γ are strictly positive, we find

$$\Gamma^{1/2} A b = \lambda \Gamma^{-1/2} b, \tag{4.51}$$

and after multiplying both sides by $\Gamma^{1/2}$, the desired result

$$\Gamma A b = B b = \lambda b, \tag{4.52}$$

is obtained. Considering two eigenvectors of matrix $V = \Gamma^{1/2} A \Gamma^{1/2}$, v_i, v_j, and the corresponding eigenvectors b_i and b_j of matrix B, we have

$$b_i^T b_j = v_i^T \Gamma v_j \neq 0, \tag{4.53}$$

which does not necessarily vanish unless $\Gamma = \gamma I_n$. □

For the case at hand, observe first that S_ϵ is positive definite and so is S_ϵ^{-1}. Using $\Theta = \theta I_m$ and correspondingly, $\Upsilon = (1 + 4\theta)^{-1} I_m$, and given that the matrices appearing in (4.47) only differ by scalar multiples and scalar multiples of the identity, the same analysis done for the scalar case in (4.50) holds, replacing the factors $\epsilon/(\lambda_S + \epsilon)$ with the scaled eigenvalues guaranteed by Lemma 4.3. What is missing here is a tight estimate of the eigenvalues of ΣS_ϵ^{-1}. In the diagonal case, there were just $\epsilon/(\lambda_S + \epsilon) \leq 1$, and this produced nice bounds even for the case where $\lambda_S = 0$.

Numerical results on the spectrum of the stepping matrix H of (4.46) are presented in Figure 4.1. Random Jacobian matrices G of size $m = 200$ were generated and the mass matrix M was set to the identity. The rank of the matrices G was made deficient in the first three cases to illustrate the point made here that regularization does resolve this issue. The last line on the plot corresponds to a matrix with full row rank which is even better behaved. Condition numbers of the Schur complement matrix S are shown on the plot for each line. These range from the moderate 10^8 to the extreme 10^{16}, which at the limit of machine precision, and this hopefully explains the spurious overshoot for the largest eigenvalue for this latter case. The perturbations ϵ_j were chosen randomly in the range $[\epsilon_{\min}, 2\epsilon_{\min}]$ and so were the stabilization parameters $\theta_j \in [2, 3]$. These simple experiments show that the spectrum of H is expected to be well behaved in general, even in cases where the proofs provided above do not apply.

A complete characterization of the spectrum of the iteration matrix would have been desirable and, hopefully, will be available in the future. For the general nonlinear case, the recurrence relation must be analyzed in terms of its contraction properties and this is also left for future work. The discussion of this section is summarized in Theorem 4.4 below which is restricted to homogeneous perturbation and damping.

Theorem 4.4. *Given a configuration manifold Q of dimension n, a constant, symmetric and positive definite $n \times n$ mass matrix M and linear homogeneous constraint functions $g : Q \mapsto \mathbb{R}^m$ with $m < n$ with $g(q) = Gq$ for a constant $m \times n$ matrix G. Given also a constant diagonal perturbation matrix $\Sigma = \mathsf{diag}(\epsilon_1, \epsilon_2, \ldots, \epsilon_m)$, where $\epsilon_j > 0, j = 1, 2, \ldots, m$, as well as diagonal stabilization matrix $\Theta = \mathsf{diag}(\tau_1/h, \tau_2/h, \ldots, \tau_m/h)$ where $\tau_j > 1$ for $j = 1, 2, \ldots, m$. The dynamics of the* SPOOK *stepper given by* (4.27) *converges to $Gq_k \to 0$ and $Gv_k \to 0$ when the projected forces vanish.*

Proof. Given that $\Sigma = \epsilon I_m$, the main matrix to analyze is $K_\epsilon = S S_\epsilon^{-1} = I_m - \epsilon S_\epsilon^{-1}$ where $m \times m$ matrices are defined in (4.37) for S, in (4.38) for S_ϵ and in (4.39) for K_ϵ. Note first that given forces f_k, the forcing term l_k of (4.42) does not contribute to the constraint stabilization dynamics as long as $[I_m - K_\epsilon] l_k = 0$. This happens when either $GM^{-1} f_k = 0$ which means that forces are acting tangentially to the constraint $g(q) = 0$, or when $l_k = h^2 GM^{-1} f_k$ is in the null space of $[I_m - K_\epsilon]$.

Next, since S and S_ϵ differ only by a constant diagonal element, they are diagonalizable with the same orthogonal transformation, U. If the eigenvalues

of S are $(\lambda_{S_1}, \lambda_{S_2}, \ldots, \lambda_{S_m})$, say, and thus, K_ϵ is also diagonalized with this so that we have

$$U K_\epsilon U^T = \mathrm{diag}\left(\frac{\lambda_{S_1}}{\lambda_{S_1}+\epsilon}, \frac{\lambda_{S_2}}{\lambda_{S_2}+\epsilon}, \ldots, \frac{\lambda_{S_m}}{\lambda_{S_m}+\epsilon}\right). \tag{4.54}$$

Now, consider an eigenvalue λ of the iteration matrix $H = A^{-1}B$ defined in (4.46) with eigenvector $w = (x^T, y^T)$. Any such root satisfies the quadratic equation (4.47). Now, decompose the vector x according to the eigenvectors $v_i, i = 1, 2, \ldots, m$ of matrix S, multiply (4.47) on the left with v_i^H (the Hermitian conjugate is needed here since the coefficients of x may be complex) and divide through with $x^H x$. Since all matrices appearing in (4.47) differ from K_ϵ by only a scalar multiple of the identity and an overall scalar factor, the vectors v_i are eigenvectors of these as well. This leads to m equations of the same form as (4.49), namely $\lambda^2 - 2\lambda\beta_i + \alpha_i$ where the real coefficients β_i, α_i, $i = 1, 2, \ldots, m$, are now

$$\begin{aligned} \beta_i &= 1 - \frac{2(\theta+1)\lambda_{S_i}}{(1+4\theta)(\lambda_{S_i}+\epsilon)}, \\ \alpha_i &= 1 - \frac{4\theta\lambda_{S_i}}{(1+4\theta)(\lambda_{S_i}+\epsilon)}. \end{aligned} \tag{4.55}$$

Repeating the computations found in (4.50) and (4.48) for each eigenvalue $\lambda_{S_i} \geq 0$ of S, each such eigenvalue λ of H is found to have modulus $|\lambda| < 1$ whenever $\theta > 0$ and $\lambda_{S_i} > 0$, and $|\lambda| = 1$ when $\theta = 0$ or $\lambda_{S_i} = 0$. The dynamics of the iterates z_k in (4.45) thus corresponds to potentially oscillating exponential decay for the relevant modes, i.e., those corresponding to nonzero eigenvalues $\lambda_{S_i} \neq 0$, and this means that $x_k = Gq_k \to 0$ and $v_k = Gq_k \to 0$ as $k \to \infty$. Since the zero eigenvalue modes do not contribute, the constraint Gq is thus stabilized in both position and velocity and the proof is complete. □

4.7 End notes

A physics based, numerically stable constraint realization algorithm was constructed for both holonomic and nonholonomic constraints. The same techniques were used to provide a stabilization algorithm for holonomic constraints. This was proven to be globally stable for the linear homogeneous case, given homogeneous regularization and stabilization parameters.

Previously known regularization schemes are not physical in the sense that they cannot be formulated within a Lagrangian mechanics framework. In addition, the numerical realization of constraint realization schemes or penalty forces has been repeatedly observed to be a disaster except for the simplest cases.

Even in the careful analysis of [166], the numerical examples have high oscillations and exhibit noise. For constraint stabilization, with all the deep analysis and beautiful results of Uri Ascher and his colleagues and students reported in [61, 26, 24, 25] and [27], there is nothing that comes close to the simplicity

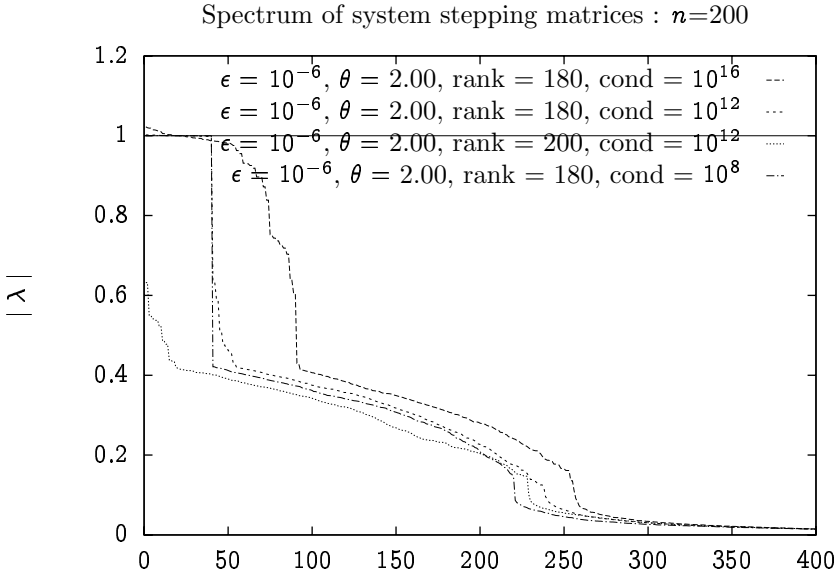

Figure 4.1: The spectrum of the stepping matrix for moderate values of condition number and degeneracy.

and physical motivation of the results we have just presented. Also, in most of Ascher's work, one uses an explicit Runge-Kutta method as the main work horse and performs extra computation to stabilize and regularize the system afterward. As discussed in Chapter 3 however, it is possible to stabilize constraints in a physical way, and still using a first order, one stage method, which is faster.

There are high order, accurate and efficient methods for DAEs [165]. These have well established application domains, and to the extent that they are not used to model physical systems which satisfy a least action principle or systems having symplectic flows, they are perfectly fine choices.

With much the same motivations behind the present thesis, low order methods Runge-Kutta methods for DAEs were developed with an eye on efficiency in the work of Cameron and Piché [59, 58]. It could be interesting to see if some of these ideas could be reused to build higher order variational methods.

For mechanical systems, discrete variational mechanics yields simple and remarkably good low order methods, and this is what is needed for interactive simulations at least, namely, relying only on linear systems, global error bounds, and strong linear stability. Indeed, for fixed time step h, discrete energy oscillates within bounds of order h^2 for the methods considered here. Since the regularization strategy here is to associate an energy cost quadratic in the constraint violation. Since the total energy budget is bounded, the allowed violations are globally bounded as well.

Also because of the strictly physics based approach, regularization is not just a lesser evil in choosing between inaccuracy, high computational cost or instabil-

ities, but a genuine physical model. Using a large regularization parameter for a given constraint amounts to introducing compliance into the system. The regularization and stabilization scheme can thus be used to model any form of stiff force in a stable way, for the cost of adding one extra row and column to the linear system (4.32). Other schemes for modeling stiff forces in multibody systems, such as [18], which amounts to using linearized implicit Euler discretization for the stiff forces, much like in [39], correspond to modifying the mass matrix with terms of the form $h^2 G^T K G$, where G is the Jacobian of the stiff force, K is the stiffness matrix, and h is the time step. When the eigenvalues of the stiffness matrix K become large, the linear systems become ill-conditioned. By contrast, in the present formulation, the corrections are of the form $GK^{-1}G^T$ and these are well behaved as the eigenvalues of the stiffness matrix become large.

Stiff forces can also be handled using adaptive time steps. A variational formulation of this for the context of smooth but stiff contact force potential is found in Modin and Führer [206].

Previously [168], I reported a different discretization scheme for the analytic constraint regularization and stabilization technique of the present chapter. This amounted to a first order backward discretization of the constraint equations and in turn, that amounts to discretizing the strong oscillatory forces using linearized first order implicit integration. In fact, that stepping scheme is in the regularized, stabilized family of Section 4.5, when choosing suitable parameters. This was used in three Master's thesis projects, two of which completed successfully. The first application was the simulation of cloth [70] and the second was for deformable solids [201, 254]. In both cases, the results compared favorably in performance to the state of the art in computer graphics literature. However, from the modeling perspective, the new strategy was superior because it did not dissipate so much energy for degrees of freedom orthogonal to the constraints. The same techniques have also been used in collaborative work with Servin, resulting in one refereed conference proceeding article [254], one article to appear in 2007 [253] and another in preparation, where the goal was to simulate light cables having both bending and torsion resistance and subject to heavy loads, for interactive simulations of cranes for training applications. Since the cable is modeled using constrained rigid bodies, it is possible to subject the hoisting cables to frictional contacts, using techniques similar to those described in Chapter 10. Prior to this work, the literature only reported the modeling of cables with pure constraints [131] and thus not allowing any contacts and not providing for bending and torsion, or using Crosserat theory [222] which allows for bending and torsion but not contacts, or with point masses [187] which allows for modeling contacts and bending but not torsion.

Constraints were only considered to first order here but higher order schemes can be built and these will be developed in the future. It is hoped that regularization and stabilization techniques discussed in the present chapter will permit stable linearization of the internal stages.

The stability result could be stronger if the spectrum of the stepping matrix could be proved to be within the unit circle at all time, and with a better

characterization of the unit eigenvalue modes of degenerate systems.

The correspondence between exact and regularized systems, both in continuous and discrete, could be analyzed to greater length. Also, a comprehensive error analysis is still missing, though important elements and techniques for doing that exist already [196].

Rayleigh dissipation functions are central to the stabilization and regularization analysis. Though they are well known in system's engineering though [177], they are conspicuously underused in the analytical mechanics literature, receiving all the attention of one paragraph in Goldstein [105], and barely mentioned in Lanczos. Their role in the optimization framework of the principle of least action, continuous and discrete, is only now being recognized.

5 Bagatelle II: Numerical Stability of Simple Harmonic Oscillator

Explicit computations of the linear stability function of integrators [114] are performed for a variety of methods for the case of the simple harmonic oscillator described in Chapter 2. This amounts to evaluating the stability function on the imaginary axis. This is argued to be a good measure of the usability of a given integration method for physical systems. The standard technique is described in Section III which contains all computations. General observations are made in Section 5.2.

5.1 Classical stability analysis and classical methods

The simple harmonic oscillator is the second simplest physical example. Indeed, if one considers a power series of potential functions $V(q)$ near an equilibrium point, one finds $V(q) = V_0 + \frac{k}{2}q^T q + \dots$ after the q variables have been changed so that $q = 0$ is the equilibrium point. The constant term V_0 does not contribute anything to the physics. The quadratic term in q produces a *linear restoration force* and is thus the prototypical example of a linear physical system.

Investigation of linear stability theory of numerical methods for ordinary differential equation rests on evaluating the result of a single step of a given method on the *canonical test function* of Dahlquist [114], namely

$$\dot{q} = \lambda q, \tag{5.1}$$

where $q, \lambda \in \mathbb{C}$, are complex numbers. Applying any given one step method to an arbitrary initial condition q_0 for a *single step*, one finds

$$q_1 = R(h\lambda)q_0, \tag{5.2}$$

where $R : \mathbb{C} \mapsto \mathbb{C}$ is a scalar complex function of the complex argument $z = h\lambda$, and is called the *stability function* of the integrator. This illustrates the fact that integrators are *linear filters*. Stability is then defined as the region of the complex plane such that $|R(z)| \leq 1$, where $|\cdot|$ is the modulus operator for complex numbers. Of course, being only linear, this analysis does not yield the complete picture. However, a method which fails to be linearly stable cannot be expected to produce good results in general. In addition, not all integration methods can be amenable to the format (5.1), and in particular, the variational

integration methods described so far do not fit in this framework, because they involve two term recurrence relations where q_2 depends on both q_1 and q_0 (of course, replacing R in (5.2) with a matrix can work).

The description of stability as the region of complex plane where $\mid R(z) \mid \leq 1$ is focused on the preservation of stability of dynamical systems under discretization, and provides little or no information on unstable systems. For instance, the perfectly valid equation $\dot{q} = 2q$ describes exponential growth and a perfect integrator would then yield $\mid R(z) \mid = 2$, and thus be considered unstable. The analysis provides useful information as long as one concentrates on the combination where the time step is positive, $h > 0$, and the real part of λ is positive so that $\Re(\lambda) < 0$. The case $\Re(\lambda) = 0$ is marginal in fact, but that is precisely the one corresponding to conservative mechanical systems.

The reason to use $\lambda \in \mathbb{C}$ is that oscillatory systems necessarily have a non-zero imaginary component in this one-dimensional framework. Indeed, a simple harmonic oscillator corresponds here to the pure imaginary case: $\lambda = i|\lambda|$. The present analysis is thus restricted to pure imaginary λ to get a better picture of some popular, simple integration methods, including the first order Euler implicit and explicit methods, as well as the second order Runge-Kutta method and the wildly popular RK4a method [113, 112], in the context of the simplest linear physical problem.

The stability functions for these are known to be [114]

$$
\begin{aligned}
R_{ee}(z) &= 1 + z, \text{ explicit Euler} \\
R_{ie}(z) &= \frac{1}{1+z}, \text{ implicit Euler ,} \\
R_{\mathrm{mid}}(z) &= \frac{1+z/2}{1-z/2} \text{ implicit midpoint} \\
R_{rk2}(z) &= 1 + z + \frac{z^2}{2}, \text{ explicit midpoint,} \\
R_{rk3}(z) &= 1 + z + \frac{z^2}{2} + \frac{z^3}{6}, \text{explicit third order Runge-Kutta ,} \\
R_{rk4}(z) &= 1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24}, \text{explicit fourth order Runge-Kutta.}
\end{aligned}
\tag{5.3}
$$

In general, an explicit Runge-Kutta methods of order p has a linear stability function $R_{rk(p)}(z)$ equal to the p first term in the Taylor series expansion of $\exp(x)$, namely, $\exp(z) = 1 + z + (1/2)z^2 + \cdots + (1/n!)z^n \ldots$. Restricting to the

case where $z = ix, x \in \mathbb{R}$, and $x > 0$, we have,

$$\begin{aligned} |R_{ee}(ix)| &= \sqrt{1+x^2} > 1, \\ |R_{ie}(ix)| &= \frac{1}{\sqrt{1+x^2}} < 1, \\ |R_{\text{mid}}(ix)| &= 1 \\ |R_{rk2}(ix)| &= \sqrt{1+\frac{x^4}{4}} > 1, \\ |R_{rk3}(ix)| &= \sqrt{1-\frac{x^4}{12}+\frac{x^6}{36}}, \\ |R_{rk4}(ix)| &= \sqrt{1-\frac{1}{72}x^6+\frac{1}{576}x^8}. \end{aligned} \tag{5.4}$$

It is clear from this that the first order explicit Euler method and the second order Runge-Kutta method are useless for physical systems without additional dissipation since the stability function is *always greater* than unity. By contrast, the implicit Euler method is, as was observed previously, unrelently dissipative and can never keep the oscillations going. The implicit midpoint method is stable for any step size and never dissipates anything. Explicit Runge-Kutta of order 3 and above do have non-zero stability region which increase in size with the order of the method.

For the third order explicit Runge-Kutta method, the linear stability function, $R_{rk3}(ix)$ for real $x \geq 0$, starts at 1 for $x = 0$. The modulus $|R_{rk3}(ix)|$ decreases monotonically to take the minimum value $\sqrt{8/9}$ at $x = \sqrt{2} \approx 1.4142$, and increases henceforth to reach value 1 for $x = \sqrt{3} \approx 1.7321$, and is greater than unity afterward. Within this range, the Verlet formula holds steady but requiring one third of the computational work.

Likewise for the fourth order explicit Runge-Kutta method, the stability function, $R_{rk4}(ix)$ for real $x \geq 0$, starts with value 1 at $x = 0$. The modulus $|R_{rk4}(ix)|$ decreases monotonically to reach a minimum value of $1/2$ for $x = \sqrt{6} \approx 2.4495$, and increases monotonically from that point onward, reaching the value 1 at $x = \sqrt{8} \approx 2.8284$, and is greater than unity after that point.

Though on the surface, it appears that using RK4a would allow greater time steps than the simple Verlet formula which holds up to $x = 2$ as seen before in Section 2.4.4. However, this is deceptive. Already from $x = 1.5$, RK4a is decreasing the modulus of the complex state vector q by nearly 6% *per step*. Just around $x = 2$, RK4a is loosing 25% of the oscillation amplitude per step whilst Verlet is still holding up. At $x = 2.5$, Verlet is useless but RK4 is now producing garbage trajectories which decay to 0 at the rate of 50% *per step*. This holds up until $x = \sqrt{8} \approx 2.8284\ldots$, at which point the solution explodes.

The extension of the stability range by a mere $\sqrt{8} - 2 \approx 0.82843\ldots$ comes at the cost of 4 function evaluations per step, and at least twice the storage requirement. Not only that but this method is slightly dissipative *everywhere* within the stability domain which means that this method eventually decays to $q = 0$ when integrating over a very long time interval. The reason for that is

that RK4a is not symplectic and it therefore does not preserve any surface area during integration.

The only interesting functions in (5.4) are those for $|R_{rk3}(ix)|$ and $|R_{rk4}(ix)|$. These are plotted in Figure 5.1 below.

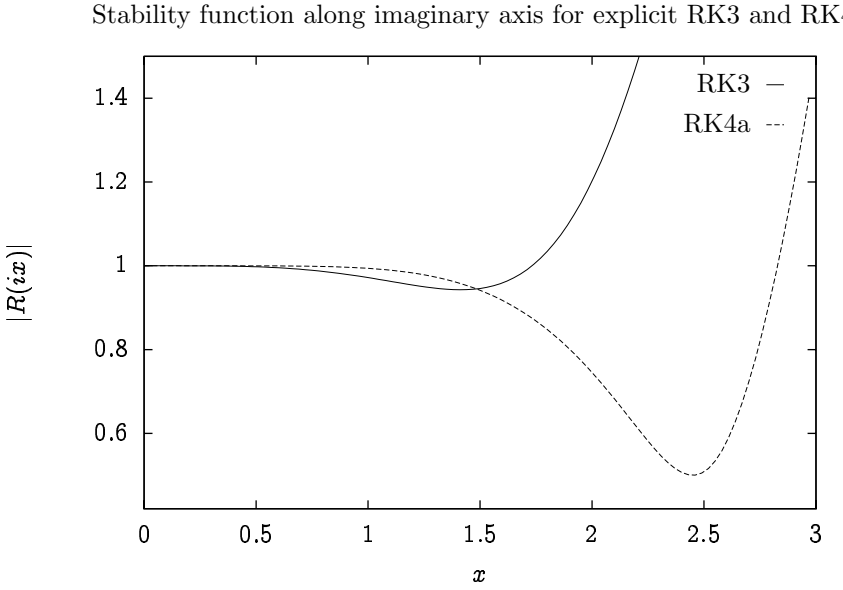

Figure 5.1: The modulus of stability functions $|R_{\mathrm{rk3}}(ix)|$ and $|R_{\mathrm{rk4}}(ix)|$ for complex arguments.

5.2 End notes

It might seem artificial to look at the stability of the simple harmonic oscillator. However, as is well known in mechanics (see [105], Chapter 6, or [22] Chapter 5, and [174] Chapter V), simple harmonic motion is the most important dynamics near any equilibrium point of any system. For a complicated system, the spectrum of frequency corresponds to the eigenvalue spectrum of the Hessian of the potential function $v(q)$, which is expected to have vanishing first derivative at an equilibrium point $q_0 \in Q$, say. When integrating at fixed time step, the danger is always present that one of the natural frequencies has a period commensurate with the time step and thus, it is always possible that the system become suddenly unstable. An a priori analysis of all possible natural frequencies of a system is prohibitively expensive in most cases.

High precision integrators based on high order Runge-Kutta formula address this issue by adapting the time step, reducing it when high frequencies or large derivatives in general are detected. As explained previously, the real-time context does not really allow for adaptive time step adjustment, even though such a

strategy might be more efficient for a given error margin.

It is also true that the simple Verlet stepping scheme, used here as an exemplar for unconstrained systems, is not unconditionally stable but it nevertheless does preserve the symplectic flow over the entire stability domain, and does so at an incredibly low computational cost. The extra precision gained by using a fourth order Runge-Kutta method for instance is not so useful because it comes at the cost of losing symplecticity.

Stability is but one of many qualitative measures of an integration method and it must be considered in the context of the target application.

6 Bagatelle III: The Simple Pendulum

The SPOOK scheme of Section 4.4 is compared to a variety of standard techniques on the simplest nonlinear constrained mechanical system, the simple pendulum. The problem is described in Section 6.1. Several well-known mathematical reformulations and integration techniques are then presented in Section 6.2, including penalty force, projection, index reduction, coordinate reduction and standard DAE techniques. The results of numerical experiments on these are illustrated in Section 6.3 and various observations are collected in Section 6.4.

6.1 Introduction

The simple pendulum in two dimensions is the simplest non-trivial example of a system with holonomic constraint. The case considered here consists of a bob of mass m moving in two dimensions at a fixed distance l from the origin subject to a uniform gravitational field of strength a_g. This example is easily and correctly reduced to a one-dimensional problem but nevertheless, it is commonly used as a test of numerical methods designed to handle arbitrary constrained systems which are not so easily reduced.

The dynamics of this system is described by the position of the bob, $q(t) \in \mathbb{R}^2$, its velocity $\dot{q}(t) \in \mathbb{R}^2$, its mass $m > 0$, the gravitational acceleration $a_g > 0$ and its orientation $w \in \mathbb{R}^2, \|w\| = 1$, and the length of the pendulum $l > 0$. The Lagrangian is simply

$$\mathcal{L}(q, \dot{q}) = \frac{m}{2}\|\dot{q}\|^2 - m a_g w^T q, \tag{6.1}$$

and this is subject to the scleronomic holonomic constraint

$$g(q) = \|q\| - l = 0, \tag{6.2}$$

or equivalently

$$g_s(q) = \frac{1}{2}\left(\|q\|^2 - l^2\right) = 0, \tag{6.3}$$

which have the Jacobian

$$G = \frac{1}{\|q\|} q^T, \quad \text{and } G_s = q^T, \tag{6.4}$$

respectively.

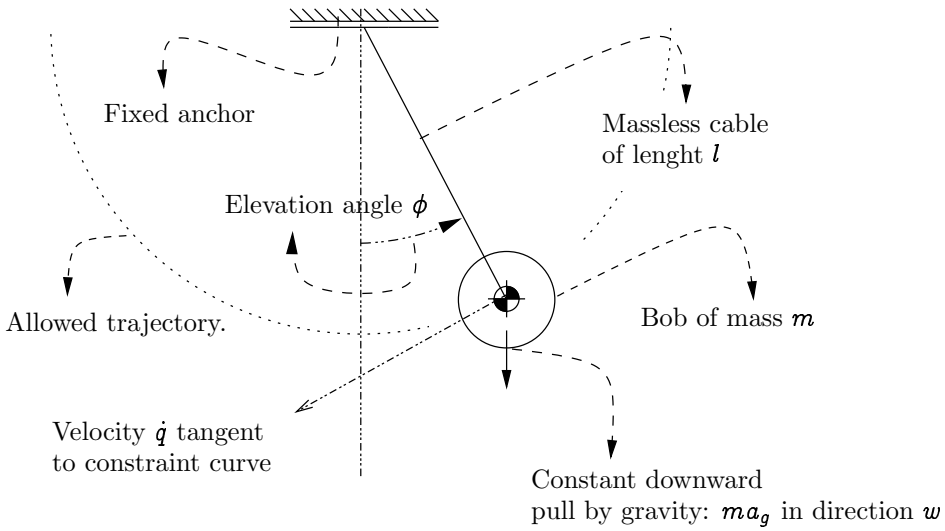

Figure 6.1: Schematics of a two-dimensional pendulum.

Obviously, the description of a point moving in two dimensions at fixed distance from the origin is easily demonstrated to be

$$q(t) = l \begin{bmatrix} \sin(\phi(t)) \\ -\cos(\phi(t)) \end{bmatrix}, \tag{6.5}$$

where the origin $\phi = 0$ was chosen to correspond to the bob at the lowest position, which is the equilibrium state.

Performing these substitutions in the Lagrangian, the following one dimensional system is recovered

$$\tilde{\mathcal{L}}(\phi, \dot{\phi}) = \frac{ml^2}{2}\dot{\phi}^2 + ma_g l\cos(\phi), \tag{6.6}$$

and this leads the equations of motion

$$\ddot{\phi} + \frac{a_g}{l}\sin(\phi) = 0. \tag{6.7}$$

For small oscillations, the approximation $\sin(\phi) \approx \phi$ can be used to recover simple harmonic oscillator motion near the equilibrium point with frequency $\omega = \sqrt{a_g/l}$. This was observed by Galileo who noted correctly that the frequency and the period of oscillations did not depend on the mass of the bob at all, but only on the length of the cable. Since there is a single free parameter ω, the length l and mass m are renormalized to 1.

For large swing, the solutions of (6.7) is a Jacobi elliptic function

$$\begin{aligned} \omega t = u &= \int_0^{\phi} \mathrm{d}\theta \frac{1}{\sqrt{1 - \sin^2\frac{\alpha}{2}\sin^2\theta}} = F(\phi \setminus \frac{\alpha}{2}) = F(\phi, \sin^2\frac{\alpha}{2}), \\ \sin(\frac{\phi}{2}) &= \mathrm{sn}(\omega t - \phi_0, \sin\frac{\alpha}{2}), \end{aligned} \tag{6.8}$$

where the frequency $\omega = \sqrt{a_g/l}$, α is the angle at maximum amplitude, ϕ_0 is the angle at initial time, and $F(x \setminus \varphi)$ is the incomplete elliptic integral of the first kind [176].

6.2 Alternative formulation and integration techniques

The two-dimensional pendulum is a great test bed for any constraint handling strategy. Out of a large number of alternatives, the following representative strategies have been selected for direct comparison. First, the distance constraint $g(q) = 0$ is replaced by a strong potential $V_\epsilon = (1/2)\epsilon^{-1}g^2(q), \epsilon > 0$, and a damping term $\mathfrak{R} = (1/2)b\dot{g}^2(q)$ with $b > 0$, and the resulting equations of motion are integrated using either the first order implicit Euler or the implicit midpoint methods. Second, the time integration is split into two stages, first stepping the bob according to the Verlet formula and following this by a projection back onto the constraint surface. Third, the library DASSL [54] is used on an augmented formulation of the problem which introduces one extra variable. Finally, the SPOOK stepper of Section 4.4 is applied to this system. The exact solution of (6.8) computed using numerical implementations of elliptic functions is used for comparison.

6.2.1 Penalty formulation and implicit Euler integration

Assuming a finite spring stiffness of $k = 1/\epsilon > 0$ and introducing the potential force

$$V(q) = \frac{1}{2\epsilon}\left(\|q\| - l\right)^2, \tag{6.9}$$

neglecting all damping terms since implicit Euler is already dissipative, the equations of motion are found to be

$$\begin{aligned} \dot{q} &= v \\ \dot{v} &= -a_g w - \frac{1}{m\epsilon}\left(1 - \frac{l}{\|q\|}\right) q, \end{aligned} \tag{6.10}$$

where w is the constant unit vector pointing along the gravitational acceleration as in (6.1). To integrate this using implicit Euler, the following nonlinear systems of equations must be solved

$$\begin{aligned} q_{k+1} - q_k - h v_{k+1} &= 0 \\ v_{k+1} - v_k + h a_g w + h\mu^2\left(1 - \frac{l}{\|q_{k+1}\|}\right) q_{k+1} &= 0 \end{aligned} \tag{6.11}$$

for the new state q_{k+1}, v_{k+1}, in which $\mu^2 = 1/(m\epsilon)$ was introduced. Using Newton-Raphson iterations, the Jacobian matrix here is

$$J_e(q) = \begin{bmatrix} I & -hI \\ h\mu^2(I - \frac{l}{\|q\|^2} q q^T) & I \end{bmatrix}. \tag{6.12}$$

The non-linear equations in (6.11) can be solved in Matlab^{TM} (from The MathWorks, `www.mathworks.com`) or Octave (`www.octave.org`, open source software released under the GNU Public License) using the function `fsolve` which, in turn, invokes the `hybrd1` routine from MINPACK [207]. It is of little consequence exactly how efficient the nonlinear solver is in this case. The point made here is that even with solving non-linear equations down to machine precision, the results obtained are worse than with the simple physics based stepping scheme with a linear approximation.

6.2.2 Penalty formulation and implicit midpoint integration

The equations of motion (6.10) of the previous section still apply here, though it is now advisable to add the damping term via $\mathfrak{R} = (b/2)\dot{g}^2(q)$ since implicit midpoint exactly preserves energy otherwise. The damping here serves as constraint stabilization. After taking the damping term into consideration, the discrete stepping equations become

$$\begin{aligned} q_{k+1} &= q_k + \frac{h}{2}(v_{k+1} + v_k) \\ \left(I + hb\frac{1}{\|q_k\|^2}q_k q_k^T\right) v_{k+1} &= v_k - ha_g w + \frac{h\mu^2}{2}\left(1 - \frac{l}{\|q_k\|}\right) q_k \\ &\quad + \frac{h\mu^2}{2}\left(1 - \frac{l}{\|q_{k+1}\|}\right) q_{k+1}, \end{aligned} \tag{6.13}$$

and the corresponding change in the definition of the Jacobian

$$J_m(q) = \begin{bmatrix} I & -\frac{h}{2}I \\ \frac{h\mu^2}{2}(I - \frac{l}{\|q\|^2}qq^T) & I + hb\frac{1}{\|q\|^2}qq^T \end{bmatrix}. \tag{6.14}$$

One could be fancy here and use a symmetric discretization for the damping term, as described in (3.86). In fact, a strict definition of the midpoint rule would lead to that. However, an interesting aspect of the variational framework is seen here in that it is entirely possible to mix and match different discretization for each individual term appearing in the Lagrangian and the Rayleigh dissipation functions.

6.2.3 Post facto projection

A very popular strategy in game physics engine due to Jakobsen [140] is to first step the system *ignoring* all constraints and then apply a projection step using a nonlinear Gauss-Seidel process. For the simple pendulum in two dimensions, the projection step is merely to adjust the length of the vector q_{k+1} to satisfy $\|q_{k+1}\| = l$. This is a *radial* projection. Since this stepping strategy is naturally dissipative in the radial direction, no additional damping term is needed.

Using Verlet integration, the stepping process is then

$$\begin{aligned} \tilde{q} &= 2q_k - q_{k-1} - \frac{h^2 a_g}{m} w \\ q_{k+1} &= \frac{l}{\|\tilde{q}\|}\tilde{q} = \alpha\tilde{q}, \text{ with } \alpha = \frac{l}{\|\tilde{q}\|}. \end{aligned} \tag{6.15}$$

This process is illustrated in Figure 6.2. If the energy of the system is measured using the symmetric formulation

$$\begin{aligned} E_k = \frac{m}{2h^2}\|q_k - q_{k-1}\|^2 + \frac{ma_g}{2} w^T (q_k + q_{k-1}) \\ = \frac{m}{2}\|v_k\|^2 + \frac{ma_g}{2} w^T (q_k + q_{k-1}), \end{aligned} \tag{6.16}$$

the energy change in one step can be found using

$$\begin{aligned} \Delta E_{k+1} &= E_{k+1} - E_k \\ &= \frac{m}{2}\left(\|v_{k+1}\| - \|v_k\|^2\right) + \frac{ma_g}{2} w^T \left((q_{k+1} - q_k) + (q_k - q_{k-1})\right) \\ &= \frac{m}{2}(v_{k+1} - v_k)^T (v_{k+1} + v_k)^T + + \frac{hma_g}{2} w^T (v_{k+1} + v_k) \\ &= \frac{m}{2}(v_{k+1} + v_k)^T \left(v_{k+1} - v_k + ha_g w\right) \\ &= \frac{(\alpha-1)m}{2}(v_{k+1} + v_k)^T q = \frac{(\alpha-1)m}{2\alpha h}(q_{k+1} - q_{k-1})^T q_{k+1} \\ &= \frac{(\alpha-1)m}{2\alpha}\left(1 - \frac{q_{k+1}^T q_{k-1}}{l^2}\right), \end{aligned} \tag{6.17}$$

and this last expression is almost always negative leading to quick dissipation. In fact, with this method, the dissipation is higher with higher velocity. Only when the pendulum is nearly upside down with low tangential velocity is the energy increasing. Illustrations of this behavior is found in Section 6.3 below.

6.2.4 Variational integration

Because strict variational integration calls for the exact solution of $g(q_{k+1})$, no damping is added here, leaving it up to the nonlinear equation solver to follow the constraint closely. The equations of motion reduce to

$$\begin{aligned} q_{k+1} &= 2q_k - q_{k-1} - h^2 a_g w^T q_k + \lambda q_k \\ \|q_{k+1}\|^2 - l^2 &= 0, \end{aligned} \tag{6.18}$$

which leads to the quadratic formula to solve for λ

$$\begin{aligned} \lambda_\pm &= -\frac{q^T q_k}{l^2} \pm \sqrt{1 + \frac{(q^T q_k)^2}{l^4} - \frac{\|q\|^2}{l^2}}, \text{ where} \\ q &= 2q_k - q_{k-1} - h^2 a_g w. \end{aligned} \tag{6.19}$$

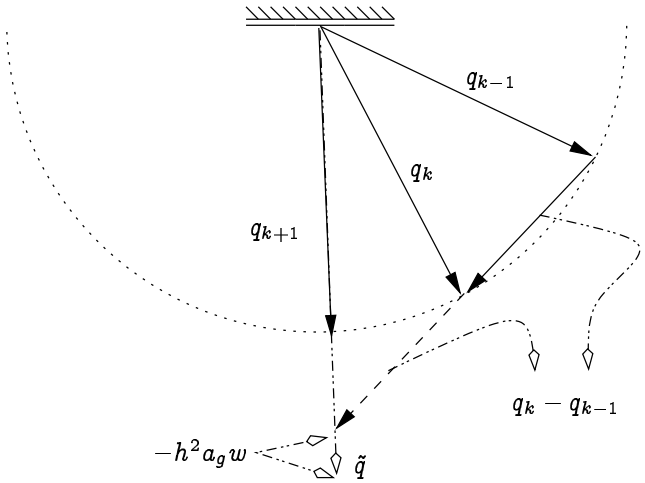

Figure 6.2: A post facto projection stepper illustrated. Note how the point $\tilde{q}$ is almost always outside of the circle.

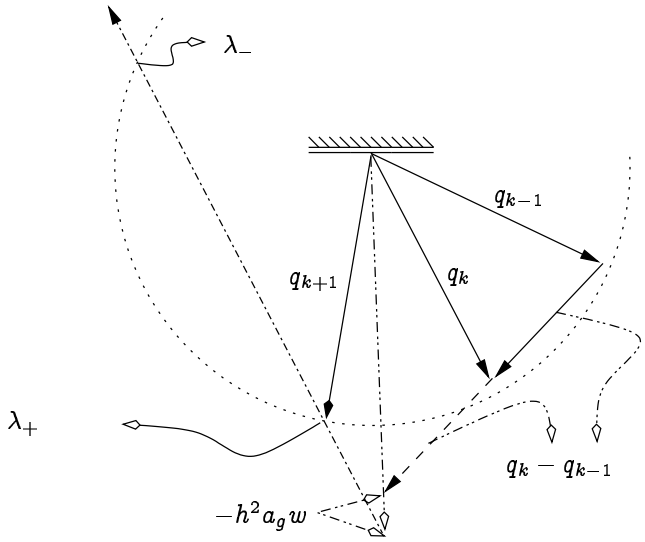

Figure 6.3: A vector diagram to illustrate the variational stepper applied to the simple pendulum in two dimensions.

In the case when $q^T q_k > 0$, which is true unless the time step h is enormous, the difference $\lambda_- < \lambda_+ < 0$ and the root λ_+ makes q_{k+1} closest to q_k and is the correct choice. A vector diagram explaining this is shown in Fig 6.3.

Measuring the total energy of the system with the symmetric formulation as in (6.16), the change in one step is now

$$\begin{aligned}
\Delta E_{k+1} &= E_{k+1} - E_k \\
&= \frac{m}{2}\left(\|v_{k+1}\|^2 - \|v_k\|^2\right) + \frac{m a_g}{2} w^T \left((q_{k+1} - q_k) + (q_k - q_{k-1})\right) \\
&= \frac{m}{2}\left(v_{k+1} - v_k\right)^T \left(v_{k+1} + v_k\right)^T + + \frac{h m a_g}{2} w^T \left(v_{k+1} + v_k\right) \\
&= \frac{m}{2}\left(v_{k+1} + v_k\right)^T \left(v_{k+1} - v_k + h a_g w\right) \\
&= \frac{\mu m}{2}\left(v_{k+1} + v_k\right)^T q_k = \frac{\mu m}{2h}\left(q_k^T q_{k+1} - q_k^T q_k\right),
\end{aligned} \tag{6.20}$$

and this expression has indeterminate sign. As shown in Section 6.3 below, the energy actually oscillates with amplitude proportional to h^2 which is a reasonable approximation.

Note that the equations of motion are integrated using a single square root operation per time step.

6.2.5 Index 2 reduction and DASSL integration

The DASSL [54] package can provably handle index 1 DAEs only, but has been observed to handle some higher index problems correctly. Given the index 3 formulation of the pendulum

$$\begin{aligned}
\dot{q} &= v \\
m\dot{v} &= \lambda q - a_g w \\
g_s(q) &= 0,
\end{aligned} \tag{6.21}$$

DASSL can process the system using the special options to increase the absolute tolerance to 10^{-3}, the relative tolerance to 10^{-1}, and using configuration parameters to the error on the algebraic equation from the convergence tests.

It is also possible to reformulate the system as an index 2 problem by adding the velocity constraint $G\dot{q} = 0$, a trick which is due to Gear and Leimkuhler [96] and reads

$$\begin{aligned}
\dot{q} &= v + \alpha q \\
\dot{v} &= \lambda q - a_g w \\
g_s(q) &= 0 \\
q^T v &= 0,
\end{aligned} \tag{6.22}$$

and the new Lagrange multiplier α corresponds to the last (redundant) equation $q^T v = 0$. Even for this formulation, as observed by Petzold et al. [54] (see

notes pp104–105 in the SIAM edition), it is necessary to remove algebraic equations from the error tests and to increase the absolute and relative tolerance to approximately 10^{-6} and 10^{-3}, respectively.

Note that the workhorse of DASSL is a BDF method and that this reduces to the implicit Euler method for first order. A fast decrease of energy is thus expected with this strategy. It would be possible in this case to add the energy equation as an extra algebraic condition, as was done by Barrlund [41] for a similar DAE solver strategy. This would indeed conserve energy but not the symplecticity of the flow. In addition, for a general system, especially one which contains some dissipative components, the nonlinear equation describing the exact energy is not available making this strategy impractical except for simple examples. Finally note that the number of nonlinear equations processed by this method is alarmingly large. Indeed, each constraint in fact introduces at least two equations using the Gear and Leimkuhler strategy and at least three using the Barrlund strategy. By contrast, the SPOOK stepper of Section 4.4 introduces one and only one equation for each constraint component. When direct methods are used for solving the linear systems, the computational work increases as $O((1+k)^3 m^3)$ where m is the number of constraint equations and k is the number of additional equations that must be processed. Each additional equation introduces almost one order of magnitude of extra work.

6.2.6 Index reduction and Baumgarte stabilization

By far the most popular method for solving constrained systems with simple methods is that of index reduction and constraint stabilization due to Baumgarte [47]. Here, the constraint $g(q) = 0$ is replaced by the mathematical equivalent $\ddot{g}(q) + \alpha\dot{g}(q) + \beta g(q) = 0$, where $\alpha, \beta > 0$ are free parameters which have to be chosen carefully. Choosing $\alpha = 1/h$, $\beta = (\alpha/2)^2$ works well for a number of cases. Nevertheless, it is not easy to find appropriate values which work well for a general problem. In fact, the simplicity of this method is entirely defeated by the impossibility of finding reliable parameter values in general, as demonstrated in [24].

In any event, the equations of motion now read

$$\begin{aligned} \dot{q} &= v \\ m\dot{v} &= \lambda q - m a_g w \\ q^T \dot{v} + 2\|v\|^2 + \alpha v^T q + \frac{\beta}{2}\left(\|q\|^2 - l^2\right) &= 0, \end{aligned} \tag{6.23}$$

and this is easily manipulated to yield a semi-implicit ODE formulation

$$\begin{aligned} \dot{q} &= v \\ \begin{bmatrix} M & -G^T \\ G & 0 \end{bmatrix} \begin{bmatrix} \dot{v} \\ \lambda \end{bmatrix} &= \begin{bmatrix} f_e \\ -\alpha G v - \beta g(q) - \dot{G} q \end{bmatrix}, \end{aligned} \tag{6.24}$$

where M is the constant mass matrix of the basic modelLagrangian (3.12), the vector of generalized forces f_e is defined as before, $f_e = -\partial V/\partial q^T$, and the

constraint functions $g_i(q) = 0, i = 1, 2, \ldots, m$, have Jacobian matrix $G = \partial g/\partial q$, or $G_{ij} = \partial g_i/\partial q_j$. Whenever G has full row-rank, this linear system is solvable for $\dot{v}$ which thus yields a standard ODE formulation for variables q, v.

This strategy for solving the DAEs of mechanical systems has prevailed in the graphics literature and some of the engineering literature. As seen in the examples, it is in fact suitable for simple systems and this is why it is included here. However, this is deceptive since tuning the stabilization parameters for larger problems can be very difficult or impossible [24].

6.2.7 Using the spook stepper

Using the SPOOK stepper from Section 4.4, one finds stepping formula

$$\begin{bmatrix} I_2 & -q_k \\ q_k^T & \frac{\epsilon}{h^2}\frac{1}{1+4d} \end{bmatrix} \begin{bmatrix} q_{k+1} \\ \lambda \end{bmatrix} = \begin{bmatrix} 2q_k - q_{k-1} - h^2 a_g w \\ -\frac{4}{1+4d} g_k - \frac{4d}{1+4d}\|q_k\|^2 + \frac{1}{1+4d} q_k^T q_{k-1} \end{bmatrix}, \tag{6.25}$$

where $\epsilon > 0$ is the compliance parameter, $d = \tau/h$ is the dimensionless damping rate. In addition, $g_k = g(q_k)$ is the constraint value at step k and I_2 is the 2×2 identity matrix. The 3×3 linear system of (6.25) is easily reduced to a single linear equation for λ, which is well behaved as long as q_k is away from the origin. The regularization parameter can in fact be set to $\epsilon = 0$ without any ill effect in this case.

Of course, linearization does introduce an error but as shown below in the results section, these are of the order of h^2 which is satisfactory.

6.3 Numerical experiments

Starting a unit length, unit mass pendulum with natural frequency $\omega^2 = 10$ horizontally, at rest, perpetual oscillations are expected in which the x coordinate varies within the interval $[-1, 1]$ and the y coordinate within the interval $[-1, 0]$. For this example, the total energy of the system is constant and exactly vanishes: $E(t) = 0$. The period of oscillation is $T = 4\sqrt{\frac{l}{a_g}} K(\sin(\pi/4)) \approx 2.3445$, where $K(k)$ is the complete elliptic integral of the first kind (see [21] for instance).

The analytic solution is easily computed for this set of initial conditions using widely available numerical implementations of the Jacobi elliptic functions and is compared with numerical results on each graph.

Given the period of a little over 2 seconds, and the maximum velocity of $\sqrt{10} \approx 3.1623$, a fixed time step of $1/60 \approx 0.0167$ should be small enough for all numerical methods. This gives more than 100 sample points per period which should be sufficient.

For the rest of this chapter, the numerical results obtained from a variety of method are explored. For each method, a picture with either three or four subplots was produced and, whenever possible, the reference solution, labeled as "exact" in the plot keys, was drawn using a dotted line. When the trajectories computed by a given numerical method coincide with the exact method, only one

solid line is visible because the two lines with different style are drawn directly on top of each other.

The scales on the subplots are all linear though in a few cases, a different scale is used on the left hand side vertical axis as opposed to the right hand side one. This is indicated with a "(left)" and "(right)" qualifier in the plot key, in the case where the Lagrange multiplier λ is on one scale and the constraint violation $g(q)$ on another in Figures 6.10 and 6.11.

The minimum energy for this system is when the bob is at rest at the bottom position and this yields $E = -10$. Negative energy is perfectly valid in the context of classical mechanics.

Consider first what can happen using a standard ODE solver (LSODE in this case [129], as this is available in Octave) directly on the reduced formulation. Fig 6.4 shows that the trajectory is in very good agreement with the exact method.

The energy subplot tells a worrisome story though as the line creeps up relentlessly. By contrast, at least for this sort of time step, integrating the reduced equation with the simple leapfrog scheme yields cyclic and bounded energy variations as seen in Fig 6.5.

Now, moving on to a simple projection scheme, as illustrated in Fig 6.6, there is a rapid decay of the energy until the pendulum is at rest, pointing down. The rate of energy loss increases with the velocity but is otherwise unpredictable. Using the SHAKE [242] algorithm though, the energy fluctuations are significant but the system keeps oscillating forever as seen in Figure 6.7.

Methods designed to handle DAEs such as DASSL [54] essentially fail in index 3 formulation as seen in Figure 6.8 but work reasonably well in index 2 reduction if care is taken to chose the parameters correctly, as seen in Figure 6.9.

SPOOK from Section 4.4) produces the data shown in Figure 6.10. The energy oscillates as was seen before in the case of the simple harmonic oscillator in chapter 2 but these oscillations stay bounded. The exact trajectory is indistinguishable from the computed one as well. Meanwhile, the constraint violation stays within $\|g(q)\| < 8 \cdot 10^{-3}$. In fact, this is true even when the regularization parameter is driven as $\epsilon \to 0$, and even when solving the non-linear equations defining this stepper (4.27). There is in fact always a residual error here of the order of $O(h^2)$.

Next, results from the Runge-Kutta 4a method using the Baumgarte stabilization scheme [47] are presented in Figure 6.11. The energy curve looks good at first with a drift of less than one part in a thousand over 400 steps, and so goes the constraint violation curve with errors of $\|g(q)\| < 3 \cdot 10^{-5}$. Trajectories are also indistinguishable from the exact solution. However, the energy is drifting up relentlessly at a linear rate and the method eventually breaks down after a large enough number of steps. In addition, four times as much work is performed at each step in comparison with SPOOK illustrated in Figure 6.10.

Comes the turn of integrating the penalty formulation of Section 6.2.2 using the implicit midpoint rule and results are presented in Figure 6.12. The spring constant here is $k = 10^4$ and the damping ration is $\zeta = 1$, which should produce

critically damped oscillations.

The energy plot here is not bad but the damping, which should act only along the radial direction, is also removing kinetic energy from the system. It is known that the implicit midpoint rule should exactly preserve energy for this case [112]. Not too surprisingly, the trajectory is in reasonably good agreement with the exact solution. There is still a noticeable drift in phase shift over time even though a fairly large spring constant is used. As the spring constant increases even more, it is expected that the nonlinear equations to solve become progressively more ill-conditioned and thus increasingly difficult to solve. Again, for the effort and the cost, the results from SPOOK in Figure 6.10 are better.

Last but not least, and apologizing for labeling such a poor method with such an illustrious name, the results of applying the implicit Euler method on the penalty formulation of Section 6.2.1 are shown in Figure 6.13. A moderate spring constant of $k = 10^4$ and damping ratio of $\zeta = 1$ were used here again. As is customary with the implicit Euler method, the energy dissipates away much faster than expected from the value of the damping constant. What is clear from the plot of the y coordinate though is that the observed *frequency* of oscillations, which is essentially determined by the value of the gravitational acceleration a_g, is wrong by a noticeable amount. Increasing the spring constant further, this rate of falling would decrease even more. Curiously, the radial force is damping motion that is orthogonal to it.

This curious phenomena gives some indication why cloth simulated using networks of point masses connected with springs and dampers, simulated using a linearized version of Euler's implicit first order method [39], appears to fall too slowly.

6.4 End notes

The numerical behavior of the simplest DAE for a mechanical system was investigated using several alternative methods. It is clear from the data that integration methods based on the variational principle presented in Chapter 3 are best in terms of qualitative behavior, and even performance as measured by the number of linear system solve operations done per step. The one exception to this being the coordinate reduction strategy since it only involves one scalar equation. In more general cases, coordinate reduction involves smaller linear systems but these are more dense and sometimes ill-conditioned [28]. In addition, computing the coordinate reduction often involves additional QR and SVD operations and thus, are not clearly faster *a priori*. Among the variational methods, the SPOOK scheme of Chapter 4 stands out for requiring only the solution of linear systems but still maintaining good agreement on the trajectories and keeping good bounds on energy.

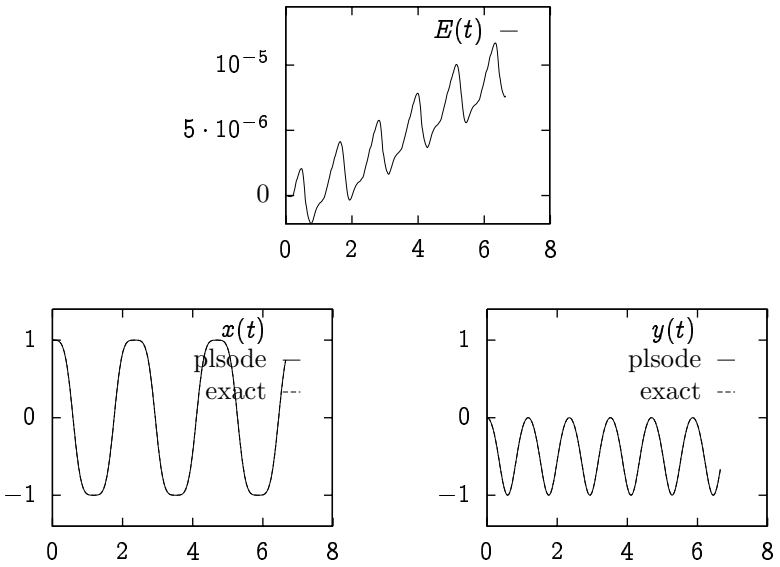

Figure 6.4: Integrating the reduced equations of motion of the planar simple pendulum with the reference integrator LSODE [129] available in Octave.

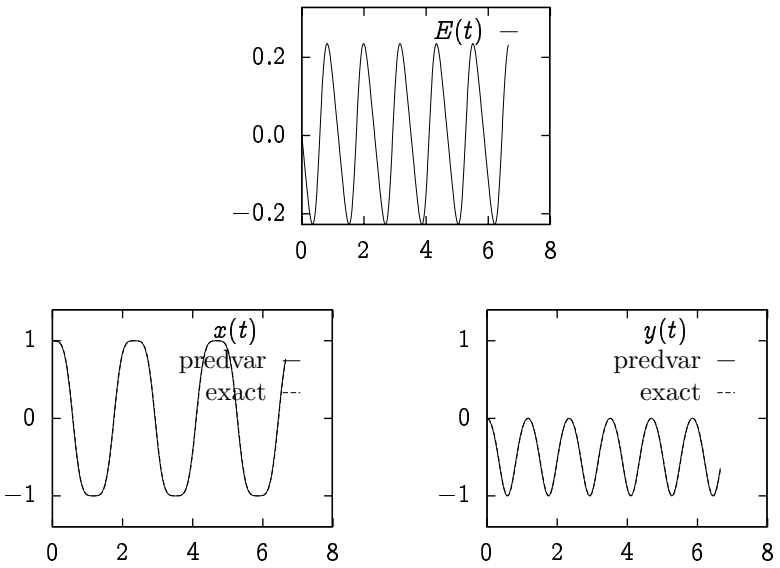

Figure 6.5: Integrating the reduced equations of motion of the planar simple pendulum using the Verlet method.

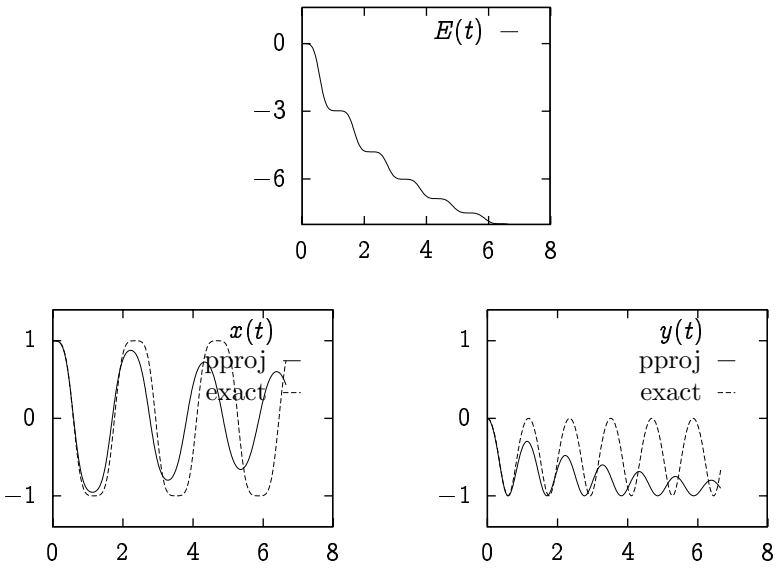

Figure 6.6: The planar simple pendulum integrated with the Verlet method followed by a pure projection back to the constraint surface.

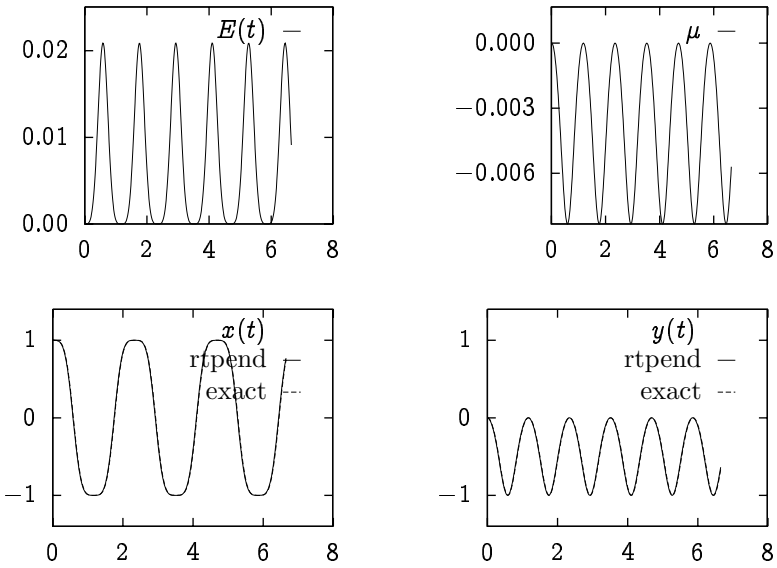

Figure 6.7: The planar simple pendulum integrated using the SHAKE algorithm.

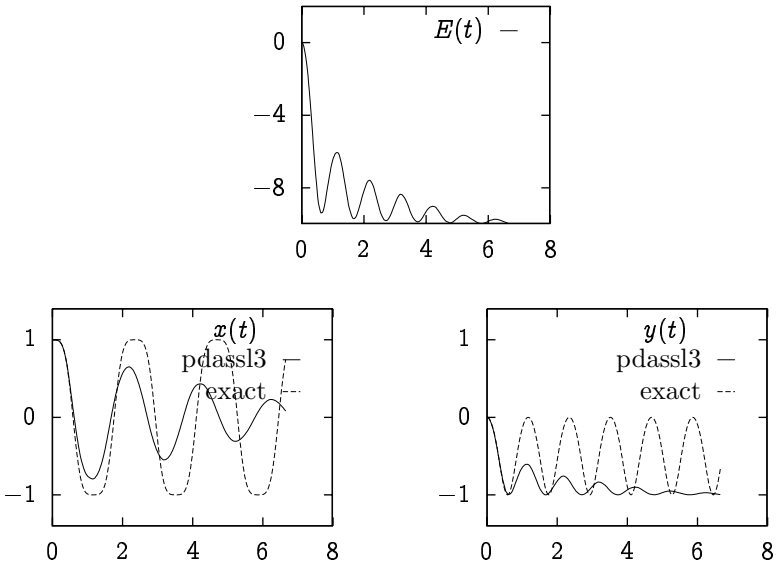

Figure 6.8: The planar simple pendulum integrated with DASSL with index 3 formulation.

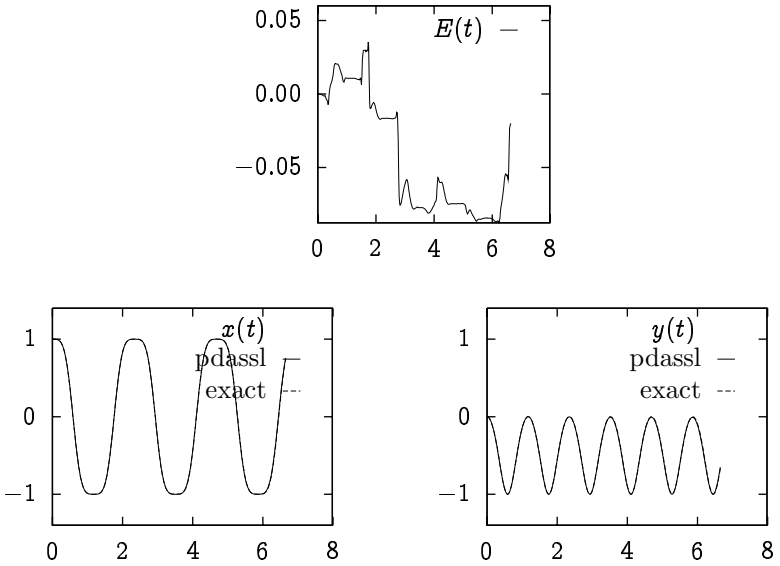

Figure 6.9: The planar simple pendulum integrated with DASSL after performing an index reduction and adjusting parameters.

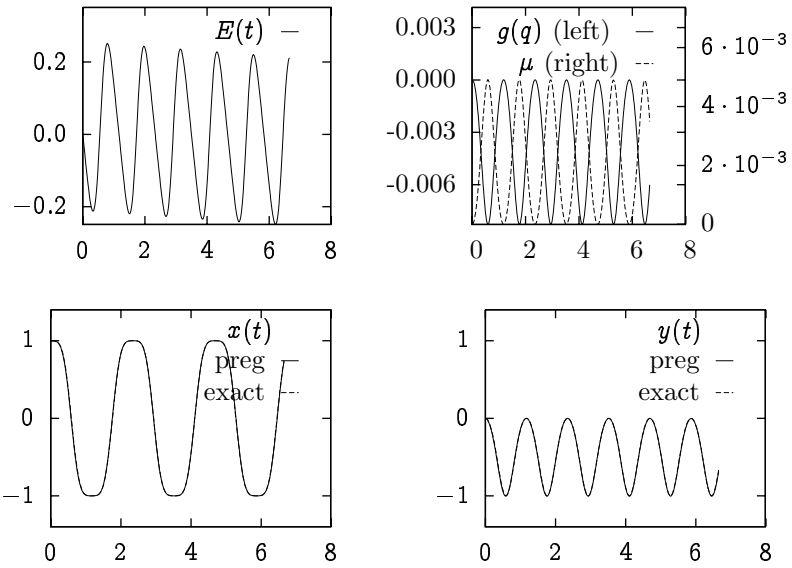

Figure 6.10: The planar simple pendulum integrated with SPOOK.

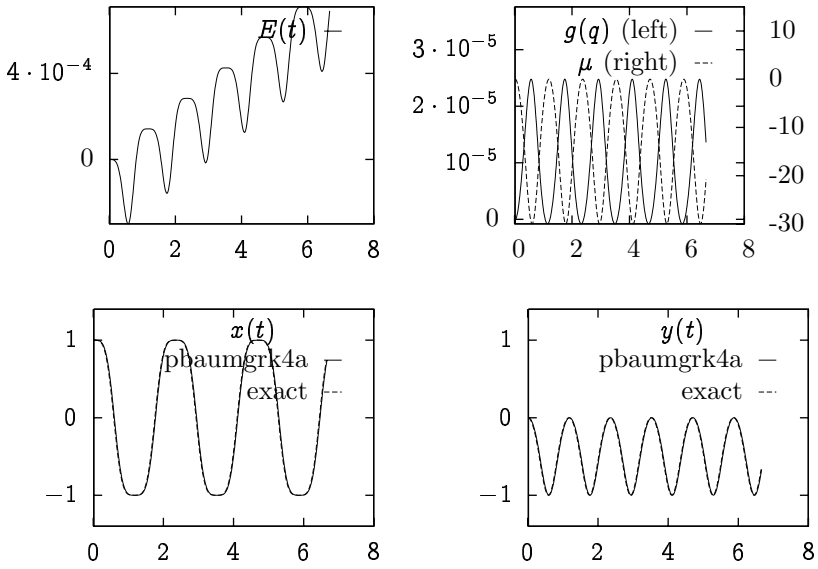

Figure 6.11: The planar simple pendulum integrated using index 1 reduction, Baumgarte stabilization, and Runge-Kutta method RK4a.

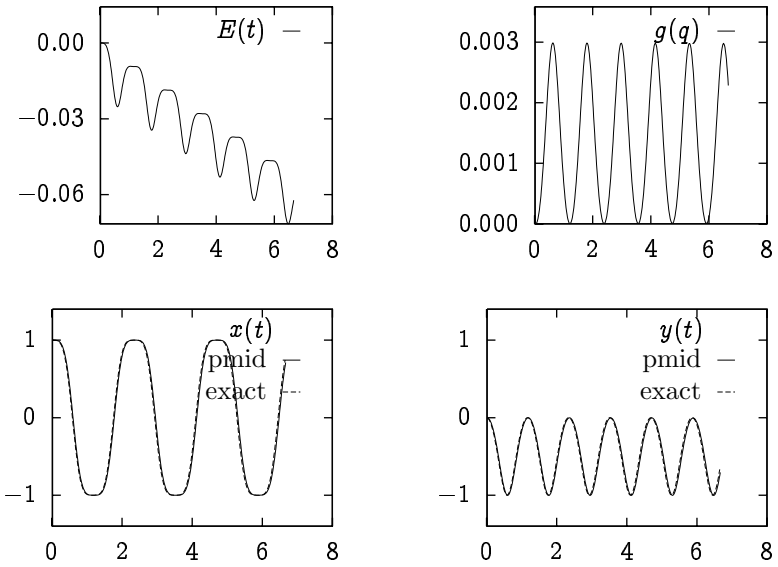

Figure 6.12: The planar simple pendulum integrated with implicit midpoint method, with a small damping coefficient of $b = 1$.

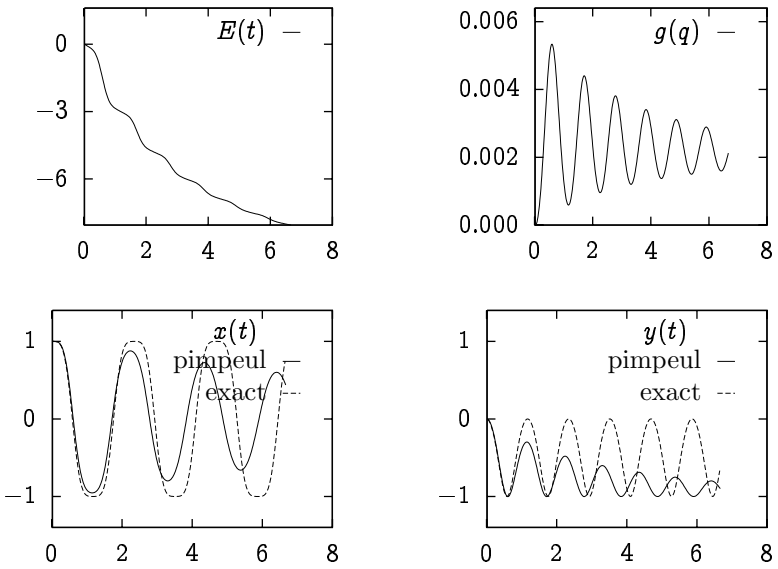

Figure 6.13: The planar simple pendulum integrated using the implicit first order Euler method to the regularized problem for moderate spring constant.

7 Bagatelle IV: The Slider Crank

The planar slider-crank mechanism is perhaps the simplest constrained mechanical system to exhibit constraint Jacobian degeneracy at isolated points in the configuration space. This is analyzed here to demonstrate how the SPOOK method of Section 4.4 resolves such problems gracefully in comparison with standard techniques. The slider-crank mechanism is described in Section 7.1 where the different parts and constraints are defined in details. The Lagrangian for this system is constructed explicitly in Section 7.2 and the singularities are precisely identified in Section 7.3. The results of numerical experiments are presented in Section 7.4 and observations are collected in Section 7.5.

7.1 Introduction

A slider-crank is a common mechanical device which converts rotational motion, as produced by the most commonly available type of motors, to linear motion. This is done by attaching one body to the shaft of the rotational driver, connecting this body with a hinge joint to a second body at one end, and attaching the other end of this second body to a linear guide in a way that allows translational motion.

The slider crank in two dimensions is also the simplest example of a system with a *constraint singularity*. When the two arms have identical lengths, the Jacobian matrix of the slider crank has two isolated singularities where it becomes rank deficient. Such singularities are integrable if the velocity does not vanish but this is cold comfort since it is easy to construct an example for which the equilibrium configuration comes to rest at the singularity, as constructed below.

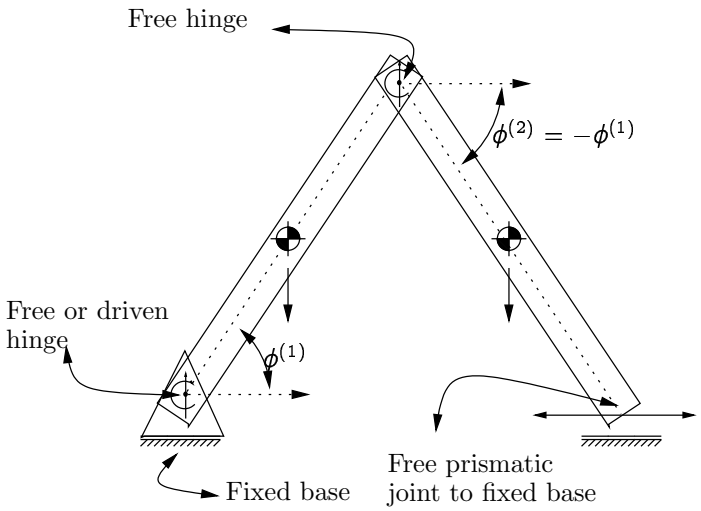

Figure 7.1: A schematic diagram of a slider crank mechanism.

The basic motion of a two-dimensional rigid body is shown in Section 12.4 to be described by the coordinates

$$q = \begin{bmatrix} x_1 \\ y_2 \\ \phi \end{bmatrix}, \tag{7.1}$$

where $x = (x_1, y_2) \in \mathbb{R}^2$ and $\phi \in \mathbb{R}$ is a scalar angle. The rigid rotation matrix defined by ϕ is denoted $R(\phi)$ and has the following definition

$$R(\phi) = \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix}. \tag{7.2}$$

The time derivative is found to be

$$\dot{R}(\phi) = \dot{\phi} J_1 R(\phi), \qquad \text{where } J_1 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}. \tag{7.3}$$

The inertia matrix in this case is a constant diagonal 3×3 matrix of the form

$$M = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & \mathcal{J}_0 \end{bmatrix} \tag{7.4}$$

where $\mathcal{J}_0$ is a scalar computed using

$$\mathcal{J}_0 = \int_\Omega \mathrm{d}^2 x \rho(x) \|x\|^2, \tag{7.5}$$

and $\Omega \in \mathbb{R}^2$ is the bounded region covered by the rigid body in rest configuration, with the center of mass at the origin, in some reference orientation. The scalar

function $\rho(x) : \mathbb{R}^2 \mapsto \mathbb{R}_+$ is the mass density at x and is non-negative, and $\|x\|^2$ is the standard Euclidean norm in two dimensions.

The slider crank mechanism requires mathematical definitions for the hinge (revolute joint) and the prismatic (slider joint) holonomic constraint, in addition to a constant rotational velocity driver, a nonholonomic constraint. These are defined below.

Note before starting that even if the lengths of the two arms are different, simple trigonometry indicates that

$$l_1 \sin(\phi^{(1)}) = -l_2 \sin(\phi^{(2)}). \tag{7.6}$$

This will be of use later when locating the singularities.

7.1.1 Two-dimensional hinge joint

To define the hinge or revolute joint in two dimensions, start with an attachment point $x^{(0)}$ that is fixed in the inertial frame. Then, consider the vector $p^{(1)}$ that is fixed in the frame of a rigid body so that $p^{(1)}(t) = x^{(0)}$ at all times. The only motion allowed is then a rotation about $x^{(0)}$ and therefore, the constraint

$$g_{H_1}(q; x^{(0)}) = x(t) + R(\phi(t))p^{(1)}(0), \tag{7.7}$$

and from this, the Jacobian matrix is extracted by computing $\dot{g}_{H_1} = G_{H_1}\dot{q}$

$$G_{H_1} = \begin{bmatrix} I_2 & J_1 R(\phi) p^{(1)} \end{bmatrix}, \tag{7.8}$$

where I is the 2×2 identity matrix.

The construction of the two body hinge constraint is similar. Consider now $p^{(1)}, p^{(2)}$, which are fixed vectors in the frame of body one and two, respectively. By forcing the location of the point on body one defined by the position of $p^{(1)}$ to coincide with that of $p^{(2)}$, then, the relative motion of these two bodies is a rotation about the location of the center of rotation which is given by both $p^{(1)}$ and $p^{(2)}$, namely,

$$g_{H_2} = x^{(1)} + R^{(1)}p^{(1)}(0) - x^{(2)} - R^{(2)}p^{(2)}(0) = 0. \tag{7.9}$$

The notation here uses $R^{(i)} = R(\phi^{(i)}$, and $p^{(i)}(0)$ denotes the fixed position of the attachment point in the frame of body i.

The Jacobian for this is found to be

$$G_{H_2} = \begin{bmatrix} I_2 & J_1 R^{(1)} p^{(1)}(0) & -I_2 & -J_1 R^{(2)} p^{(2)}(0) \end{bmatrix}. \tag{7.10}$$

7.1.2 A constant angular velocity constraint

Since for a rigid body, the angular velocity $\dot{\phi}$ is the same for all points attached to it, an angular velocity driver is defined simply by the constraint

$$g_\omega(q) = \dot{\phi} - \omega(t) = 0, \tag{7.11}$$

where $\omega(t)$ is the driving velocity at the attachment point. The Jacobian for this constraint is the constant projection matrix

$$G_\omega = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}. \tag{7.12}$$

This can also be made into a two body constraint if needed.

7.1.3 The two-dimensional prismatic constraint

A prismatic constraint forces a point fixed on a rigid body to move along a given axis. For the slider crank mechanism, the axis is defined in the inertial frame only and so a representation for the line is chosen as

$$v^T x + a = 0, \tag{7.13}$$

where $x \in \mathbb{R}^2$ is any point on the line, $v \in \mathbb{R}^2$ is a unit vector normal to the direction of the line, and the scalar a is the intercept at the origin. Given a reference point $p^{(1)}$ fixed in the rigid body coordinate, the definition of the prismatic constraint is then

$$g_P(q) = v^T x + v^T R(\phi) p^{(1)} + a = 0, \tag{7.14}$$

where $x(t)$ is the position of the center of mass. The Jacobian is easily computed

$$J_P = \begin{bmatrix} v^T & v^T J_1 R(\phi) p^{(1)} \end{bmatrix}. \tag{7.15}$$

7.2 The slider-crank Lagrangian

Agglomerating the variables as follows,

$$q = \begin{bmatrix} x^{(1)} \\ \phi^{(1)} \\ x^{(2)} \\ \phi^{(2)} \end{bmatrix}, \qquad M = \begin{bmatrix} M^{(1)} & 0 \\ 0 & M^{(2)} \end{bmatrix}, \qquad M^{(i)} = \begin{bmatrix} m^{(i)} & 0 & 0 \\ 0 & m^{(i)} & 0 \\ 0 & 0 & \mathcal{J}_0{}^{(i)} \end{bmatrix}, \tag{7.16}$$

leads to the free Lagrangian

$$\mathcal{L}(q, \dot{q}) = \frac{1}{2} \dot{q}^T M \dot{q} - g q^T M w \tag{7.17}$$

where g is magnitude of the gravitational acceleration and w is the constant vector in the direction opposite to gravitational acceleration

$$w = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}^T. \tag{7.18}$$

The equations of motion for both the continuous and discrete cases are constructed from this using the techniques of Chapter 3.

7.3 Singularities

Choosing a configuration such that both arms have lengths l and the attachment points for the hinges and prismatic joints are $p^{(i)}{}_{\mp} = \mp(l/2)u, i = 1, 2$, for the first and second attachments, respectively, with $u = (1, 0)^T$ is a unit vector in the x direction, the follow Jacobian matrix G is found

$$G = \begin{bmatrix} I_2 & -(l/2)J_1 R^{(1)} u & 0 & 0 \\ I_2 & (l/2)J_1 R^{(1)} u & -I_2 & (l/2)J_1 R^{(2)} u \\ 0 & 0 & u^T & (l/2)J_1 v^T R^{(2)} u, \end{bmatrix}, \tag{7.19}$$

which evaluates to

$$G = \begin{bmatrix} 1 & 0 & (l/2)\sin(\phi^{(1)}) & 0 & 0 & 0 \\ 0 & 1 & -(l/2)\cos(\phi^{(1)}) & 0 & 0 & 0 \\ 1 & 0 & -(l/2)\sin(\phi^{(1)}) & -1 & 0 & -(l/2)\sin(\phi^{(2)}) \\ 0 & 1 & (l/2)\cos(\phi^{(1)}) & 0 & -1 & (l/2)\cos(\phi^{(2)}) \\ 0 & 0 & 0 & 0 & 1 & (l/2)\cos(\phi^{(2)}) \end{bmatrix}. \tag{7.20}$$

This has a row rank deficiency for $\phi^{(1)} = \pm\pi/2$ at which point

$$G(\pm\pi/2) = \begin{bmatrix} 1 & 0 & \pm(l/2) & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & \mp(l/2) & -1 & 0 & \pm(l/2) \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{7.21}$$

Adding row 5 to row 4 and then subtract row 2 from the new row 4 creates a zero row.

7.4 Numerical experiments

Since the singularity of the slider crank mechanism is isolated, it is rather difficult to hit it precisely during a simulation unless starting precisely on it. Alternately, a viscous drag force of the form $-\gamma\dot{q}, \gamma > 0$ can be added and the mechanism can be left to stabilize in the degenerate configuration.

Simulations where performed using the plain variational formulation, equivalent to RATTLE, the SPOOK stepper of Chapter 4, as well as Baumgarte stabilized Runge-Kutta methods of order two and four as described in Section 6.2.6.

The first set of runs were performed without any form of friction. This is illustrated in Figure 7.2 which contains two panels. The top panel shows the time evolution of the x coordinate of the first body and the bottom panel indicates the logarithm of the condition number of the matrix used during the integration process, either $GM^{-1}G^T$ or a perturbation thereof.

For these runs, the slider crank was started at an angle $\phi = \pi/4$ at rest, and left to drop freely under the action of gravity. The slider crank then oscillates

like the pendulum. The time step is $h = 1/60$ as usual. Notice that the condition number shoots up each time the system comes near the singularity, though since it is crossed at non-zero velocity, there is no real problem. The maximum value of the condition number here is 10^6 which is modest. The trajectories produced by the different methods do not differ significantly over a period. For SPOOK, the regularization was chosen to be $\epsilon = 10^{-6}$ and the constraint stabilization parameter was $\tau/h = 2$. For the Baumgarte methods, the parameters were chosen as $\alpha = 1/h$ and $\beta = (\alpha/2)^2$, as in Section 6.2.6.

For the second set of runs illustrated in Figure 7.3, some small amount of viscous damping was added so the slider crank settles precisely on the singularity at $\phi = -\pi/2$. No driver was added here and the system was started at rest at $\phi = \pi/4$, adding a viscous drag force of $f = -0.3\dot{q}$. After oscillating for a while, the slider crank settles near $\phi = -\pi/2$ and the condition number of the stepping matrices increase toward 10^{-20}, at which point the linear and nonlinear solvers in Octave issue warning that SVD is being used to find a minimum norm solution. Of course, the matrix processed by SPOOK has maximum condition number of $1/\epsilon = 10^6$ and this poses no problem. The trajectories are all close to each other.

7.5 End notes

Point singularities where Jacobian matrices become rank deficient are not the worst possible numerical problem but SPOOK definitely removes these without any additional cost. By contrast, an exact variational method, RATTLE in the present case, can have severe difficulty at or near a singularity. Other methods might simply just fail where the solutions of the stepping equations become ill-behaved near or at the singularities.

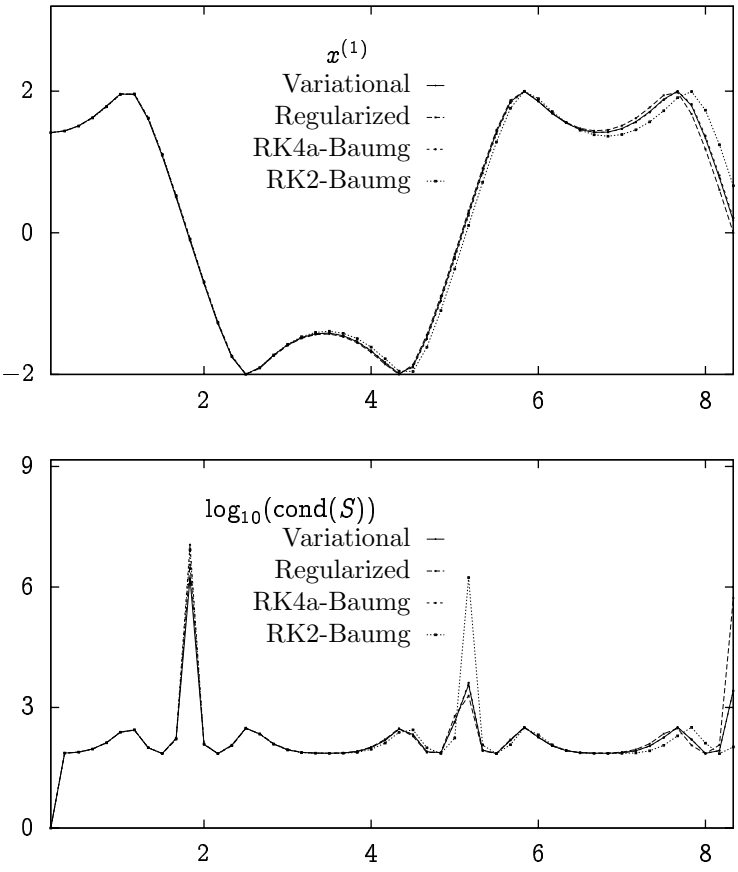

Figure 7.2: Integration of the planar slider crank without viscous friction using four different methods. The initial conditions are $\phi = \pi/4$ and $\dot{q} = 0$. See text for details on the methods used.

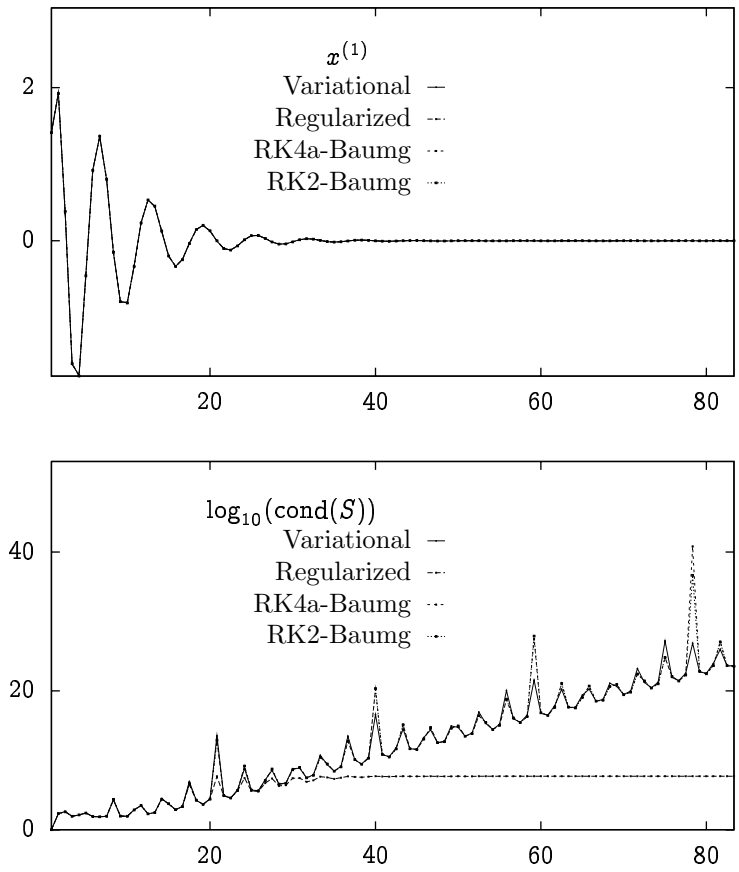

Figure 7.3: Integration of the planar slider crank with viscous friction of magnitude $\gamma = 0.3$, using four different methods. The initial conditions are $\phi = \pi/4$ and $\dot{q} = 0$. See text for details on the methods.

8 Bagatelle V: High Oscillations

The seminal constraint realization paper of Rubin and Ungar [241] contains a simple illustrative example in two dimensions which captures all the essential aspects of the limit behavior of strong forces. The same example is investigated here to give flesh to the theory presented in the previous chapters. A simple highly oscillatory planar mechanical system is described in Section 8.1 along with its known analytic solution. Section 8.2 illustrates how the SPOOK stepper of Section 4.4 removes the highly oscillatory parts with numerical experiments. General observations are collected in Section 8.3.

8.1 High oscillation example

The simplest possible case is a two-dimensional particle constrained to move on a straight line subject to a constant force normal to that line as illustrated in Figure 8.1.

Consider a point particle of unit mass moving in two dimensions with coordinates $q(t) : \mathbb{R} \mapsto \mathbb{R}^2, q = (q^{(0)}(t), q^{(1)}(t))^T$. Subject this particle to a linear constraint, i.e., restrict it to move on the line $g(q) = q^{(0)} - q^{(1)} = 0$. This constraint has constant Jacobian $G = \partial g/\partial q = [1, -1]$. Add the constant force $f = [-1, 1]^T$, corresponding to the potential $V(q) = q^{(0)} - q^{(1)}$, which acts trying to veer it off the constraint surface.

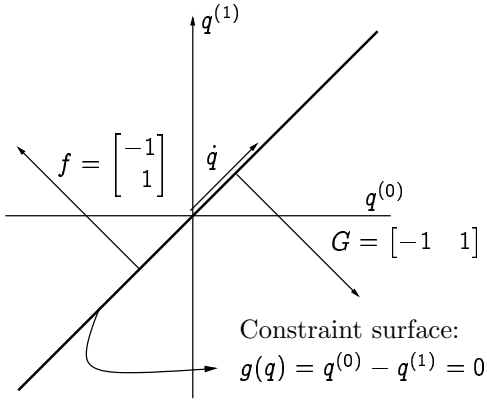

Figure 8.1: Schematics of a two-dimensional linearly constrained system.

This is now a system subject to a linear constraint equation and a constant force that is orthogonal to the constraint surface. Without any other applied force, the particle should move with constant velocity along the line $q^{(0)}(t) - q^{(1)}(t) = 0$.

The question now is what happens to the trajectory if the constraint is replaced by a strong force, and, in particular, how does the particle respond to the constant unit force which acts against the constraint force.

First consider the exactly constrained problem. The analytic formulation of the equations of motion starts from the constant mass model Lagrangian of (3.12) using the mass matrix $M = I_2$, i.e., the 2×2 identity matrix, and the constrained equations of motion of (3.92) to yield

$$\begin{bmatrix} \ddot{q}^{(0)} \\ \ddot{q}^{(1)} \end{bmatrix} - \lambda \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \qquad q^{(0)} - q^{(1)} = 0. \tag{8.1}$$

This can be solved by inspection by setting $\lambda = 1$. But for completeness, differentiating the constraint equation twice with respect to time yields $G\ddot{q} = 0$, and this leads to the linear system of equations

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} \ddot{q}^{(0)} \\ \ddot{q}^{(1)} \\ \lambda \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix}, \tag{8.2}$$

and these can easily be solved to yield the solution $\lambda = 1$, producing the constraint force $G^T\lambda = -f$, exactly balancing the applied force. This leaves the particle free to move along the line $q^{(0)}(t) = q^{(1)}(t)$ according to the reduced equations of motion

$$\ddot{q}^{(0)} = \ddot{q}^{(1)} = 0, \tag{8.3}$$

the solutions of which have constant velocity $v = \dot{q}^{(0)} = -\dot{q}^{(1)}$, as per initial conditions.

To realize the constraint as the limit of a strong force, introduce the potential function $V_\epsilon = \frac{1}{2\epsilon} g^2(q)$, and impose initial conditions $q(0) = 0, \dot{q} = [1, 1]^T$. This leads to the equations of motion

$$\begin{aligned} \ddot{q}^{(0)} + \left(1 + \frac{1}{\epsilon}(q^{(0)} - q^{(1)})\right) &= 0 \\ \ddot{q}^{(1)} - \left(1 + \frac{1}{\epsilon}(q^{(0)} - q^{(1)})\right) &= 0. \end{aligned} \tag{8.4}$$

Introducing the variables $x = q^{(0)} + q^{(1)}$ and $y = \epsilon + q^{(0)} - q^{(1)}$, the ODEs of motion are transformed to the decoupled equations

$$\begin{aligned} \ddot{x} &= 0, \\ \ddot{y} + \tfrac{2}{\epsilon} y &= 0, \end{aligned} \tag{8.5}$$

which have solution $x = \alpha t, y = \beta \sin(\omega_\epsilon t) + \gamma \cos(\omega_\epsilon t)$, where $\omega_\epsilon^2 = 2/\epsilon$, and α, β and γ are determined from the initial conditions to be $\alpha = 2, \beta = 0, \gamma = \epsilon$. Therefore, the complete solution of (8.4) reads

$$\begin{aligned} q_\epsilon^{(0)}(t) &= t - \frac{\epsilon}{2}\left(1 - \cos\left(\sqrt{\tfrac{2}{\epsilon}}t\right)\right), \\ q_\epsilon^{(1)}(t) &= t + \frac{\epsilon}{2}\left(1 - \cos\left(\sqrt{\tfrac{2}{\epsilon}}t\right)\right). \end{aligned} \tag{8.6}$$

The value of $g(q)$ and $\lambda = -\epsilon^{-1} g(q)$ are thus computed to be

$$\begin{aligned} g_\epsilon(t) &= q^{(0)} - q^{(1)} = -\epsilon\left(1 - \cos\left(\sqrt{\tfrac{2}{\epsilon}}t\right)\right), \\ \lambda_\epsilon(t) &= -\frac{1}{\epsilon} g(t) = 1 - \cos\left(\sqrt{\tfrac{2}{\epsilon}}t\right). \end{aligned} \tag{8.7}$$

It is evident from (8.7) that $\lim_{\epsilon \to 0} g_\epsilon = 0$. However, λ_ϵ does not have a limit and it oscillates more and more rapidly in the interval $[0, 2]$ as $\epsilon \to 0$.

A standard integration method is bound to fail on this problem unless these oscillations are damped somehow. The problem is illustrated in Figure 8.2 which portraits the analytic solution of the constraint force which quickly becomes intractable. The SPOOK stepper of Section 4.2, without the stabilization and damping terms introduced in Section 4.4 fares slightly better as shown in Figure 8.3. The averaging process used to construct this numerical method does work in decreasing the amplitude of the oscillations but not their frequency, which is problematic. The next section introduces damping and illustrates how this helps the numerical method as well.

8.2 Damping the high frequency oscillations

Adding a Rayleigh function of the form $\mathfrak{R} = \eta^{-1} \dot{g}^2(q)$ produces the drag force $-\eta^{-1} G G^T \dot{q}$ and after a few computations, using the same change of variables as previously, the new equations of motion are

$$\begin{aligned} \ddot{x} &= 0 \\ \ddot{y} + \frac{2}{\eta}\dot{y} + \frac{2}{\epsilon} &= 0. \end{aligned} \tag{8.8}$$

To analyze these new equations of motion, set $\omega_\epsilon = \sqrt{2/\epsilon}$ and introduce the damping ratio $\zeta = \eta^{-1}\omega_\epsilon^{-1}$. Changing the time variable to $\tau = \omega_\epsilon t$, and introducing $u(\tau) = y(\omega_\epsilon t)$, the differential equation for y transforms to:

$$\ddot{u} + 2\zeta\dot{u} + u = 0, \tag{8.9}$$

where the $\dot{u} = \mathrm{d}u/\mathrm{d}\tau$ is now the natural time derivative. The solutions of (8.9) is easily verified to be

$$u(\tau) = \exp(-\zeta\tau)\left(u(0)\cos(\xi\tau) + \frac{\dot{u}(0) + \zeta u(0)}{\xi}\sin(\xi\tau)\right), \tag{8.10}$$

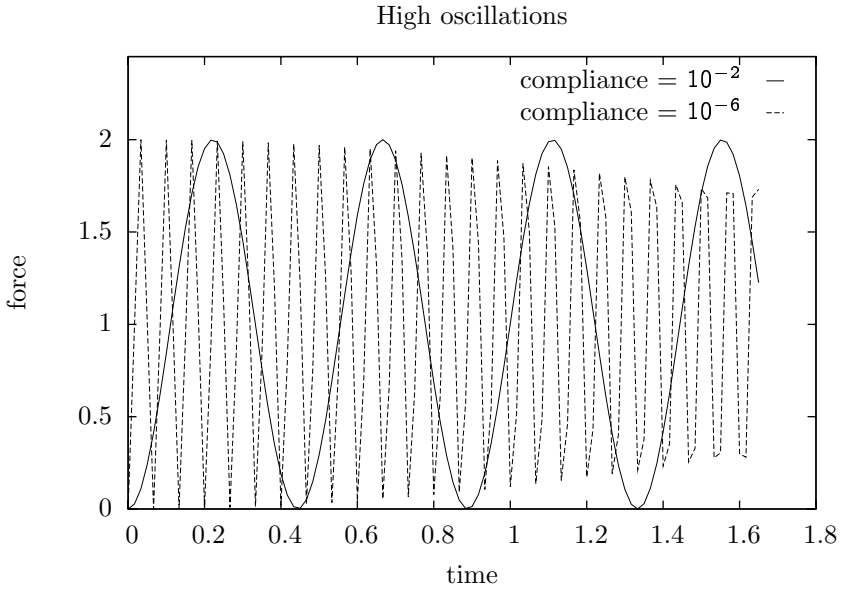

Figure 8.2: High oscillations in the analytic solution of a constraint realization problem in two dimensions.

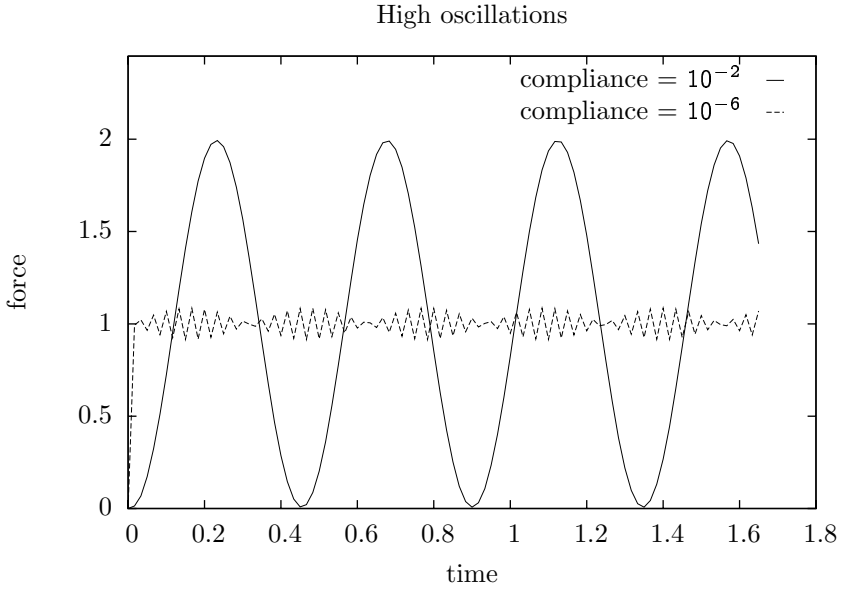

Figure 8.3: High oscillations in a regularized but unstabilized stepping scheme.

where $\tau = \omega_\epsilon t$ is the natural time, and where $\xi = \sqrt{1-\zeta^2}$ is allowed to take imaginary values. For the case $\xi = 0$, the second term in (8.10) reduces to $\dot{u}(0) + u(0)\tau$.

Note that the variable $x(t)$ is unaffected by the damping term. The solution of the problem now involves the following variables

$$\begin{aligned} y(t) &= u(\tau), \\ g(q(t)) &= y(t) - \epsilon = u(\tau) - \epsilon, \\ \dot{g}(q(t)) &= \dot{y}(t) = \omega_\epsilon \dot{u}(\tau), \\ \lambda_\epsilon &= -(1/\epsilon) g(t) - (1/\eta)\dot{g}(t) \\ &= 1 - \frac{\omega_\epsilon^2}{2}\left(u(\tau) - 2\zeta\dot{u}(\tau)\right). \end{aligned} \tag{8.11}$$

By taking $\epsilon \to 0$ limits of various terms in (8.11) for fixed damping ratio ζ (and fixed $\xi = \sqrt{1-\zeta^2}$), the conclusion is that for *any* $\zeta > 0$, the functions $u_\epsilon(\tau), y_\epsilon(t), g_\epsilon(t), \lambda_\epsilon(t)$ all converge uniformly as $\epsilon \to 0$. All that is needed is the known fact that the following expression

$$\lim_{\omega \to \infty} \omega^n \exp(-\omega t \alpha) \mapsto 0, \tag{8.12}$$

converges uniformly, over any finite interval $[0, T]$, for any power $n > 0$, and for any scalar α with positive real part.

To compare this with the stabilization strategy of Section 4.4, note the following identity

$$\frac{1}{\eta} = \zeta\omega_\epsilon = \zeta\sqrt{\frac{2}{\epsilon}}, \tag{8.13}$$

and from this, the parameter τ of Section 4.4 is computed to be

$$\tau = \frac{\epsilon}{\eta} = \zeta\sqrt{2\epsilon}, \tag{8.14}$$

so that a fixed value of the numerical damping rate τ/h corresponds to an increasing damping ratio $\zeta = \alpha/\sqrt{\epsilon}$. Likewise, if the damping ratio ζ is fixed, the numerical damping rate τ/h decreases as the square root of the compliance parameter ϵ.

Simulations were made using the SPOOK stepper for different values of compliance and damping ratio, and compared with the exact result. Several cases are now illustrated in which the initial conditions are either consistent or not.

For moderate frequency, using $\epsilon = 10^{-2}$ and damping ratios of either $\zeta = 1$ or $\zeta = 0.1$, the exact trajectories are shown in Figure 8.4 and the result of simulations with the SPOOK stepper in Figure 8.5. These graphs offer little surprise, since oscillations are suppressed as one would expect from looking at the equations. Numerically, these equations are not very stiff either.

However, when the compliance is reduced to $\epsilon = 10^{-6}$, the picture becomes more interesting. Of course, the analytic solution immediately reaches equilibrium in Figure 8.6 but so does the numerical method in Figure 8.7, provided the

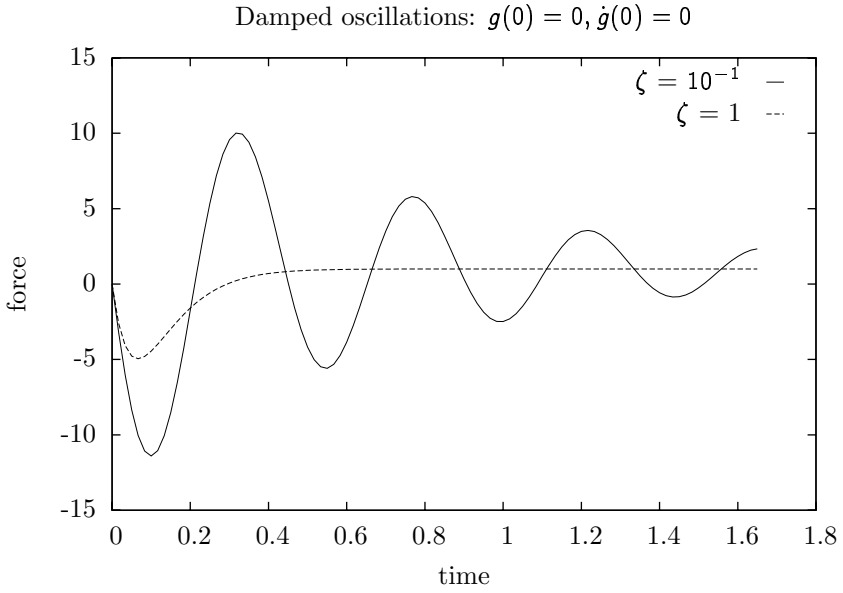

Figure 8.4: The effect of damping with regularization $\epsilon = 10^{-2}$ for the exact solution.

damping ration is set to $\zeta = 1$. Note also that in the numerical implementation, the case $\zeta = 1$ is not critically damped but slightly under damped. Nevertheless, oscillations quickly die out.

Fixing the compliance to $\epsilon = 10^{-4}$, Figures 8.8, 8.9 and 8.10 illustrate the behavior of the exact solution for the cases where the initial position, initial velocity, or both initial position and velocity violate the constraint. As expected from the analytic formulation, the transients quickly die out, especially for critical damping $\zeta = 1$, and the unperturbed solution is recovered.

For the same value $\epsilon = 10^{-4}$, Figures 8.11, 8.12, and 8.13 illustrate the numerical solution computed by the SPOOK stepper and these reflect precisely the behavior of the exact solution. The value of ϵ could easily be decreased much further in this case, as demonstrated in the analysis of Section 4.6, without any concern for stability. The graphs would not be so instructive though as all the stabilization dynamics would occur in one or two steps when using $\zeta = 1$.

This is an example of a *bona fide* stiff system of differential equations as it has both high oscillations and high damping. Discretization of such systems requires great care to avoid instabilities and to separate the fast dissipative modes from the rest of the dynamics.

But the interesting conclusion is that suitable damping will extract the correct slow motion from the highly oscillatory parts and remove sensitivity to the initial conditions.

The SPOOK stepper of Section 4.4 is successful at extracting the correct slow motion without much tuning.

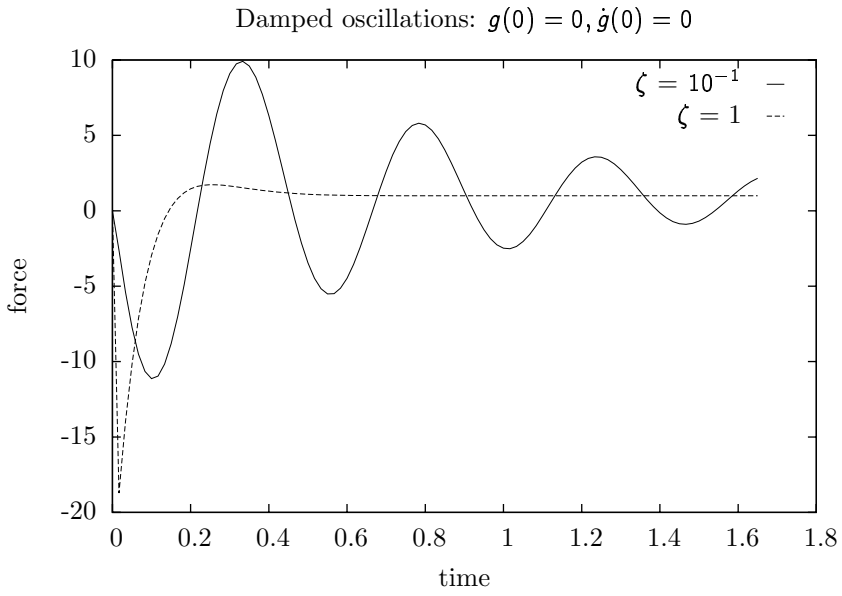

Figure 8.5: The effect of damping with regularization $\epsilon = 10^{-2}$ for the numerical solution of the SPOOK stepper.

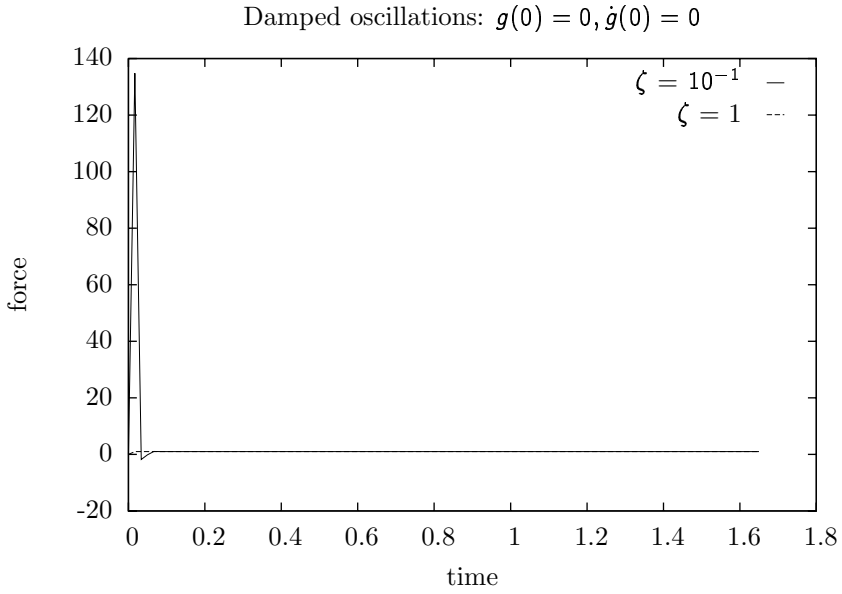

Figure 8.6: The effect of damping with regularization $\epsilon = 10^{-6}$ for the exact solution.

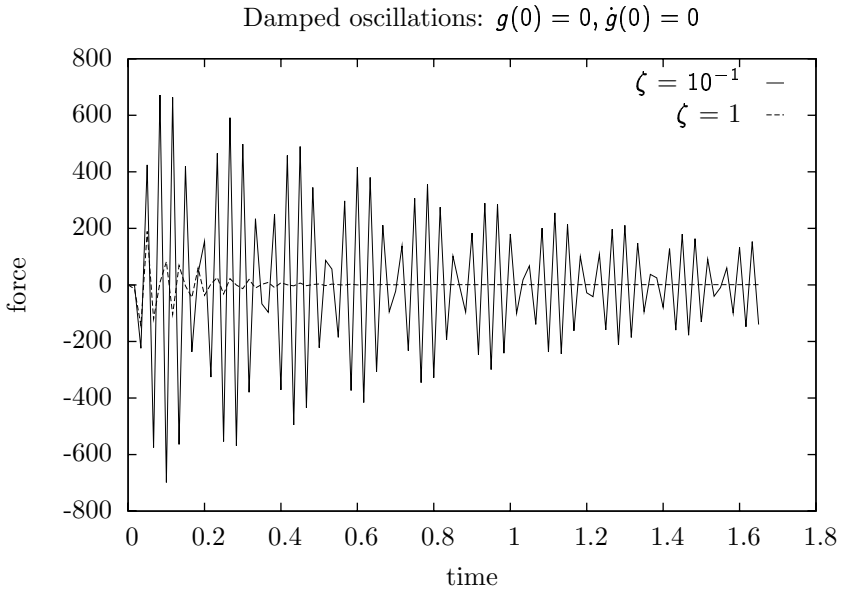

Figure 8.7: The effect of damping with regularization $\epsilon = 10^{-6}$ for the numerical solution of the SPOOK stepper.

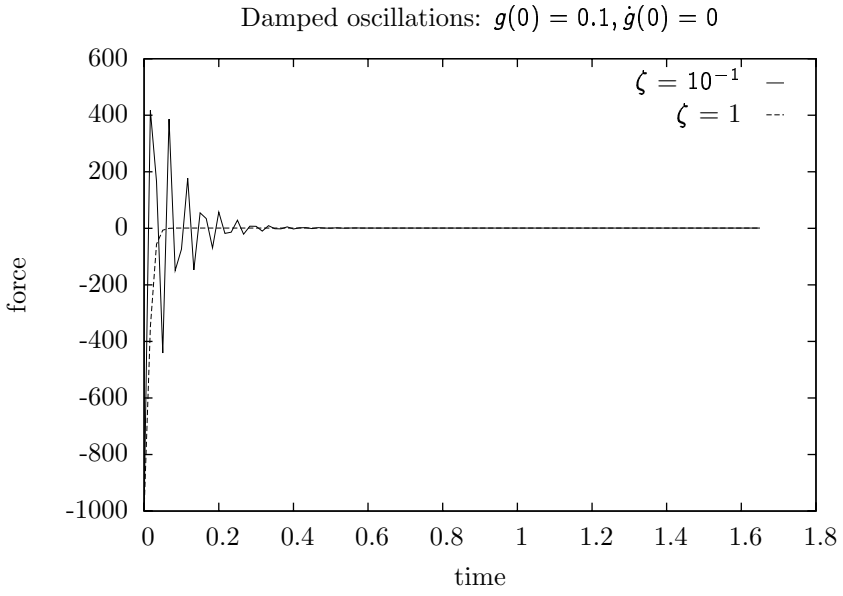

Figure 8.8: The effect of damping with regularization $\epsilon = 10^{-4}$ on the exact solution when starting from an inconsistent initial condition.

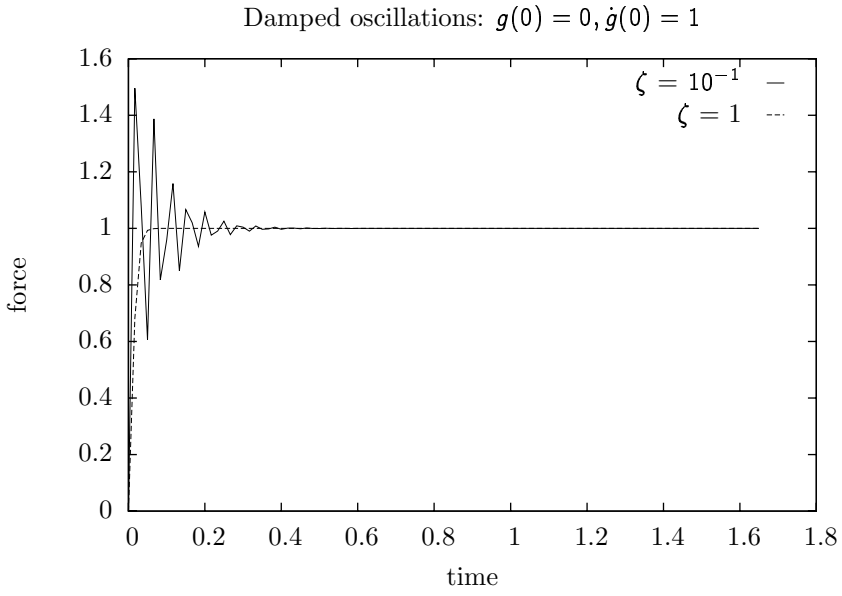

Figure 8.9: The effect of damping with regularization $\epsilon = 10^{-4}$ on the exact solution in the case of inconsistent initial velocity.

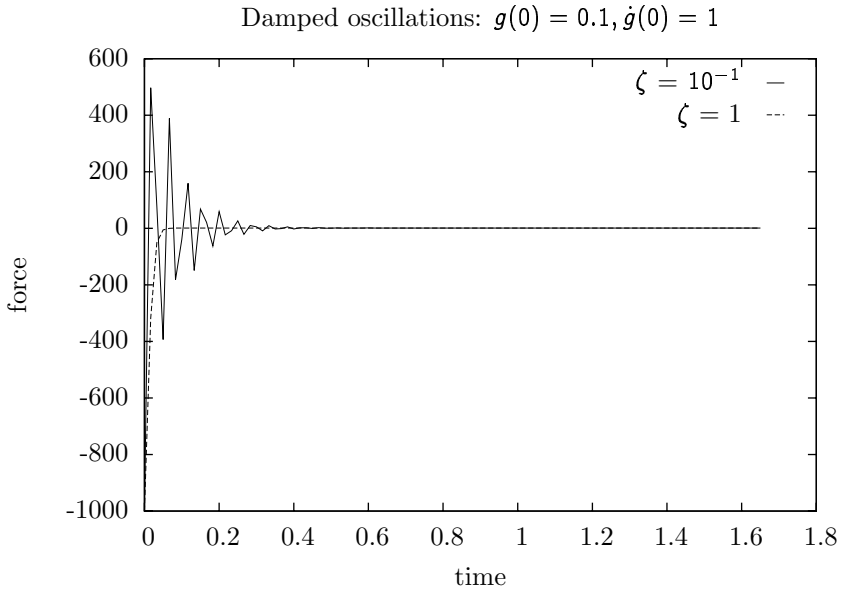

Figure 8.10: The effect of damping with regularization $\epsilon = 10^{-4}$ on the exact solution when both the initial position and velocity are inconsistent.

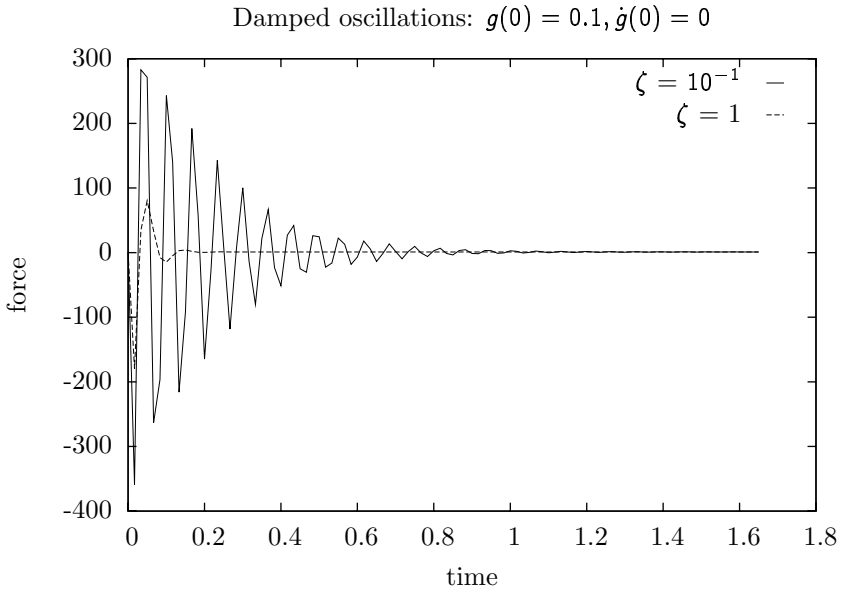

Figure 8.11: The effect of damping with regularization $\epsilon = 10^{-4}$ on the variational solution when the initial position is inconsistent.

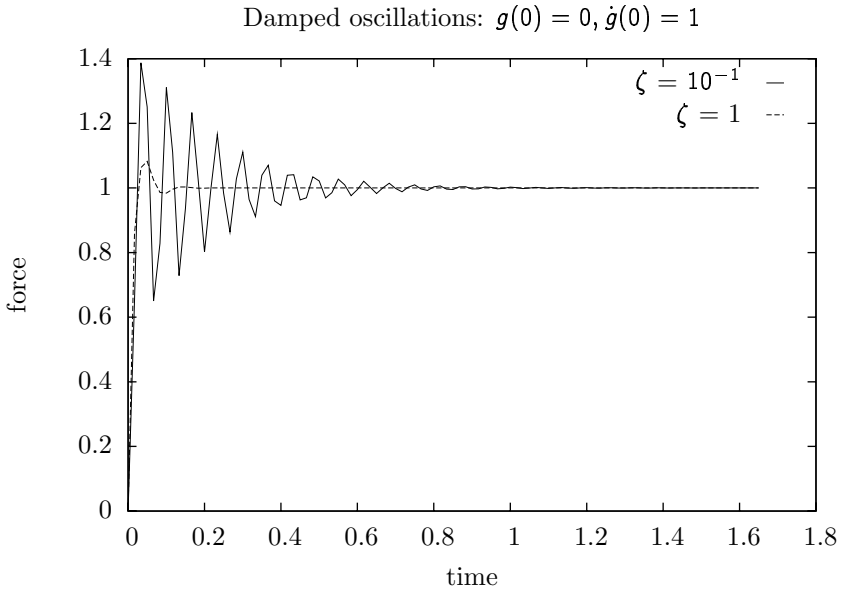

Figure 8.12: The effect of damping with regularization $\epsilon = 10^{-4}$ on the variational solution when the initial velocity is inconsistent.

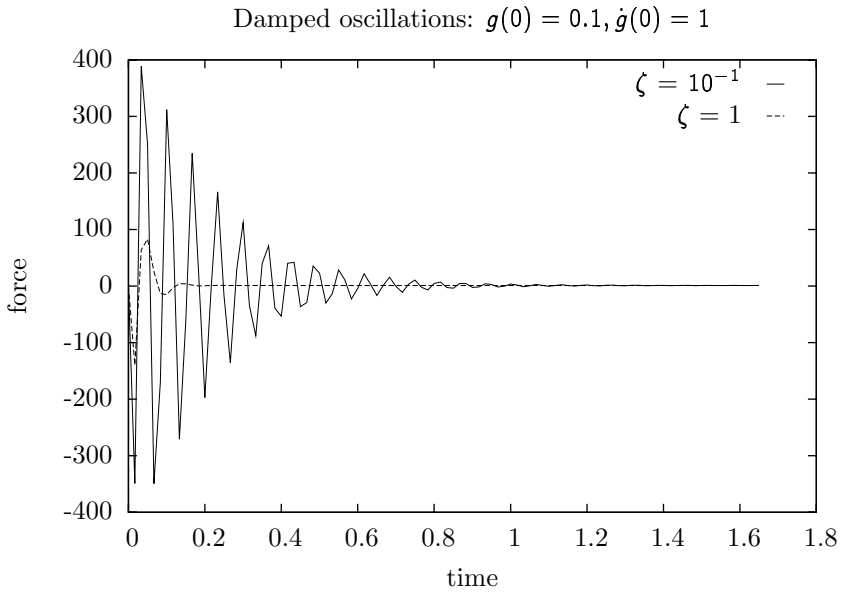

Figure 8.13: The effect of damping with regularization $\epsilon = 10^{-4}$ on the variational solution when both the initial position and velocity are inconsistent.

8.3 End notes

The theory of constraint realization of Rubin and Ungar [241] predicts that penalty forces suffer from high oscillations in the absence of damping. As this was considered in the development of the constraint stabilization scheme of Section 4.4, simple numerical experiments show that indeed, the SPOOK stepper does suppress these oscillations in a stable way. This shows that, at least for linear cases, the SPOOK stepper can handle constraint inconsistencies and recover from large constraint violations.

9 Bagatelle VI: Smooth Impacts

Impacts are physical phenomena best idealized as instantaneous events resulting in discontinuous velocity changes, at least in the context of interactive simulations. A simple damped harmonic oscillator model is introduced to illustrate how it is possible to summarize the result of fast dynamics into macroscopic constitutive laws. Section 9.1 presents the model and the analytic solution. This is followed by analysis for the analytic behavior for the oscillatory regime in Section 9.2, the critically damped regime in Section 9.3, and the overdamped regime in Section 9.4. The limitations of the model and the need for the nonsmooth models of Chapter 10 are discussed in Section 9.5.

9.1 Introduction

The surface of solids is a physical system which is at equilibrium under normal circumstances. But when the surfaces of two solids come into near proximity, the disturbances produce forces which act to prevent any penetration. For many solids such as steel, wood, or glass, these forces become so strong so quickly that hardly any visible deformation is produced. When two billiard balls collide for instance, the time taken by the surface forces to change the velocities is of the order of microseconds and therefore, essentially instantaneous. The deformations are so small that a linear analysis of surface forces is sufficient.

When two solids impact at a high incident velocity, some of the energy of the impacting momentum is transferred to vibrational modes and from there, to sound and heat. Though the incident velocities are considered large, the analysis still starts with a linear dissipation process as this is sufficient for the present purpose.

Consider a point particle with mass m and position $q : \mathbb{R} \mapsto \mathbb{R}$ in one spatial dimension. Subject this particle to a one-sided spring damper system with stiffness k and damping constant b so the force generated is

$$f = \begin{cases} -kq - b\dot{q} \text{ when} & q < 0 \\ 0 & \text{otherwise.} \end{cases} \tag{9.1}$$

Consider the initial conditions $q(0) = 0, \dot{q}(0) = -v_0, v_0 > 0$, i.e., the particle starts at the origin but moving towards the $q < 0$ forbidden region. In this region, the spring-damper force is active and the trajectory for this system is now computed from $t = 0$ up to the time when the force becomes inactive again,

i.e., when $q(t) = 0, \dot{q}(t) > 0$. This corresponds to one full cycle of the spring-damper system.

Using Newton's second law of motion (3.2), the trajectory must satisfy the equation

$$m\ddot{q} + b\dot{q} + kq = 0. \tag{9.2}$$

To simplify the notation of the solution, write

$$\gamma = b/m, \quad \omega^2 = k/m, \quad \tau = \omega t, \quad \zeta = \frac{\gamma}{2\omega} = \frac{b}{2\sqrt{k}}\frac{1}{m}. \tag{9.3}$$

After changing variables to $x(\tau) = x(\omega t) = q(t)$, the canonical equation is recovered

$$\ddot{x} + 2\zeta\dot{x} + x = 0, \tag{9.4}$$

where the dots denote derivatives with respect to the natural time variable, τ, and where ζ is called the damping ratio. Solutions for this are

$$x(\tau) = ae^{\tau\lambda_+} + be^{\tau\lambda_-}, \tag{9.5}$$

where $\lambda_\pm$ are the roots of the indicial equation

$$\begin{aligned} \lambda^2 + 2\zeta\lambda + 1 &= 0 \\ \lambda_\pm &= -\zeta \pm \sqrt{\zeta^2 - 1} = -\zeta \pm \xi \\ \xi &= \sqrt{\zeta^2 - 1}. \end{aligned} \tag{9.6}$$

For the initial conditions under considerations, this leads to

$$x(\tau) = \frac{v_0}{\xi} e^{-\zeta\tau} \sinh(\xi\tau). \tag{9.7}$$

In turn, this produces an oscillatory regime for $0 \leq \zeta < 1$, a critically damped regime for $\zeta = 1$, and an overdamped regime for $\zeta > 1$.

9.2 Oscillatory regime

When the damping ratio is in the range $0 \leq \zeta < 1$, the solution satisfying the initial conditions $x(0) = 0, \dot{x}(0) = -v_0/\omega$ is

$$x(\tau) = -\frac{v_0\sqrt{1-\zeta^2}}{\omega} e^{-\zeta\tau} \sin(\sqrt{1-\zeta^2}\tau). \tag{9.8}$$

The second zero occurs when $\sqrt{1-\zeta^2}\tau^+ = \pi$ at which point the velocity is

$$v^+ = \dot{q}(\omega^{-1}\tau^+) = \omega\dot{x}(\tau^+) = v_0 e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}}. \tag{9.9}$$

Setting the restitution coefficient $r < 1$ as the exponential in this expression, labeling $v^- = -v_0$, the Newton impact restitution law is recovered

$$v^+ = -rv^-, \quad r = \exp\left(-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}\right). \tag{9.10}$$

Now, the damping ratio is defined as

$$\zeta = \frac{b}{2\sqrt{k}}\frac{1}{m}, \tag{9.11}$$

so that for fixed m, it is possible to let $b \to \infty$, $k \to \infty$ with constant ζ without changing any of the behavior. Therefore, provided the response time is small compared to other time scales in the system—and it is—the contact physics which changes the incident velocity v^- to v^+ can be considered as instantaneous. The maximum penetration is proportional to v_0/ω and is of course small for a high frequency oscillator.

It is also clear that using a simple spring-damper model for contacts and impacts is a bad idea because of the mass dependence of the damping ratio and the very high frequencies involved. If the frequencies are allowed to be moderate, then, the penetration will be large. A mass-independent solution is better in general and this is constructed in Chapter 10.

9.3 Critical damping

When the damping ratio reaches unity, $\zeta = 1$, the solution is an exponential decay

$$x(\tau) = -\frac{v_0}{\omega}\tau e^{-\tau} \leq 0. \tag{9.12}$$

Therefore, this contact never becomes free again and the velocity goes to 0 asymptotically, with a maximum penetration at $\tau = 1$

$$x(1/\omega) = -\frac{v_0}{\omega}\frac{1}{e}, \tag{9.13}$$

which can be made arbitrarily small by choosing a very large frequency.

9.4 Overdamping

When the damping ratio ζ increases beyond unity, the solutions of (9.4) is

$$x(\tau) = -\frac{v_0}{\omega\sqrt{\zeta^2 - 1}}e^{-\zeta\tau}\sinh(\sqrt{\zeta^2 - 1}\tau), \tag{9.14}$$

and again, this is always negative and never escapes. With this process, the incident velocity quickly vanishes and the point particle stays near $q = 0$.

9.5 End notes

Though a linear spring and damper model of contacts is easily analyzed and easily implemented, correctly yielding known phenomenology of impacts, the frequencies typically involved are too high for simple explicit or semi-implicit integration methods. In addition, the behavior of this model depends critically

on the mass of the bodies in contact, since both the frequency ω and damping ratio ζ depend on the impinging mass, and thus requires careful tuning. For multibody systems, though the mass of each component is known, the effective mass at a contact point depends on the configuration of the system and can vary greatly between that of a single component and that of all of them. For instance, the effective mass of a well aligned stack of books resting on a table is the sum of the individual masses. But if this is knocked away from equilibrium, the mass effective inertia felt at the point of contact with the table decreases to that of the books at the bottom of the pile.

But the spring and damper model teaches one important lesson. The exact details of the contact dynamics are irrelevant and the net effect can be modeled using a single mass independent impact parameter. There are alternatives for impact models [56] but the Newtonian restitution coefficient and the impact law (9.10) suffice for the present purposes. This is good news for the applicability of the SPOOK stepper as well since the linear stability analysis of Section 4.6 shows that it can in fact process linear systems with both high frequencies and high damping coefficients in a stable way, without tuning. Chapter 10 goes into the details of this.

10 Nonsmooth Problems: Contacts and Friction

Nonsmooth phenomena arising from impacts and dry friction are investigated in the context of the variational method. After introducing the general context of contacts and other nonsmooth problems in Section 10.1, a geometric formulation of nonpenetration conditions and the description of impact discontinuities are described in Section 10.2. This is followed by a review of essential elements of nonsmooth analysis in Section 10.2, and this theory is then used to present the exact variational impact resolution method of [87] in Section 10.4. This is then approximated to yield a two stage impact resolution process that is guaranteed to be dissipative in Section 10.5 and the results are illustrated and compared with numerical experiments in Section 10.6.

Nonsmooth forces and nonideal constraints are introduced in Section 10.7 using the variational framework developed in Section 3.12. This is used to build dry friction models in stages. Velocity limit constraints are first considered in Section 10.8 which are found to be strictly dissipative in Section 10.9. This is followed with an analysis of ghost constraints—general constraints on the ghost variables themselves—in Section 10.10. Both the phenomenology of dry friction and the variational modeling using all the previously defined elements are presented in Section 10.11 which contains a novel, isotropic, regularized and solvable dry friction model, and a solvability analysis for it. Short descriptions of alternative dry friction models are provided in Section 10.12. The results of a well-known one dimensional numerical experiment and general comments are provided in Section 10.13

10.1 Introduction

Surface interactions between electrically neutral solids are combinations of electrostatic and quantum-mechanical phenomena of great complexity as the source of much scientific and technological interest. However, these interactions take place over such small distances and time scales that they are entirely irrelevant in the context of interactive simulations except in their geometrical and constitutive aspects, such as non-penetration conditions and dry friction laws. The exact time and length scales are not relevant here but they are several orders of magnitude smaller than the ten millisecond and ten millimeter resolution used in the interactive simulations. In this context, impacts are instantaneous and occur in-place. This means that when two bodies meet with a non-vanishing normal velocity for instance and, very shortly thereafter depart or stick together, the

change in velocity is considered to happen instantaneously and without producing any change in position.

Also, solid bodies in resting contacts are subject to normal separation forces as well as dry friction forces which lie in the plane tangent to the contact normal. The normal of this plane is given by the gradient of the separating distance function. Dry friction exhibits two modes, namely, a *stiction, rolling, or static* mode, wherein the relative tangential velocity vanishes, and a *friction, sliding, or kinetic* mode mode wherein the tangential contact forces have magnitude proportional to the normal contact forces and directly oppose the finite relative tangential velocity. The normal forces are presumed to arise from localized surface deformations but these are assumed to be so small as to be entirely negligible.

Impacts and dry friction are nonsmooth phenomena, producing jump discontinuities in the velocities of the system. They are naturally best analyzed from a *discrete* viewpoint dealing only with the continuous positions and impacts—time integrals of forces. This is precisely the framework provided by the discrete variational formulation exposed in Chapter 3.

This formulation is extended in the present chapter to introduce an economical, dissipative impact resolution algorithm, as well as an isotropic friction formulation based on NCP which is proved to be solvable. The linearization of this NCP is showed to correspond to previously known linear complementarity problem (LCP) models except for the physics motivated diagonal regularization terms.

10.2 Normal contact forces and impacts

When the surfaces of solids come into close proximity, the very strong microscopic interactions and bulk elastic forces translate essentially to unilateral constraints at the macroscopic level, imposing non-penetration conditions between the bodies. Considering any two bodies, given a point $p^{(0)}$ fixed on body 0, say, and $p^{(1)}$ fixed on body 1, the *signed* distance between these two points should be non-negative: $c(p^{(0)}, p^{(1)}) \geq 0$. Such inequality constraints necessarily lead to *impacts*, i.e., forces which are so large and vary so rapidly as to cause jump discontinuities in the velocities $\dot{q}$, at least for any reasonable observation scale. To clearly separate the time scales, impacts are idealized as true mathematical jump discontinuities and so that if an impact occurs at time t_i, then

$$\lim_{t \uparrow t_i} \dot{q}(t) = \dot{q}_- \neq \dot{q}_+ = \lim_{t \downarrow t_i} \dot{q}(t), \tag{10.1}$$

i.e., the velocity before and after the impact differ.

For such impacts, the trajectories $q(t)$ remain continuous but the velocities $\dot{q}(t)$ exhibit jump discontinuities, implying that accelerations $\ddot{q}(t)$ are not defined at impact times, t_i.

The first thing to consider is the treatment of the impact impulses and contact forces arising from the non-penetration conditions $c(p^{(i)}, p^{(j)}) \geq 0$ between any two points $p^{(i)}, p^{(j)}$ fixed on bodies i and j, respectively. Though this constraint is expressed in terms of body coordinates, it is not strictly holonomic since there

is no global change of coordinates which can remove this degree of freedom from the system. Consider a spherical rigid body and an infinite plane for instance. From the convexity of the geometries involved, the closest point between these two objects can be identified uniquely. The signed distance function can be written directly in terms of the rigid body center of mass position, $x(t)$, say, as $c(q(t)) = \tilde{c}((x(t)) \geq 0$. When the sphere contacts the plane so that $\tilde{c}(x(t)) = 0$, the vertical motion of the sphere can be eliminated. When the sphere is in free flight, the vertical motion must be added. Such an approach would lead to two configuration manifolds, namely, $Q_1 \subset \mathbb{R}^2 \times SO(3)$ for the planar motion, and $Q_2 \subset \mathbb{R}^3 \times SO(3)$ for the spatial motion, and where $SO(3)$ is the group of rigid rotations in three dimensions. Selecting between these two manifolds is cumbersome and becomes quickly intractable if one considers a large collection of contacting rigid bodies. This is where the extended coordinate formulation becomes advantageous for its simplicity and uniformity.

Continuing with the simple example of a solid sphere and an infinite plane, our configuration manifold remains $Q \in \mathbb{R}^3 \times SO(3)$, but a Lagrange multiplier $\nu \in \mathbb{R}$ must also be added. This multiplier is the magnitude of the force acting in the normal direction at the contact. However, a further restriction is imposed on ν given the inequality constraint $c(q) \geq 0$. To see this, start from the regularized constraint formulation of (4.7). Choose $\epsilon > 0$, so that $c(q)$ is now allowed to become negative, producing a restoring force $\nu = -\epsilon^{-1}c(q)$, which shows again that ϵ is much like the inverse of a spring constant. But typical contact forces have very little adhesion and it is expected that $\nu = 0$ whenever the separation is positive, i.e., $c(q) > 0$.

Another perspective on this is the consideration of the augmented discrete Lagrangian and the discrete principle of least action. Ignoring regularization for the moment and discretizing the coupling term with

$$\int_0^h \mathrm{d}s \nu^T c(q(s)) \approx h {\nu_0}^T c_0, \tag{10.2}$$

the least action principle demands that the discrete action $\mathbb{S}_d(q_0, \ldots, q_N, h)$ be minimized over the *allowed* trajectories, namely, $c(q_k) \geq 0$. Using the well known theorem of Karush-Kuhn-Tucker [49], the stationarity conditions read

$$\begin{aligned} D_1\mathbb{L}_d(q_k, q_{k+1}, h) + D_2\mathbb{L}_d(q_{k-1}, q_k, h) + h{C^T}_k\nu_k = 0 \\ 0 \leq c_{k+1} \quad \perp \quad \nu_k \geq 0, \end{aligned} \tag{10.3}$$

where the perpendicularity sign $\perp$ is understood component-wise in the case where $c : Q \mapsto \mathbb{R}^m, m \leq \dim Q$, is a vector function leading to $\nu \in \mathbb{R}^m_+$.

Now, though the formulation of (10.3) clearly imposes the desired constraint $c_k \geq 0$ and clearly demonstrates the complementarity condition between the components of the Lagrange multiplier ν and the components of the distance function $c(q)$, this strategy does not conserve energy at all. In addition, it does not allow for a description of the physics of impacts which usually includes energy dissipation.

This can be remedied in several different ways. For instance, in the strategy exposed in [87], the times t_i at which the signed distance function $c(q)$ crosses zero are located using a binary search and an impact resolution stage is performed at these instants so that the energy is kept constant before and after, $E(t_i^+) = E(t_i^-)$, and the outbound velocity is in the normal cone of the restriction manifold $c(q) \geq 0$ at the impact location, i.e., $C(q(t_i))v(t_i^+) \geq 0$. In other words, the final velocity is receding and the energy is preserved. Such a strategy is expensive to implement as observed by the authors of [87].

Instead, the intention here is to construct a fixed time-step strategy. This implies that penetration constraint violation are detected *post facto* or, if a predictor estimate is used, preemptively, since impacts have zero probability to occur precisely at time kh, for some integer k. This has two implications. The first is that a constraint stabilization strategy is required so that violations $c(q_k) < 0$ are quickly stabilized back to $c(q_{k+j}) \geq 0$ for some integer $j > 0$. The second is that impacting velocity must be resolved without regards to the actual penetration depth as the two aspects are treated independently.

The strategy proposed here hinges on the dissipative physical constraint stabilization scheme introduced in Section 4.4 and the known complementarity conditions of (10.3). The idea here is that a stabilization parameter τ is chosen so that $\tau/h = 1$, there should be almost instantaneous dissipation of the incident velocity $C\dot{q} > 0$ which violates the constraint at the time of impact. The constraint stabilization scheme should then produce $c(q_k) \approx 0$ after a few steps. Modifying the stepping equations for the SPOOK steeper of (4.27) to restrict both the Lagrange multiplier and the constraint violation to be non-negative, as done further below, should yield a reasonable approximation for non-adhesive contact forces. The drawback is that this model leads to totally inelastic collisions. This remedied by an impact stage during which a simplified version of the energy preserving impact resolution of [87] is performed. The overall scheme is proven to be is strictly dissipative, though experiments show that the artificial dissipation rate is small. Given the linear stability Theorem 4.4, the constraint violations are expected to quickly disappear in a stable way.

10.3 Essential notions of nonsmooth analysis

Before delving in the analysis of nonsmooth contacts, it is necessary to understand the generalization of concepts of tangents and normals to cover nonsmooth surfaces. A very short overview is provided here. The reader is referred to the monograph of Clarke [66], from which the present section is adapted, for a thorough treatment of optimization problems on nonsmooth sets.

Considering that all geometrical objects of standard 3D graphics used in a virtual environment (VE) are defined in terms of polygons and that these are nonsmooth at all edges and all vertices, a suitable definition of contacting surfaces for the scope of the present thesis must include the nonsmooth case.

First consider the a scalar function $f : \mathbb{R}^n \to \mathbb{R}$ which satisfies a Lipschitz

condition at point x as follows

$$\|f(x'') - f(x')\| \leq K\|x'' - x'\|, \text{ for all } x', x'' \in x + \epsilon B, \tag{10.4}$$

where B is the unit radius ball centered at the origin, and $\epsilon > 0$ is a small positive number. Here, x' merely means "a point other than x", and x'' means " a point other than x or x'".

Note that f need not be continuous to have this Lipschitz property. Next, instead of using standard one sided derivative, the notion of a generalized directional derivative of the function f evaluated at the point x in the direction of $v \in \mathbb{R}^n$ is introduced

$$f^\circ(x; v) = \limsup_{\substack{y \to x \\ \lambda \downarrow 0}} \frac{f(x + \lambda v) - f(y)}{\lambda}. \tag{10.5}$$

This limit exists at any point x where the function $f(x)$ satisfies the Lipschitz property (10.4). The properties of the subderivative (10.5) are such that a generalized gradient can be defined as

$$\partial f(x) = \{y \in \mathbb{R}^n \mid f^\circ(x; v) \geq v^T x \text{ for all } v \text{ in } \mathbb{R}^n\}. \tag{10.6}$$

For a piecewise continuous function which has discontinuities in the derivative at a point x, and such that it has m distinct one sided gradients, namely, the row vectors $\xi_1, \xi_2, \ldots, \xi_m$, the definition in (10.6) expresses the fact that ∂f is the convex hull of the row vectors $\xi_i, i = 1, 2, \ldots, m$.

Consider that function $f(x)$ satisfies the Lipschitz condition (10.4) for a given ball $x + \epsilon B$, and has countably many discontinuities within that same ball. These discontinuities are located in a set Ω_f which has measure zero so that f is differentiable almost everywhere in the Lebesgue sense. Assume also that there is another set S of measure zero which is to be avoided (for reasons clarified in [66]), then, the generalized gradient can be expressed as

$$\partial f(x) = \text{co}\{\lim_{i \to \infty} \partial f(x_i)/\partial x_i \mid x_i \to x, x_i \notin \Omega_f, x_i \notin S\}, \tag{10.7}$$

where $\text{co}\{\cdot\}$ is the convex hull of a given set.

The meaning of this definition is that the generalized gradient is the convex hull of all the gradients found within a small ball of a given neighborhood of dimension ϵ. Of course, for a continuous function, the gradient is the same within $O(\epsilon)$ for all sample points x_i and thus, the generalized gradient coincides with the standard one.

To apply this to the familiar notion of tangents and normals, start with the definition of the distance to a given set $\mathcal{A}_c \in \mathbb{R}^n$, defined, for instance, as the following set

$$\mathcal{A}_c = \{x \in \mathbb{R}^n \mid c_i(x) \geq 0, i = 1, 2, \ldots, m\}, \tag{10.8}$$

for m continuous functions $c_i : \mathbb{R}^n \mapsto \mathbb{R}$. With this or any other definition of an *allowed* region $\mathcal{A}_c$, the distance function $d_{\mathcal{A}_c}$ is defined as follows

$$d_{\mathcal{A}_c}(x) = \min_{y \in \mathcal{A}_c} (\|x - y\|). \tag{10.9}$$

The notion of a tangent to a smooth surface defined with $c(x) = 0$ is the normalized vector v which can be followed so that $c(x + \lambda v) = 0$ for small enough λ. In other words, the directional derivative of $c(x)$ in the direction of the tangent vanishes. Using the generalized directional derivative, the following definition for the *tangent cone* is produced

$$T_{\mathcal{A}_c}(x) = \{v \in \mathbb{R}^n \mid d^\circ_{\mathcal{A}_c}(x;\, v) = 0\}. \tag{10.10}$$

Now, the usual definition of a normal is that it should be orthogonal to the tangent. But the strict orthogonality must be relaxed here and replaced with an inequality to obtain the normal cone definition

$$N_{\mathcal{A}_c}(x) = \{\xi \mid \xi v \leq 0 \text{ for all } v \text{ in } T_{\mathcal{A}_c}(x)\}, \tag{10.11}$$

where ξ are now row vectors. The set $N_{\mathcal{A}_c}(x)$ can also be understood to be the closed convex cone of $\partial d_{\mathcal{A}_c}(x)$ using the definition (10.6), so that all directions making $d_{\mathcal{A}_c}$ increase should be included.

The definition (10.11) is important in that it establishes a fundamental inequality for the product between the row vectors in $N_{\mathcal{A}_c}(x)$ and the vectors $v \in T_{\mathcal{A}_c}$ and at a given point. However, Clarke [66] does provide a more intuitive definition of this set based on the notion of perpendicularity. Consider a point x' which is outside of the manifold $\mathcal{A}_c$. If point x is the unique vector on the boundary $\partial\mathcal{A}_c$ closest to x', then vector $u = x' - x$ is perpendicular to $\mathcal{A}_c$ at x, and this is written as $u_i \perp \mathcal{A}_c$.

Now, choosing a point x on the boundary $\partial\mathcal{A}_c$, collect all such perpendicular vectors u_i obtained from points $x_i \in x + \epsilon B, i = 1, 2, \ldots$, in the vicinity of x and build the closure of the convex hull

$$N_{\mathcal{A}_c}(x) = \bar{\text{co}}\{\lambda \frac{u_i^T}{\|u_i\|} \mid \lambda \geq 0, u_i \perp \mathcal{A}_c \text{ at } x,\, x_i \to x, u_i \to 0\}, \tag{10.12}$$

where $\bar{\cdot}$ denotes the closure of a set.

The inequality in the original definition of (10.11) is now easily understood. Consider one of the candidate points x_i in the definition of the set (10.12). The distance of that point to the set $\mathcal{A}_c$ is simply the norm $\|u_i\|$. If point $x \in \mathcal{A}_c$ is the closet point to x_i, it follows that for any vector in the tangent cone, $v_k \in T_{\mathcal{A}_c}(x)$, the distance between x_i and a point $x_\alpha = x + \alpha y \in \mathcal{A}$ is greater than the distance $\|x_i - x\|$. Now, for any vector in the tangent cone, $v_k \in T_{\mathcal{A}_c}(x)$, the vector $x_\alpha = x + \alpha v_k, \alpha > 0$ is also in $\mathcal{A}_c$ for sufficiently small $\alpha > 0$. The nonnegativity of α follows from the definition of the tangent cone as the generalized directional derivative (10.5). Explicitly, the following inequality must hold

$$\|x_i - (x + \alpha v_k)\|^2 \geq \|x_i - x\|^2 = \|u_i\|^2, \tag{10.13}$$

for small $\alpha > 0$, any vector $u_i \perp \mathcal{A}_c$, and any vector $v_k \in T_{\mathcal{A}_c}$. Now, expanding the terms on the left hand side of (10.13),

$$\|u_i\|^2 + \alpha^2\|v_k\| - 2\alpha u_i^T v_k \geq \|u_i\|^2, \tag{10.14}$$

and after neglecting the term in α^2, this yields

$$\alpha u_i^T v_k \leq 0, \tag{10.15}$$

for any normal vector $u_i \perp \mathcal{A}_c(x)$ and any tangent vector $v_k \in T_{\mathcal{A}_c}(x)$, at any given boundary point $x \in \partial\mathcal{A}_c$.

A picture of the the different cases which can be encountered is provided in Figure 10.1.

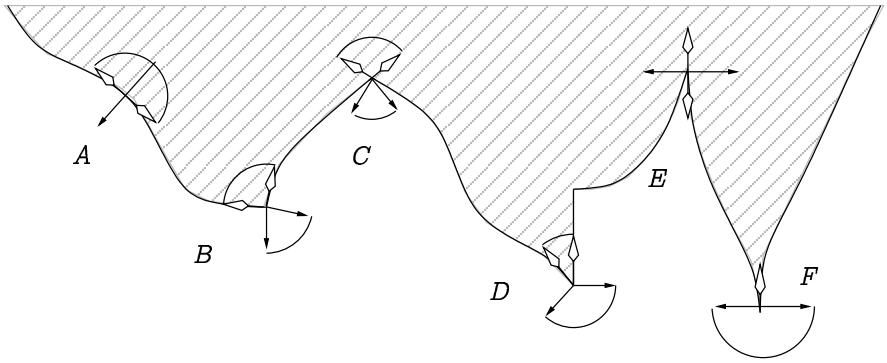

Figure 10.1: The different cases for tangent and normal sets of a nonsmooth curve. Here, perpendicular vectors to $\mathcal{A}_c$ have solid black arrows and tangent vectors have white arrows. This picture is adapted from Fig 1.5 in [66].

Referring to Figure 10.1, the first case with label A is the usual one, namely, there is a unique normal and a unique tangent at q_A so that the normal cone is the single row vector $N_{\mathcal{A}_c}(q_A) = \{\partial c(q_A)/\partial q\}$ and likewise for the tangent cone $T_{\mathcal{A}_c}(q_A)$.

The points labeled q_B, q_C and q_F illustrate different types of kinks in which the normal cone $N_{\mathcal{A}_c}(q)$ is the convex hull between the normals of the curve $c(q)$ evaluated on each side of the discontinuity, and similarly for the tangent cone $T_{\mathcal{A}_c}(q)$ at these points.

The point labeled q_E is an extreme case where the normal and tangential cones consist of a single vector but they are in the opposite direction of what would normally be expected, the normal being the horizontal line and the tangent the vertical one.

As exposed in [66], optimization problems are readily generalized to nonsmooth and nonconvex sets such as the boundary illustrated in Figure 10.1. The reason to adopt these concepts here is to emphasize that the result of [87] are valid for the case of a nonsmooth boundary.

These observations now serve to establish an equality condition in the discrete, nonsmooth principle of least action of Section 10.4.

10.4 Accurate discrete nonsmooth mechanics

In a recent paper by Fetecau et al. [87], the discrete variational principle is extended to include impacts. This formulation is important because it is strictly energy preserving and establishes a solvable numerical procedure to compute impact impulses. In particular, the formulation covers the case where a point particle is restricted to move in a region of space defined by a *nonsmooth* boundary. The case of nonsmooth contact surfaces was already analyzed by Marsden and co-workers in [152] and in [225], where a representation of Coulomb's dry friction model is constructed as well.

A different strategy for handling impacts and contacts, also based on variational principle, is found in Modin and Führer [206] where the time scales are not separated as is done here. Instead, very strong potentials are combined with an adaptive time step strategy to resolve the fast dynamics. As argued previously, this is not well-suited for the interactive physics context though.

A simplified version of the construction found in [87] is presented here with emphasis on the computational method, and with the explicit intention of constructing the approximation presented in Section 10.5. The details of the extension of the variational principle and its discretization to include finitely many impact points—isolated instants in time at when the velocity $\dot{q}$ suffers a jump discontinuity—are well covered in [87] and it would not be particularly useful to reproduce the long, technical proof here.

Consider a sub manifold of the configuration space Q so that $\mathcal{A}_c \subset Q$. This manifold of *allowed configurations* is constructed with a piecewise continuous function $c(q)$ of the configuration variables $q \in Q$. For simplicity and to closely follow close the argument in [87], first consider a single scalar function $c : Q \mapsto \mathbb{R}$. The generalization to a vector valued function $c : Q \mapsto \mathbb{R}^{n_c}$, with piecewise continuous components $c_i(q)$ is straight forward.

The restriction manifold $\mathcal{A}_c$ is the set of points $q \in Q$ such that the function $c(q)$ is non-negative, i.e.,

$$\mathcal{A}_c = \{q \in Q \mid c(q) \geq 0\}, \tag{10.16}$$

and the boundary of this manifold is written

$$\partial\mathcal{A}_c = \{q \in Q \mid c(q) = 0\}. \tag{10.17}$$

Note that this is a manifold with a closed boundary.

Consider a trajectory:

$$(q_0, t_0), (q_1, t_1), \ldots, (q_i, t_i), (\tilde{q}, \tilde{t}), (q_{i+1}, t_{i+1}), \ldots, (q_N, t_N), \tag{10.18}$$

so that $\tilde{q} \in \partial C$, i.e., $\tilde{t}$ is the location of the impact. The extended variational principle applies as before but contributions from the variation of $\tilde{q}$ and $\tilde{t}$ must now be included so that $\tilde{q} + \delta\tilde{q} \in \partial C$. Likewise, the variation of $\tilde{t}$ will lead to conservation of energy at the impact location. Simplify this by setting $t_i = ih$, where h is the fixed time step for all steps other than the one containing the

impact. The impact time $\tilde{t}$ is then parametrized as $\tilde{t} = (i-1)h + \alpha h$, with $\alpha \in (0,1)$. The action then reads

$$\begin{aligned}\mathbb{S}_d(q_0, q_1, \ldots, q_N, \tilde{q}, \alpha) = \sum_{k=0}^{i-2} \mathbb{L}_d(q_k, q_{k+1}, h) + \mathbb{L}_d(q_{i-1}, \tilde{q}, \alpha h) \\ + \mathbb{L}_d(\tilde{q}, q_i, (1-\alpha)h) + \sum_{k=i}^{N-1} \mathbb{L}_d(q_k, q_{k+1}, h). \quad (10.19)\end{aligned}$$

In order to compute the variation of the action, follow the procedure developed in Section 3.14.3 for holonomic constraints where restrictions on the allowed variations δq were considered. Concentrating on the collision event at $k = i$, the discretized action principle requires that variations of the action $\mathbb{S}_d(q_0, q_1, \ldots, q_N, \tilde{q}, \alpha)$ vanishes with respect to the allowed variations of each of the variables of interest, namely $\delta q_{i-1}, \delta\tilde{q}, \delta q_i$ and $\delta\alpha$. A straight forward computation on the definition (10.19) produces the following four terms

$$\begin{aligned}\big(D_2\mathbb{L}_d(q_{i-2}, q_{i-1}, h) + D_1\mathbb{L}_d(q_{i-1}, \tilde{q}, \alpha h)\big)\,\delta q_{i-1} = 0, \\ \big(D_2\mathbb{L}_d(q_{i-1}, \tilde{q}, \alpha h) + D_1\mathbb{L}_d(\tilde{q}, q_i, (1-\alpha)h)\big)\,\delta\tilde{q} \leq 0, \\ \big(D_3\mathbb{L}_d(q_{i-1}, \tilde{q}, \alpha h) - D_3\mathbb{L}_d(\tilde{q}, q_i, (1-\alpha)h)\big)\,h\delta\alpha = 0, \\ \big(D_2\mathbb{L}_d(\tilde{q}, q_i, (1-\alpha)h) + D_1\mathbb{L}_d(q_i, q_{i+1}, h)\big)\,\delta q_i = 0,\end{aligned} \quad (10.20)$$

which must be satisfied simultaneously. The inequality of the second line is the Fourier principle of virtual work stating that when the configuration space has a closed boundary, virtual work must be nonpositive [173]. To extract the exact meaning of these equations, it is necessary to compute the allowed variations in terms of unconstrained variables, as was done in Section 3.14 for the derivation of the constrained, discrete, Euler-Lagrange equations (3.96).

Of course, for all other time steps $k \neq i$, the standard stepping equations (3.22) apply. Now, in the first line of (10.20), the variations δq_i are unrestricted since it is assumed that q_i is away from the constraint surface. For $n = \dim Q$, this produces n nonlinear equations to be solved for $\tilde{q}$ and α. However, it is already known that $\tilde{q} \in \partial\mathcal{A}_c$ and so, when this boundary has codimension 1, there are $n+1$ equations to solve for the n components of vector $\tilde{q}$ and the scalar α, namely, the n equations from the first line of (10.20) and one equation for $c(\tilde{q}) = 0$.

Next, since $\tilde{q} \in \partial\mathcal{A}_c(\tilde{q})$, the allowed variations of $\delta\tilde{q}$ must therefore be in the tangent cone $T_{\mathcal{A}_c}(\tilde{q})$ defined in (10.10). Using the definitions of the normal cone $N_{\mathcal{A}_c}(q)$ from (10.11), the second line of (10.20) will be satisfied when

$$D_2^T\mathbb{L}_d(q_{i-1}, \tilde{q}, \alpha h) + D_1^T\mathbb{L}_d(\tilde{q}, q_i, (1-\alpha)h) \in N_{\mathcal{A}_c}(\tilde{q}). \quad (10.21)$$

For a smooth boundary, this can be rewritten as

$$\begin{aligned}D_2^T\mathbb{L}_d(q_{i-1}, \tilde{q}, \alpha h) + D_1^T\mathbb{L}_d(\tilde{q}, q_i, (1-\alpha)h) + \frac{\partial c(\tilde{q})}{\partial q^T}\nu = 0 \\ 0 \leq C(\tilde{q})(q_i - \tilde{q}) \quad \perp \quad \nu \geq 0,\end{aligned} \quad (10.22)$$

if the restriction manifold is defined as $c(q) \geq 0$. If the boundary is nonsmooth, then, there are multiple tangent Jacobians $C^{(j)}$ and multiple positive Lagrange multipliers $\nu^{(j)}$.

Next, consider the third line in (10.20). There are no restrictions on α other than $\alpha \in [0,1]$ and thus, the following scalar relation holds

$$D_3\mathbb{L}_d(q_{i-1}, \tilde{q}, \alpha h) - D_3\mathbb{L}_d(\tilde{q}, q_i, (1-\alpha)h) = 0. \tag{10.23}$$

This is a nonlinear condition between for q_i, since α and $\tilde{q}$ are known at this point. Recalling the theory of Section 3.10 and the definition of discrete energy of (3.61), the second and third line of (10.20) have a clear meaning, namely, that we are looking for a position vector q_i such that the discrete velocity $h^{-1}(q_i - \tilde{q})$ lies in the normal cone $N_{\mathcal{A}_c}(\tilde{q})$, and so that the discrete energy is preserved.

Finally, the fourth line of (10.20) tells us how to restart the standard stepping at time index $k = i + 1$, using the corrected velocity appearing in the term

$$D_2^T\mathbb{L}_d(\tilde{q}, q_i, (1-\alpha)h). \tag{10.24}$$

Applying this analysis on the constant mass matrix Lagrangian (3.12) with the simple discretization of (3.25), the procedure is as follows. First, find $\tilde{q} \in \partial\mathcal{A}_c$ and $\alpha \in (0,1)$ to solve

$$\begin{aligned} M\left(\frac{\tilde{q} - q_{i-1}}{\alpha h}\right) - M\left(\frac{q_{i-1} - q_{i-2}}{h}\right) + \alpha h \nabla V(q_{i-1}) &= 0 \\ c(\tilde{q}) &= 0. \end{aligned} \tag{10.25}$$

Defining the *incident velocity* as: $v_- = \alpha^{-1}h^{-1}(\tilde{q} - q_{i-1})$, this step computes the exact location of the impact as well as the incident velocity v_-.

The next step is the impulse response which will compute the outward velocity v_+ so that it lies in the normal cone $N_{\mathcal{A}_c}(\tilde{q})$ and so that the energy before and after the impact are identical. Of course, a dissipative energy model could be added here as well. What is thus needed is the solution of

$$\begin{aligned} M\left(\frac{q_i - \tilde{q}}{(1-\alpha)h}\right) - M\left(\frac{\tilde{q} - q_{i-1}}{\alpha h}\right) + (1-\alpha)h\nabla V(\tilde{q}) \quad &\in N_C(\tilde{q}) \\ \frac{1}{2}{v_+}^T M v_+ + V(\tilde{q}) = \frac{1}{2}{v_-}^T M v_- + V(q_{i-1}) + W(v_-), \end{aligned} \tag{10.26}$$

where $W(v_-)$ is a dissipative term which is normally dependent on the incident velocity.

Finally, the stepping continues with

$$M\left(\frac{q_{i+1} - q_i}{h}\right) - M\left(\frac{q_i - \tilde{q}}{(1-\alpha)h}\right) = -h\nabla V(q_i). \tag{10.27}$$

The different processes are illustrated in Figure 10.2.

The theory exposed in [87] justifying this model is impeccable. However, locating all impacts one at a time in this fashion is far from desirable. In addition, the occurrence of multiple simultaneous impacts leaves much to be desired. Also, exactly locating $\tilde{q}$ is not feasible except for very simple cases such as collisions with a plane for instance. Simplifications of this model are now investigated.

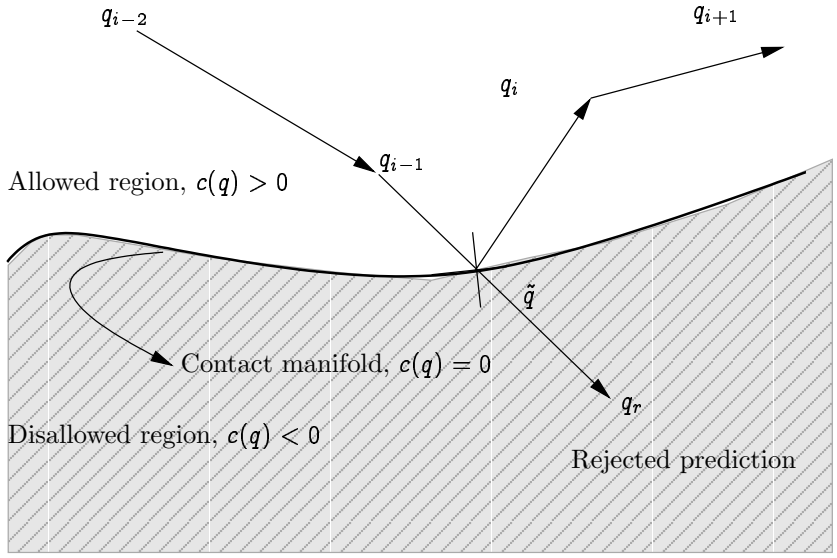

Figure 10.2: Schematics of the exact impulse model described in this section.

10.5 A two step approximate impulse model

An approximation strategy is now constructed to avoid locating $\tilde{q}$ precisely and skip many of the steps to compute the incident and outbound velocities, v_- and v_+. The first model considered is a *post facto* impulse resolution so that at step i, q_i lies inside the restricted region. The constraint stabilization mechanism could be used by itself to operate for a few steps to bring q_{i+n} near the boundary $\partial\mathcal{A}_c$. However, this leads to nearly zero outbound velocity, $v_+ = 0$, independently of the incident velocity v_- which is nice for stability but not good for accurate modeling of elastic or nearly elastic impacts.

If the violation at q_i is not too big, a good approximation of the incident velocity is:

$$v_- \approx \frac{q_i - q_{i-1}}{h}. \tag{10.28}$$

To estimate the outbound velocity v_+, first impose a discrete impulsive Newton impact law [56] on the system. In one spatial dimension, this law is usually stated as

$$v_+ = -\psi v_-, \tag{10.29}$$

where $\psi \in [0, 1]$ is called the *restitution coefficient*. A perfectly elastic collision corresponds to $\psi = 1$ in which case the outbound velocity is just a reflection of the incoming one. Now, given that impulses occur at *fixed location* q, the only change in energy is a *kinetic energy* change. For the Newton-Poisson impact law in the context of a one-dimensional impact of a point particle of mass m, the change in energy is have:

$$E_+ - E_- = T(q, v_+) - T(q, v_-) = (\psi^2 - 1)T(q, v_-) \leq 0, \tag{10.30}$$

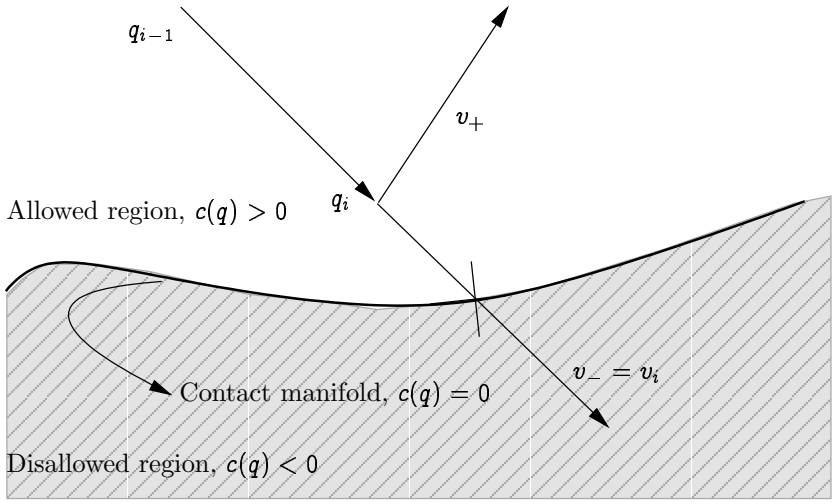

Figure 10.3: Schematics of the approximate impulse model described in this section. The case illustrated here is the *preemptive* impact resolution where the incident velocity v_- is changed before impact has occurred.

for an impact at location q and where $T(v)$ is the kinetic energy, defined earlier in (3.11).

This is illustrated schematically in Figure 10.3 for the preemptive impact detection case and in Figure 10.4 for post facto detection.

To generalize this to higher dimensions, an estimate of the part of the incident velocity that is normal to—or approximately so—the contact manifold is needed. This is precisely $C_k v_-$ where C_k is the $n_c \times n$ Jacobian matrix of the n_c contact surface evaluated at the discrete time k. A strict equivalent of the Newton impact law would state $Cv_+ = -\Psi C v_-$, where Ψ is a diagonal matrix of size $n_c \times n_c$, $\Psi = \mathsf{diag}(\psi_1, \psi_2, \ldots, \psi_{n_c})$, $\psi_i \in [0, 1]$ being the Newton restitution parameter of the ith contact manifold. However considering that impulsive contact forces must be non-adhesive leads directly to a complementarity formulation (an analysis of this is found in [224]). Thus, the second stage of the exact procedure presented in the last section, assuming also the presence of other equality constraints with agglomerated Jacobian G, and adding regularization parameters with diagonal, non-negative matrices Σ and Ξ, for the equality and contact constraints, respectively, the discrete Euler-Lagrange equations (10.26) are approximated with the following LCP

$$\begin{aligned} Mv_+ - G^T{}_k\lambda - C^T{}_k\nu &= Mv_- \\ G_k v_+ + \Sigma\lambda &= G_k v_- \\ C_k v_+ + \Psi C v_- + \Xi\nu &= w \\ 0 \le \nu \quad \perp \quad w &\ge 0, \end{aligned} \tag{10.31}$$

where the potential forces $h^2\nabla V$ were neglected. Indeed, impulsive forces suffi-

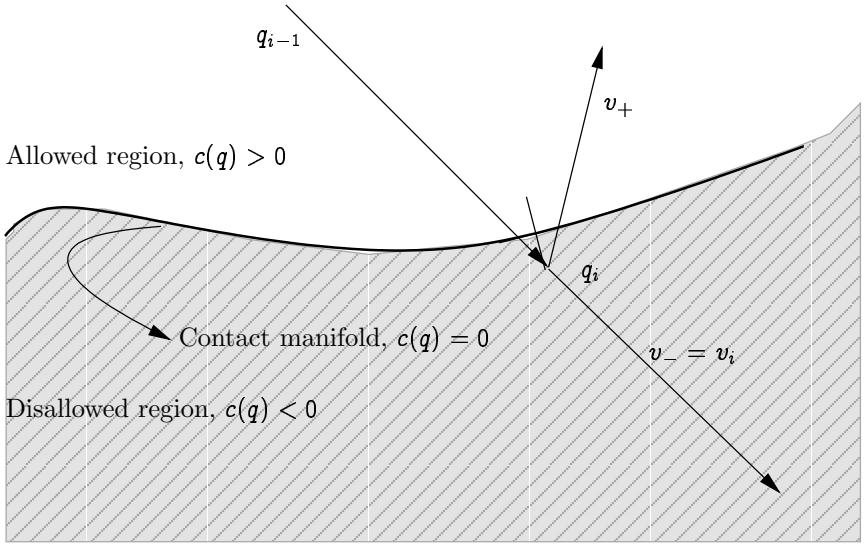

Figure 10.4: Schematics of the approximate impulse model described in this section. The case illustrated here is the *post facto* impact resolution where the incident velocity v_- is changed after impact has occurred. Constraint stabilization is then used to stabilize back to the contact surface.

cient to revert the incident velocity must be very large in comparison to other forces in the system. Note that this update only affects the velocity variables so the energy change from this update is restricted to kinetic energy change. This is now evaluated:

$$\begin{aligned} T_+ = v_+^T M v_+ &= v_+^T M v_-^T + v_+^T G^T \lambda + v_+^T C^T \nu \\ &= v_-^T M v_- + \lambda^T G v_- + \nu^T C v_- + \nu^T C v_+ + v_+^T G^T \lambda \\ &= T_- - \lambda^T \Sigma \lambda + \nu^T (C v_+ + \Psi C v_-) + \nu^T (I - \Psi) C v_- \\ &= T_- - \lambda^T \Sigma \lambda + \nu^T w - \nu^T \Xi \nu + \nu^T (I - \Psi) C v_- \\ &\leq T_-. \end{aligned} \tag{10.32}$$

The last inequality is derived from the following facts

$$\begin{aligned} -\lambda^T \Sigma \lambda \leq 0 \quad & \text{since } \Sigma \text{ is symmetric and positive definite,} \\ \nu^T w = 0 \quad & \text{from the complementarity condition in (10.32),} \\ -\nu^T \Xi \nu \leq 0 \quad & \text{since } \Xi \text{ is symmetric and positive definite,} \\ C v_- \leq 0 \quad & \text{by assumption on the contact conditions,} \\ (I - \Psi) C v_- \leq 0 \quad & \text{from the definition of } \Psi \text{ since } 0 \leq \psi_j \leq 1, j = 1, 2, \ldots, n_c, \\ \nu \geq 0 \quad & \text{in the solution of LCP (10.32).} \end{aligned} \tag{10.33}$$

Therefore, this impulsive stage can only decrease the kinetic energy. Since the positions are not changed in this stage, the total energy can only decrease.

Once the impulsive stage is computed and velocities updated, the integration proceeds using the computed velocities, v_+, and the previous positions.

Observe that impact conditions are caught *a posteriori*, i.e., after detecting a non-zero penetration for one or several of the nonpenetration conditions. Stabilization back to the contact surface is handled using the constraint stabilization mechanism described in Chapter 4. This strategy is not very good for cases involving a combination of high incident velocities and deep penetration. For some cases, one can implement a *preemptive* impact strategy. This is left for future work.

This model still lacks friction forces in the direction tangential to the contact plane. The friction model derived in Section 10.11.4 can be added to the present formulation without changing the dissipative properties.

Other work related to impacts and friction is discussed in Section 10.12.

10.6 Numerical comparison

The behaviors of different impact models described in the previous sections are now illustrated. First in Figure 10.5, a short simulation using the exact model of Section 10.4 is portrayed. For a one-dimensional problem, this method is fast and very simple to implement. Modifying the energy conservation stage (10.26) to dissipate a fraction of the incident kinetic energy is easily done, requiring that the outgoing kinetic energy be a fraction of the incident one. This amounts to Newton's law of impacts. This is parametrized with a restitution coefficient, $\psi \in [0, 1]$. Data on the graphs has been sub-sampled (due to a technical problem) which is why the lines do not touch the plane $x = 0$ exactly.

In Figure 10.6, the long term behavior for unit restitution as computed by the exact, preemptive and post facto methods is illustrated. The approximate methods loose energy linearly in the case of post facto detection and exponentially for the preemptive method. For the post facto case, the decay is small and tolerable since in practice, there is no perfectly elastic collision. Preliminary experiments show that this decay can be made very small if one makes a better estimate of the impact location parameter α.

In Figure 10.7, the regularized stepper of Section 4.2 is used without any constraint stabilization. This leads to unpredictable impact restitution, depending on exactly when and how deep a penetrating configuration is caught. This is clearly not a usable scheme.

In Figure 10.8, different values of the normalized constraint stabilization parameter $d = \tau/h$ of Section 4.4 are used to stabilize the regularized contact constraint. This is strongly dissipative from $d \geq 2$ and can in fact be made to absorb all the impact energy for $d = 100$, if desired. Such an impact processing scheme is stable for $d = 2$ but the observed restitution coefficient is random, and very close to 0.

Finally, in Figure 10.9, one step before and a few step after impact are illustrated with each of the methods at zero restitution $\phi = 0$. For the post facto

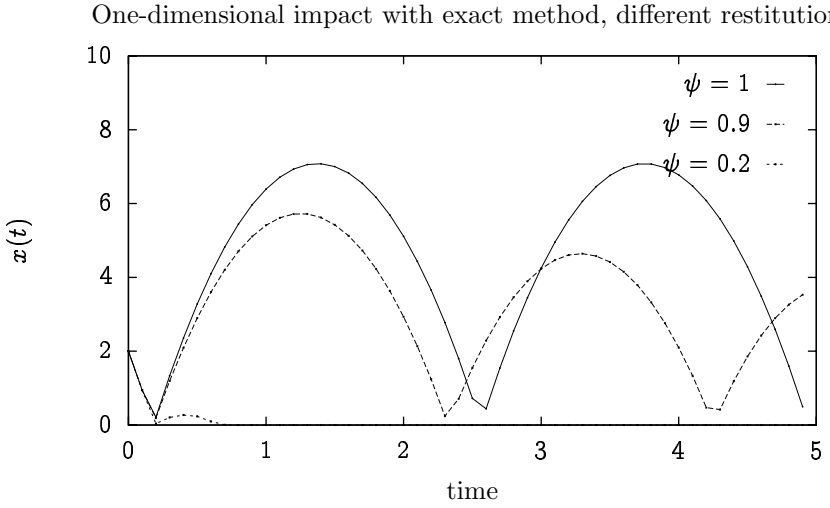

Figure 10.5: One-dimensional impact simulated with the exact method of Section 10.4 but with different restitution coefficients. Initial conditions are $v(0) = -10, x(0) = 2$, and the time step is $h = 1/60$.

method, the point penetrates deeply and is then stabilized, with some comparatively small overshoot. For the preemptive case, the trajectory is modified to avoid penetration and is made to graze the contact surface instead. The exact method locates the time step when penetration would have occurred and puts the particle on the surface $x = 0$ and the end of that step, with zero energy.

The post facto strategy offers a good compromise. For complicated cases with very fast moving objects, a combined preemptive and post-facto scheme should be considered, and this is left for future work.

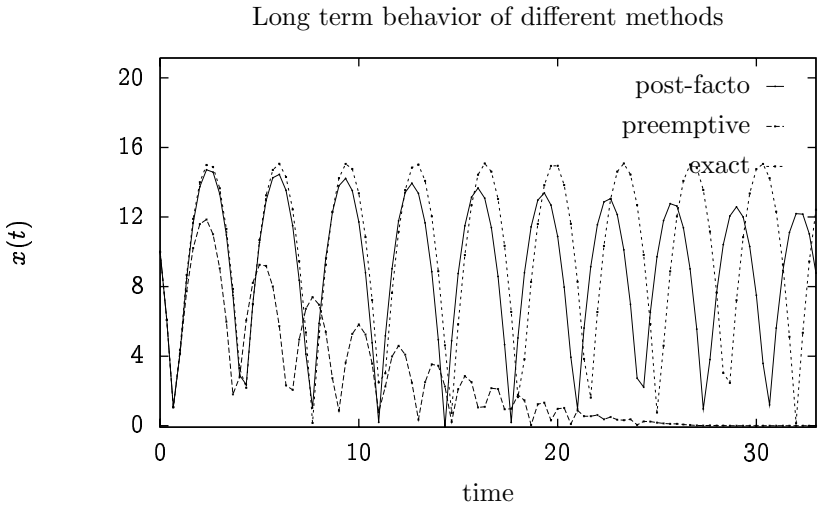

Figure 10.6: Long time integration of the exact, preemptive and post facto methods using unit restitution, $\psi = 1$, which leads to elastic impacts. The post facto method still looses energy slowly but does a better job than the preemptive method.

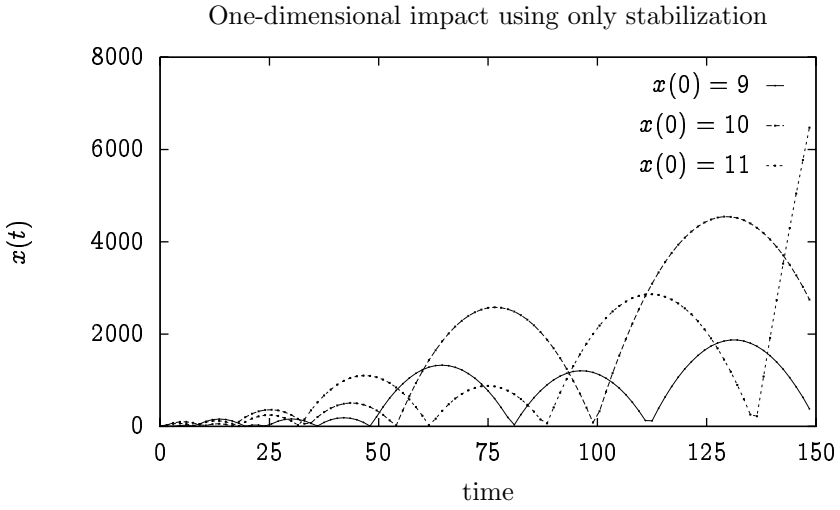

Figure 10.7: Long time simulation using only the regularized stepper. Even with zero restitution, this produces erratic behavior which can be unstable. Note that changing the initial position causes wildly different behavior. The initial velocity is fixed at $v(0) = -10$.

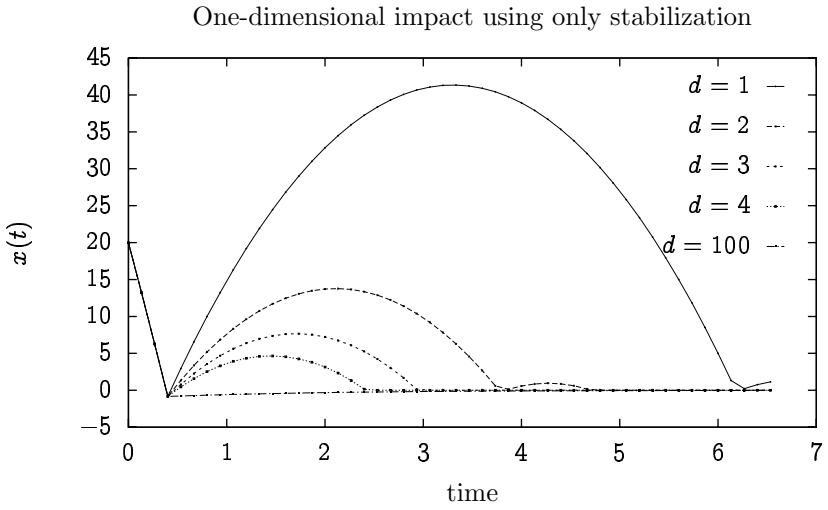

Figure 10.8: Integrating over impacts using only the SPOOK stepper of Section 4.4, the stabilization parameter $d = \tau/h$ is varied. Note that for $d > 2$, the collisions are completely inelastic.

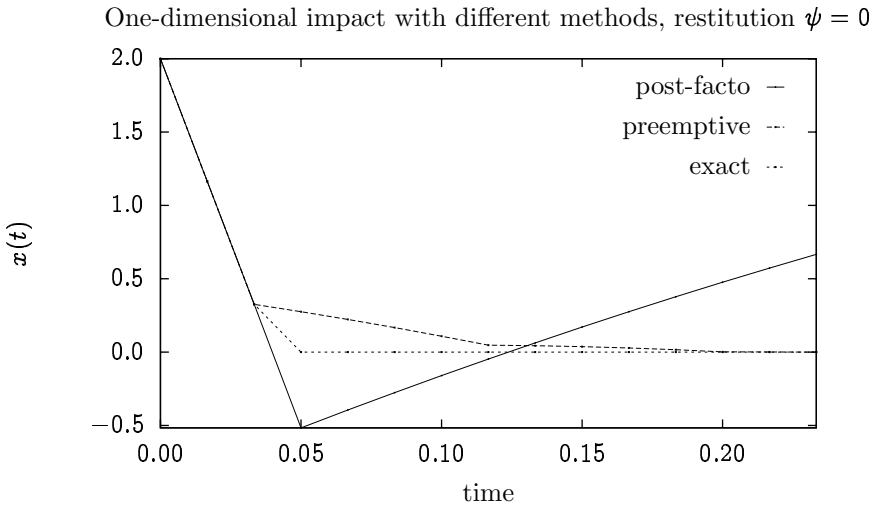

Figure 10.9: A few integration steps before and after an impact with zero restitution, illustrating how the different methods behave. As expected, the exact method finds the contact manifolds and stays on it. The other methods quickly converge to it.

10.7 Nonsmooth forces and nonideal constraints

Impacts arising from non-penetration conditions are but one instance of a nonsmooth phenomenon in physics and they are distinguished for being based on position only. In this respect, non-penetration conditions are very much like holonomic constraints when they are active. By contrast, friction forces and other nonsmooth phenomena are expressed in terms of velocities and this fact suggests that the correct strategy is to construct appropriate Rayleigh dissipation functions.

Noting that for a given Rayleigh function $\mathfrak{R}(q, \dot{q}, t)$, the effect on the dynamics of a system with Lagrangian function $\mathcal{L}(q, \dot{q}, t)$ is via the *integral* term appearing in d'Alembert's principle, namely:

$$-\int_0^h \mathrm{d}s \frac{\partial \mathfrak{R}}{\partial \dot{q}^T} \delta q = f_d^{(-)}(q_0, q_1, h)\delta q_0 + f_d^{(+)}(q_0, q_1, h)\delta q_1 . \tag{10.34}$$

This integral term is well behaved even for nonsmooth functions $\mathfrak{R}$, as long as the velocity partial derivatives are integrable functions.

Also, when applying d'Alembert's principle to nonsmooth systems which have hard boundaries, so the virtual displacements δq might be restricted, the equality sign of (3.77) is replaced by Fourier's inequality [173], namely

$$\delta \int_{t_0}^{t_1} \mathrm{d}s \mathcal{L}(q(s), \dot{q}(s)) + \int_{t_0}^{t_1} \mathrm{d}s f^T \delta q \leq 0. \tag{10.35}$$

In the following sections, several nonsmooth Rayleigh functions are constructed to model different forms of dry friction.

10.8 Velocity limits

First consider a dissipation force which restricts the *magnitude* of the velocity vector $\dot{q}$ so that $\|\dot{q}\| \leq \rho, \rho > 0$. Clearly, it is possible force $\|\dot{q}\| = 0$ using the dissipation function $\mathfrak{R}(q, \dot{q}, \alpha, \dot{\alpha}) = -\tau \left[(\gamma/2)\dot{\alpha}^T\dot{\alpha} + \dot{\alpha}^T\dot{q}\right]$, where the ghost variables α have the same dimensionality as q. However, limiting the magnitude of $\dot{q}$ away from $\dot{q} = 0$ requires only a one-dimensional Lagrange multiplier.

What is needed here is a dissipation force which is switched on whenever $\|\dot{q}\|$ exceeds the limit ρ. The regularization parameter $\delta \geq 0$ will control how quickly the velocity is brought back within bounds. To do this, introduce the nonsmooth function:

$$\theta_+ = \begin{cases} 0 & \text{when } x < 0, \\ x & \text{otherwise .} \end{cases} \tag{10.36}$$

This function extracts the positive part of any scalar $x \in \mathbb{R}$. The derivative of θ_+ is well defined away from the origin but nonsmooth analysis [66] reveals that

the proper definition is

$$\frac{\mathrm{d}\theta_+(x)}{\mathrm{d}x} = \theta'_+(x) = \begin{cases} 0 & \text{when } x < 0, \\ 1 & \text{when } x > 0, \\ [0,1] & \text{when } x = 0. \end{cases} \tag{10.37}$$

In other words, $\theta'_+(0)$ is undetermined as it can take any value in the interval $[0,1]$. This appears problematic at first but it is shown below that the regularization procedure picks a unique value for $\theta'_+(0)$. Note carefully the use of $\theta'_+(x)$ to denote the total x derivative in order to avoid confusion with the time derivative which is still written as $\dot{\phi} = \mathrm{d}\phi/\,\mathrm{d}t$.

The constraint being imposed now is simply $\theta_+(\|\dot{q}\| - \rho) = 0$ and therefore, the regularized Rayleigh function needed here is

$$\mathfrak{R}_\zeta = -\frac{\delta\dot{\zeta}^2}{2} + \dot{\zeta}\theta_+(\|\dot{q}\| - \rho), \tag{10.38}$$

where $\zeta \in \mathbb{R}$ is a scalar, and this generates forces on the q variables according to (10.35)

$$\begin{aligned} -\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}^T}\right) + \frac{\partial \mathcal{L}}{\partial q^T} - \frac{\partial \mathfrak{R}_\zeta}{\partial \dot{q}^T} &= 0, \\ f_q = -\frac{\partial \mathfrak{R}_\zeta}{\partial \dot{q}^T} &= -\dot{\zeta}\theta'_+(\|\dot{q}\| - \rho)\frac{1}{\|\dot{q}\|}\dot{q}, \end{aligned} \tag{10.39}$$

and for the ghost $\dot{\zeta}$,

$$\begin{aligned} -\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \mathcal{L}}{\partial \dot{\zeta}^T}\right) + \frac{\partial \mathcal{L}}{\partial \zeta^T} - \frac{\partial \mathfrak{R}_\zeta}{\partial \dot{\zeta}^T} = f_\zeta &= 0 \\ f_\zeta = -\frac{\partial \mathfrak{R}_\zeta}{\partial \dot{\zeta}} &= \delta\dot{\zeta} - \theta_+(\|\dot{q}\|^2 - \rho) = 0. \end{aligned} \tag{10.40}$$

Note here that Fourier's inequality (10.35) is not used since there is no restriction on ζ or $\dot{\zeta}$. Combining (10.39) and (10.40) leads to a nonsmooth nonlinear system of equations to be solved when discretized though, in fact, it is simpler to use an equivalent NCP formulation, as already suggested by Fourier's inequality. Indeed, the ghost equation (10.40) is identical to the complementarity condition

$$0 \le \delta\dot{\zeta} - \|\dot{q}\| + \rho \perp \dot{\zeta} \ge 0. \tag{10.41}$$

The inequality sign to the left of (10.41) accounts for the fact that $\theta_+(x) \ge 0$. Now, the Rayleigh function in (10.38) produces change in energy as shown in (3.88) and given the choice of signs in the definition of $\mathfrak{R}_\zeta$ in (10.38), it follows that restricting $\dot{\zeta} \ge 0$ leads to dissipation, which is the current intention. Next, whenever $\|\dot{q}\| < \rho$, the value of $\dot{\zeta}$ vanishes from the f_ζ equation in (10.39). Therefore, as long as $\dot{\zeta} = 0$, it follows that $\delta\dot{\zeta} - \|\dot{q}\| + \rho > 0$.

To formulate the equations of motion (10.39) as a *bona fide* NCP, the derivative term θ'_+ in formulation of f_q must be considered carefully. Note first that if $\dot{\zeta} > 0$, the f_ζ equation in (10.39) yields

$$\dot{\zeta} = \frac{1}{\delta}\left(\|\dot{q}\| - \rho\right) = \frac{1}{\delta}\theta_+(\left(\|\dot{q}\| - \rho\right)), \tag{10.42}$$

which in turn implies that the f_q force is

$$f_q = -\frac{1}{\delta\|\dot{q}\|}\theta'_+(x)\theta_+(x)\dot{q}, \tag{10.43}$$

where $x = \|\dot{q}\| - \rho$. But since $\theta'_+(0)\theta_+(0) = 0$, at which point $\dot{\zeta} = 0$, it follows that $\dot{\zeta}\theta'_+(x) = \dot{\zeta}$ everywhere and the force term reduces to

$$f_q = -\frac{1}{\|\dot{q}\|}\dot{\zeta}\dot{q}, \tag{10.44}$$

whenever $\delta > 0$. Note that this force opposes the direction of motion and is therefore dissipative.

For discretization, several choices are available as was described in Section 3.12. Choosing the fully implicit discretization for the forces $f_d^{(\pm)}(q_0, q_1, h)$, leads to the stepping equations

$$\begin{aligned} D_1^T\mathbb{L}_d(q_k, q_{k+1}, h) + D_2^T\mathbb{L}_d(q_{k-1}, q_k, h) - W_{k+1}^T\zeta = 0 \\ 0 < -W_{k+1}\left(q_{k+1} - q_k\right) + \rho + \delta\zeta \perp \zeta \geq 0, \end{aligned} \tag{10.45}$$

with the definition:

$$W_{k+1} = \frac{1}{\|q_{k+1} - q_k\|}\left(q_{k+1} - q_k\right)^T = \frac{1}{\|v_{k+1}\|}v_{k+1}. \tag{10.46}$$

For the special case where the discrete Lagrangian $\mathbb{L}_d(q_0, q_1, h)$ has the form,

$$\mathbb{L}_d(q_0, q_1, h) = \frac{1}{2h}\left(q_1 - q_0\right)^T M\left(q_1 - q_0\right) - hV(q_0), \tag{10.47}$$

and approximating $W_{k'} \approx W_{k+1}$, the stepping equations in velocity form are the solvable mixed linear complementarity problem (MLCP):

$$\begin{bmatrix} M & W_{k'}^T \\ -W_{k'} & \frac{\delta}{h} \end{bmatrix}\begin{bmatrix} v_{k+1} \\ \zeta \end{bmatrix} + \begin{bmatrix} -Mv_k + hV_k \\ \rho \end{bmatrix} = \begin{bmatrix} 0 \\ \chi \end{bmatrix} \tag{10.48}$$
$$0 \leq \zeta \quad \perp \quad \chi \geq 0,$$

where the slack variable χ was introduced. In practice, it is sufficient to take $W_{k'} = W_k$. Obviously, it would have been possible to define the Rayleigh function using only parts of the velocity vector $\dot{q}$, using, e.g., $v = D(q)\dot{q}$ where $D(q)$ is a projection operator. This would only change the definition of the matrix W in (10.46) but not the rest of the analysis and in particular, not the form of the NCP for the stepping equations.

In addition, the velocity variables $\dot{q}$ can be replaced any ghost velocity $\dot{\lambda}$ and thus impose a restriction on the magnitude of Lagrange multipliers corresponding to nonholonomic constraints. In that latter case however, the sign of the Rayleigh function must be reversed to preserve the dissipative property. This is shown below in Section 10.9. Note finally that the $\theta(x)$ function can in fact be removed by using the restriction $\dot{\zeta} \geq 0$ with the Rayleigh function

$$\tilde{\mathfrak{R}}_\zeta = \frac{\delta}{2}\dot{\zeta}^2 + \dot{\zeta}(\rho - \|\dot{q}\|), \tag{10.49}$$

and using the Fourier inequality (10.35) leads directly to

$$\begin{aligned} f_q &= -\dot{\zeta}\frac{1}{\|\dot{q}\|}\dot{q} \\ 0 \leq -f_\zeta &= \delta\dot{\zeta} + \rho - \|\dot{q}\| \qquad \perp \dot{\zeta} \geq 0. \end{aligned} \tag{10.50}$$

Both the unrestricted formulation using the filter function $\theta_+(x)$ and the restricted formulation with $\dot{\zeta} \geq 0$ have their advantages.

10.9 Dissipative properties of velocity limits

Velocity limits described in the previous section are now shown to be strictly dissipative. It was shown in Section 3.12, particularly in (3.88), that for any Lagrangian system with time independent Lagrangian function $\mathcal{L}(z, \dot{z})$ and any Rayleigh function $\mathfrak{R}$, the energy dissipation rate is

$$\frac{\mathrm{d}E}{\mathrm{d}t} = -\dot{z}^T \frac{\partial \mathfrak{R}}{\partial \dot{z}^T}. \tag{3.88$'$}$$

For the case at hand, the generalized coordinates z consist of $z = (q, \zeta)$ and thus the dissipation rate is

$$\begin{aligned} \frac{\mathrm{d}E}{\mathrm{d}t} &= -\dot{q}^T \frac{\partial \mathfrak{R}_\zeta}{\partial \dot{q}^T} - \dot{\zeta}\frac{\partial \mathfrak{R}_\zeta}{\partial \dot{\zeta}} \\ &= -\dot{\zeta}\theta'_+(\|\dot{q}\| - \alpha)\|\dot{q}\| + \dot{\zeta} f_\zeta \\ &\leq -\dot{\zeta}\|\dot{q}\| + 0 \leq 0, \end{aligned} \tag{10.51}$$

since $f_\zeta = 0$ and $\dot{\zeta} \geq 0$ by construction, and $\theta_+(x) \geq 0$.

In the discrete case, given initial conditions v_0 and q_0, write v_1 and q_1 for the velocity and position that would be found without the additional term, and set the actual computed updates as

$$\begin{aligned} v &= v_1 - M^{-1}W_{k'}^T \zeta \\ q &= q_1 - hM^{-1}W_{k'}^T \zeta. \end{aligned} \tag{10.52}$$

Then, the difference between the energy that is found at v, q and that which would be found at v_1, q_1 if the velocity limit was not active, given the discrete

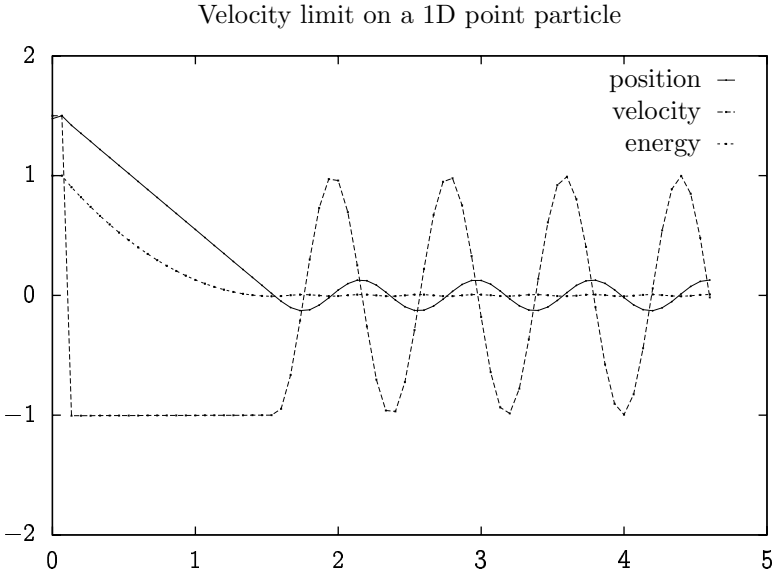

Figure 10.10: Simulation of the dynamics of a one-dimensional particle subject to a speed constraint and a harmonic potential.

Lagrangian of (3.25) and the corresponding discrete energy (3.61), is computed to be

$$\begin{aligned}\Delta E &= v^T M v - v_1^T M v_1^T - h f_1^T W_{k'}^T \zeta \\ &= -\frac{\delta}{h}\zeta^T\zeta - \zeta^T\left(W_{k'}M^{-1}W_{k'}^T\zeta + \alpha + hW_{k'}M^{-1}f_0\right) \\ &= -\frac{\delta}{h}\zeta^T\zeta - \zeta^T\chi \\ &\leq 0,\end{aligned} \tag{10.53}$$

where $f_0 = -\partial V(q_0)/\partial q^T$. Since $\zeta \geq 0$, and since the term in parenthesis on the second line is recognized as the reduced equation for ζ after eliminating the velocity equation in (10.48), the expression is strictly negative when the constraint is active and when $\delta > 0$.

Thus, this formulation is strictly dissipative and therefore models some form of friction phenomena.

An illustration of the effect of this type of velocity limit constraint is provided in Figure 10.10. Here, a point particle with unit mass is hooked to a simple harmonic oscillator with unit frequency at position $x = 3/2$ and $v = 3/2$, moving away from the origin. The velocity limit imposes a speed of $\|\dot{q}\| \leq 1$. The velocity is too high at $t = 0$ and the constraint activates and reverses it. The particle reaches back toward the origin with velocity $\dot{q} = -1$ until it is within a region where $\|\dot{q}\| \leq 1$ under the action of the harmonic oscillator.

10.10 Range limits on pseudo-particle coordinates

An analysis of range limits on position variables was presented in Sections 10.2, 10.4, and 10.5 which included details of the kinetic energy before and after impacts. Such limits arise naturally from non-penetration constraints for instance. However, it is also useful to model range limits on the coordinates of the pseudo-particles themselves. The need for this sort of model is clear when considering the idealization of a massless rope with finite break tension for instance. If such a rope attaches two rigid bodies for instance, it will keep them at a distance $d \leq q_0$ as long as the tension λ is less than the break threshold λ_0 but will cease to generate any force if $\lambda(t) > \lambda_0$ at any point in time as the rope is then broken.

As an alternative example, consider a hinge joint between two rigid bodies made out of some metal assembly. Metals typically behave elastically under stress up to some maximum—the yield point. Thus, hard bounds on constraints are desirable and these can be enforced with a technique identical to standard impacts, though with even simpler Jacobians since in general, what is imposed is the simple constraint $\|\lambda\| < \lambda_{\max}$ or just $\lambda_{\min_i} \leq \lambda_i \leq \lambda_{\max_i}$, there λ is the ghost variable of a given constraint.

These additional constraints are dissipative in general using the two stage impulse model described in Section 10.5. Adding them to the stepping model is straight forward so it is safe to omit the full details.

10.11 Dry friction and the Coulomb model

Rayleigh dissipation functions of the form $\mathfrak{R} = (1/2)\dot{q}^T D\dot{q}$ were introduced in Section 3.12 as the archetype *polygenic* force. This Rayleigh function generates a force of the form $f_{\mathfrak{R}} = -D\dot{q}$ which is linear in the velocities $\dot{q}$ and therefore, generates no force at all when $\dot{q} = 0$. This type of friction proportional to velocity is an example of a *viscous drag* which is typical of the dissipation forces encountered when moving a solid body through a fluid such as air or water where this is known as Stoke's law [175]. More accurate viscous drag forces for high velocity motion in a fluid involves higher powers of the velocity but the fact remains that viscous drag forces vanish for zero velocity.

By contrast, dry friction forces are nearly independent of velocity and in particular, they do not vanish at zero contact velocity. The usefulness of this in daily life is tremendous since otherwise, it would be impossible to walk, sit still on a chair, hold a glass of wine, or keep clothing on.

10.11.1 Phenomenology of dry friction

Dry friction arises when two solids come in contact and no fluid is present at the contact area. The exact details of the contact forces can be very complicated especially if one accounts for elastic and plastic deformations and in fact, a complete explanation of dry friction starting from molecular forces is still missing though much progress has been made in that direction recently [231]. Not too

surprisingly, the explanation of dry friction comes from quantum mechanics as all other classical explanations, based on surface roughness for instance, have proved to be wrong. Since the details of such models are not particularly relevant to simulation of the macroscopic phenomena, they are not pursued further. Good macroscopic constitutive models describing the tangential contact forces of contacting bodies do exist as described now.

The most striking characteristic of dry friction is that it exhibits two distinct modes. The first is know alternately as *stiction*, *stick mode*, *rolling friction*, or *static friction*, and refers to the situation where the tangential contact forces maintain zero relative contact velocity between two contacting bodies. Different names are used in different branches of the literature and can be conflicting, though static friction is universally understood. Rolling friction, in particular, means *both* static friction—which allows rolling as opposed to sliding [266]—and the additional friction forces observed when a circular object rolls steadily on a contacting plane. Since stiction forces maintain zero relative velocity, they produce no work. The second is known alternately as *sliding*, *kinetic friction*, or *dynamic friction*, and refers to the situation where the relative contact velocity is non-zero and where the tangential contact force has constant magnitude, directly opposing the sliding velocity. The transition between the two modes is known as the *stick slip* transition.

The principles of dry friction were suggested by Leonardo da Vinci and presented first by Guillaume Amontons in 1699 and verified by by Charles-Augustin Coulomb in 1781. The following conclusions were reached regarding the tangential friction forces at the interface between two contacting bodies:

Area independence: the net friction force is nearly independent of the contacting area;

Kinetic friction force depends on magnitude of normal: the magnitude of the kinetic friction force is directly proportional to the normal force between the contacting bodies;

Stick-slip transition depends on magnitude of normal: the magnitude of the tangential force when the contacting bodies start to move is directly proportional to the normal force;

Kinetic friction depends on sliding speed: the magnitude of the friction force decreases slightly when the relative sliding speed at the contact increases;

Kinetic friction is maximally dissipative: the kinetic friction force acts in a direction *directly opposing* the sliding velocity in the case of isotropic frictional contacts. In the anisotropic case, kinetic friction acts in a direction causing maximum dissipation.

Modern investigations of dry friction have provided partial explanations of these observation [231]. It is reported that Coulomb did in fact suspect the velocity dependence of the kinetic friction coefficient but his data was not accurate enough to make the claim at the time.

To formulate this more accurately, consider a mechanical system with generalized coordinates $q : \mathbb{R} \mapsto Q$, where Q is an n-dimensional manifold. Assume the existence of a contact point at time t_0 so the contact constraint is active: $c(q(t_0)) = 0$. Now, assume that $D^{(c)}(q) : \mathbb{R}^n \mapsto \mathbb{R}^2$ is a projection operator so that $v^{(c)} = D^{(c)}\dot{q}$ is the relative *sliding velocity* at the contact point $q(t_0)$. Obviously, the rows of matrix $D^{(c)}$ are orthogonal to the contact normal: $C = \partial c(q(t_0)/\partial q$. Let $f_n^{(c)}$ be the magnitude of the normal force at the contact point and let $f_t^{(c)}$ be the tangential force acting in the contact plane. With this notation, dry friction is formulated as follows

$$\begin{aligned} v^{(c)} = D^{(c)}\dot{q} = 0 \iff \|f_t^{(c)}\| \leq \mu_s^{(c)}\|f_n^{c)}\|, \text{ static friction} \\ \frac{\dot{q}^T {D^{(c)}}^T f_t^{(c)}}{\|D^{(c)}\dot{q}\|\|f_t^{(c)}\|} = -1 \iff \|f_t^{(c)}\| = \mu_k^{(c)}\|f_n^{(c)}\|, \text{ kinetic friction,} \end{aligned} \tag{10.54}$$

where $\mu_s^{(c)} > 0$ is the *coefficient of static friction*, and $\mu_k^{(c)} > 0$ is the *coefficient of kinetic friction*. These relations are true for each contact point. In general $\mu_s^{(c)} \geq \mu_k^{(c)}$, and the discrepancy is of the order of 10-20% [232].

The formulation (10.54) is obviously a complementarity condition as either one of the two sets of constraint is active at any given time. The case of $D^{(c)}\dot{q} = 0$ can be handled using the methods of Section 3.14.5 and Section 4.3, but the second condition and the transition between the two requires new analysis.

A reasonable approximation of the dependence of the friction coefficient on the relative contact velocity $v \in \mathbb{R}^2$, is given by either of the following two models [103]:

$$\begin{aligned} &\text{analytic model: } \mu_a(v) = \mu_s \left(1 - \alpha \frac{\|v\|}{\sqrt{\eta^2 + \|v\|^2}}\right), \\ &\text{discrete model: } \mu_d(v) = \begin{cases} \mu_s \left(1 - \alpha \frac{\|v\|}{\eta}\right) & \text{when } \|v\| \leq \eta \ , \\ \mu_s(1-\alpha) & \text{when } \|v\| > \eta, \end{cases} \end{aligned} \tag{10.55}$$

where, in both cases, the kinetic and static friction coefficients are related by: $\mu_k = \mu_s(1-\alpha), \alpha \in (0,1)$, and the parameter $\eta > 0$ is a *creep threshold*. Of course, all these parameters are dependent on the specific contact properties. Note that the transition from μ_s to μ_k is very rapid and in fact, at the time scale used in interactive simulations, it might be too quick to catch and thus, one can generally use μ_s if the contact is in stiction, and μ_k otherwise. Nevertheless, the conclusion to draw is that the friction coefficient is a velocity dependent parameter.

The complementarity conditions described in (10.54) and (10.55) define the *friction cone* in which the contact forces are constrained to be. To see this, think of the contact plane with orthogonal directions t_1 and t_2 as shown in Figure 10.11 below. Normal to that surface is the normal force of contact, f_{nor}, pointing upward since the contacts are assumed to be non-adhesive. The magnitude of the normal force, $\|f_{\text{nor}}\|$ provides the budget for the magnitude of the tangential

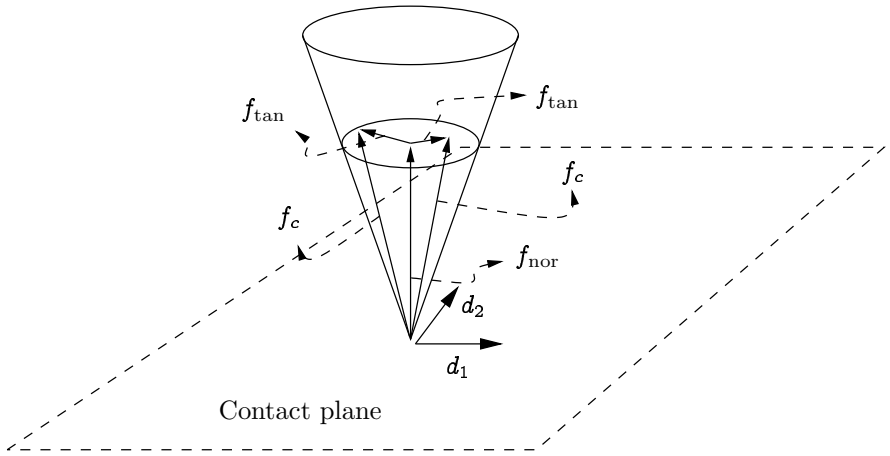

Figure 10.11: The Coulomb friction cone.

force according to $\|f_{\tan}\| \leq \mu_s \|f_{\text{nor}}\|$, which means that tangential forces are within a disk of radius $\mu_s \|f_{\text{nor}}\|$ for the isotropic case. The net contact force is the vector sum $f_{\text{nor}} + f_{\tan}$ which must lie within the cone. Two cases are illustrated in Figure 10.11, namely, one in which the tangential force $f_{\tan}$ lies strictly in the interior of the cone—a case of static friction—, and one which lies on the boundary—a case of kinetic friction.

The Coulomb *friction* cone poses difficulties because it is quadratic, unlike the linear *convex cones* defined in Section 16.3. Several approximations of this cone can simplify the problem of solving for friction forces in multibody multicontact configurations, as described below in Sections 10.11.3 and 10.12.

10.11.2 An analytic model of Coulomb friction.

A novel model for Coulomb friction using Rayleigh dissipation functions is now derived. This model is the first instance of a standard NCP formulation which is motivated by physics and solvable for any value of friction coefficients. Previously known models are often restricted to a small friction coefficient near zero as in the Trinkle, Pang, Sudarsky, and Lo model [266], or constructed directly from the stepping equations as in the models of Anitescu, Potra, Stewart, and Trinkle [17, 261, 19], this latter family of models being solvable for all positive values of friction coefficient, when the problems are nondegenerate.

In its nonlinear form, the new model is rigorously isotropic but could easily be also extended to anisotropic friction as well—fabrics such as velvet are notorious for being strongly anisotropic. It also includes the possibility of velocity dependence of the friction coefficients as described in (10.55) for instance. Finally, in the nonlinear formulation, the number of complementarity conditions in the new model is minimal when compared with the solvable linear formulation [19]. Nevertheless, a linearization of the new model does reduce to a slight modification

of the standard LCP model [19], which is then regularized with small positive terms along the diagonal. This makes it solvable by the Lemke algorithm [179] in all cases, including degenerate ones.

To make this section simpler to read, all subscripts indicating contact index are dropped. A single contact constraint of the form $c(q) \geq 0$ which produces the Lagrange multiplier $\nu \geq 0$ is considered. The projection into the plane tangent to the contact is the $2 \times n$ projection operator $D(q)$ so that $v = D(q)\dot{q}$ is the relative tangential velocity at the contact point. An extension to multiple simultaneous contacts is described at the end of this section.

First recall that a Rayleigh function of the form

$$\mathfrak{R}(q, \dot{q}) = \frac{1}{2\gamma} \|D(q)\dot{q} - w(t)\|^2, \tag{10.56}$$

eventually produces the constraint $D(q)\dot{q} = w(t)$ if the damping constant γ^{-1} is large enough. This is true for any generalized velocity, including the velocities corresponding to the ghost variables. The driving velocity $w(t)$ will be set to 0 in what follows but it is an interesting addition to the modeling vocabulary, as it can be used to model a drive belt such as that found on grocery store checkout counters. This is easily reintroduced at the end of the derivation though.

Given the phenomenology described in Section 10.11.1, a pair of complementary constraints must be imposed as follows

Zero contact velocity: impose $v = D(q)\dot{q} \approx 0$ using ghost velocity $\dot{\beta} \in \mathbb{R}^2$;

Friction cone condition: impose $\|\dot{\beta}\| \leq \mu(\dot{q})\nu$ where μ is the friction coefficient and ν is the normal force for the given contact $c(q) \geq 0$.

For the first constraint, introduce the regularized Rayleigh function

$$\mathfrak{R}_\beta = -\left(\frac{\gamma \|\dot{\beta}\|^2}{2} + \dot{\beta}^T D(q)\dot{q} \right), \tag{10.57}$$

which was used before to model the kinematic constraint $D(q)\dot{q} = 0$.

To impose a limit on the magnitude of the friction force, introduce the *negative* of the Rayleigh function constructed in Section 10.8, using the form given in (10.49) which restricts the ghost $\dot{\sigma} \geq 0$

$$\mathfrak{R}_\sigma = \frac{\delta \dot{\sigma}^2}{2} + \dot{\sigma}(\nu\mu(\dot{q}) - \|\dot{\beta}\|). \tag{10.58}$$

As shown below, this negative sign is necessary to preserve the dissipation property in the case where the velocity limit is applied to a ghost, the variable β in this case, as opposed to the coordinates q directly.

The combination of these dissipation functions generates the following nonlinear complementarity force terms on the generalized variables q and ghosts ν, β,

and σ, after considering Fourier's inequality (10.35), equations are found

$$\begin{aligned} f_q &= D^T\dot{\beta} - \frac{\nu\mu'}{\|D(q)\dot{q}\|}D(q)\dot{q}\dot{\sigma}, \\ f_\beta &= \gamma\dot{\beta} + D(q)\dot{q} + \dot{\sigma}\frac{1}{\|\dot{\beta}\|}\dot{\beta} = 0, \\ 0 \le -f_\sigma &= \delta\dot{\sigma} - \|\dot{\beta}\| + \nu\mu(\dot{q}) \qquad \perp \dot{\sigma} \ge 0, \\ f_\nu &= 0, \end{aligned} \tag{10.59}$$

where it is assumed that the Lagrangian does not depend on the ghosts β and σ to justify $f_\beta = 0$ and $-f_\sigma > 0$. The nonlinearity is confined to the Jacobian matrices W and B defined as follows:

$$\begin{aligned} W &= (1/\|D(q)\dot{q}\|)D(q)\dot{q}^T, \\ B &= (1/\|\dot{\beta}\|)\dot{\beta}^T, \end{aligned} \tag{10.60}$$

so that the rows of W and B are normalized row vectors.

A few observations can be made on this model at this point. First off, it is strictly dissipative since a simple computation reveals

$$\begin{aligned} \frac{\mathrm{d}E}{\mathrm{d}t} &= \dot{q}^T f_q + \dot{\beta}^T f_\beta + \dot{\sigma} f_\sigma + \dot{\nu} f_\nu \\ &= \nu\mu'\|D(q)\dot{q}\|\dot{\sigma} - \gamma\|\dot{\beta}\|^2 - \|\dot{\beta}\|\dot{\sigma} + \dot{\sigma} f_\sigma \\ &\le 0. \end{aligned} \tag{10.61}$$

The friction coefficient derivative is non-positive, a known experimental fact, so that $\mu' \le 0$. The term $-\gamma\|\dot{\beta}\|^2$ is negative for $\gamma > 0$ as assumed here, and the same goes for $-\dot{\sigma}\|\dot{\beta}\|^2 < 0$ since $\dot{\sigma} \ge 0$. Fourier's inequality guarantees that $f_\sigma \le 0$ but in fact here, $\dot{\sigma} f_\sigma = 0$ by the complementarity rule.

Next, the following two sets of identities are revealed using simple algebraic manipulations on (10.59), first for the case of stiction when $\dot{\sigma} = 0$ the following conditions hold

$$\begin{aligned} \dot{\sigma} &= 0, \\ \|\dot{\beta}\| &< \nu\mu(\dot{q}), \\ D(q)\dot{q} &= -\gamma\dot{\beta} = 0 + O(\gamma), \\ 0 < \|D(q)\dot{q}\| &< \gamma\nu\mu(\dot{q}), \\ \dot{\beta}^T D(q)\dot{q} &= -\|\dot{\beta}\|\|D(q)\dot{q}\|, \end{aligned} \tag{10.62}$$

and then for the sliding case when $\dot{\sigma} = 0$ and the following conditions hold instead

$$\begin{aligned} \dot{\sigma} &> 0, \\ \|\dot{\beta}\| &= \delta\dot{\sigma} + \nu\mu(\dot{q}) = \nu\nu(\dot{q}) + O(\delta), \\ \|D(q)\dot{q}\| &= (1 + \gamma\delta)\dot{\sigma} + \gamma\nu\mu = \dot{\sigma} + O(\gamma), \\ \dot{\beta}^T D(q)\dot{q} &= -\|\dot{\beta}\|\|D(q)\dot{q}\|. \end{aligned} \tag{10.63}$$

Taken together, these two sets of relations exactly match the definition of Coulomb friction of (10.54) when $\delta = \gamma = 0$. In the case where $\gamma > 0$, there is no true stiction mode there is a *creep* velocity of magnitude $\|D\dot{q}\| = \gamma\|\dot{\beta}\|$. However since the regularization parameters are only introduced for numerical reasons and since they are kept near $\gamma \approx 10^{-6}$, this anomaly is not a big issue and definitely commensurate with observed creep as reported in [231], Chapter 11, for instance. The direct opposition of the friction force to the velocity is always exactly verified however, no matter the values of the regularization parameters.

This analytic formulation of the Coulomb friction problem is a standard NCP formulation with a single scalar complementarity condition. Indeed, the full equations of motion become

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathcal{L}}{\partial \dot{q}^T} - \frac{\partial \mathcal{L}}{\partial q^T} - f_q &= 0 \\ f_\beta &= 0 \\ 0 \le c(q) \quad \perp \quad \nu &\ge 0 \\ 0 \le -f_\sigma \quad \perp \quad \dot{\sigma} &\ge 0, \end{aligned} \tag{10.64}$$

where the definitions of f_q, f_β and f_σ of (10.59) were used. The system (10.64) is a standard NCP in the variables $\dot{q}, \ddot{q}, \nu, \dot{\beta}$, and $\dot{\sigma}$. In fact, at this stage, time derivatives for β and σ are deleted in the notation, since they play no role whatsoever in the definition of the equations of motion, though they were essential in finding the correct formulation via Rayleigh functions and d'Alembert's principle. In doing this, a set of non-linear differential algebraic inequalitys (DAI) in the dynamics variables of the physical system is recovered.

By contrast, the formulation of Pang et al. [266] and subsequent analysis by Pang and co-workers [62, 8, 269] was never formulated as a standard NCP.

Ignoring the contribution of the derivative of the friction coefficient μ' in the definition of f_q of (10.59), the conclusion is that the NCP (10.64) is solvable for Lagrangians $\mathcal{L}$ which have positive definite kinetic energy since for any approximation of the Jacobian B, as defined in the second line of (10.60), the corresponding MLCP is solvable. The full solvability proof is deferred so that it can be presented together with the equivalent result of the discretized problem as they are similar.

10.11.3 Linearized analytic Coulomb friction model

Nonlinear complementarity problems are a phenomenal beast to wrestle and thus, it makes sense to revise the analysis to introduce a linearized version of the analytic friction model. The culprit for nonlinearity is the norm operator $\|\beta\|$ that was needed to limit the tangential friction coefficient within the *friction cone*. This can be linearized with the following clever trick, known as the *polygonal* approximation. Consider a *non-orthogonal* basis in $d^{(i)} \in \mathbb{R}^2, i = 1, 2, \ldots, n_d$ such that any vector v can be decomposed as a non-negative linear combination

$$v = \sum_i \alpha_i d_i, \alpha_i \ge 0, i = 1, 2, \ldots, n_d, \tag{10.65}$$

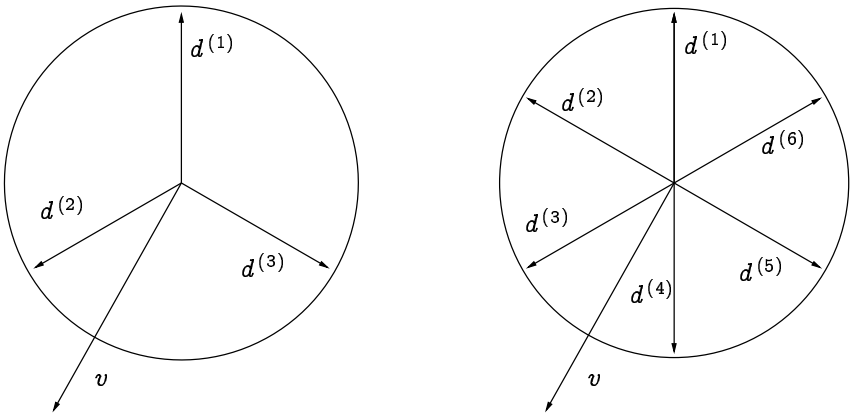

Figure 10.12: Non-orthogonal bases in $\mathbb{R}^2$ and the decomposition of a vector v on the basis elements. Only three vectors are needed but having a larger basis improves the approximation of the norm of v as the sum of the nonnegative projections.

so the Euclidean norm $\|v\|$ is approximately

$$\|v\| \approx \sum_i \alpha_i = E^T \alpha, \tag{10.66}$$

with the definitions

$$\alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{n_d} \end{bmatrix}, \qquad \text{and} \qquad E = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}. \tag{10.67}$$

This is illustrated in Figure 10.12. The difference between the exact norm depends on how closely the vector v is to a basis vector. Given a large basis, this error is small. Now, modify the definition of the projection operator $D(q)\dot{q}$ to read $\overline{D(q)}\dot{q}$ which, instead of being a projection in $\mathbb{R}^2$ decomposed on the usual orthonormal basis, is now the vector containing the n_d projections of the tangential contact velocity—the sliding velocity—along n_d basis vectors $d^{(i)}$. Let the norm of $\|\dot{\beta}\|$ be simply $E^T\dot{\beta}$, which is correct when $\dot{\beta} \geq 0$, the linearized Rayleigh functions (10.57) and (10.58) defining the friction model are now replaced with

$$\overline{\mathfrak{R}}_\beta = -\left(\frac{\gamma \|\dot{\beta}\|^2}{2} + \dot{\beta}^T \overline{D(q)}\dot{q} \right), \tag{10.68}$$

and the dissipation for the σ ghost restricted to $\dot{\sigma} > 0$ is

$$\overline{\mathfrak{R}_\sigma} = \frac{\delta \dot{\sigma}^2}{2} + \dot{\sigma}(\nu\mu(\dot{q}) - E^T\dot{\beta}). \tag{10.69}$$

Given that both ghosts are now restricted with $\dot{\beta} \geq 0$ and $\dot{\sigma} \geq 0$, the forces satisfy the following system

$$\begin{aligned} f_q &= \overline{D}^T \dot{\beta} + \nu \mu' \overline{W}^T \dot{\sigma} \\ 0 \leq -f_\beta &= \gamma \dot{\beta} + \overline{D(q)} \dot{q} + E \dot{\sigma} \qquad \perp \quad \dot{\beta} \quad \geq 0 \\ 0 \leq -f_\sigma &= \delta \dot{\sigma} - E^T \dot{\beta} + \nu \mu(\dot{q}) \qquad \perp \quad \dot{\sigma} \quad \geq 0, \end{aligned} \tag{10.70}$$

according to Fourier's inequality (10.35). This is recognized as the Anitescu-Potra-Stewart-Trinkle solvable LCP model [259, 262, 19, 261], with the exception of the regularization parameters γ, δ, which correspond to small diagonal perturbation. This difference is investigated further in Section 10.11.5.

10.11.4 A discretized model of Coulomb friction

As in the previous sections dealing with polygenic forces and dissipation forces, e.g., Section 3.12, the main issue is to choose an appropriate discretization of the *integral* $-\int_0^h \partial \mathfrak{R} / \partial \dot{z} \delta z$, where z is the agglomerated vector of regular and ghost coordinates. As was done in Section 4.3 in the discretization of strong Rayleigh functions corresponding to regularized holonomic constraints, an *implicit* discretization is chosen for most terms with the exception of the term proportional to the derivative of the friction coefficient, μ', in (10.59). The reason is that this term is small and that, as shown in the next section, discretizing it implicitly breaks a symmetry in the NCP or LCP formulation, jarring with the solvability argument.

The Rayleigh function $\mathfrak{R}_\beta$ of (10.57) was discretized in Section 4.3 and the same method is used here. Looking at the force terms generated by $\mathfrak{R}_\sigma$ of (10.58), the following discretizations are used

$$\begin{aligned} f_{d\,\sigma,q}^{(+)} &= \frac{1}{2}(\nu_1 + \nu_0)\mu'(v_1) W_1 \sigma_1, \\ f_{d\,\sigma,q}^{(-)} &= 0, \\ f_{d\,\sigma,\beta}^{(+)} &= 0, \\ f_{d\,\sigma,\beta}^{(-)} &= B_1^T \sigma_1, \\ -f_{d\,\sigma,\sigma}^{(+)} &= 0, \\ -f_{d\,\sigma,\sigma}^{(-)} &= \delta \dot{\sigma}_1 + \frac{1}{2}(\nu_1 + \nu_0)\mu(v_0) - B_1 \dot{\beta}_1, \end{aligned} \tag{10.71}$$

with the definitions $\dot{\sigma}_1 = h^{-1}(\sigma_1 - \sigma_0)$ and similarly for $\dot{\beta}_1$, as well as the velocity $v_1 = h^{-1}(q_1 - q_0)$, and the normalized velocity projection matrices $W_1 = \|v_1\|^{-1} v_1^T$ and $B_1 = |\beta_1\|^{-1} \beta_1^T$. The negative sign in the last two equations are there for convenience in writing the complementarity condition. Since there are no other terms in the Lagrangian involving σ, this is purely notational.

For the linearized model of Section 10.11.3, the force terms are computed

similarly to yield

$$\begin{aligned}
\overline{f}^{(+)}_{d\,\sigma,q} &= \frac{1}{2}(\nu_1+\nu_0)\mu'(v_1)W_1\sigma_1,\\
\overline{f}^{(-)}_{d\,\sigma,q} &= 0,\\
\overline{f}^{(+)}_{d\,\sigma,\beta} &= 0,\\
\overline{f}^{(-)}_{d\,\sigma,\beta} &= E\sigma_1,\\
-\overline{f}^{(+)}_{d\,\sigma,\sigma} &= 0,\\
-\overline{f}^{(-)}_{d\,\sigma,\sigma} &= \delta\dot{\sigma}_1 + \frac{1}{2}(\nu_1+\nu_0)\mu(v_0) - E^T\dot{\beta}_1,
\end{aligned} \tag{10.72}$$

and all terms with the exception of W_1 involve only linear operators.

10.11.5 Solvability of analytic and discretized Coulomb friction models

First introduce the notation necessary to consider multiple frictional contact points. Consider a set of nonpenetration conditions given by n_c scalar functions $c^{(i)}(q) \geq 0$, each describing a single contact with index i whenever $c^{(i)}(q) = 0$. Contact surface i has normal $C^{(i)} = \partial c^{(i)}/\partial q$ and local tangent projection operator $D^{(i)}(q)$ or $\overline{D}^{(i)}(q)$. For the latter case of non-orthonormal projections, the number of directions for each contact is $n_d^{(i)}$. The physical properties are given by the friction coefficient function $\mu^{(i)}$ and the regularization parameters are given by $\epsilon^{(i)}, \gamma^{(i)}$ and $\delta^{(i)}$ for the contact, zero velocity and friction cone dissipation functions, respectively. These parameters are agglomerated in the diagonal matrices U, Σ, Δ, and Γ, respectively, and have the explicit form

$$\begin{aligned}
U &= \mathrm{diag}(\mu^{(1)}, \mu^{(2)}, \ldots, \mu^{(n_c)}),\\
\Sigma &= \mathrm{diag}(\epsilon^{(1)}, \epsilon^{(2)}, \ldots \epsilon^{(n_c)}),\\
\Delta &= \mathrm{diag}(\delta^{(1)}, \delta^{(2)}, \ldots, \delta^{(n_c)}),\\
\Gamma &= \mathrm{diag}(\gamma^{(1)} I_2, \gamma^{(2)} I_2, \ldots \gamma^{(n_c)} I_2),\\
\overline{\Gamma} &= \mathrm{diag}(\gamma^{(1)} I_{n_1^{(d)}}, \gamma^{(2)} I_{n_2^{(d)}}, \ldots, \gamma^{(n_c)} I_{n_{n_c}^{(d)}}).
\end{aligned} \tag{10.73}$$

Each active contact has a tangential velocity $v^{(i)} = D^{(i)}\dot{q}$ and a tangential force $\beta^{(i)}$, and in the linearized formulation, a norm estimator $E^{(i)} = (1, 1, \ldots, 1)^T$ which is an $n_d^{(i)} \times 1$ matrix, leading to the diagonal block matrices

$$\begin{aligned}
W &= \mathrm{diag}(W^{(1)}, W^{(2)}, \ldots W^{(n_c)}),\\
B &= \mathrm{diag}(B^{(1)}, B^{(2)}, \ldots B^{(n_c)}),\\
E &= \mathrm{diag}(E^{(1)}, E^{(2)}, \ldots E^{(n_c)}),
\end{aligned} \tag{10.74}$$

where W and B are matrices of size $n_c \times 2n_c$ and E is of size $\sum_i n_d^{(i)} \times n_c$. Putting all these elements together, the stepping equations take the quasilinear

form

$$\begin{bmatrix} M & -D_k^T & -C_k^T & 0 \\ D_{k'} & \Gamma & 0 & B_{k'}^T \\ C_{k'} & 0 & \Sigma & 0 \\ 0 & -B_{k'} & U & \Delta \end{bmatrix} \begin{bmatrix} x \\ \beta \\ \nu \\ \sigma \end{bmatrix} + \begin{bmatrix} q_x \\ q_\beta \\ q_\nu \\ q_\sigma \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ b \\ c \end{bmatrix} \tag{10.75}$$
$$0 \le \nu \perp b \ge 0$$
$$0 \le \sigma \perp c \ge 0,$$

for the case of the nonlinear friction model,

$$\begin{bmatrix} M & -\overline{D}_k^T & -C_k^T & 0 \\ \overline{D}_{k'} & \Gamma & 0 & E \\ C_{k'} & 0 & \Sigma & 0 \\ 0 & -E^T & U & \Delta \end{bmatrix} \begin{bmatrix} x \\ \overline{\beta} \\ \nu \\ \sigma \end{bmatrix} + \begin{bmatrix} q_x \\ q_\beta \\ q_\nu \\ q_\sigma \end{bmatrix} = \begin{bmatrix} 0 \\ \overline{a} \\ \overline{b} \\ \overline{c} \end{bmatrix} \tag{10.76}$$
$$0 \le \overline{\beta} \perp a \ge 0$$
$$0 \le \nu \perp b \ge 0$$
$$0 \le \sigma \perp c \ge 0,$$

for the linearized model.

Block vectors x, q_β, q_ν and q_σ have different forms depending on whether the position or velocity formulation is used, as discussed is Section 3.15 and Section 4.5. This leads to either using $x = q_{k+1}$ or $x = v_{k+1}$, and the block components q_x, q_β, q_ν and q_σ vary accordingly. The ghost coordinates β, ν, and σ also include either a factor of h^2 for the position formulation and h for the velocity formulation.

The system (10.75) is quasi-linear in the sense that block matrices $D_{k'}, C_{k'}$ and $B_{k'}$ should in fact be evaluated at the solution q_{k+1}, β, ν and σ.

To understand the solvability of the MLCP defined in (10.75), start with the case where all block matrices are evaluated at the beginning of the time step so that $k' = k$. This introduces much symmetry in the block matrix on the left hand side of (10.75) and leads to a solvable MLCP. Indeed, in reference to the solvability theory discussed in Section 16.4, what is needed for either the quasilinear or the linear formulation is an estimate of the size of $z^T H z$ where H is the square matrix appearing in (10.75) or in (10.76). The estimates are

$$z^T H_{\text{nonlin}} z = x^T M x + \beta^T \Gamma \beta + \nu^T \Sigma \nu + \sigma^T \Delta \sigma + \nu^T U \sigma \\ - x^T (D_k - D_{k'}) \beta - x^T (C_k - C_{k'}) \nu, \tag{10.77}$$

for the nonlinear case, and

$$\overline{z}^T H_{\text{lin}} \overline{z} = x^T M x + \overline{\beta}^T \overline{\Gamma \beta} + \nu^T \Sigma \nu + \sigma^T \Delta \sigma + \nu^T U \sigma \\ - x^T (\overline{D}_k - \overline{D}_{k'}) \beta - x^T (C_k - C_{k'}) \nu, \tag{10.78}$$

for the linear case. Choosing the approximation $C_{k'} = C_k$, $D_{k'} = D_k$, and $\overline{D}_{k'} = \overline{D}_k$, the conclusion is reached that unless the diagonal block matrices all vanish simultaneously, there cannot be an exceptional family of elements, as per Definition 16.12, for these complementarity problems and thus, they are both solvable. The same conclusion is reached by applying the friction feasibility theory of Pang and Stewart [227], which hinges also in the absence of an exceptional family of elements, but requires also a number of additional assumptions. Since solvability is proven already, there is no need to go over these.

A proof of the existence of a solution is hardly a solvability proof, however. Solvability is now demonstrated for each model, starting with the nonlinear model of (10.75). First, symmetrize by setting $C_{k'} = C_k$ and similarly, $D_{k'} = D_k$ in (10.75). Drop all k and k' prime subscripts in the block matrices B, C and D, and then take the Schur complement of the first line of (10.75), solving for x, and evaluating and evaluating the following substitutions

$$\begin{aligned} x &= -M^{-1}q_x + M^{-1}D^T\beta + M^{-1}C\nu, \\ Dx &= DM^{-1}D^T\beta + DM^{-1}C\nu - DM^{-1}q_x, \\ Cx &= CM^{-1}D^T\beta + CM^{-1}C\nu - CM^{-1}q_x. \end{aligned} \tag{10.79}$$

This yields the following reduced system

$$\begin{bmatrix} S_{DD} & S_{DC} & B \\ S_{DC}^T & S_{CC} & 0 \\ -B^T & U & \Delta \end{bmatrix} \begin{bmatrix} \beta \\ \nu \\ \sigma \end{bmatrix} + \begin{bmatrix} q_\beta - DM^{-1}q_x \\ q_\nu - CM^{-1}q_x \\ q_\sigma \end{bmatrix} = \begin{bmatrix} 0 \\ b \\ c \end{bmatrix} \tag{10.80}$$
$$0 \le \nu \perp b \ge 0$$
$$0 \le \sigma \perp c \ge 0,$$

where the following definitions were used

$$\begin{aligned} S_{DD} &= DM^{-1}D^T + \Gamma, \\ S_{CC} &= CM^{-1}C^T + \Sigma, \\ S_{DC} &= DM^{-1}C^T. \end{aligned} \tag{10.81}$$

Now, since S_{DD} is symmetric and positive definite, it is possible to compute another Schur complement, solving for β and substituting, to get

$$\begin{bmatrix} H_{11} & -H_{12}^T \\ H_{12} + U & H_{22} \end{bmatrix} \begin{bmatrix} \nu \\ \sigma \end{bmatrix} + \begin{bmatrix} q_\nu - CM^{-1}q_x - S_{DC}^T S_{DD}^{-1} q_\beta \\ q_\sigma + NS_{DD}^{-1}q_\beta \end{bmatrix} = \begin{bmatrix} b \\ c \end{bmatrix} \tag{10.82}$$
$$0 \le \nu \perp b \ge 0$$
$$0 \le \sigma \perp c \ge 0,$$

with following definitions for the block H_{ij} matrices

$$\begin{aligned} H_{11} &= S_{CC} - S_{DC}^T S_{DD}^{-1} S_{DC} \\ H_{21} &= B^T S_{DD}^{-1} S_{DC} \\ H_{22} &= \Delta + B^T S_{DD}^{-1} B. \end{aligned} \tag{10.83}$$

The problem defined by (10.82) is a pure LCP. This is solvable by Lemke's algorithm [179] because the matrix $H_U = H + \tilde{U}$ is copositive plus, being the sum of a positive definite matrix

$$H = \begin{bmatrix} H_{11} & -H_{21}^T \\ H_{21} & H_{22}, \end{bmatrix} \tag{10.84}$$

and a matrix with non-negative entries, $[\tilde{U}]_{ij} \geq 0$, where $\tilde{U}$ is defined blockwise as

$$\tilde{U} = \begin{bmatrix} 0 & 0 \\ U & 0 \end{bmatrix}, \tag{10.85}$$

and the blocks of matrix $\tilde{U}$ have dimensions matching the blocks of matrix H. The proof that H_U is copositive plus is found in [69] and is used as the main proof in [14] but for a slightly different problem which is now described.

Consider now the linear model of (10.76). The matrix appearing in this model is nearly identical to that of the Anitescu, Potra,Trinkle, and Stewart model [17, 261, 259, 19], except for the addition of a diagonal perturbation with non-negative elements, and the neglect of holonomic constraints. Repeating the procedure just described for the nonlinear case but taking only one Schur complement, eliminating only the x variables, the reduced system reads

$$\begin{bmatrix} \overline{S}_{DD} & \overline{S}_{DC} & E \\ \overline{S}_{DC}^T & \overline{S}_{CC} & 0 \\ -E^T & U & \Delta \end{bmatrix} \begin{bmatrix} \overline{\beta} \\ \nu \\ \sigma \end{bmatrix} + \begin{bmatrix} q_\beta - \overline{D} M^{-1} q_x \\ q_\nu - C M^{-1} q_x \\ q_\sigma \end{bmatrix} = \begin{bmatrix} \overline{a} \\ \overline{b} \\ \overline{c} \end{bmatrix}$$
$$0 \leq \overline{\beta} \perp \overline{a} \geq 0$$
$$0 \leq \nu \perp \overline{b} \geq 0$$
$$0 \leq \sigma \perp \overline{c} \geq 0, \tag{10.86}$$

with the definitions

$$\begin{aligned} \overline{S}_{DD} &= \overline{D} M^{-1} \overline{D}^T + \overline{\Gamma}, \\ \overline{S}_{CC} &= C M^{-1} C^T + \Sigma, \\ \overline{S}_{DC} &= \overline{D} M^{-1} C^T. \end{aligned} \tag{10.87}$$

Now defining

$$\overline{S}_0 = \begin{bmatrix} \overline{S}_{DD} & \overline{S}_{DC} & E \\ \overline{S}_{DC}^T & \overline{S}_{CC} & 0 \\ -E^T & 0 & \Delta \end{bmatrix}, \qquad \text{and } \overline{U} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & U & 0 \end{bmatrix}$$
$$\overline{S}_U = \overline{S}_0 + \overline{U}, \tag{10.88}$$

it follows that $\overline{S}_0$ is positive definite, and since $\overline{U}$ contains only nonnegative elements, $\overline{S}_U$ is co-positive plus and thus solvable using the Lemke algorithm [179]. By contrast, the Anitescu-Potra-Stewart-Trinkle model matrix is only co-positive

since it does not have any of the positive diagonal perturbations of the present model. The Lemke algorithm is more robust when applied to co-positive plus matrices since there cannot be any form of cycling then [69].

If the multibody system is subject to a mix of locally linearized, regularized holonomic and nonholonomic constraints, as described in Section 4.5, the analysis above holds with the replacement of the mass matrix M with the positive definite matrix H defined in (4.32), and the vector x is replaced with $(x^T, \alpha^T, \lambda^T)^T$ of (4.29) for the position formulation or (4.31) for the velocity formulation, as the case applies. In either case, all the conclusions just derived survive.

Solving either the nonlinear model (10.75) or the linearized version (10.76) efficiently is a challenging task and often requires symmetric approximation and splitting, as discussed in Chapter 17 and in Anitescu and Hart [16]. However, having both an existence and a solvability proof is a good thing, indeed. As shown in Chapter 11, other formulations of the Coulomb friction problem do not enjoy a guaranteed existence proof and this allows to the existence paradoxical configurations which do not have finite solutions for the contact forces.

10.12 Survey of numerical models of dry friction

Modeling of Coulomb friction using complementarity formulation in multibody systems has a long history, going back to the work of Lötstedt [189] in the early 1980s, followed by Baraff [34, 33] in the graphics literature, and Panagiotopoulos [1, 224], Glocker and Pfeiffer [232], and Anitescu, Stewart, Trinkle, and Pang [229, 266, 228, 262, 17, 19], in the mathematical and engineering literature. Unfortunately, these are not solvable except in very low friction. Only the formulations of Anitescu, Stewart, Trinkle, and Potra [17, 261, 259, 19], are actually solvable for arbitrary non-negative friction coefficients.

NCP models have also been proposed by Pang and co-workers [266, 229, 269, 62], among many others, but without any proof that the NCP was solvable, except in the limit of *very* small friction coefficients. Existence and solvability are still open problems for the case of deformable bodies.

The problem here is that dry friction produces impulses during stick-slip transition and this actually means that accelerations are not well defined. A classical two-dimensional example due to Painlevé [223], involving a rod contacting a plane, can be used to demonstrate the existence of configurations with no finite solutions and others with multiple solutions, in addition to regular cases with unique solutions. However, this paradox is lifted if an implicit time stepping formulation in terms of velocities is chosen, and the dry friction model is discretized judiciously, as demonstrated in the work of Stewart and Trinkle [261, 259], and refined by Anitescu, Potra, and Stewart [17, 19]. Existence and solvability is established for these models but only for non-degenerate cases. The diagonal perturbations corresponding to the regularization of the Rayleigh functions of Section 10.11.2 remove this non-degeneracy requirement, as shown in Section 10.11.5.

Except for the work of Pandolfi, Kane, Marsden, and Oritz [225], none of the previously cited time stepping schemes are derived from a variational principle. The formulation of Pandolfi et al. [225] shares similarity to the present in that it also uses Rayleigh dissipation functions. However, instead of introducing the tangential force explicitly with Lagrange multipliers, the Rayleigh functions are defined as

$$\mathfrak{R} = \sum_i \mu^{(i)} \nu^{(i)} |D^{(i)}(q)\dot{q}|, \tag{10.89}$$

where the sum extends over all contacts i, each of which has friction coefficient $\mu^{(i)}$ and tangential velocity projection $D^{(i)}(q)$. This is possible since their approach relies on a nonsmooth optimization scheme which minimizes the Rayleigh functions (10.89) at each step. The main issue with this work is that one needs to solve nonsmooth, nonconvex, nonlinear optimization problems. A proof of the existence of a solution was not provided in [225].

New friction models are routinely published in the literature. The reader is referred to the monograph of Brogliato [56] for a broad survey.

Some notable approximations of dry frictional contacts are worth mentioning though. The simplest is to only compute the normal forces and to then apply a force $f_{\tan} = -\mu\nu\,\mathtt{sgn}(D(q)\dot{q})$ for each contact. The problem here is to decide what to do with $\mathtt{sgn}(0)$. One solution for this is to replace $\mathtt{sgn}(x)$ with a similar function that is not discontinuous, such as $\mathtt{tanh}(\kappa x)$ for instance, with $\kappa > 0$. For large values of κ, this produces something similar to $\mathtt{sgn}(x)$ though it is always the case that $\mathtt{tanh}(\kappa x) = 0$ for $x = 0$ and this leads to significant *creep* velocities at contacts, an undesirable feature. Trying to fix this problem leads necessarily to a nonsmooth function and this, in turn, requires solution of LCPs, as shown in [258].

Assuming that one chooses a truly nonsmooth formulation, simplifications come in two categories, namely, relaxing the coupling between the tangential force budget and the magnitude of the normal force of (10.54), or decoupling the components of the tangential force when estimating the magnitude. This leads to the following models.

Using the usual Euclidean norm, bound the tangential contact force $f_{\tan}$ as $\|f_{\tan}\| \leq \mu\overline{\nu}$, for some $\overline{\nu} > 0$, such as the last known value, for instance. This replaces the friction cone of Figure 10.11 with a cylinder. Depending on the size of the current normal, this approximation produces an overestimate or underestimate of the tangential force budget, as shown in Figure 10.13. The resulting stepping scheme for this is still nonlinear though. The main issue here is that for a stacking problem, unless the estimates $\overline{\nu}$ are adjusted dynamically, the tangential force budget for items at the bottom of the pile will tend to be insufficient, whilst those at the top will be excessive.

One can also use a friction pyramid approximation to the friction cone, bounding *each* component of the tangential force individually along the tangential directions d_1 and d_2 specified by the projection operator $D(q)\dot{q}$, so that $f_{\tan\, i} \leq \mu\nu, i = 1, 2$, and ν is the normal force of the current computation. This is done in [228] for instance. Figure 10.14 illustrates the situation. The coupling between

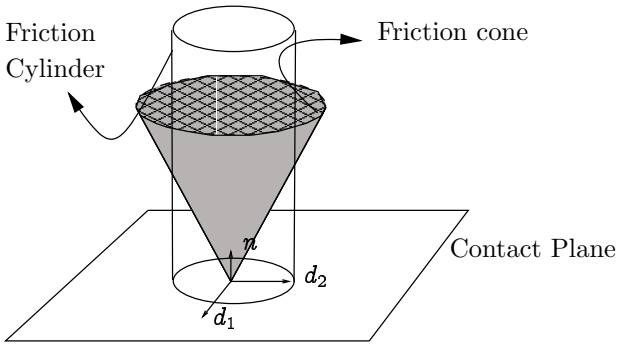

Figure 10.13: The cylindrical approximation to the Coulomb friction cone.

the magnitude of the tangential force budget and the normal survives, but there can be excessive friction when the external forces are aligned toward the corners of the pyramid.

Combining the previous two strategies leads to *box* approximation of the friction cone in which each component of the tangential force is bounded by an estimate of the normal force, so that $f_{\tan\, i} \leq \mu\overline{\nu}, i = 1, 2$, and $\overline{\nu}$ is an estimate of the corresponding normal force. This leads to anisotropy as in the case of pyramidal friction above, and the absence of scaling of the tangential force budget described in the cylindrical approximation approximation to the friction cone. However, the MLCP describing this problem is now a simple boxed, mixed linear complementarity problem, and the matrix defining it is now positive definite, provided the perturbation parameters are all positive.

For both the box and cylindrical friction models, the problems to solve are equivalent to quadratic programs, as shown in Chapter 17. It is also possible to perform iterations so the estimate $\overline{\nu}$ is refined, and thus the cylinder or the box can be scaled. Such schemes have been used several times in the literature at least in [191, 273, 236, 2, 238, 142, 143], and probably numerous other instances. The convergence of these techniques is often assumed [77], but as shown in Chapter 17, it is not guaranteed.

For completeness, an illustration of the polygonal approximation of the friction cone described in Section 10.11.3 is found in Figure 10.16. The directions d_i can be chosen to reduce the anisotropy found in the pyramidal model, and strictly scale the tangential force budget with the normal.

The cylindrical, pyramidal, box, and polygonal friction cone approximation described above, are all solvable when discretized with the techniques of Section 10.11.4, and moreover, they are all strictly dissipative, since they can all be formulated as approximations of the Rayleigh functions (10.57) and (10.58), which are dissipative analytically as shown in Section 10.11.2, and also when discretized using a short calculation similar to that of (10.53).

In addition to the models provided above, several other models and algorithms applicable to the quasi-static problem of deformable bodies were reviewed in

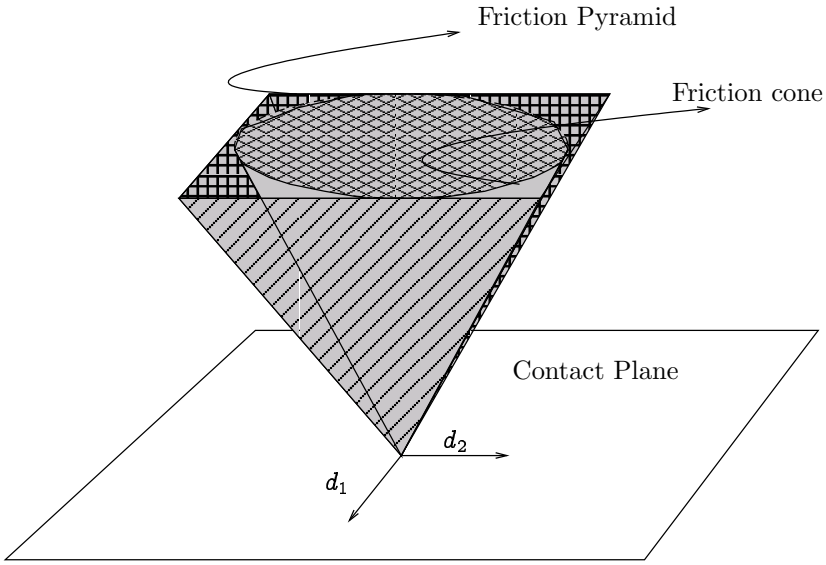

Figure 10.14: A pyramidal approximation model to the Coulomb friction cone.

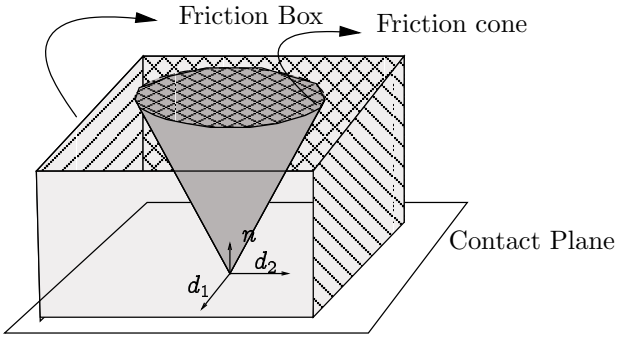

Figure 10.15: The box approximation model of the Coulomb friction cone.

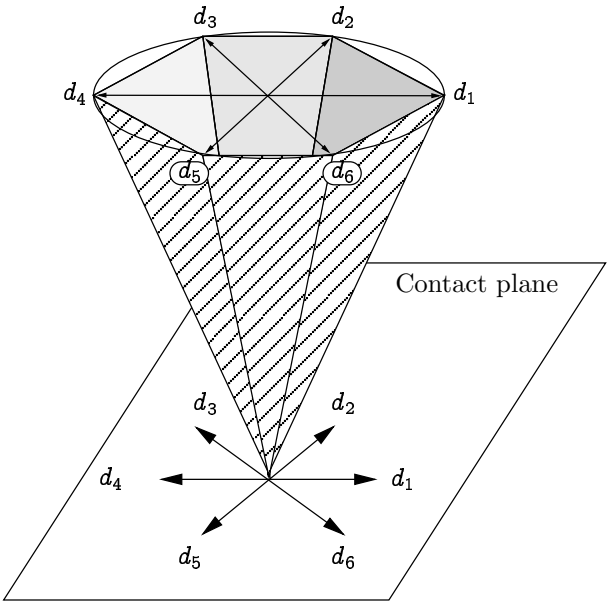

Figure 10.16: The Anitescu-Potra-Trinkle-Stewart approximation model of the Coulomb friction cone.

Christensen, Klarbring, Pang, and Strömberg [62, 63, 64]. Generally, these algorithms are reported to work well only when the friction coefficients are very small.

More recently, Glowinski, Shiau, LieJune, Ming, and Nasser, constructed dry friction models and solution methods in a series of articles [101, 102, 255, 103] based on a non-standard complementarity formulation. The idea here is to compute a Lagrange multiplier λ such that, stated here in one dimension for simplicity,

$$\lambda \dot{q} = |\dot{q}|. \tag{10.90}$$

This type of condition is not at all similar to the complementarity formulation used here. It is not clear when such nonlinear conditions can be solved, even though the idea itself is sound and correctly captures the physics. Whether this formulation resolves the Painlevé paradox remains to be seen.

10.13 End notes

To get to grips with the issues related with friction, consider a classical example of a point particle moving in one dimension subject the combined force of a harmonic oscillator, dry friction, and a sinusoidal force. This example is found in [113], example 6.5, for instance, and used recurrently in the literature. The point particle has unit mass, it is attached to the origin with a spring of unit strength, and it is subject to a viscous drag force of $-\gamma\dot{q}$ where $\gamma > 0$. The particle is also subject to dry friction which is represented here as $\mu\,\mathrm{sgn}(\dot{q})$,

where μ is the effective friction coefficient and $\mathtt{sgn}(x)$ is the signum function. Using Newton's law directly, the equation of motion reads

$$\ddot{q} + 2\gamma\dot{q} + \mu\,\mathtt{sgn}(\dot{q}) + q = A\cos(\omega t). \tag{10.91}$$

The dry friction force is obviously nonsmooth, producing the term $\pm\mu$ whenever $\dot{q} \neq 0$. Resolving what happens when $\dot{q} = 0$ is the main issue when solving dry friction problems. Since formally, the signum function at the origin can take all values in the interval $[-1, 1]$, system (10.91) is in fact a *differential inclusion*, which should be written as $\ddot{q} \in \mathcal{F}$, where $\mathcal{F}$ is the set of possible values of the forcing terms. An analysis of these in the particular context described here can be found in in [258]. A simple strategy consists of analyzing the transition and trying to pick the right value of $\mathtt{sgn}(0)$ using consistency arguments.

The solution constructed in [113], for instance, requires adaptive time stepping and switch conditions at each transition, as well as the solution of a small LCP to pick the suitable value for $\mathtt{sgn}(0)$ whenever the velocity crosses zero. By contrast, using either the smoothed nonlinear or the LCP formulation for the friction model presented in Section 10.11.4, one can compute a solution indistinguishable from the one presented in [113], but using a large time step $h = 1/60$. The solution computed using the nonlinear model or LCP model are shown in Figure 10.17. The data is identical though it is much faster to compute a solution to the LCP directly, at least for this case. Also, in comparison to alternative models such as those of Glowinski [103] and references provided therein, there is no high oscillation in the Lagrange multiplier term.

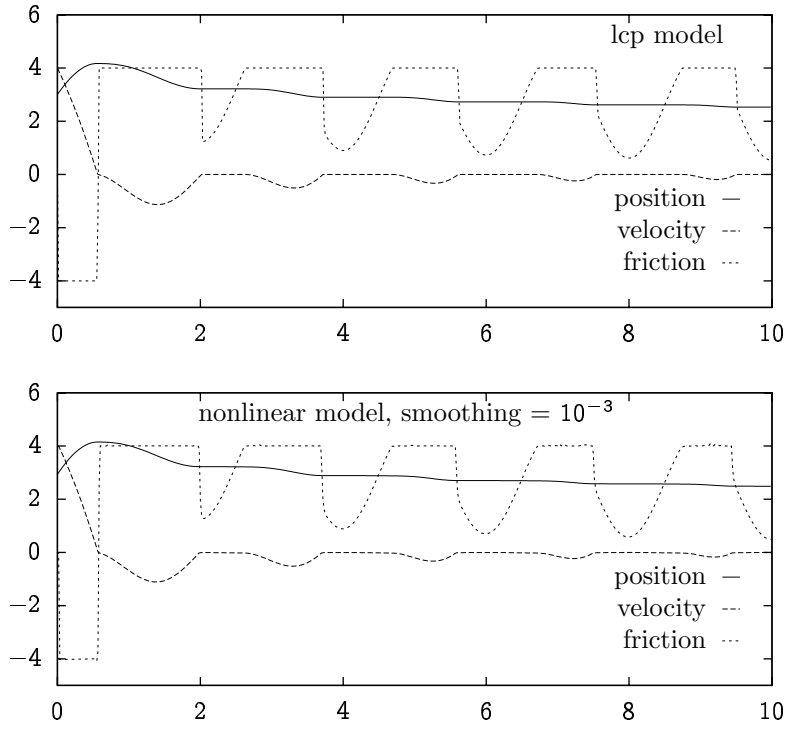

Figure 10.17: A one-dimensional dry friction example, integrated using both the nonlinear formulation the linear complementarity problem reduction.

11 Bagatelle VII: The Painlevé Paradox

Dry friction is a non-standard force model defined as a mix of nonholonomic and nonideal constraints. This formulation leads to paradoxical configurations where, apparently, it is not possible to compute the acceleration of simple dynamical systems. This paradox is easily analyzed in the case of a planar rod in dry frictional contact with a line in the state of kinetic (sliding) friction.

Historical background of the problem is provided in Section 11.1, followed by a kinematic description of the mechanical system. The equations of motion are then derived for both the static and kinetic friction states in Section 11.3. The paradox is discussed in details in Section 11.4 where numerical experiments are also presented to illustrate that the dry friction formulation of Chapter 10 and the discretization of Section 10.11.4 successfully resolve the problem, as was proved in Section 10.11.5. Other resolution techniques and models are discussed in Section 11.5.

11.1 Introduction

Then French mathematician and politician Paul Painlevé observed that Coulomb's law of dry friction can be used to construct simple examples in two dimensions which have either no finite solution, several solutions, or a unique solution, depending on the details of the configuration, as first reported in [223]. This example is presented here in order to illustrate how the variational stepping scheme combined with the friction model developed in Section 10.11.4 leads to a resolution of this paradox.

The paradox is realized by a simple two-dimensional rod of length $2l$ and mass m subject to downward gravity and Coulomb friction at one contact point as described below and in [259, 232, 34] for instance. A different version of this paradox which has more to do with differential inclusions is also found in [108]. The two formulations will be compared briefly in Section 11.5.

11.2 Basic configuration

Consider a two-dimensional thin rod of length $2l$, width $2w$ and mass m, with uniform density. The inertia of this system is easily computed to be

$$\begin{aligned} \mathcal{J}_0 &= \int_{-w}^{w} \mathrm{d}x_2 \int_{-l}^{l} \mathrm{d}x_1 \frac{m}{4lw} \left(x_1^2 + x_2^2 \right) \\ &= \frac{ml^2}{3} + \frac{mw^2}{3}, \end{aligned} \tag{11.1}$$

where x_1, x_2 are the Cartesian coordinates of the points interior to the rod. In the limit of vanishing thickness, this reduces to

$$\lim_{w \to 0} \mathcal{J}_0 = \frac{ml^2}{3}. \tag{11.2}$$

When the density is nonuniform and concentrated toward the center, the value of $\mathcal{J}_0$ can be made smaller and, as we show in Section 11.3.2, a smaller value of $\mathcal{J}_0$ allows to construct paradoxical examples for lower values of friction coefficients. The value $\mathcal{J}_0 = ml^2/3$ makes the algebra slightly simpler.

The coordinates of the system are the planar coordinates of the center of mass $x(t) \in \mathbb{R}^2$ as well as the elevation angle θ. This is agglomerated in the coordinate vector $q \in \mathbb{R}^3$

$$q = \begin{bmatrix} q^{(1)} \\ q^{(2)} \\ q^{(3)} \end{bmatrix} = \begin{bmatrix} x \\ \phi \end{bmatrix}, \; x = \begin{bmatrix} x^{(1)} \\ x^{(2)} \end{bmatrix}. \tag{11.3}$$

The 3×3 mass matrix for this system is

$$M = \mathrm{diag}(m, m, \mathcal{J}_0), \tag{11.4}$$

and is constant. Introduce the unit vector $u(\phi) \in \mathbb{R}^2$ as

$$u(\phi) = \begin{bmatrix} \cos\phi \\ \sin\phi \end{bmatrix}. \tag{11.5}$$

The 2D Cartesian coordinates of the two extremities of the rod are labeled with a $\pm$ label are are found at

$$p^{(\pm)} = x \pm l u(\phi), \tag{11.6}$$

and these two points have velocity:

$$v\pm = \dot{\phi} l J_1 u(\phi), \tag{11.7}$$

where we introduced the 2×2 orthonormal matrix

$$J_1 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \tag{11.8}$$

which is a root of the identity, i.e., $J_1^2 = I_2$, where I_2 is the 2×2 identity matrix.

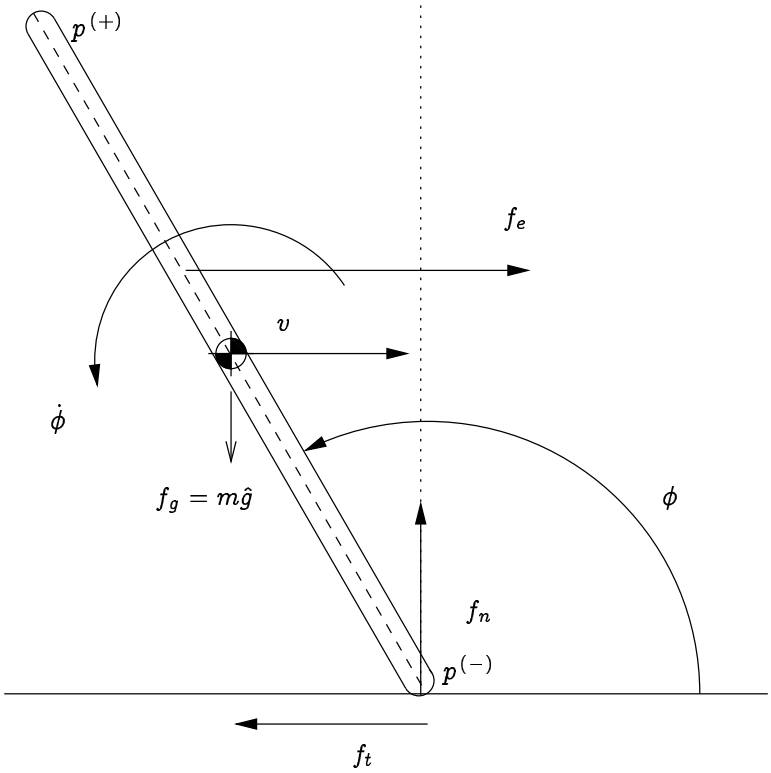

Figure 11.1: Schematics of a two-dimensional rod in frictional contact with a plane. The contact point is either touching or separating in the normal direction, and either sliding or sticking in the tangential one.

To impose non-penetration constraints at the extremities of the rod, we simply impose that the y coordinates of the extremities $p^{(\pm)}$ be non-negative, namely,

$$g^{(\pm)}(q) = q^{(2)} \pm l \sin(q^{(3)}) = q^{(2)} \pm l \sin\phi \geq 0, \tag{11.9}$$

which produces the Jacobians

$$G^{(\pm)} = \begin{bmatrix} 0 & 1 & \pm l\cos(q^{(3)}) \end{bmatrix} = \begin{bmatrix} 0 & 1 & \pm l\cos\phi \end{bmatrix}. \tag{11.10}$$

For any of these constraints, the projected tangential velocity at the contact point along the $q^{(1)}$ or x direction is given by $D^{(\pm)}\dot{q} = (1,0)p^{(\pm)}$ where

$$D^{(\pm)} = \begin{bmatrix} 1 & 0 & \mp l\sin(q^{(3)}) \end{bmatrix} = \begin{bmatrix} 1 & 0 & \mp l\sin\phi \end{bmatrix}. \tag{11.11}$$

Since an acceleration formulation is used in what follows, the second time

derivatives are needed. A simple computation yields the following terms

$$\ddot{g}^{(\pm)}(q) = G^{(\pm)}(q)\ddot{q} + \dot{G}^{(\pm)}(q)\dot{q} = G^{(\pm)}(q)\ddot{q} + b^{(\pm)}(q,\dot{q}), \tag{11.12}$$

$$b^{(\pm)}(q,\dot{q}) = \mp l\dot{\phi}^2 \sin\phi, \tag{11.13}$$

$$\frac{d}{dt}\left[D^{(\pm)}(q)\dot{q}\right] = D^{(\pm)}(q)\ddot{q} + \dot{D}^{(\pm)}(q)\dot{q} = D^{(\pm)}(q)\ddot{q} + c^{(\pm)}(q,\dot{q}), \tag{11.14}$$

$$c^{(\pm)}(q,\dot{q}) = \mp l\dot{\phi}^2 \cos\phi. \tag{11.15}$$

In addition, the values of the following matrices will be needed in what follows

$$A = G^{(\pm)}M^{-1}G^{(\pm)^T} = \frac{1}{m}\left[1 + \frac{l^2 m}{\mathcal{J}_0}\cos^2\phi\right] = \frac{1}{m}\left[1 + 3\cos^2\phi\right], \tag{11.16}$$

$$B = D^{(\pm)}M^{-1}D^{(\pm)^T} = \frac{1}{m}\left[1 + \frac{l^2 m}{\mathcal{J}_0}\cos^2\phi\right] = \frac{1}{m}\left[1 + 3\sin^2\phi\right] \tag{11.17}$$

$$C = D^{(\pm)}M^{-1}G^{(\pm)^T} = G^{(\pm)}M^{-1}D^{(\pm)^T} = -\frac{l^2}{\mathcal{J}_0}\sin\phi\cos\phi = -\frac{3}{m}\sin\phi\cos\phi, \tag{11.18}$$

where the final expression on each line corresponds to the specific case where $\mathcal{J}_0 = ml^2/3$. The configuration discussed in [34, 259, 232] corresponds to a simple case where $g^{(-)}(q) \geq 0$ and imposing Coulomb friction at that contact point. This is illustrated in Figure 11.1.

11.3 Equations of motion in acceleration form

Let us assume that there is a contact point so that $g^{(-)}(q) = 0$, which implies that $\phi \in [0, \pi]$. We drop the $(-)$ superscript form now on to simplify the notation. In fact, most of the results derived here carry over to the analysis of the other contact point with $g^{(+)}(q) = 0$, as expected from symmetry.

When the contact $g(q) = 0$ is active, there is net generalized normal force $f_n = G^T\nu$ where $\nu \in \mathbb{R}_+$, and $f_n \in \mathbb{R}^3$. This is the net effect—including the torque—of the force $\bar{f}_n = (0, \nu)^T$ applied at point $p^{(\pm)}$, i.e., the generalized force f_n includes both the force and the torque. The magnitude of the contact force is thus $\|\bar{f}_n\| = \nu$. Likewise, the generalized tangential force produced by friction at the contact point is $f_t = D^T\dot{\beta}$. Recall that the ghost velocities are used for nonholonomic constraint forces, as shown in Section 3.14.5 and the description of the friction model in Section 10.11.2. The magnitude of the tangent force applied to extremity point is thus p is $\|\bar{f}_t\| = \dot{\beta}$. The Coulomb relation thus reads

$$|\dot{\beta}| \leq \mu\nu, \tag{11.19}$$

where μ is the friction coefficient. For simplicity, the same coefficient is used for both static and kinetic friction here.

Assuming the only external influence is that of a constant gravitational field producing the force $f_g = M\hat{g}$ with $\hat{g} = (0, g, 0)^T$, and $g > 0$ is the gravitational

acceleration, the equations of motion read

$$\begin{aligned} M\ddot{q} &= M\hat{g} + G^T\nu + D^T\dot{\beta} \\ D\ddot{q} + c(q,\dot{q}) &= \dot{\sigma} \\ 0 \le g(q) \ &\perp \ \nu \ge 0 \\ 0 \le \mu\nu - \dot{\beta} \ &\perp \ \sigma \le 0 \\ 0 \le \mu\nu + \dot{\beta} \ &\perp \ \sigma \ge 0. \end{aligned} \tag{11.20}$$

These equations are usually simplified by explicitly distinguishing two cases, namely, the stiction case for which $D\dot{q} = 0$ and the sliding case for which $\dot{\beta} = \mu\nu$, and therefore $D\dot{q} \neq 0$.

Explicit equations are now derived for each of the cases and the conditions for transition between them are characterized. The case of stiction is analyzed in section 11.3.1 and the case of sliding motion is covered in section 11.3.2. The nature of the Painlevé paradox is then analyzed in Section 11.4 which also contains a discussion of how this is resolved using the techniques of Chapter 10, as well as a numerical example.

11.3.1 Stiction case

Assuming that $D\dot{q} = 0$, we keep the equation $D(q)\ddot{q} + \dot{D}(q)\dot{q} = 0$ satisfied and this leads to the following MLCP

$$\begin{bmatrix} M & -D^T & -G^T \\ D & 0 & 0 \\ G & 0 & 0 \end{bmatrix} \begin{bmatrix} \ddot{q} \\ \dot{\beta} \\ \nu \end{bmatrix} + \begin{bmatrix} -f \\ c(q,\dot{q}) \\ b(q,\dot{q}) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \rho \end{bmatrix} \tag{11.21}$$
$$0 \le \nu \ \perp \ \rho \ge 0,$$

where f is the generalized external force applied on the system, including the pull of gravity and any other force. Eliminating the variable $\ddot{q}$ by taking a Schur complement reduces this system to a LCP involving only the variable ν

$$\begin{aligned} K\nu + r &= \rho \\ 0 \le \nu \ &\perp \ \rho \ge 0, \end{aligned} \tag{11.22}$$

where the 1×1 (a scalar) matrix K is the Schur complement

$$K = \left[A - C^T B^{-1} C\right] = \frac{1 + ml^2/\mathcal{J}_0}{m} \frac{1}{1 + 3\sin^2\phi} = \frac{4}{m} \frac{1}{1 + 3\sin^2\phi} \ge \frac{1}{m} > 0, \tag{11.23}$$

using the definitions of A from (11.16), B from (11.17), and C from (11.18), respectively, and using the definition of (11.2) for $\mathcal{J}_0$. The one-dimensional

vector r, also a scalar

$$\begin{aligned} r &= b(q,\dot{q}) + (G - CB^{-1}D)M^{-1}f - CB^{-1}c(q,\dot{q}) \\ &= g\left(\mp\left(\frac{l\dot{\phi}^2}{g}\right)\frac{1+\alpha}{1+\alpha\sin^2\phi}\sin\phi - 1\right) \\ &= -g\left(\left(\frac{l\dot{\phi}^2}{g}\right)\frac{4\sin\phi}{1+3\sin^2\phi} + 1\right) \end{aligned} \tag{11.24}$$

where use was made that $f = (f_1, f_2, f_3)^T = (0, -mg, 0)$. The second line in (11.24) elicits the general case and the third line specializes to $\alpha = 3$ and a contact at $p^{(-)}$.

The full solution is then recovered using

$$\begin{aligned} \dot{\beta} &= -B^{-1}C\nu - B^{-1}DM^{-1}f - B^{-1}c^{(\pm)} \\ &= \frac{\cos\phi}{1+3\sin^2\phi}\left(\nu\sin\phi \pm ml\dot{\phi}^2\right), \\ \ddot{q} &= M^{-1}f + M^{-1}G^T\nu + M^{-1}D^T\dot{\beta}, \end{aligned} \tag{11.25}$$

$$\begin{aligned} \nu &= \max(0, -r/K) \\ \dot{\beta} &= \mathrm{mid}(-\mu\nu, \frac{\cos\phi}{1+3\sin^2\phi}\left(\nu\sin\phi \pm ml\dot{\phi}^2\right), \mu\nu), \end{aligned} \tag{11.26}$$

where the mid function is defined thus

$$\mathrm{mid}(x,y,z) = \begin{cases} x & \text{if } y \le x \le z \text{ or } z \le x \le y \\ y & \text{if } x \le y \le z \text{ or } z \le y \le z \\ z & \text{if } x \le z \le y \text{ or } y \le z \le x. \end{cases} \tag{11.27}$$

Since the 1×1 matrix K is in fact a positive scalar, the value of the normal force is positive whenever $r < 0$

Of course, if it is found that the magnitude of the tangential force is outside of the allowed range, $|\dot{\beta}| \ge \mu\nu$, then, this solution is not valid and a different problem must be solved as described next.

11.3.2 Sliding case

Once the magnitude of the tangential force reaches the maximum, i.e., $|\beta| = \mu\nu$, the contact point switches to *sliding mode* and there is only one variable to determine, namely, the value of the normal force ν. Now, the net constraint force is easily computed as

$$f_c = G^T\nu + D^T\dot{\beta} = \left[G^T - \mu\,\mathrm{sgn}(D\dot{q})D^T\right]\nu, \tag{11.28}$$

which means that the MLCP to solve is now:

$$\begin{bmatrix} M & -G^T + \mu\,\mathrm{sgn}(D\dot{q})D^T \\ G & 0 \end{bmatrix} \begin{bmatrix} \ddot{q} \\ \nu \end{bmatrix} + \begin{bmatrix} -f \\ -\dot{G}\dot{q} \end{bmatrix} = \begin{bmatrix} 0 \\ \rho \end{bmatrix} \tag{11.29}$$
$$0 \leq \nu \ \perp \ \rho \geq 0.$$

Eliminating the variable $\ddot{q}$, the pure LCP reduction is found to be

$$\begin{aligned} S\nu + s &= \rho \\ 0 \leq \nu \ &\perp \ \rho \geq 0, \end{aligned} \tag{11.30}$$

with:

$$\begin{aligned} S &= GM^{-1}G^T - \mu\,\mathrm{sgn}(\dot{q})GM^{-1}D^T \\ &= \frac{1}{m}\left[1 + 3\cos^2\phi + 3\mu\,\mathrm{sgn}(D\dot{q})\cos\phi\sin\phi\right] \\ s &= GM^{-1}f + \dot{D}\dot{q} = -g + c^{(\pm)} = -g \mp l\dot{\phi}^2\cos\phi. \end{aligned} \tag{11.31}$$

This LCP has a unique solution as long as $S \geq 0$ but, however, there are infinitely many solutions for $S = 0, s \geq 0$, and no solution for $S < 0, s < 0$. This can happen, for instance, with the configuration shown in Figure 11.1, where $\cos\phi < 0$, $\mathrm{sgn}(D(q)\dot{q}) = 1$, choosing a large enough friction coefficient, and an angle ϕ such that

$$\begin{aligned} \mu &\geq \arg\min_{\phi} \frac{1 + 3\cos^2\phi}{3\sin\phi\cos\phi} = \frac{4}{3}, \\ \phi &\in \{\phi \mid \frac{1 + 3\cos^2\phi}{3\sin\phi\cos\phi} < \mu\}. \end{aligned} \tag{11.32}$$

To recover a solution, one would need to annihilate the *finite* sliding velocity $D\dot{q} \to 0$ instantly or let the contact separate.

This is illustrated in Figure 11.2 with the same parameters that we have used so far in the analysis. This figure is similar to that found in [232]. Discussion for this is provided in Section 11.4.

11.4 The paradox and its resolution

As observed in [232], for friction coefficient $\mu = 5/3$—which corresponds to the "path" marker in Figure 11.2—, a rod contacting the plane at $p^{(-)}$ with angle $\phi = 3\pi/4 + \delta, \delta \in (0, \pi/4)$, sliding along towards the right is in a feasible region so that LCP (11.31) has a unique solution with $\nu = -s/S, \beta = -\mu\nu$. Assuming now that the rotational velocity is negative but not too large, i.e., $\dot{\phi} < 0$, then the inclination angle ϕ will increase to reach $\phi = 3\pi/4$ at which point the LCP defined in (11.31) has no finite solution. This will happen in finite time if the tangential velocity $v_t > 0$ is large enough initially. The value of $\mu = 5/3$ might seem unnaturally high but the paradox can also be reproduced for values of friction $\mu \leq 1$ if one chooses a different mass distribution for the rod to yield a

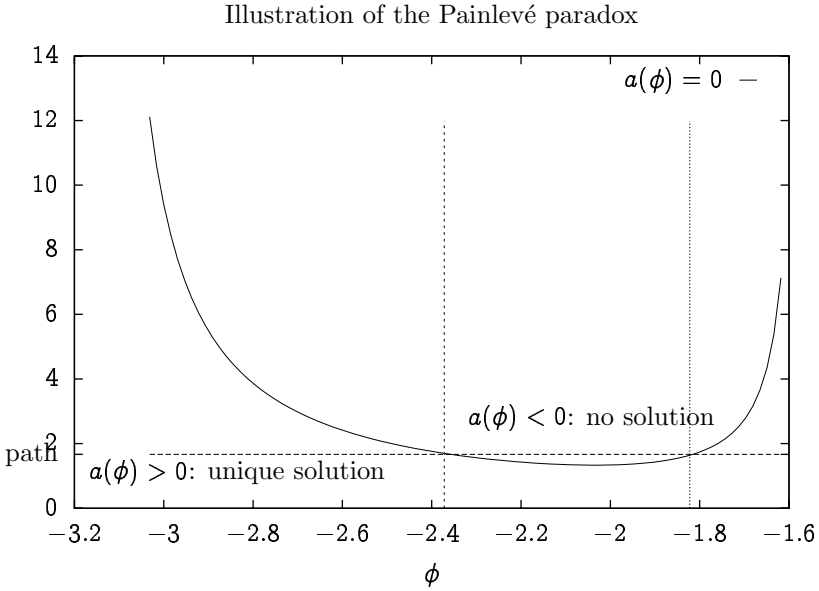

Figure 11.2: The paradoxical region for the problem of a planar rod in sliding dry frictional contact.

lower value of $\mathcal{J}_0$. For instance, the paradox can be observed for $\mathcal{J}_0 = ml^2/32$ and $\mu = 1$ as reported by Stewart [260].

The formulation of the contact problem given in (10.75) is however always solvable and thus, no paradox is ever observed. To see this, a few frames of an animation driven by the SPOOK stepper using the new Coulomb friction model—which reduces to the LCP formulation of Anitescu et al. [17, 19] for the two dimensional case—are presented in Figure 11.3. The friction coefficient is chosen to be $\mu = 5/3$ to conform with the treatment in Pfeiffer and Glocker [232]. Since they could not simulate this problem with their methods, experimentation produced the following set of initial conditions. The elevation angle starts at $\phi = 5\pi/6$ and the angular velocity is $\dot{\phi} = -2$. This configuration puts the point $p^{(-)}$ in contact with the plane. The linear velocity is chosen as $v = (1.7, l_0\dot{\phi}\cos\phi)$ so the contact velocity at $p^{(-)}$ is zero in the vertical direction and 0.7 in the horizontal direction. Using a fairly large time step of $h = 1/60 \approx 0.01667$ and simulating for 120 steps, taking pictures every 20 steps, yields the sequence of Figure 11.3. The rod is pictured with a simple line. At each step in the simulation, the coefficients of the LCP formulation based on acceleration, which leads to the paradox, were compute to see if the current configuration would be soluble or not. As predicted by Pfeiffer and Glocker, the configuration is feasible initially as the rod slides to the right and raises in altitude for a short while, then becomes infeasible while it is near vertical and the contact velocity comes to a halt, to become feasible again once the rod starts to fall back toward the left. The important point to note here is that the discrete time-stepping scheme

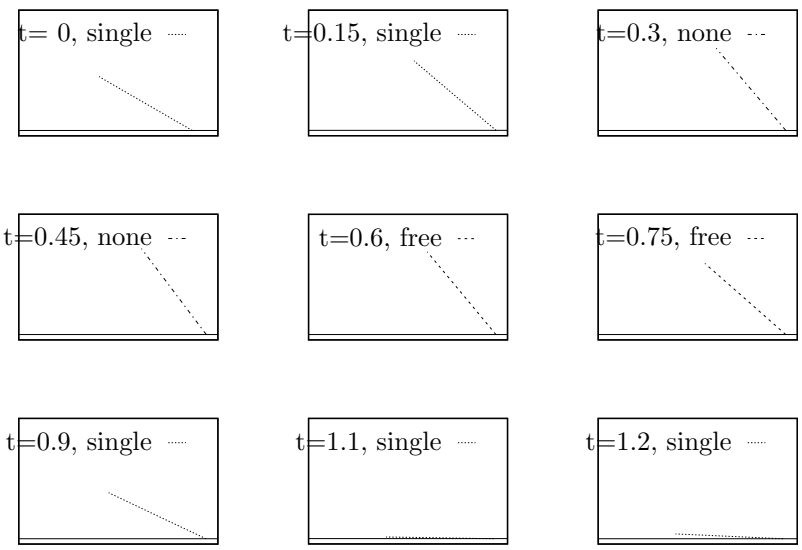

Figure 11.3: Numerical illustration of the Painlevé paradox. Each frame is a snapshot of the simulation, evenly spaced in time. The key on each subfigure indicates the time and whether or not there is a classical solution for the given configuration.

is always solvable and steps right over the problematic cases.

Part of the reason for the paradox is the acceleration formulation. Indeed, the principle of least action leading to the differential equations of motions hinges on the assumption that the trajectory $q(t)$ is twice continuously differentiable. But friction is a non-smooth process which leads to impacts, i.e., sudden changes in $\dot{q}$ at isolated times t_k, at which points the acceleration $\ddot{q}$ is not well defined.

However, in approximating the trajectory with discrete samples q_k, we have made no assumption on the velocity and the resulting discrete stepping equations are well behaved and uniquely defined everywhere. In other words, the fact that less smoothness is assumed in the derivation of the discrete stepping equations from the variational principle removes potential singularity and non-uniqueness.

Note also that the present formulation does not require a formulation of the equations of motion in terms of differential inclusions but enjoys the very same benefits that were demonstrated by Stewart [259].

11.5 End notes

The configuration of [108] corresponds to fixing $p^{(+)}(q) = 0$ and attaching the other end $p^{(-)}(q)$ on a wire that runs parallel to the ground. This removes $\dot{\phi}$ from the equations.

For the simpler one-dimensional case of a point particle with coordinate $q(t)$, subject to the friction force $\nu\mu\,\mathtt{sgn}(\dot{q})$, there is a simpler paradox in that the zero velocity mode $\dot{q} = 0$ is a *trap*. For this case, the differential equations of motion can be processed correctly if one interprets the signum function $\mathtt{sgn}(0)$ correctly, i.e., assigning it a value $\mathtt{sgn}(0) \in [-1, 1]$ which makes the problem consistent. This is done in [113] and in [258], for instance. The one-dimensional example does contain discontinuities but these are mild and limited to $\dot{q} = 0$.

By contrast, the Painlevé paradox concerns the case where a finite sliding velocity leads to a well behaved solution at one instant and nonexistence at the next, if the elevation angle happens to be on the boundary of the forbidden curve shown in Figure 11.2.

12 Rigid Bodies I: Fundamentals

The kinematic analysis of the motion of a rigid body is presented. In Section 12.1, the body is first considered to be composed of point particles which are constrained in their motion so that all pairwise distances between them are preserved. This results in identifying the configuration manifold of rigid body motion as the product $Q = \mathbb{R}^3 \times SO(3)$, where $SO(3)$ is the Lie group of special orthogonal transformations in three dimensions. The form of the kinetic energy term then provided in Section 12.2 and special attention is paid to the configuration dependent mass matrix in Section 12.3. Two-dimensional rigid bodies are then described in Section 12.4 and general discussion is provided in Section 12.5.

12.1 Basic motion of a rigid aggregate

Following a strategy credited to Euler, as mentioned in [105], consider a finite collection of point particles. Each such particle has a constant mass, $m^{(i)}$, and a time-dependent position, $x^{(i)}(t) \in \mathbb{R}^3$. For such a collection of points, rigidity means that the Euclidean distance between any two points is preserved by the motion so that $d^{(i,j)}(t) = \|x^{(j)}(t) - x^{(i)}(t)\| = d^{(i,j)}(0)$, where $\|\cdot\|$ is the Euclidean norm. This definition implies that the motion described by the points is an isometry of $\mathbb{R}^3$ with respect to the Euclidean norm.

According to the Mazur-Ulam theorem [198], "every bijective isometry $f : E \mapsto F$ between normed spaces is affine" (as cited in [270]). Therefore, the most general motion of a collection of rigid points is the affine transformation

$$x^{(i)}(t) = x(t) + R(t)x^{(i)}(0), \tag{12.1}$$

where $x(t) \in \mathbb{R}^3$ is some reference point, and the 3×3 matrix $R(t)$ is orthonormal, so that $R^T R = I_3$. From this definition, the distance between any two point particles i and j at time t is

$$\begin{aligned} d^{(i,j)}(t) &= \|x^{(i)}(t) - x^{(j)}(t)\| = \|R(t)(x^{(i)}(0) - x^{(j)}(0))\| \\ &= d^{(i,j)}(0), \end{aligned} \tag{12.2}$$

as desired, provided $R^T R = I_3$. Assuming the motion is continuous near the origin at $t = 0$ implies that $\det(R) = +1$. Thus, R is a proper rotation and so $R \in SO(3)$, where $SO(3)$ is the Lie group of special orthogonal transformations in three dimensions. An introduction to Lie groups and $SO(3)$ in particular is found in [193] for instance. The configuration manifold of rigid body motion is

thus $Q = \mathbb{R}^3 \times SO(3)$, in contrast with previous examples in this thesis where the configuration manifold was usually $\mathbb{R}^n$.

Define the *center of mass*, $x^{(0)}$ and the *center of mass coordinates* of each particle, $q^{(i)}$, as follows

$$m = \sum_i m^{(i)}, \quad x^{(0)} = m^{-1} \sum_i m^{(i)} x^{(i)}, \text{ and } \quad q^{(i)} = x^{(i)} - x^{(0)}. \tag{12.3}$$

Taken together, these definitions produce the identity $\sum_i m^{(i)} q^{(i)} = 0$. Assuming that the point particles move according to an isometry as in (12.1), the time evolution of the vectors $q^{(i)}(t)$ satisfies

$$q^{(i)}(t) = x(t) + R(t) q^{(i)}(0), \tag{12.4}$$

but since $\sum_i m^{(i)} q^{(i)} = 0$, for any time t, it follows that $x(t) = 0$. Thus the vectors $x^{(i)}(t)$ have the following form:

$$x^{(i)}(t) = x^{(0)}(t) + R(t) q^{(i)}(0). \tag{12.5}$$

This defines the *body* or *material* coordinates $q^{(i)}(0)$, which are fixed in the *body frame* of reference. Any of the point particles has *inertial frame* coordinates given by (12.5).

Since the kinematics of any of the point particles in the aggregate is related ultimately to that of $R(t)$, the time derivative of the latter is needed to compute velocities. Matrix $R(t)$ is orthogonal, $R(t)R^T(t) = 1$ at all times t and thus

$$0 = \dot{R}R^T + R\dot{R}^T = \dot{R}R^T + \left(\dot{R}R^T\right)^T, \tag{12.6}$$

which means the 3×3 matrix $\widehat{\omega} = \dot{R}R^T$ is antisymmetric. Any such matrix can be written as

$$\widehat{\omega} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}, \tag{12.7}$$

where $a \in \mathbb{R}^3$. In other word, any vector $x \in \mathbb{R}^3$ can be promoted to the corresponding 3×3 antisymmetric matrix $\widehat{x}$. With this notation, the time rate of change of the rotation matrix R is then

$$\dot{R} = \widehat{\omega} R. \tag{12.8}$$

This defines the *angular velocity* vector $\omega \in \mathbb{R}^3$, which, as seen in Chapter 13, is an axial vector defined in the inertial frame. In consequence, any point particle in the aggregate with position $x^{(i)}(t)$ defined in (12.5) has velocity

$$v^{(i)}(t) = \dot{x}^{(i)}(t) = \dot{x}^{(0)}(t) + \widehat{\omega} q^{(i)}(t). \tag{12.9}$$

Since the velocity of any of the point particle is linearly related to that of the center of mass as well as the angular velocity of the aggregate, it is possible to express the total kinetic energy in terms of $\dot{x}^{(0)}(t)$ and $\omega(t)$, the topic of the next section.

12.2 Kinetic energy

A simplifying feature of the motion of a rigid aggregate is the strict independence of the kinetic energy of linear motion of the center of mass and that of rotations about the center of mass. To see this, first recall that the kinetic energy of a point mass is simply $T^{(i)} = \frac{1}{2} m^{(i)} \|\dot{x}^{(i)}\|^2$, i.e., half the mass times the squared magnitude of velocity, and that kinetic energy is additive so that $T = \sum_i T^{(i)}$. Consider the squared magnitude of velocity (12.9) for one of the point mass in our aggregate

$$\|\dot{x}^{(i)}\| = \|\dot{x}^{(0)} + \widehat{\omega} q^{(i)}\|^2 = \|\dot{x}^{(0)}\|^2 + 2\dot{x}^{(0)^T} \widehat{\omega} q^{(i)} + \left(\widehat{\omega} q^{(i)}\right)^T \left(\widehat{\omega} q^{(i)}\right). \quad (12.10)$$

The first two terms are easily summed to yield

$$\sum_i m^{(i)} \|\dot{x}^{(0)}\|^2 = m \|\dot{x}^{(0)}\|^2, \text{ and} \quad (12.11)$$

$$\sum_i m^{(i)} 2\dot{x}^{(0)^T} \widehat{\omega} q^{(i)} = 2\dot{x}^{(0)^T} \widehat{\omega} \sum_i m^{(i)} q^{(i)} = 0. \quad (12.12)$$

For the last term, we use the well known identity $\widehat{x}y = -\widehat{y}x$ and $\widehat{x}^T = -\widehat{x}$ (this is proven as Lemma 13.3 below) to arrive at the sum

$$\sum_i m^{(i)} \left(\widehat{\omega} q^{(i)}\right)^T \left(\widehat{\omega} q^{(i)}\right) = -\omega^T \left(\sum_i m^{(i)} \widehat{q}^{(i)} \widehat{q}^{(i)} \right) \omega = \omega^T \mathcal{I} \omega, \quad (12.13)$$

where $\mathcal{I}$ is the inertia tensor defined as

$$\mathcal{I} = -\sum_i m^{(i)} \widehat{q}^{(i)} \widehat{q}^{(i)}. \quad (12.14)$$

Defining the reference inertia tensor $\mathcal{I}_0 = \mathcal{I}(0)$, and then using the identity $\widehat{Rx} = R^T \widehat{x} R$ from Lemma 13.5 below, the inertia tensor has the form

$$\mathcal{I} = -R^T \left(\sum_i m^{(i)} \widehat{q}^{(i)}(0) \widehat{q}^{(i)}(0) \right) R = R^T \mathcal{I}_0 R. \quad (12.15)$$

Collecting the results and summing over all the particles in the aggregates yields the kinetic energy

$$T = T_{\text{trans}} + T_{\text{rot}} = \frac{1}{2} m \|\dot{x}^{(0)}\| + \frac{1}{2} \omega^T \mathcal{I} \omega, \quad (12.16)$$

which demonstrates that translational and rotational parts of the kinetic energy are decoupled.

As an alternative derivation of the kinetic energy of rotation, following [178], first note that for any two vectors $u, v \in \mathbb{R}^3$, the following identity holds

$$u^T v = \sum_{i=1}^{3} u_i v_i = \text{tr}(uv^T) = \text{tr}\left(\begin{bmatrix} u_1 v_1 & u_1 v_2 & u_1 v_3 \\ u_2 v_1 & u_2 v_2 & u_2 v_3 \\ u_3 v_1 & u_3 v_2 & u_3 v_3 \end{bmatrix} \right). \quad (12.17)$$

Considering the terms on the left hand side of (12.13) and noting that

$$\widehat{\omega} q^{(i)}(t) = \widehat{\omega} R(t) q^{(i)}(0) = \dot{R} q^{(i)}(0), \tag{12.18}$$

the kinetic energy can be expressed in terms of the time rate of change of the rotation matrix R itself

$$T_{\text{rot}} = \frac{1}{2} \operatorname{tr}(\dot{R} \mathcal{K}_0 \dot{R}^T), \tag{12.19}$$

with the definition

$$\mathcal{K}_0 = \sum_i m^{(i)} q^{(i)}(0) q^{(i)}(0)^T. \tag{12.20}$$

Using the identity $\widehat{x}\widehat{x} = xx^T - x^T x I_3$ proved in Lemma 13.4, the following identity relates $\mathcal{I}_0$ and $\mathcal{K}_0$

$$\mathcal{I}_0 = \left(\sum_i m^{(i)} \| q^{(i)}(0) \|^2 \right) I_3 - \mathcal{K}_0. \tag{12.21}$$

Definition (12.19) for the kinetic energy was used in the first variational formulation of the discrete rigid body equations in [208]. This makes the nine coefficients of the orthonormal matrix R the configuration variables which must be subjected to the six constraint equations contained in the requirement that $RR^T = I_3$. A different strategy is adopted in Chapter 15 which relies on quaternion algebra instead, and this motivates the content of Chapter 13.

12.3 The inertia tensor

The inertia tensor is the first instance of a configuration dependent mass matrix. Some of its properties are now analyzed.

The result of Lemma 13.4 states that given any two vector $x, y \in \mathbb{R}^3$, then $\widehat{x}\widehat{y} = -x^T y I_3 + y x^T$. It follows that the inertia tensor can be expressed as

$$\mathcal{I} = \sum_i m^{(i)} \left(\| q^{(i)} \| I_3 - q^{(i)} q^{(i)T} \right), \tag{12.22}$$

from which it is clear that $\mathcal{I}$ is symmetric. The same inertia tensor $\mathcal{I}_0$ is also positive definite provided the rigid aggregate contains at least three point masses which are not collinear. To see this, take a vector $x \in \mathbb{R}^3$ and evaluate the product $x^T \mathcal{I} x$

$$x^T \mathcal{I} x = \sum_i m^{(i)} \left(\|x\| \| q^{(i)} \| - (x^T q^{(i)})^2 \right). \tag{12.23}$$

The term in parenthesis is non-negative from the Cauchy Schwartz identity. This term vanishes if and only if $x = \lambda q^{(i)}$ for all i, which is not possible if the points masses are not collinear.

Now, considering a rigid body in three dimensions occupying a finite volume, it must be possible to find sample points $q^{(i)}(0)$ which are not all collinear or coplanar. Otherwise, the body would have zero volume. Thus, the inertia tensor

of a rigid body with finite volume is symmetric and positive definite. Though it can be interesting for certain applications to consider linear and planar bodies, this is not considered further here.

Since the inertia tensor $\mathcal{I}$ of a rigid body with finite volume is symmetric positive definite, and it can therefore be diagonalized as

$$\mathcal{I}_0 = \iota^{(1)} u u^T + \iota^{(2)} v v^T + \iota^{(3)} w w^T, \tag{12.24}$$

where the orthonormal eigenvectors u, v and w are called the *principal axes* of the rigid bodies, and the eigenvalues $\iota^{(i)}, i = 1, 2, 3$ are called the principal inertiae.

12.4 The two-dimensional rigid body

It is often useful to consider two-dimensional examples to illustrate various aspects of the theory of multibody systems. The analysis of Sections 12.1 and 12.2 is thus repeated for two dimensions.

The observations made in Section 12.1 still apply though now, the rigid transformation matrix R is a member of $SO(2)$ which is a one-dimensional Lie group with representation

$$R(\phi) = \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix}. \tag{12.25}$$

For any $\phi \in \mathbb{R}$, matrix $R(\phi)$ is orthonormal and has positive determinant. The time derivative is much simpler here

$$\begin{aligned} \dot{R}(\phi) &= \dot{\phi} \begin{bmatrix} -\sin(\phi) & -\cos(\phi) \\ \cos(\phi) & -\sin(\phi) \end{bmatrix} \\ &= \dot{\phi} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} -\sin(\phi) & -\cos(\phi) \\ \cos(\phi) & -\sin(\phi) \end{bmatrix} \\ &= \dot{\phi} J_1 R(\phi), \text{ with:} \\ J_1 &= \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \end{aligned} \tag{12.26}$$

where matrix J_1 is the infinitesimal generator for the Lie group $SO(2)$.

With this definition, the velocity of one of the points in the aggregate is now

$$v^{(i)}(t) = \dot{x}^{(i)}(t) = \dot{x}^{(0)}(t) + \dot{\phi} J_1 q^{(i)}(t), \tag{12.27}$$

and after a simple computation, the kinetic energy derived in Section 12.2 now reads:

$$T = \frac{1}{2} M \left\| \dot{x}^{(0)}(t) \right\|^2 + \frac{\dot{\phi}^2}{2} \mathcal{J}_0, \tag{12.28}$$

where the inertia *scalar* was introduced as:

$$\mathcal{J}_0 = \sum_i m^{(i)} \left\| q^{(i)} \right\|^2. \tag{12.29}$$

The scalar nature of the angular degrees of freedom here greatly simplifies all the analysis of two-dimensional rigid bodies as compared to their three dimensional counterparts. Note in particular that the inertia tensor matrix is *constant*, i.e., independent of the configuration of the rigid body. This contrasts the three-dimensional case where the inertia tensor is configuration dependent as shown in (12.14).

12.5 End notes

The configuration manifold of three-dimensional rigid bodies is $\mathbb{R}^3 \times SO(3)$ and this warrants and analysis of $SO(3)$ to find suitable parametrization of the kinematics variables, since these are no longer simple n-dimensional real vectors.

The kinetic energy of a three-dimensional rigid body decouples into translational and angular terms which involve the total mass as well as the inertia tensor, the latter being configuration dependent. The consequences of this on the motion of a free rigid body are discussed in Chapter 15.

13 Rigid Bodies II: Kinematics and the Quaternion Algebra

As shown in Chapter 12, the rigid body configuration manifold is $\mathbb{R}^3 \times SO(3)$ and so the kinematics analysis requires an understanding of $SO(3)$ and its tangent bundle, wherein the velocities needed to express the kinetic energy terms lie. A representation for this using a combination of quaternion and matrix algebra is constructed in the present chapter. This representation is useful for the systematic analysis of both translational and rotational modes of rigid multibody system. The motivation to use the four-dimensional quaternions to parametrize the three dimensional Lie group $SO(3)$ is that $SO(3)$ is not a flat manifold and thus, cannot be represented globally with three parameters only without encountering singularities. The quaternion representation is singularity free, however.

The main objectives are an explicit parametrization of proper orthonormal 3×3 matrices $R(q)$ in terms of four real parameters $q_i, i = 0, 1, 2, 3$ restricted with $\sum_i q_i^2 = 1$, and an explicit, everywhere invertible relationship between the observed angular velocity vector $\omega \in \mathbb{R}^3$ and the time derivatives $\dot{q}_i$, so that the $q_i(t)$ can be integrated given known angular velocities $\omega(t)$. To make this theory useful for implementation purposes, care is taken to represent all operations involving the parameters q_i as matrix-vector products, treating $q = (q_0, q_1, q_2, q_3)^T$ either as a plain four-dimensional vector, or as a parameter defining the elements of classes of matrices.

After introducing some historical background in Section 13.1, a number of algebraic identities and simple theorems are derived in Section 13.2 before introducing the quaternion algebra in Section 13.3, and characterizing the three-dimensional vector subspace of it which corresponds to $\mathbb{R}^3$ in Section 13.4. The quaternion algebra is then represented in terms of real 4×4 matrices in Section 13.5 with both left and right isomorphisms, a feature which allows reordering factors in complicated expressions. The representation of isometries in $\mathbb{R}^3$ using the quaternion algebra is established in Section 13.6, with special emphasis on the matrix representation. The special form of the rotation matrices and factorization thereof is analyzed further in Section 13.8. After considering the differential calculus of quaternions in Section 13.9, with special emphasis on the consequences of the non-commutativity of the product operation, the connection between quaternion velocity and the angular velocity vector of proper 3×3 orthonormal matrices is established in Section 13.10. Discussion of other representations of $SO(3)$ is found in Section 13.11. A summary of the most important

formulae is provided in Section 13.12.

13.1 Historical background and motivation

Quaternions are an invention of William H. Hamilton in 1845. He was looking for a type of vectors for which a division operation could be defined, especially in three dimensions. The aim was to find a representation of rotations, similar to what is found in two dimensions where complex vectors of unit magnitudes can be used to perform planar rigid rotations using the multiplication operator of complex algebra.

Hamilton did not quite succeed with his program and in fact, he started a controversy which has lasted to this day [4]. As a consequence, the elements of quaternion algebra which are useful for the study of the group of rigid rotations in three dimensions are not often found in standard texts on mechanics and for this reason, they are presented here.

There is more to quaternion algebra than the representation of the rotation group suitable for parametrizing rigid body kinematics, but that is not covered here.

Contrary to what is customary in 3D graphics literature, the aim here is to describe quaternion algebra in terms of standard linear algebra operations. This is done so that it is straightforward to translate the relevant formulae into computer code for execution, even though it is not the most mathematically elegant presentation.

13.2 Preliminary algebraic identities

A number of elementary identities are useful in manipulating the algebra of quaternions, especially when using the 4×4 real matrix representation constructed in Section 13.5. The reader can skip this section on first reading and refer back to it as needed.

Lemma 13.1. *For any vector $x \in \mathbb{R}^3$, the vector $\widehat{x}x = 0$.*

Proof. Using the definition of $\widehat{x}$ and direct computation,

$$\widehat{x}x = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -x_3x_2 + x_2x_3 \\ x_3x_1 - x_1x_3 \\ -x_2x_1 + x_1x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{13.1}$$

□

The converse also holds namely, that

Lemma 13.2. *Given $x, w \in \mathbb{R}^3$, $\widehat{x}w = 0$ iff $w = \alpha x, \alpha\mathbb{R}$.*

Proof. If $w = \alpha x$, then, since $\widehat{x}$ is a linear operator, $\widehat{x}(\alpha x) = \alpha \widehat{x} x = 0$.

For the converse, assume without loss of generality that $\|x\| = 1$ and construct a right handed orthonormal basis x, y, z, $\|x\| = \|y\| = \|z\| = 1$ so that $\widehat{x} y = z$ and $\widehat{x} z = -y$. Decompose the vector $w \in \mathbb{R}^3$ according to this basis so that

$$w = \alpha x + \beta y + \gamma z, \alpha, \beta, \gamma \in \mathbb{R}, \tag{13.2}$$

and then, evaluate $\widehat{x} w = -\gamma y + \beta z$. Given the assumed linear independence of the basis vectors x, y, z, it follows that $\widehat{x} w = 0$ if and only if $\gamma = \beta = 0$ which means that $w = \alpha x$. □

Lemma 13.3. *Given two vectors $x, y \in \mathbb{R}^3$, then, $\widehat{x} y = -\widehat{y} x$.*

Proof. Using the definition of $\widehat{x}$ and computing both products directly,

$$\begin{aligned} \widehat{x} y &= \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} -x_3 y_2 + x_2 y_3 \\ x_3 y_1 - x_1 y_3 \\ -x_2 y_1 + x_1 y_2 \end{bmatrix}, \text{ and} \\ \widehat{y} x &= \begin{bmatrix} 0 & -y_3 & y_2 \\ y_3 & 0 & -y_1 \\ -y_2 & y_1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -y_3 x_2 + y_2 x_3 \\ y_3 x_1 - y_1 x_3 \\ -y_2 x_1 + y_1 x_3 \end{bmatrix}, \end{aligned} \tag{13.3}$$

and the result is found by inspection. □

Considering column vectors as $n \times 1$ matrices and using the standard multiplication rules, the outer product $x y^T$ can be formed for any two vectors $x, y \in \mathbb{R}^3$. Definition 13.1 provides the explicit form of the resulting rank 1 matrix.

Definition 13.1. *Given two vectors, $x, y \in \mathbb{R}^3$, the products $x y^T$ and $y^T x$ are defined as*

$$x y^T = \begin{bmatrix} x_1 y_1 & x_1 y_2 & x_1 y_3 \\ x_2 y_1 & x_2 y_2 & x_3 y_3 \\ x_3 y_1 & x_3 y_2 & x_2 y_3 \end{bmatrix}, \quad y x^T = \begin{bmatrix} y_1 x_1 & y_1 x_2 & y_1 x_3 \\ y_2 x_1 & y_2 x_2 & y_3 x_3 \\ y_3 x_1 & y_3 x_2 & y_2 x_3 \end{bmatrix}. \tag{13.4}$$

This notation is used for the next identity on products of the form $\widehat{x}\widehat{y}$.

Lemma 13.4. *Given two vectors $x, y \in \mathbb{R}^3$, the following identity holds: $\widehat{x}\widehat{y} = -x^T y I_3 + y x^T$.*

Proof. Using the definition and direct computations,

$$\begin{aligned} \widehat{x}\widehat{y} &= \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \begin{bmatrix} 0 & -y_3 & y_2 \\ y_3 & 0 & -y_1 \\ -y_2 & y_1 & 0 \end{bmatrix} \\ &= \begin{bmatrix} -(x_2 y_2 + x_3 y_3) & y_1 x_2 & y_1 x_3 \\ y_2 x_1 & -(x_1 y_1 + x_3 y_3) & y_2 x_3 \\ y_3 x_1 & y_3 x_2 & -(x_1 y_1 + x_2 y_2) \end{bmatrix} \\ &= -x^T y I_3 + y x^T. \end{aligned} \tag{13.5}$$

The last line is obtained by noting that each term on the diagonal is of the form $-x^T y + x_i y_i, i = 1, 2, 3$. □

The identification of an antisymmetric 3×3 matrix $A = -A^T$ with the vector a via $A = \widehat{a}$ is slightly misleading in view of Lemma 13.5 below. Indeed, transforming a coordinate system via a 3×3 orthonormal matrix R, the vector a is not transformed to $a \mapsto Ra$ like any other vector but rather, it is transformed to an axial equivalent, the exact orientation of which depends on the handedness $\det(R)$ of the orthogonal transformation R.

Lemma 13.5. *Given a vector $x \in \mathbb{R}^3$ and a 3×3 orthonormal matrix R, then, $R\widehat{x}R^T = \det(R)\widehat{Rx}$.*

Proof. Without loss of generality, assume that $\|x\| = 1$. Now, $\widehat{x}$ is matrix of size 3×3 and thus, it defines a linear transformation $\widehat{x} : \mathbb{R}^3 \mapsto \mathbb{R}^3$. Change the basis according to the orthonormal transformation R. The matrix representation of this linear operator then becomes $\bar{X} = R\widehat{x}R^T$, using the change of basis theorem and observing that $R^{-1} = R^T$.

Next, observe that matrix $\bar{X}$ is antisymmetric since

$$\bar{X}^T = \left(R\widehat{x}R^T\right)^T = R\widehat{x}^T R^T = -R\widehat{x}R^T = -\bar{X}^T, \tag{13.6}$$

which means that $\bar{X} = \widehat{a}$ for some vector $a \in \mathbb{R}^3$.

Observe however that for $w = \lambda Rx$, we have $\bar{X} = R\widehat{x}R^T(\lambda Rx) = \lambda R\widehat{x}x = 0$, which, using Lemma 13.2, that $w = \alpha a = \lambda Rx$. Combining these two relations,

$$\bar{X} = \widehat{a},\ a = \sigma Rx, \text{ for some scalar } \sigma \in \mathbb{R}. \tag{13.7}$$

Computing the product $\bar{X}\bar{X}$,

$$\begin{aligned} \bar{X}\bar{X} &= -\|a\|^2 I_3 + aa^T = -\sigma^2\|x\|^2 + \sigma^2 Rxx^T R^T \\ &= \sigma^2 R\left(\|x\|^2 I_3 + xx^T\right)R^T \\ &= \sigma^2 R\widehat{x}\widehat{x}R^T = \sigma^2 R\widehat{x}R^T R\widehat{x}R^T \\ &= \sigma^2 \bar{X}\bar{X}, \end{aligned} \tag{13.8}$$

which means that $\sigma^2 = 1$ and so $\sigma = \pm 1$.

To compute σ, consider the natural right handed basis with $x = (1, 0, 0)^T$, $y = (0, 1, 0)^T$ and $z = (0, 0, 1)^T$, and its transformation $u = Rx, v = Ry, w = Rz$. The vectors u, v, w are obviously the columns of matrix R. Given that for any 3×3 matrix A with columns $a, b, c \in \mathbb{R}^3$, the determinant is the triple product $\det(A) = c^T \widehat{a} b$,

$$z^T \widehat{x} y = 1 = \det \begin{bmatrix} u & v & w \end{bmatrix}, \tag{13.9}$$

and after transforming each vector, this implies

$$1 = z^T R^T R\widehat{x}R^T Ry = w(\sigma\widehat{u})v = \sigma \det \begin{bmatrix} x & y & z \end{bmatrix} = \sigma \det(R), \tag{13.10}$$

Therefore, $\sigma^{-1} = \sigma = \det(R)$, and the proof is complete. □

Lemma 13.6. *Given any two vectors $x, y \in \mathbb{R}^3$, then, $\widehat{\widehat{x}y} = yx^T - xy^T$.*

Proof. First, write $v = \widehat{x}y$ so that

$$v = \begin{bmatrix} x_3y_2 - x_2y_3 \\ -x_3y_1 + x_1y_3 \\ -x_2y_1 + x_1y_2 \end{bmatrix}, \tag{13.11}$$

and then build the matrix

$$\widehat{\widehat{x}y} = \widehat{v} = \begin{bmatrix} 0 & x_2y_1 - x_1y_2 & x_3y_1 - x_1y_3 \\ x_1y_2 - x_2y_1 & 0 & x_3y_2 - x_2y_3 \\ x_1y_3 - x_3y_1 & x_2y_3 - x_3y_2 & 0 \end{bmatrix}. \tag{13.12}$$

The result is found by inspection using the definitions in (13.4). □

Since $\widehat{x}x = 0$, the matrix $\widehat{x}$ is singular and therefore, the determinant vanishes so $\det(\widehat{x}) = 0$. Likewise, matrices of the form xy^T are singular since all the columns are scalar multiples of the vector x. In fact, these latter matrices have rank one for any dimension though the former have rank 2. This rank deficiency allows to compute the determinants of matrices with related forms.

Lemma 13.7. *Given $x \in \mathbb{R}^3$ and $\lambda \in \mathbb{R}$, the matrix $A(\lambda) = \lambda I_3 + \widehat{x}$ has determinant $\det(A(\lambda)) = \lambda^3 + \lambda x^T x$.*

Proof. The determinant of $A(\lambda)$ is computed by expanding the minors down the first column

$$\begin{aligned} \det(\lambda I + \widehat{x}) &= \begin{vmatrix} \lambda & -x_3 & x_2 \\ x_3 & \lambda & -x_1 \\ -x_2 & x_1 & \lambda \end{vmatrix} \\ &= \lambda \begin{vmatrix} \lambda & -x_1 \\ x_1 & \lambda \end{vmatrix} - x_3 \begin{vmatrix} -x_3 & x_2 \\ x_1 & \lambda \end{vmatrix} - x_2 \begin{vmatrix} -x_3 & x_2 \\ \lambda & -x_1 \end{vmatrix} \\ &= \lambda(\lambda^2 + x_1^2) - x_3(-x_3\lambda - x_1x_2) - x_2(x_3x_1 - \lambda x_2) \\ &= \lambda^3 + \lambda(x_1^2 + x_2^2 + x_3^2) \\ &= \lambda^3 + \lambda x^T x. \end{aligned} \tag{13.13}$$

□

Lemma 13.8. *For any two vectors $x, y \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}$, x is a right eigenvector and y^T is a left eigenvector of the matrix $A(\lambda) = \lambda I_n + xy^T$, both with the same eigenvalue $\lambda + x^T y$.*

Proof. Direct computation shows that $A(\lambda)x = (\lambda + x^T y)x$ and $y^T A(\lambda) = y^T(\lambda + x^T y)$. □

For $x, y \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}$, the matrix $\lambda I_n + xy^T$ has non-zero determinant as computed in Lemma 13.9 below.

Lemma 13.9. *Given $\lambda \in \mathbb{R}$ and $x, y \in \mathbb{R}^n$, the matrix $A(\lambda) = \lambda I_n + xy^T$ has determinant $\det(A(\lambda)) = \lambda^{n-1}(\lambda + x^T y)$.*

Proof. We proceed by induction, first noting the result for $n = 2$ where

$$\begin{aligned} \det(\lambda I_2 + xy^T) &= \begin{vmatrix} \lambda + x_1 y_1 & x_1 y_2 \\ x_2 y_1 & \lambda + x_2 y_2 \end{vmatrix} \\ &= \lambda^2 + \lambda(x_1 y_1 + x_2 y_2) = \lambda^2 + \lambda x^T y. \end{aligned} \tag{13.14}$$

Next, partition the matrix $A(\lambda)$ as follows

$$A(\lambda) = \begin{bmatrix} \lambda I_{n-1} + \bar{x}\bar{y}^T & y_n \bar{x} \\ x_n \bar{y}^T & \lambda + x_n y_n \end{bmatrix}, \tag{13.15}$$

where the bar signifies projection by truncation of the last element, e.g., $\bar{u} = (u_1, u_2, \ldots, u_{n-1})^T, u \in \mathbb{R}^n$. Matrix $A(\lambda)$ can be factored as

$$A(\lambda) = BC = \begin{bmatrix} D & 0 \\ \bar{w}^T & \sigma \end{bmatrix} \begin{bmatrix} I_{n-1} & \bar{z} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} D & D\bar{z} \\ \bar{w}^T & \bar{w}^T \bar{z} + \sigma \end{bmatrix}, \tag{13.16}$$

with the definitions

$$\begin{aligned} D &= \lambda I_{n-1} + \bar{x}\bar{y}^T, \qquad \bar{z} = \frac{y_n}{\lambda + \bar{x}^T \bar{y}} \bar{x} \\ \bar{w} &= x_n \bar{y}, \qquad \text{and } \sigma = \frac{\lambda x^T y + \lambda^2}{\lambda + \bar{x}^T \bar{y}}. \end{aligned} \tag{13.17}$$

The result from of Lemma 13.8 was used in computing $\bar{z}$. Now, $\det(BC) = \det(B)\det(C)$ for any square matrices B, C, and since $\det(C) = 1$, because matrix C is upper triangular with unit diagonal,

$$\det(A(\lambda)) = \det(B(\lambda)) = \sigma \det(D(\lambda)). \tag{13.18}$$

Now, by the induction hypothesis, $\det(D(\lambda)) = \det(\lambda I_{n-1} + \bar{x}\bar{y}^T) = \lambda^{n-1} + \lambda^{n-2}\bar{x}^T\bar{y}$ and therefore, we find that

$$\det(A(\lambda)) = \frac{\lambda x^T y + \lambda^2}{\lambda + \bar{x}^T \bar{y}} \left(\lambda^{n-1} + \lambda^{n-2} \bar{x}^T \bar{y} \right) = \lambda^n + \lambda^{n-1} x^T y, \tag{13.19}$$

proving the result holds for any $n > 2$. □

Another useful identity concerns 4×4 real antisymmetric matrices of the fol-

lowing forms

$$A_+ = \begin{bmatrix} 0 & a_1 & a_2 & a_3 \\ -a_1 & 0 & -a_3 & a_2 \\ -a_2 & a_3 & 0 & -a_1 \\ -a_3 & -a_2 & a_1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & a^T \\ -a & \widehat{a} \end{bmatrix},$$

and (13.20)

$$A_- = \begin{bmatrix} 0 & a_1 & a_2 & a_3 \\ -a_1 & 0 & a_3 & -a_2 \\ -a_2 & -a_3 & 0 & a_1 \\ -a_3 & a_2 & -a_1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & a^T \\ -a & -\widehat{a} \end{bmatrix},$$

where $a \in \mathbb{R}^3$ is a real vector. The minimum polynomial of these matrices is computed explicitly in Lemma 13.10 below.

Lemma 13.10. *Given a real vector $a \in \mathbb{R}^3$, the antisymmetric matrices $A_\pm$ defined in* (13.20) *have the minimum polynomial:*

$$m(\lambda) = \lambda^2 + a^T a. \tag{13.21}$$

Proof. The computation is carried out explicitly only for A_+ since it is similar to that for A_-. Expanding the determinant $\det A_+ - \lambda I_4$ along the first column,

$$\begin{aligned} \det(A_+ - \lambda I_4) = -\lambda \det(\widehat{a} - \lambda I_3) + a_1 \det \begin{bmatrix} a_1 & a_2 & a_3 \\ a_3 & -\lambda & -a_1 \\ -a_2 & a_1 & -\lambda \end{bmatrix} \\ - a_2 \det \begin{bmatrix} a_1 & a_2 & a_3 \\ -\lambda & -a_3 & a_2 \\ -a_2 & a_1 & -\lambda \end{bmatrix} + a_3 \det \begin{bmatrix} a_1 & a_2 & a_3 \\ -\lambda & -a_3 & a_2 \\ a_3 & -\lambda & -a_1 \end{bmatrix}. \end{aligned} \tag{13.22}$$

After using Lemma 13.7 for the first term and simple algebraic manipulations,

$$\det(A_+ - \lambda I_4) = (\lambda^2 + a^T a)^2, \tag{13.23}$$

and therefore, the minimum polynomial is $m(\lambda) = \lambda^2 + \|a\|^2$. Repeating the computation for A_- produces the desired result. □

The Cayley Hamilton theorem can be used to compute any power of the matrices $A_\pm^n, n > 1$, by recursively substituting $A_\pm^2 = -\|a\|^2 I_4$, leading to Corollary 13.11 below.

Corollary 13.11. *Given a vector $a \in \mathbb{R}^3$ and the matrices $A_\pm$ defined above in* (13.20)*, the following identities hold*

$$\begin{aligned} A_\pm^2 &= -\|a\|^2 I_4, \\ A_\pm^{2n} &= (-)^n \|a\|^{2n}, \\ A_\pm^{2n+1} &= (-)^n \|a\|^{2n} A_\pm. \end{aligned} \tag{13.24}$$

Proof. The Cayley-Hamilton theorem states that any complex matrix A satisfies it's minimum polynomial equation. For matrices $A_\pm$, this is $m(\lambda) = \lambda^2 + \|a\|^2$, meaning that $A_\pm^2 = -\|a\|^2 I_4$. □

This can in turn be used to look at matrices of the form $B_\pm = a_0 I_4 + A_\pm$ since for these, the minimum polynomial is simply $n(\lambda) = (\lambda - a_0)^2 + \|a\|^2$. Agglomerate a_0 and $a \in \mathbb{R}^3$ into a four dimensional vector $b = [a_0, a_1, a_2, a_3]^T \in \mathbb{R}^4$. For unit b vector, the minimum polynomial simply reads: $\lambda^2 - 2b_0\lambda + 1$. And for this case, we can compute the powers of $B_\pm^n$ in terms of the Chebyschev polynomials of the second kind, $U_n(\tau)$ for some $\tau \in \mathbb{R}, |\tau| \leq 1$, defined as

$$U_n(\tau) = \frac{\sin([n+1]\phi)}{\sin\phi}, \text{ with the definition } \cos\phi = \tau, \tag{13.25}$$

as is done in [21], or for instance. These observations prove Lemma 13.12 below.

Lemma 13.12. *Given a unit vector $b = [b_0, b_1, b_2, b_3] \in \mathbb{R}^4$, the matrices*

$$B_+ = \begin{bmatrix} b_0 & b_1 & b_2 & b_3 \\ -b_1 & b_0 & -b_3 & b_2 \\ -b_2 & b_3 & b_0 & -b_1 \\ -b_3 & -b_2 & b_1 & b_0 \end{bmatrix}, \qquad B_- = \begin{bmatrix} b_0 & b_1 & b_2 & b_3 \\ -b_1 & b_0 & b_3 & -b_2 \\ -b_2 & -b_3 & b_0 & b_1 \\ -b_3 & b_2 & -b_1 & -b_0 \end{bmatrix} \tag{13.26}$$

satisfy:

$$B_\pm^n = U_{n-1}(b_0)B - U_{n-2}I_4, \tag{13.27}$$

for any integer $n \geq 2$.

Proof. From the Cayley-Hamilton theorem, $B_\pm^2 = 2b_0 B_\pm - I_4$. Proceeding by induction, assume for the moment that $B_\pm^n = p_{n-1}B_\pm + q_{n-2}I_4$ where p_n, q_n, are polynomials in b_0. From the definition of the Chebyschev polynomials [21], $U_0(x) = 1, U_1(x) = 2x$. Write $x = b_0$ and verify that $p_1 = U_1(b_0)$ and $q_0 = -U_0(b_0)$. Now, $B_\pm^{n+1}$ can be computed as

$$\begin{aligned} B_\pm^{n+1} &= p_{n-1}B_\pm^2 + q_{n-2}B_\pm = p_{n-1}\left[2b_0 B_\pm - I_4\right] + q_{n-2}B_\pm \\ &= (2b_0 p_{n-1} + q_{n-2})B - p_{n-1}I_4. \end{aligned} \tag{13.28}$$

From this, it follows that the recurrence relations are

$$\begin{aligned} q_n &= -p_{n-1}, \text{ and} \\ p_n &= 2b_0 p_{n-1} - p_{n-2}. \end{aligned} \tag{13.29}$$

This is precisely the recurrence relation which applies for the Chebyschev polynomials of the first and second kind. Since the result is valid for $n = 2$, the recurrences (13.29) validate the result by induction. □

If vector b is not normalized, simply set $x = b_0/\|b\|$ and recompute the recurrence to find the following.

Corollary 13.13. *For a non-zero real vector $b \in \mathbb{R}^4$ and the matrices $B_\pm$ considered in Lemma 13.12, setting $x = b_0/\|b\|$ yields*

$$B_\pm^n = \|b\|^{n-1} U_{n-1}(x) B_\pm - \|b\|^n U_{n-2}(x) I_4. \tag{13.30}$$

Proof. Since $\tilde{b} = (1/\|b\|)b$ and $\tilde{B}_\pm = (1/\|b\|)B_\pm$ satisfy all the conditions of Lemma 13.12, the result follows by direct substitution. □

When manipulating the rigid body equations of motion, some matrices of the form $\Sigma + A_+(a)$, where $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \sigma_3, \sigma_4), \sigma_i > 0, i = 1, 2, 3, 4$, is a positive definite diagonal matrix, and the antisymmetric 4×4 matrix $A_+(a)$ was defined in (13.20). Of particular interest are the eigenvalues of such matrices which are computed in Lemma 13.14 below.

Lemma 13.14. *Given an antisymmetric $n \times n$ matrix $A_+(a)$ as defined in* (13.20) *for some real vector $a \in \mathbb{R}^3$, and a diagonal positive definite 4×4 matrix $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \sigma_3, \sigma_4), \sigma_i > 0, i = 1, 2, 3, 4$, the four eigenvalues of the 4×4 matrix $C = \Sigma + A_+(a)$ are given by the following formulae*

$$\lambda_{\pm,\pm'} = 1 \pm' \frac{\sqrt{2}}{2} \sqrt{\|v + w\|(\|v + w\| \pm \|v - w\|) - 2v^T w}, \tag{13.31}$$

where the following definitions were used

$$\begin{aligned} \tilde{\Sigma} &= \sigma_1 \,\mathrm{diag}(\sigma_2, \sigma_3, \sigma_4), \\ v &= \tilde{\Sigma}^{1/2} a, \\ w &= \det(\Sigma)^{1/2} \tilde{\Sigma}^{-1/2} a. \end{aligned} \tag{13.32}$$

Proof. First note that $\Sigma^{-1/2} B \Sigma^{-1/2} = I_4 + \Sigma^{-1/2} A \Sigma^{-1/2}$. Applying this transformation to the determinant of $B - \lambda I_4$ yields

$$\begin{aligned} \det(B - \lambda I_4) &= \det(\Sigma) \det(\Sigma{-}1/2(B - \lambda I_4 \Sigma^{-1/2})) \\ &= \det(\Sigma) \det(\Sigma{-}1/2 A_+ \Sigma^{-1/2} - (\lambda - 1) I_4), \end{aligned} \tag{13.33}$$

and therefore, if μ is an eigenvalue of $\tilde{A}_+ = \Sigma{-}1/2 A_+ \Sigma^{-1/2}$, then, $\lambda = \mu + 1$ is an eigenvalue of matrix $B = \Sigma + A_+$. Now, a simple computation yields

$$\tilde{A} = \Sigma^{-1/2} A \Sigma{-}1/2 = \begin{bmatrix} 0 & v^T \\ v & \widehat{w} \end{bmatrix}, \tag{13.34}$$

with the definitions of (13.32). Repeating the computation of Lemma 13.10 with the vectors v, w instead produces the principal polynomial

$$p(\mu) = \mu^4 + \mu^2(\|v\|^2 + \|w\|^2) + (v^T w)^2, \tag{13.35}$$

and after some simple algebraic manipulations, the roots $\lambda_{\pm,\pm'}$ of (13.31) are found. Note that at least one of the roots of $p(\mu)$ in (13.35) is positive and so at least two of the eigenvalues of matrix B are real. □

The final type of special matrices we consider is the case of 3×3 real matrices of the forms $B = I_3 + \alpha A$ and $C = I_3 + \beta A^2$, where α and β are scalars and the 3×3 matrix A has either the form $H\widehat{x}$ or $\widehat{x}H$ for some real vector $x \in \mathbb{R}^3$ and a symmetric positive definite 3×3 matrix H. These matrices frequently appear in the analysis of discrete stepping equations for the free rigid body problem.

Observe first that since H is symmetric, it can be diagonalized as $H = QDQ^T$, where the real 3×3 matrix D is diagonal and real 3×3 matrix Q is orthogonal so that $Q^TQ = QQ^T = I_3$. Define the vector vector $y \in \mathbb{R}^3$, $y = Q^Tx$ and so according to Lemma 13.5, $\widehat{y} = Q^T\widehat{x}Q$. Using $A = H\widehat{x}$, it follows that $A = QDQ^T\widehat{x} = QD\widehat{y}Q^T$. Matrix A is thus the similarity transform of a 3×3 matrix of the form $G = D\widehat{y}$, where $D = \mathsf{diag}(d_1, d_2, d_3), d_i > 0, i = 1, 2, 3$, is diagonal and positive definite. The same holds if using $A = \widehat{x}H$ instead.

Under the same similarity transformation by an orthogonal 3×3 matrix Q, the scalar $\kappa = \mathsf{det}(H)x^TH^{-1}x$ is invariant since $\mathsf{det}(QDQ^T) = \mathsf{det}(D)$ and $x^TQD^{-1}Q^Tx = y^TD^{-1}y$. Combining these observations yields Lemma 13.15 below.

Lemma 13.15. *Given a symmetric, positive definite 3×3 matrix H, a vector $x \in \mathbb{R}^3$, and real scalars $\alpha, \beta \in \mathbb{R}$. Define the matrix $A = H\widehat{x}$ and the scalar $\kappa = \mathsf{det}(H)x^TH^{-1}x$. Then, the following two identities hold*

$$\left(I + \beta A^2\right)^{-1} = I - \frac{\beta}{1 + \beta\kappa}A^2, \text{ and} \tag{13.36}$$

$$(I + \alpha A)^{-1} = I - \frac{\alpha}{1 + \alpha^2\kappa}A + \frac{\alpha^2}{1 + \alpha^2\kappa}A^2. \tag{13.37}$$

Proof. Consider first a positive definite diagonal 3×3 matrix of the form $D = \mathsf{diag}(d_1, d_2, d_3)$ with $d_i > 0$, and a real vector $y \in \mathbb{R}^3$. The matrix $D\widehat{y}$ has the following characteristic polynomial, as found from a short computation

$$m(\lambda) = \mathsf{det}(D\widehat{y} - \lambda I_3) = \lambda\left(\lambda^2 + \mathsf{det}(D)y^TD^{-1}y\right). \tag{13.38}$$

From the Cayley-Hamilton theorem, it follows that matrix $M = D\widehat{y}$ is a matrix root of $m(\lambda)$ so that $M^3 = -\kappa M$, where $\kappa = \mathsf{det}(D)y^TD^{-1}y$. Since D is diagonal with positive entries, the inverse is $D^{-1} = \mathsf{diag}(1/d_1, 1/d_2, 1/d_3)$. Since the characteristic polynomial of a matrix is invariant under similarity transforms, and since the scalar κ is also invariant under similarity transforms, the result applies to any positive definite 3×3 matrix H and real vector $x \in \mathbb{R}^3$ using the observations preceding this lemma. □

This completes the set of identities needed to construct the matrix representation of the quaternion algebra.

13.3 Elementary quaternion algebra

The *quaternion division ring* $\mathbb{H}$, is a four-dimensional algebra over $\mathbb{R}^4$ in which non-zero elements have a multiplicative inverse.

To construct the ring $\mathbb{H}$, start with the elements $q \in \mathbb{R}^4$ with the standard vector addition, subtraction, and scalar multiplication operations. Therefore, given two elements $p, q \in \mathbb{H}$, define addition and scalar multiplication as follows

$$q = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}, \quad p = \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix}, \qquad q + \alpha p = \alpha p + q = \begin{bmatrix} q_0 + \alpha p_0 \\ q_1 + \alpha p_1 \\ q_2 + \alpha p_2 \\ q_3 + \alpha p_3 \end{bmatrix}, \text{ for any } \alpha \in \mathbb{R}. \tag{13.39}$$

The addition operation is commutative.

For the multiplication operation, consider four basic elements, $\mathtt{h}, \mathtt{i}, \mathtt{j}, \mathtt{k} \in \mathbb{H}$, and define the following multiplication rules

$$\begin{aligned} &\mathtt{h}^2 = \mathtt{h} = 1, \ \mathtt{hi} = \mathtt{i}, \ \mathtt{hj} = \mathtt{j}, \ \mathtt{hk} = \mathtt{k}, \\ &\mathtt{i}^2 = \mathtt{j}^2 = \mathtt{k}^2 = -\mathtt{h} = -1 \\ &\mathtt{ij} = -\mathtt{ji} = \mathtt{k}, \quad \mathtt{jk} = -\mathtt{kj} = \mathtt{i}, \quad \mathtt{ki} = -\mathtt{ik} = \mathtt{j}. \end{aligned} \tag{13.40}$$

These multiplication rules are summarized Table 13.1 below.

	h	i	j	k
h	h	i	j	k
i	i	−1	k	−j
j	j	−k	−1	i
k	k	j	−i	−1

Table 13.1: The quaternion algebra multiplication table.

Clearly, the element $\mathtt{h}$ is the multiplicative unit and it will sometimes be written as $\mathtt{h} = 1$ as there is little chance of confusion.

The vector representation of these elements is defined as:

$$\mathtt{h} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathtt{i} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathtt{j} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad \mathtt{k} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \tag{13.41}$$

which implies the following representation for any element $q \in \mathbb{H}$

$$q = q_0\mathtt{h} + q_1\mathtt{i} + q_2\mathtt{j} + q_3\mathtt{k}, \tag{13.42}$$

where $q_i \in \mathbb{R}, i \in \{0, 1, 2, 3\}$.

To avoid confusion with regular vectors, we introduce the operator $\star$ to denote multiplication between arbitrary quaternions and this operation is now defined by distribution over the basic elements $\mathtt{h}, \mathtt{i}, \mathtt{j}, \mathtt{k}$. Given two elements $p, q \in \mathbb{H}$,

defined as $p = p_0\mathrm{h} + p_1\mathrm{i} + p_2\mathrm{j} + p_3\mathrm{k}$, and $q = q_0\mathrm{h} + q_1\mathrm{i} + q_2\mathrm{j} + q_3\mathrm{k}$, we find

$$\begin{aligned} p \star q &= (p_0\mathrm{h} + p_1\mathrm{i} + p_2\mathrm{j} + p_3\mathrm{k})(q_0\mathrm{h} + q_1\mathrm{i} + q_2\mathrm{j} + q_3\mathrm{k}) \\ &= p_0\mathrm{h}(q_0\mathrm{h} + q_1\mathrm{i} + q_2\mathrm{j} + q_3\mathrm{k}) + p_1\mathrm{i}(q_0\mathrm{h} + q_1\mathrm{i} + q_2\mathrm{j} + q_3\mathrm{k}) \\ &+ p_2\mathrm{j}(q_0\mathrm{h} + q_1\mathrm{i} + q_2\mathrm{j} + q_3\mathrm{k}) + p_3\mathrm{k}(q_0\mathrm{h} + q_1\mathrm{i} + q_2\mathrm{j} + q_3\mathrm{k}) \\ &= (p_0q_0 - p_1q_1 - p_2q_2)\mathrm{h} + (p_0q_1 + p_1q_0 + p_2q_3 - p_3q_2)\mathrm{i} \\ &+ (p_0q_2 + p_2q_0 + p_3q_1 - p_1q_3)\mathrm{j} + (p_0q_3 + p_3q_0 + p_1q_2 - p_1q_2)\mathrm{k}. \end{aligned} \tag{13.43}$$

The complex conjugate of $q \in \mathbb{H}$ is denoted as $q^\dagger$ and is defined as follows

$$\begin{aligned} &\mathrm{h}^\dagger = \mathrm{h},\ \mathrm{i}^\dagger = -\mathrm{i},\ \mathrm{j}^\dagger = -\mathrm{j},\ \mathrm{k}^\dagger = -\mathrm{k}, \\ &q^\dagger = q_0\mathrm{j} - q_1\mathrm{i} - q_2\mathrm{j} - q_3\mathrm{k}. \end{aligned} \tag{13.44}$$

From these definitions, it follows that

$$q \star q^\dagger = q_0^2 + q_1^2 + q_2^2 + q_3^2, \tag{13.45}$$

so that $q \star q^\dagger$ is real and non-negative. In fact, $q \star q^\dagger$ is the same as the squared Euclidean norm of $q \in \mathbb{R}^4$. For any quaternion $q \in \mathbb{H}$, we write $\|q\|^2 = q \star q^\dagger$.

It is often convenient to partition quaternions into *scalar* and *vector* parts as follows

$$q = \begin{bmatrix} q_s \\ q_v \end{bmatrix}, \quad q_s = q_0, \quad q_v = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}. \tag{13.46}$$

Alternately, q_s is called the *real* part and q_v is called the imaginary part. This nomenclature makes sense from the definition of the complex conjugate and the observation that $q_s^\dagger = q_s$, and $q_v^\dagger = -q_v$.

Using this partition, the quaternion product can be rewritten in terms of standard vector operators as follows

$$p \star q = \begin{bmatrix} p_s q_s - p_v^T q_v \\ p_s q_v + q_s p_v + \widehat{p}_v q_v \end{bmatrix}. \tag{13.47}$$

Now, since $q^\dagger \in \mathbb{H}$, and since $q \star q^\dagger > 0$ for $q \in \mathbb{H}, q \neq 0$, it follows that

$$q \star \left(\frac{1}{\|q\|} q^\dagger \right) = 1. \tag{13.48}$$

This means that any non-zero quaternion $q \in \mathbb{H}$ has a unique multiplicative inverse in $\mathbb{H}$, written q^{-1}. In particular, for an element $q \in \mathbb{H}$ with unit norm, $\|q\| = 1$, the inverse is the complex conjugate $q^{-1} = q^\dagger$, as is the case for complex numbers.

However, as can be seen from the multiplication table, since $\mathrm{ij} = -\mathrm{ji} = \mathrm{k}$, quaternion multiplication is non-commutative (though it is associative), except for the case of scalar multiplication, i.e., for $q \in \mathbb{H}, q = q_0\mathrm{h} + 0\mathrm{i} + 0\mathrm{j} + 0\mathrm{k}$. Therefore, $\mathbb{H}$ is a *division ring* but not a *field* [138], since fields must have a commutative multiplication operator.

The following two identities are easily verified using elementary algebraic manipulations:

$$(p \star q)^\dagger = q^\dagger \star p^\dagger \tag{13.49}$$

$$(p \star q)^{-1} = q^{-1} \star p^{-1}. \tag{13.50}$$

The second of these follows from the first and the definition of the multiplicative inverse.

13.4 A three-dimensional subspace

Consider a set $\mathcal{V} \in \mathbb{H}$ containing all quaternions $q \in \mathbb{H}$ such that $q^\dagger = -q$. Obviously, since $0 = -0$, we have $0 \in \mathcal{V}$. Now, given $q = q_1\mathrm{i} + q_2\mathrm{j} + q_3\mathrm{k}$ where $q_i \in \mathbb{R}$, then, $q^\dagger = -q$. Likewise, given any scalar $\sigma \in \mathbb{R}$ and $q \in \mathcal{V}$, $\sigma q \in \mathcal{V}$. Finally, given $q, p \in \mathcal{V}$ and $\sigma, \tau \in \mathbb{R}$, $\sigma q + \tau p \in \mathcal{V}$. Therefore, we conclude that the subset $\mathcal{V} \in \mathbb{H}$ with the addition operator forms a three-dimensional vector space over $\mathbb{R}$ isomorphic to $\mathbb{R}^3$.

This consideration is in fact what motivated Hamilton originally to associate vectors in $\mathbb{R}^3$ with pure imaginary quaternions, and persists to this day in the notation $x = x_1\mathrm{i} + x_2\mathrm{j} + x_3\mathrm{k}$, for $x \in \mathbb{R}^3$, which is common in vector calculus [195].

13.5 Matrix representation of quaternion algebra

A matrix representation of the quaternion product defined in (13.47) can be read off directly producing either

$$p \star q = Q(p)q = \begin{bmatrix} p_0 & -p_1 & -p_2 & -p_3 \\ p_1 & p_0 & -p_3 & p_2 \\ p_2 & p_3 & p_0 & -p_1 \\ p_3 & -p_2 & p_1 & p_0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} p_s & -p_v^T \\ p_v & p_s I_3 + \widehat{p}_v \end{bmatrix} \begin{bmatrix} q_s \\ q_v \end{bmatrix}, \tag{13.51}$$

or

$$p \star q = P(q)p = \begin{bmatrix} p_0 & -p_1 & -p_2 & -p_3 \\ p_1 & p_0 & p_3 & -p_2 \\ p_2 & -p_3 & p_0 & p_1 \\ p_3 & p_2 & -p_1 & p_0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} q_s & -q_v^T \\ q_v & q_s I_3 - \widehat{q}_v \end{bmatrix} \begin{bmatrix} p_s \\ p_v \end{bmatrix}, \tag{13.52}$$

where the first form defines a right ordered product, and the second defines a left ordered product. In addition, the complex conjugate operation can be expressed as

$$q^\dagger = \begin{bmatrix} q_0 \\ -q_1 \\ -q_2 \\ -q_3 \end{bmatrix} = \begin{bmatrix} q_s \\ -q_v \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -I_3 \end{bmatrix} \begin{bmatrix} q_s \\ q_v \end{bmatrix} = Cq. \tag{13.53}$$

where 4×4 matrix C is defined as

$$C = \begin{bmatrix} 1 & 0 \\ 0 & -I_3 \end{bmatrix}. \tag{13.54}$$

Using the definitions for $Q(q), P(q)$ and C, two isomorphisms can be constructed between $\mathbb{H}$ and $GL_4(\mathbb{R})$, so that $\mathbb{H}$ is represented as a sub-algebra of the set of 4×4 matrices over $\mathbb{R}$ with the usual addition and multiplication rules.

Consider first the right ordered product defined by the map $\phi_r : \mathbb{H} \mapsto GL_4(\mathbb{R})$ with the definition

$$\begin{aligned} \phi_r(q) &= Q(q), \\ \phi_r(q \star p) &= Q(q)Q(p) = Q(q \star p). \end{aligned} \tag{13.55}$$

The second equality will be demonstrated in what follows.

It is clear from the definitions of matrices Q and P in (13.51) and (13.52) that that $Q(q)+Q(p) = Q(p+q)$ and therefore, ϕ_r defines a representation of $\mathbb{H}$ using standard linear algebra.

Remains to show that $Q(p \star q) = Q(p)Q(q)$. First, compute directly the matrix product

$$\begin{aligned} Q(p)Q(q) &= \begin{bmatrix} p_s & -p_v^T \\ p_v & p_s I_3 + \widehat{p}_v \end{bmatrix} \begin{bmatrix} q_s & -q_v^T \\ q_v & q_s I_3 + \widehat{q}_v \end{bmatrix} \\ &= \begin{bmatrix} p_s q_s - p_v^T q_v & -p_s q_v^T - q_s p_v^T - p_v^T \widehat{q}_v \\ p_s q_v + q_s p_v + \widehat{p}_v q_v & p_s q_s I_3 - p_v q_v^T + \widehat{p}_v \widehat{q}_v + p_s \widehat{q}_v + q_s \widehat{p}_v \end{bmatrix} \\ &= \begin{bmatrix} \alpha & -v^T \\ v & B \end{bmatrix}, \end{aligned} \tag{13.56}$$

with the definitions

$$\begin{aligned} \alpha &= (p \star q)_s, \text{ and} \\ v &= (p \star q)_v, \end{aligned} \tag{13.57}$$

as per (13.47). Remains to show that matrix B satisfies the identity $B = r_s I_3 - \widehat{r}_v$ for $r = p \star q$. This follows from the two tensor identities of Lemma 13.4 and Lemma 13.6, leading to

$$\begin{aligned} B &= p_s q_s I_3 - p_v q_v^T + \widehat{p}_v \widehat{q}_v + p_s \widehat{q}_v + q_s \widehat{p}_v \\ &= \left(p_s q_s - p_v^T q_v \right) I_3 - \left(p_v q_v^T - q_v p_v^T - p_s \widehat{q}_v - q_s \widehat{p}_v \right) \\ &= \alpha I_3 - \widehat{v} = r_s I_3 - \widehat{r}_v, \end{aligned} \tag{13.58}$$

as required. This completes the proof and therefore, the mapping ϕ_r is an isomorphism from $\mathbb{H}$ into a subgroup of $GL_4(\mathbb{R})$

Next, consider the map $\phi_l : \mathbb{H} \mapsto GL_4(\mathbb{R})$ with the definition

$$\begin{aligned} \phi_l(q) &= P(q) \\ \phi_l(p \star q) &= P(q)P(p) = P(q \star p). \end{aligned} \tag{13.59}$$

As in the case for the map ϕ_r, the map ϕ_l preserves the additive structure of $\mathbb{H}$. However, the map ϕ_l reverses the order of the product and that feature will be useful in the definition of rigid rotations in what follows. The proof that $P(q)P(p) = P(q \star p)$ is almost identical to the one above for the ϕ_r isomorphism so it is omitted.

Note also the following identities which follow directly from the definition

$$\begin{aligned} Q(q^\dagger) &= Q^T(q), \text{ and} \\ P(q^\dagger) &= P^T(q), \end{aligned} \tag{13.60}$$

and the useful order exchanging formulae

$$\begin{aligned} Q^T(q)p &= CQ^T(p)q, \\ P^T(q)p &= CP^T(p)q, \end{aligned} \tag{13.61}$$

for any two quaternions $p, q \in \mathbb{H}$.

To simplify computations, matrices $P(q)$ and $Q(q)$ are often partitioned

$$Q(q) = \begin{bmatrix} q_s & -q_v^T \\ q_v & q_s I_3 + \widehat{q}_v \end{bmatrix} = \begin{bmatrix} q & \mathcal{G}^T(q) \end{bmatrix}, \tag{13.62}$$

and

$$P(q) = \begin{bmatrix} q_s & -q_v^T \\ q_v & q_s I_3 - \widehat{q}_v \end{bmatrix} = \begin{bmatrix} q & \mathcal{E}^T(q) \end{bmatrix}, \tag{13.63}$$

which defines the 3×4 real matrices $\mathcal{E}(q)$ and $\mathcal{G}(q)$:

$$\begin{aligned} \mathcal{E}(q) &= \begin{bmatrix} -q_v & q_s I_3 + \widehat{q}_v \end{bmatrix}, \\ \mathcal{G}(q) &= \begin{bmatrix} -q_v & q_s I_3 - \widehat{q}_v \end{bmatrix}. \end{aligned} \tag{13.64}$$

It is easy to verify that $\mathcal{E}(q)q = 0$, and similarly, $\mathcal{G}(q)q = 0$. This fact will be used extensively.

The matrices $P(q)$ and $Q(p)$ actually commute with each other for any $q, p \in \mathbb{H}$ as proved in Theorem 13.16 below.

Theorem 13.16. *The matrices $P(p), Q(q)$ defined in* (13.62)*, commute for any quaternions $p, q \in \mathbb{H}$.*

Proof. Explicitly computing the product in both order, the result is

$$\begin{aligned} P(p)Q(q) &= \begin{bmatrix} p_s & -p_v^T \\ p_v & p_s I_3 - \widehat{p}_v \end{bmatrix} \begin{bmatrix} q_s & -q_v^T \\ q_v & q_s I_3 + \widehat{q}_v \end{bmatrix} \\ &= \begin{bmatrix} p_s q_s - p_v^T q_v & -p_s q_v^T - q_s p_v^T - p_v^T \widehat{q}_v \\ q_s p_v + p_s q_v^T - \widehat{p}_v q_v & -p_v q_v^T + p_s q_s I_3 + p_s \widehat{q}_v - q_s \widehat{p}_v - \widehat{p}_v \widehat{q}_v \end{bmatrix} \\ &= \begin{bmatrix} \alpha & u^T \\ v & A \end{bmatrix}, \end{aligned} \tag{13.65}$$

and similarly

$$\begin{aligned} Q(q)P(p) &= \begin{bmatrix} q_s & -q_v^T \\ q_v & q_s I_3 + \widehat{q}_v \end{bmatrix} \begin{bmatrix} p_s & -p_v^T \\ p_v & p_s I_3 - \widehat{p}_v \end{bmatrix} \\ &= \begin{bmatrix} q_s p_s - q_v^T p_v & -q_s p_v^T - p_s q_v^T + q_v^T \widehat{p}_v \\ p_s q_v + q_s p_v + \widehat{q}_v p_v & -q_v p_v^T + q_s p_s I_3 - q_s \widehat{p}_v + p_s \widehat{q}_v - \widehat{q}_v \widehat{p}_v \end{bmatrix} \\ &= \begin{bmatrix} \beta & w^T \\ x & B \end{bmatrix}. \end{aligned} \tag{13.66}$$

Inspection yields $\alpha = \beta$, $u = w$, and $v = x$. By inspection as well, matrices A and B are equal as the following identity holds

$$p_v q_v^T + \widehat{p}_v \widehat{q}_v = q_v p_v^T + \widehat{q}_v \widehat{p}_v, \tag{13.67}$$

as per Lemma 13.4. □

This fact is very interesting since it allows reordering quaternion product expressions.

Finally, the determinants matrices P and Q matrices are computed to be $\det(Q(q)) = \det(P(q)) = \|q\|^4$ as seen from (13.23) with $\lambda = q_0$. This means that as long as $\|q\| = 1$, all products involving $Q(q)$ or $P(q)$ are well behaved.

13.6 Length preserving transformations

Consider an element $q \in \mathbb{H}$ with unit length, $\|q\| = 1$, and the isomorphism $\psi_q : \mathbb{H} \mapsto \mathbb{H}$ defined with

$$\psi_q(p) = q \star p. \tag{13.68}$$

The norm of the transformed element is $\|\psi_q(p)\|^2 = q \star p \star p^\dagger \star q^\dagger = \|p\|^2 \|q\|^2 = \|p\|^2$, and is thus unchanged by the action of ψ_q. However, the subspace $\mathcal{V}$ is not invariant under the action of ψ_q since for $p = \mathrm{i} \in \mathcal{V}$ and $q = -\mathrm{i} \in \mathbb{H}$, $\|q\| = 1$ and $\psi_q(p) = 1 \notin \mathcal{V}$. As reported in[4], this is what caused controversy. The correct

formulation which leaves $\mathcal{V}$ invariant is the map $\phi_q : \mathbb{H} \mapsto \mathbb{H}$, for an element $q \in \mathbb{H}$ such that $\|q\| = 1$, defined with

$$\phi_q(p) = q \star p \star q^\dagger. \tag{13.69}$$

This is length preserving since

$$\|\phi_q(p)\|^2 = q \star p \star q^\dagger \star q \star p^\dagger \star q^\dagger = q(\star p \star p^\dagger) \star q^\dagger = \|p\|^2 q \star q^\dagger = \|p\|^2. \tag{13.70}$$

In addition, this transformation leaves the linear sub-space $\mathcal{V}$ invariant since for any $p \in \mathcal{V}$, the conjugate of the transformed element is

$$\phi_q(p)^\dagger = \left(q \star p \star q^\dagger\right)^\dagger = q \star p^\dagger \star q^\dagger = -q \star p \star q^\dagger = -\phi_q(p). \tag{13.71}$$

Therefore, if $p \in \mathcal{V}$, then, $\phi_q(p) \in \mathcal{V}$ for any $q \in \mathbb{H}$ and the transformation is length preserving when $\|q\| = 1$.

The matrix representation of the map ϕ_q is computed using the matrices $Q(q), P(q)$ defined above. First, note that the left-most part of the product can be represented as the *left* product: $r = p \star q^\dagger = P(q^\dagger)p = P^T(q)p$, and then, $q \star p \star q^\dagger = q \star r$

$$\phi_q(p) = q \star (p \star q^\dagger) = Q(q)(P(q^\dagger)p) = \left(Q(q)P^T(q)\right)p. \tag{13.72}$$

Given the definitions for matrices $Q(q)$ and $P(q)$, the following explicit formula for the product is found

$$\begin{aligned} Q(q)P^T(q) = P^T(q)Q(q) &= \begin{bmatrix} q^T \\ \mathcal{E}(q) \end{bmatrix} \begin{bmatrix} q & \mathcal{G}^T(q) \end{bmatrix} \\ &= \begin{bmatrix} q^T q & q^T \mathcal{G}^T(q) \\ \mathcal{E}(q)q & \mathcal{E}(q)\mathcal{G}^T(q) \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & \mathcal{E}(q)\mathcal{G}^T(q) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & R(q) \end{bmatrix}, \end{aligned} \tag{13.73}$$

with the definition $R(q) = \mathcal{E}(q)\mathcal{G}^T(q)$, and where the facts that $\mathcal{E}(q)q = \mathcal{G}(q)q = 0$ and $q^T q = 1$, where used. Expanding this further,

$$\begin{aligned} R(q) = \mathcal{E}(q)\mathcal{G}^T(q) &= \begin{bmatrix} -q_v & q_s I_3 + \widehat{q}_v \end{bmatrix} \begin{bmatrix} -{q_v}^T \\ q_s I_3 + \widehat{q}_v \end{bmatrix} \\ &= \left(q_s^2 - q_v^T q_v\right) I_3 + 2 q_v q_v^T + 2 q_s \widehat{q}_v \\ &= (2q_s^2 - 1) I_3 + 2 q_v q_v^T + 2 q_s \widehat{q}_v, \end{aligned} \tag{13.74}$$

where $q^T q = 1$ was used again. The last expression in (13.74) is a useful definition when performing algebraic manipulations, but it can also be simplified further by using Lemma 13.4 to yield

$$R(q) = I_3 + 2 q_s \widehat{q}_v + 2 \widehat{q}_v \widehat{q}_v. \tag{13.75}$$

Using this form makes it easy to verify that $R(q)q_v = \lambda q_v$ with $\lambda = 1$.

Also from (13.75), one can easily recover the limit of small rotations in which $q_s = 1 - O(\epsilon^2)$ and $q_v = \epsilon u$, where $u \in \mathbb{R}^3$ is a unit vector, yielding the limit

$$R(\epsilon) = I_3 + 2\epsilon\widehat{u}. \tag{13.76}$$

Given the known fact that $\mathrm{d}R/\mathrm{d}\epsilon \to \widehat{\omega}$ as $\epsilon \to 0$ and where ω is the angular velocity vector in the inertial frame of reference, it is already possible to deduce that $2\epsilon^{-1}q(\epsilon)_v \to \omega$ as $\epsilon \to 0$.

For implementation purposes and to provide for explicitly computing a quaternion q which corresponds to a given orthonormal matrix R, the components of either (13.75) or (13.74) can be resolved further in terms of the elements of q as follows

$$R(q) = \begin{bmatrix} (2q_0^2 - 1) + 2q_1^2 & 2(q_1q_2 - q_0q_3) + & 2(q_0q_2 + q_1q_3) \\ 2(q_0q_3 + q_2q_1) & (2q_0^2 - 1) + 2q_2^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_0q_1 + q_3q_2) & (2q_0^2 - 1) + 2q_3^2 \end{bmatrix}. \tag{13.77}$$

The properties of the result of two consecutive transformation are established in Theorem 13.17 and Corollary 13.18 below.

Theorem 13.17. *The transformations $\phi_q : \mathbb{H} \mapsto \mathbb{H}$ defined with $\phi_q(p) = q \star p \star q^\dagger$ form a group so that $\phi_r \circ \phi_s = \phi_{r\star s}$.*

Proof. We start with the quaternion representation of the transformation and evaluate the map $\phi_r \circ \phi_s$ on $p \in \mathbb{H}$

$$\begin{aligned} \phi_r(\phi_s(p)) &= r \star (s \star p \star s^\dagger) \star r^\dagger \\ &= (r \star s) \star p \star (s^\dagger \star r^\dagger) \\ &= (r \star s) \star p \star (r \star s)^\dagger \\ &= \phi_{r\star s}(p). \end{aligned} \tag{13.78}$$

□

Corollary 13.18. *The matrix representation of the map $\phi_q : \mathbb{H} \mapsto \mathbb{H}$ given by $Q(q)P^T(q)$ satisfies the group property.*

Proof. Compute the action of the map $\phi_r \circ \phi_s$ on $p \in \mathbb{R}^4$

$$\begin{aligned} \phi_r(\phi_s(p)) &= Q(r)P^T(r)(Q(s)P^T(s)p) \\ &= Q(r)Q(s)P^T(r)P^T(s)p \\ &= Q(r \star s)\big(P(s)P(r)\big)^T p \\ &= Q(r \star s)\big(P(r \star s)\big)^T p \\ &= \phi_{r\star s}(p), \end{aligned} \tag{13.79}$$

where use was made of the identities $Q(r)Q(s) = Q(r \star s)$ and $P(s)P(r) = P(r \star s)$. □

The representation of the rotation matrix given in (13.74) can be reformulated in terms of a unit vector $n = q_v/\|q_v\|$ and an angle ϕ with $\cos(\phi/2) = q_v$. This representation implies that $\|q_v\| = \sin(\phi/2)$ and so we have the following identity

$$R(q) = (2\cos^2(\phi/2) - 1)I_3 + 2\sin^2(\phi/2)nn^T + 2\cos(\phi/2)\sin(\phi/2)\widehat{n}. \quad (13.80)$$

Using trigonometric identities, this can be rewritten as

$$\begin{aligned} R(q) &= \cos\phi I_3 + (1-\cos\phi)nn^T + \sin\phi\widehat{n} \\ &= I_3 + (\cos\phi - 1)\left(I_3 - nn^T\right) + \sin\phi\widehat{n}. \end{aligned} \quad (13.81)$$

This is a well known formula for a rotation by ϕ about a unit axis $n \in \mathbb{R}^3, \|n\| = 1$. This representation makes it clear that n is an eigenvector of $R(q)$ with unit eigenvalue. The other two eigenvalues are generally complex conjugates.

The effect of the transformation on a given vector is to rotate the component orthogonal to the normal vector n by the angle phi in the orthogonal plane. To see this, define a right handed coordinate basis $u, v, n \in \mathbb{R}^3$, with $\|u\| = \|v\| = \|n\| = 1$ and $\widehat{u}v = n$, $\widehat{n}u = v$, and $\widehat{n}v = -u$. Then, decompose an arbitrary vector $x \in \mathbb{R}^3$ according to this basis so that $x = \alpha u + \beta v + \gamma n$. Applying the transformation (13.81) to this yields

$$\begin{aligned} R(q)x &= \alpha u + \beta v + \gamma n + \sin\phi(\alpha v - \beta u) + (\cos\phi - 1)(\alpha u + \beta v) \\ &= \gamma n + \alpha(\cos\phi u + \sin\phi v) + \beta(-\sin\phi u + \cos\phi v), \end{aligned} \quad (13.82)$$

which is easily recognized as a plane rotation by angle ϕ in the $u - v$ plane.

This is illustrated in Figure 13.1 where the axis of the cone is collinear with the vector n. The same axis is the normal of the orthogonal plane which lies at the top of the cone. The tip of the original vector x, showed with the dashed line, is rotated in the plane by the angle ϕ to result in the final vector $y = R(q)x$ showed with a solid line. The projection onto the normal plane, shown with a dotted line is what gets rotated by the angle.

13.7 Properties of the rotation matrices

The set of matrices $\{R(q) | q \in \mathbb{R}^4, \|q\| = 1\}$ defined in (13.74) is the restriction on $\mathcal{V}$ of length preserving linear transformations on $\mathbb{H}$. They form a group under multiplication as per Corollary 13.18.

The correspondence between unit quaternions and the rotation group is not a bijection however, since $R(q) = R(-q)$ which is clear from (13.77). The quaternion group therefore provides a double coverage of $SO(3)$ Theorem 13.19 below establishes the direct correspondence between any given real orthonormal 3×3 matrix Q of unit determinant and unit quaternion $\pm q$.

Theorem 13.19. *Given any real* 3×3 *orthonormal* Q*, there are two* $q \in \mathbb{H}$ *with* $\|q\| = 1$ *such that* $R(q) = Q$*.*

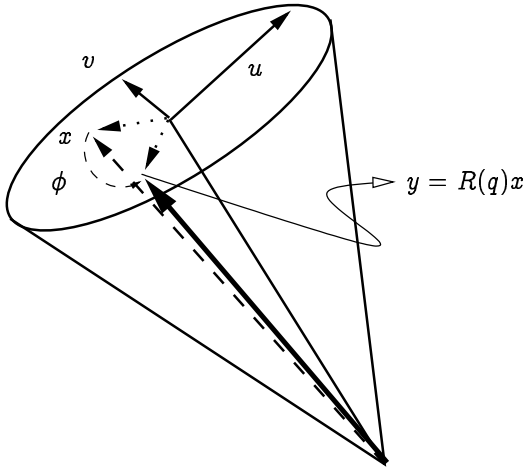

Figure 13.1: A three-dimensional rotation by an angle ϕ about an axis n.

Proof. Direct computation yields the following formulae

$$\begin{aligned} q_0^2 &= \frac{1}{4}(\mathrm{tr}(R)+4), & q_0 q_1 &= \frac{1}{4}(R_{32}-R_{23}), \\ q_0 q_2 &= \frac{1}{4}(R_{13}-R_{31}), & q_0 q_3 &= \frac{1}{4}(R_{21}-R_{12}), \end{aligned} \tag{13.83}$$

which can be solved for q_0, q_1, q_2, q_3 provided $q_0 \neq 0$. There are two solutions to this system depending on the sign of q_0.

If $q_0 = 0$, the value of the diagonal elements becomes

$$\begin{aligned} R_{11}+1 &= 2q_1^2, \\ R_{22}+1 &= 2q_2^2, \\ R_{33}+1 &= 2q_3^2. \end{aligned} \tag{13.84}$$

Now, if $q_0 = 0$, at least one of these must be non-zero since $\sum_i q_i^2 = 1$. Assume that $R_{kk}+1 = 2q_k^2 \neq 0$ for some $k \in \{1,2,3\}$, then, the other two elements can be recovered from the relations

$$R_{ki} = 2q_k q_i \tag{13.85}$$

where $i \in \{j \in \{1,2,3\} | j \neq k\}$. There are two solutions to this system corresponding whether the plus or minus sign is chosen in solving q_k from R_{kk}.

This cover all cases. Note that for any $q \in \mathbb{H}, \|q\| = 1$, the following identity holds: $R(q) = R(-q)$ and therefore, the multiplicative group of unit quaternions is a double coverage of the group $SO(3)$. □

13.8 Algebraic identities of rotation factors

The factor matrices $\mathcal{E}(q), \mathcal{G}(q)$ of (13.62) obey a number of identities which are useful in the construction of kinematic constraints and so they are presented

here.

Lemma 13.20. *For any two vectors $p, q \in \mathbb{R}^4$, the following identities hold*

$$\mathcal{G}(q)p = -\mathcal{G}(p)q, \quad \mathcal{E}(q)p = -\mathcal{E}(p)q. \tag{13.86}$$

Proof. First look at the expression involving $\mathcal{E}(q)$

$$\begin{aligned}
\mathcal{E}(q)p &= \begin{bmatrix} -q_v & q_s I_3 + \widehat{q}_v \end{bmatrix} \begin{bmatrix} p_s \\ p_v \end{bmatrix} \\
&= -p_s q_v + q_s p_v + \widehat{q}_v p_v \\
&= q_s p_v - p_s q_v - \widehat{p}_v q_v \\
&= - \begin{bmatrix} -p_v & p_s I_3 + \widehat{p}_v \end{bmatrix} \begin{bmatrix} q_s \\ q_v \end{bmatrix} \\
&= -\mathcal{E}(p)q.
\end{aligned} \tag{13.87}$$

A nearly identical computation yields the companion identity $\mathcal{G}(q)p = -\mathcal{G}(p)q$. □

Corollary 13.21. *For any vector $q \in \mathbb{R}^4$, $\mathcal{G}(q)q = 0$ and $\mathcal{E}(q)q = 0$.*

Proof. Since $\mathcal{E}(q)q = -\mathcal{E}(q)q$ as per Lemma 13.20, all components of $r \in \mathbb{R}^3, r = \mathcal{E}(q)q$, must vanish. □

The converse result holds as well now shown.

Lemma 13.22. *Given unit quaternions $p, q \in \mathbb{H}, \|q\| = \|p\| = 1$, $\mathcal{G}(p)q = \mathcal{E}(p)q = 0$ if and only if $q = \pm p$.*

Proof. Concentrate on $\mathcal{E}(q)$ matrices since the corresponding result for $\mathcal{G}(q)$ has an identical proof. The "if" part is given from Corollary 13.21. For the "only if part", note first that if $q_s = 0$, then, $\mathcal{E}(q)p = -p_s q_v + \widehat{q}_v p = 0$. But since p_v is orthogonal to the cross product $\widehat{q}_v p$, it must be that $p_s = 0$ as well. In turn, this implies that $\widehat{q}_v p = 0$ and therefore, $q_v = \lambda p_v, \lambda \in \mathbb{R}$. Since $\|q_v\| = \|p_v\| = 1$, $\lambda = \pm 1$.

Likewise, if $q_s = 1$, then, $q_v = 0$ since $\|q\| = 1$. Therefore, we have $\mathcal{E}(q)p = p_v = 0$, since the other two terms are linear in q_v and thus vanish. This implies that $p_s = \pm 1$ since $\|q\| = |p_s = 1$. So, again, $q = \pm p$.

Now, assuming that both $q_s, p_s \in (0, 1)$, then, the scalar products of q_v and p_v with $\mathcal{E}(q)p$ yield the two relations

$$q_s \|p_v\|^2 = p_s q_v^T p_v, \text{ and } q_s q_v^T p_v = p_s \|q_v\|^2. \tag{13.88}$$

Since $\|q_v\|^2 = (1 - q_s^2)$ and $\|p_v\|^2 = (1 - p_s^2)$, a simple manipulation yields the relation

$$\frac{q_s^2}{1 - q_s^2} = \frac{p_s^2}{1 - p_s^2}. \tag{13.89}$$

Given that the scalar function $f(x) = x/(1-x)$ is monotone increasing on the interval $x \in (0,1)$, since $f'(x) = 1/(1-x)^2$, we have $f(x) = f(y) \Rightarrow x = y$, for all $x, y \in (0,1)$. Therefore, $q_s^2 = p_s^2$ and so $q_s = \pm p_s$. This implies that $\|p_v\| = \|q_v\|$ as well.

Writing $p_v^T q_v = \|p_v\|\|q_v\| \cos\theta$ for some angle θ, the first identity of (13.88) now reads

$$1 = \pm \cos\theta. \tag{13.90}$$

Now, assume that $p_s = \pm q_s$ and $p_v = \pm' q_v$. The product $\mathcal{E}(q)p$ now reads

$$\mathcal{E}(q)p = \pm' q_s q_v - \pm q_s q_v = 0, \tag{13.91}$$

which means that $\pm = \pm'$, which completes the proof. □

Lemma 13.23. *For any vector $q \in \mathbb{R}^4$, the following identities hold*

$$\begin{aligned} \mathcal{E}^T(q)\mathcal{E}(q) &= \mathcal{G}^T(q)\mathcal{G}(q) = \|q\|^2 I_4 - qq^T, \text{ and} \\ \mathcal{E}(q)\mathcal{E}^T(q) &= \mathcal{G}(q)\mathcal{G}^T(q) = \|q\|^2 I_3 \end{aligned} \tag{13.92}$$

Proof. The algebraic manipulations are performed for $\mathcal{E}$ matrices only since the results are easy to verify for $\mathcal{G}$ matrices. Expand the products as follows

$$\begin{aligned} \mathcal{E}^T(q)\mathcal{E}(q) &= \begin{bmatrix} -q_v^T \\ q_s I_3 - \widehat{q}_v \end{bmatrix} \begin{bmatrix} -q_v & q_s I_3 + \widehat{q}_v \end{bmatrix} = \begin{bmatrix} q_v^T q_v & -q_s q_v^T \\ -q_s q_v & q_s^2 I_3 - \widehat{q}_v \widehat{q}_v \end{bmatrix} \\ &= \begin{bmatrix} q^T q - q_s^2 & -q_s q_v^T \\ -q_s q_v & q^T q - q_v q_v^T \end{bmatrix} \\ &= \|q\|^2 I_4 - qq^T, \end{aligned} \tag{13.93}$$

and likewise for the other ordering

$$\begin{aligned} \mathcal{E}(q)\mathcal{E}^T(q) &= \begin{bmatrix} -q_v & q_s I_3 + \widehat{q}_v \end{bmatrix} \begin{bmatrix} -q_v^T \\ q_s I_3 - \widehat{q}_v \end{bmatrix} \\ &= q_v q_v^T + q_s^2 I_3 - \widehat{q}_v \widehat{q}_v \\ &= \|q\|^2, \end{aligned} \tag{13.94}$$

where use was made of Lemma 13.4. The same algebraic identities carry over for the case of $\mathcal{G}(q)$ matrices. □

Lemma 13.24. *For any vector $q \in \mathbb{R}^4$, the following identities hold*

$$\mathcal{E}(q) = \mathcal{E}(1)P^T(q), \quad \mathcal{G}(q) = \mathcal{G}(1)Q^T(q), \tag{13.95}$$

with the shorthand $1 = [1,0,0,0]^T = \mathtt{h}$.

Proof. From the definitions (13.64), $\mathcal{E}(1) = \mathcal{G}(1) = \begin{bmatrix} 0 & I_3 \end{bmatrix}$, where 0 is a 3×1 zero block here, and therefore, the result follows from the definitions of $P(q)$ and $Q(q)$ since

$$\mathcal{E}(1)P^T(q) = \begin{bmatrix} 0 & I_3 \end{bmatrix} \begin{bmatrix} q^T \\ \mathcal{E}(q) \end{bmatrix} = \mathcal{E}(q), \tag{13.96}$$

and similarly for $\mathcal{G}(q)$. □

Lemma 13.24 produces yet another useful identity.

Lemma 13.25. *For any vectors $p, q \in \mathbb{R}^4$, the following holds*

$$\begin{aligned} \mathcal{E}(q)\mathcal{E}^T(p) &= \mathcal{E}(1)\mathcal{E}^T(p \star q^\dagger) \\ \mathcal{G}(q)\mathcal{G}^T(p) &= \mathcal{G}(1)\mathcal{G}^T(q^\dagger \star p). \end{aligned} \tag{13.97}$$

Proof. For the $\mathcal{E}$ matrix, using Lemma 13.24 yields

$$\begin{aligned} \mathcal{E}(q)\mathcal{E}^T(p) &= \mathcal{E}(1)P^T(q)P(p)\mathcal{E}^T(1) = \mathcal{E}(1)P(q^\dagger)P(p)\mathcal{E}^T(1) \\ &= \mathcal{E}(1)P(p \star q^\dagger)\mathcal{E}^T(1) = \mathcal{E}(1)\Big(\mathcal{E}(1)P^T(p \star q^\dagger)\Big)^T \\ &= \mathcal{E}(1)\mathcal{E}^T(p \star q^\dagger). \end{aligned} \tag{13.98}$$

□

Another interesting point is that the action of matrix $R(q) = \mathcal{E}(q)\mathcal{G}^T(q)$ converts $\mathcal{G}$ matrices to $\mathcal{E}$ matrices and vice versa.

Lemma 13.26. *Given any unit vector, $q \in \mathbb{R}^4, \|q\|^2 = 1$, matrices $\mathcal{E}(q), \mathcal{G}(q)$ as defined in* (13.62)*, and $R(q) = \mathcal{E}(q)\mathcal{G}^T(q)$, then*

$$R(q)\mathcal{G}(q) = \mathcal{E}(q), \quad \text{and } R^T(q)\mathcal{E}(q) = \mathcal{G}(q). \tag{13.99}$$

Proof. Direct computation yields

$$\begin{aligned} R(q)\mathcal{G}(q) &= \mathcal{E}(q)\mathcal{G}^T(q)\mathcal{G}(q) = \mathcal{E}(q)\left[\|q\|^2 I_4 - qq^T\right] \\ &= \mathcal{E}(q) - \mathcal{E}(q)qq^T = \mathcal{E}(q) - \big(\mathcal{E}(q)q\big)\,q^T \\ &= \mathcal{E}(q), \end{aligned} \tag{13.100}$$

since $\mathcal{E}(q)q = 0$ as shown in the corollary Lemma 13.20, and since $\|q\|^2 = 1$ by hypothesis. A similar computation yields the other result. □

The value of the product determinant $\det \mathcal{E}(q)\mathcal{G}^T(p)$ for two different quaternions $p, q \in \mathbb{H}$ can also be computed explicitly, and so is that of matrices of products of the form $P^T(q)Q(p)$. This will be useful in analyzing joint Jacobians later on. The result is as follows.

Lemma 13.27. *Given $p, q \in \mathbb{H}$, the matrix $B = \mathcal{E}(q)\mathcal{G}^T(p)$ has determinant $\det B = \|q\|^2\|p\|^2 q^T p$.*

Proof. First note that matrices $P(q), Q(p)$ defined in (2.29) are square and therefore, $\det P^T(q)Q(p) = \det P(q) \det Q(p) = \|q\|^4\|p\|^4$ as per (13.23). Now, the product $P^T(q)Q(p)$ can be partitioned and factored as follows

$$\begin{aligned} P^T(q)Q(p) &= \begin{bmatrix} q^T p & q^T\mathcal{E}^T(p) \\ \mathcal{G}(q)p & \mathcal{G}(q)\mathcal{E}^T(p) \end{bmatrix} \\ &= \begin{bmatrix} \alpha & u^T \\ v & A \end{bmatrix} = \begin{bmatrix} \alpha - u^T A^{-1} v & u^T \\ 0 & A \end{bmatrix} \begin{bmatrix} 1 & 0 \\ A^{-1}v & I_3 \end{bmatrix}, \end{aligned} \tag{13.101}$$

with definitions $\alpha = q^T p$, $u = \mathcal{E}(p)q$, $v = \mathcal{G}(q)p$, and $A = \mathcal{G}(q)\mathcal{E}^T(p)$. Set $w = A^{-1}v = -(1/p^T q)\mathcal{E}(p)q = -(1/p^T q)u$ as it is verified with

$$\begin{aligned} Aw &= -(1/p^T q)\mathcal{G}(q)\mathcal{E}^T(p)\mathcal{E}(p)q \\ &= -(1/p^T q)\mathcal{G}(q)[\|p\|^2 I_4 - pp^T]q = \mathcal{G}(q)p \\ &= v, \end{aligned} \tag{13.102}$$

where the Lemma 13.23 was used for the product $\mathcal{E}^T(p)\mathcal{E}(p)$. Then, set $\alpha - u^T A^{-1}v = p^T q - u^T w = q^T q + (1/q^T p)u^T u$, and finally, this simplifies to $\|q\|^2\|p\|^2/(p^T q)$. Therefore,

$$\det P^T(q)Q(p) = \|q\|^4\|p\|^4 = \frac{\|q\|^2\|p\|^2}{p^T q}\det A \tag{13.103}$$

and thus, $\det A = p^T q\|p\|^2\|q\|^2$. □

13.9 Differential calculus of quaternions

Because quaternion multiplication is non-commutative, many of the important results of differential calculus do not carry over. For instance, using the matrix representation of the quaternion algebra for instance, the following sum and product rules apply to quaternions functions $q(t), p(t) \in \mathbb{H}$

$$\begin{aligned} \frac{d}{dt}(q+p) &= \dot{q} + \dot{p} \\ \frac{d}{dt}(q \star p) &= \dot{q} \star p + q \star \dot{p}. \end{aligned} \tag{13.104}$$

However, the chain rule does not hold because of the non-commutative quaternion product.

Analytic functions of quaternions can also be defined in terms of known power series, though the results are sometimes surprising. For instance, using Lemma 13.12 to compute the powers of matrix $Q(q), q \in \mathbb{H}$ yields

$$Q^n(q) = \|q\|^{n-1}U_{n-1}(x)Q(q) - \|q\|^n U_{n-2}(x)I_4, \tag{13.105}$$

where $x = q_0/\|q\|$. An analytic function $f : \mathbb{H} \mapsto \mathbb{H}$, can be evaluated using the coefficients of the power series of f on Q and (13.105) for the powers Q^n, summing the results in matrix $f(Q)$, and map the result back to quaternion space. This procedure yields Lemma 13.28 below.

Lemma 13.28. *Given an analytic function $f : \mathbb{C} \mapsto \mathbb{C}$, defined over the complex plane with the series expansion $f(x) = \sum_{n=0} a_n x^n$, then, the function can be extended to $f : \mathbb{H} \mapsto \mathbb{H}$ via the series*

$$f(q) = \left[1 - \sum_{n=0}^{\infty} a_{n+2}\|q\|^{n+2}U_n(x)\right]\mathrm{h} + \left[\sum_{n=0}^{\infty} a_{n+1}\|q\|^n U_n(x)\right]q, \tag{13.106}$$

where $x = q_0/\|q\|$, and $U_n(x)$ is the Chebyschev polynomial of the second kind.

Proof. Using (13.105) yields the result in terms of matrix $Q(q)$. From the definition of matrix $Q(q)$ in (13.62), it follows that a matrix of the form $\alpha Q(q) + \beta I_4$ corresponds to the quaternion $\alpha q + \beta \mathrm{h}$, for real scalars, $\alpha, \beta \in \mathbb{R}$. □

Example 13.1. *To get an idea of how this works, consider a quaternion $q \in \mathbb{H}$ with $q_0 = 0$ so that $x = q_0/\|q\| = 0$, and the exponential function $\exp(x) = \sum x^n/n!$. This will keep the illustration simple. Using the trigonometric definition of the Chebyschev polynomials*

$$U_n(x) = \frac{\sin([n+1]\theta)}{\sin(\theta)}, \qquad \textit{where } x = \cos(\theta), \tag{13.107}$$

explicitly compute

$$U_n(0) = \frac{\sin([n+1]\pi/2)}{\sin(\pi/2)} = \begin{cases} 0 & \textit{if } n \textit{ is odd} \\ (-)^j & \textit{for } n = 2j \ . \end{cases} \tag{13.108}$$

Applying (13.106) *with $a_n = 1/n!$ yields*

$$\begin{aligned} \exp(q) &= \left[1 - \|q\|^2 \sum_{n=0}^{\infty} a_{n+2} \|q\|^n U_n(0)\right] \mathrm{h} + \left[\sum_{n=0}^{\infty} a_{n+1} \|q\|^n U_n(0)\right] q \\ &= \left[1 + \sum_{n=0}^{\infty} (-)^{n+1} \frac{\|q\|^{2n+2}}{(2n+2)!}\right] \mathrm{h} + \left[\frac{1}{\|q\|} \sum_{n=0}^{\infty} (-)^n \frac{\|q\|^{2n+1}}{(2n+1)!}\right] q \\ &= \cos(q) + \frac{\sin(q)}{\|q\|} q. \end{aligned} \tag{13.109}$$

It follows that for any scalar $\alpha \in \mathbb{R}$,

$$\exp(\alpha q) = \cos(\alpha \|q\|)\mathrm{h} + \frac{\sin(\alpha \|q\|)}{\|q\|} q. \tag{13.110}$$

In particular, if $\alpha = t$ and q is constant, the time derivative is evaluated as

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} \exp(tq) &= -\|q\| \sin(t\|q\|) + \cos(t\|q\|) q \\ &= q \star \exp(tq) = \exp(tq) \star q \end{aligned} \tag{13.111}$$

since for a pure imaginary quaternion q we have $q \star q = -\|q\|$. This is the familiar rule for the exponential function.

Taking a time-dependent pure imaginary quaternion such as

$$q(t) = \begin{bmatrix} 0 \\ \sin(\alpha t) \\ \cos(\alpha t) \\ 0 \end{bmatrix}, \tag{13.112}$$

then, $\mathtt{exp}(q(t)) = \mathtt{sin}(1)\mathtt{h} + \mathtt{cos}(1)q$ *and we find that* $\frac{\mathrm{d}}{\mathrm{d}t}\,\mathtt{exp}(q) = \mathtt{cos}(1)\dot{q}$. *However,*

$$q \star \dot{q} = \alpha \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \tag{13.113}$$

and so

$$\mathtt{exp}(q)\dot{q} = \mathtt{sin}(1)\dot{q} + \alpha\,\mathtt{cos}(1) \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \neq \mathtt{cos}(1)\dot{q} = \frac{\mathrm{d}}{\mathrm{d}t}\,\mathtt{exp}(q). \tag{13.114}$$

This example demonstrates that simple quaternion differential equations cannot be solved using well known analytic methods such as integrating factors.

13.10 Angular velocity

The connection between the time derivatives of the quaternions and the angular velocity of the corresponding rotation matrix $R(q(t))$ is now established. Consider a time dependent quaternion $q(t) \in \mathbb{H}, q^\dagger \star q = q \star q^\dagger = 1$. Since $q^\dagger \star q = 1$ for all times t, assuming that $q(t)$ is a differentiable function of time, it follows that

$$0 = \frac{d}{dt}\left(q \star q^\dagger\right) = \dot{q} \star q^\dagger + q \star \dot{q}^\dagger = \dot{q} \star q^\dagger + (\dot{q} \star q^\dagger)^\dagger, \tag{13.115}$$

and similarly for $q \star q^\dagger = 1$. Define $r, s \in \mathbb{H}$, with $r = \dot{q} \star q^\dagger$, and $s = q^\dagger \star \dot{q}$. It follows from (13.115) that both r and s are pure imaginary, i.e., $r^\dagger = -r$ and $s^\dagger = s$.

Next, consider the 4×4 matrix product $A(q) = Q(q)P^T(q)$ for a time dependent unit quaternion $q(t) \in \mathbb{H}, t \in \mathbb{R}$. The time derivative of $A(q(t))$ is computed as follows:

$$\begin{aligned} \dot{A} &= \dot{Q}P^T + Q\dot{P}^T \\ &= \dot{Q}Q^T Q P^T + Q\dot{P}^T P P^T \\ &= \left(\dot{Q}Q^T + \dot{P}^T P\right) A \\ &= \left(Q(\dot{q} \star q^\dagger) + P^T(\dot{q} \star q^\dagger)\right) A(q), \end{aligned} \tag{13.116}$$

using $QQ^T = Q^T Q = PP^T = P^T P = I_4$ for a unit quaternion q, the result of Theorem 13.16 to commute matrices P and Q, as well as (13.55) and (13.59) in the last step. Now, given any quaternion $r \in \mathbb{H}$,

$$\begin{aligned} Q(r) + P^T(r) &= \begin{bmatrix} r_s & -r_v^T \\ r_v & r_s I_3 + \widehat{r}_v \end{bmatrix} + \begin{bmatrix} r_s & r_v^T \\ -r_v & r_s I_3 + \widehat{r}_v \end{bmatrix} \\ &= \begin{bmatrix} 2r_s & 0 \\ 0 & 2r_s + 2\widehat{r}_v. \end{bmatrix} \end{aligned} \tag{13.117}$$

Specializing to the case where $r_s = 0$ which is the case for $r = \dot{q} \star q^\dagger$ as shown in (13.117), the result is

$$\begin{aligned} \dot{A} &= \begin{bmatrix} 0 & 0 \\ 0 & 2\widehat{r}_v \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \mathcal{G}\mathcal{E}^T \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 \\ 0 & 2\widehat{r}_v R(q) \end{bmatrix} \end{aligned} \tag{13.118}$$

and this can be processed further to read $\dot{R} = 2\widehat{r}_v R$, where $r = \dot{q} \star q^\dagger$ and $R(q)$ is defined as before. This gives the immediate identity $2r_v = \omega$, where $\omega \in \mathbb{R}^3$ is the angular velocity vector, and this is expanded further to yield the differential equation

$$\begin{aligned} \dot{q} &= q \star r = Q(q)r \\ &= \begin{bmatrix} q_s & -q_v^T \\ q_v & q_s I_3 + \widehat{q}_v \end{bmatrix} \begin{bmatrix} 0 \\ r_v \end{bmatrix} = \frac{1}{2}\mathcal{E}^T(q)\omega. \end{aligned} \tag{13.119}$$

Observe now that with $r = \dot{q} \star q^\dagger$ and $s = q^\dagger \star \dot{q}$, we have $r = q \star s \star q^\dagger$, which corresponds to having $\omega = R(q)\omega'$, where ω' is the angular velocity as seen in the body frame. Simple manipulations yield

$$\dot{q} = s \star q = P(q)r = \frac{1}{2}\mathcal{G}^T(q)\omega'. \tag{13.120}$$

The content of (13.119) and (13.120) is the main result of this section establishing a vector differential equation relating a quaternion with the angular velocity vector.

Example 13.2. *A simple example of the use of the previous equation is to solve for the motion of a rigid frame rotating at fixed angular velocity* $\omega \in \mathbb{R}^3, \omega = const.$ *Write* $r \in \mathbb{H}$ *so that* $r_s = 0, r_v = \omega$. *Set* $q = \exp[(t/2)r]$ *so that according to* (13.111),

$$\begin{aligned} \dot{q} &= \frac{1}{2}\exp[(t/2)r] \star r = \frac{1}{2} q \star r \\ &= \frac{1}{2}Q(q)r = \frac{1}{2}\begin{bmatrix} q & \mathcal{E}^T(q) \end{bmatrix} \begin{bmatrix} 0 \\ \omega \end{bmatrix} = \frac{1}{2}\mathcal{E}^T(q)\omega. \end{aligned} \tag{13.121}$$

13.11 Other representations of the rotation matrices

Rotation matrices are often represented with a set of 3 angles—the Euler angles. As shown in this section, there are 24 possible conventions for these angles, each of which having singular configurations so that multiple angles define the same rotation matrix. This prevents integration of the differential equation $\dot{R} = \widehat{\omega}R$ in terms of Euler angles unless the convention is changed near when approaching a singularity.

Starting from (13.81), the cases where n is a basis vectors are easily computed to be

$$R_1(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix}, \quad R_2(\psi) = \begin{bmatrix} \cos(\psi) & 0 & \sin(\psi) \\ 0 & 1 & 0 \\ -\sin(\psi) & 0 & \cos(\psi) \end{bmatrix},$$
$$R_3(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{13.122}$$

A general rotation matrix R can be expressed as the product of three rotation matrices either as $R = R_{i_1}(\phi)R_{i_2}(\theta)R_{i_3}(\psi)$ with three different $i_j, j = \{1,2,3\}$, or $R = R_{i_1}(\phi)R_{i_2}(\theta)R_{i_1}(\psi)$. There are $3 \times 2 \times 1 = 6$ possibilities for the case with three different axes and $3 \times 2 = 6$ more for the case with the repeated axes. There are also two sign conventions for the angles corresponding to whether they are measured with respect to the fixed or rotated frame, bringing the total to 24 different conventions.

Next, given that $\dot{R} = \widehat{\omega} R$, and given that R is a function of the time dependent angles, $\phi(t), \theta(t)$ and $\psi(t)$ say, expression of the form

$$T(\phi, \theta, \psi) \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \omega, \tag{13.123}$$

is anticipated, where matrix $T(\phi, \theta, \psi)$ has size 3×3. Of course, when matrix T is degenerate, it is not possible to integrate the angles.

To construct matrix $T(\phi, \theta, \phi)$ in (13.123), fist note that for the matrices $R_i(\phi)$, the time derivatives are

$$\dot{R}_i(\phi) = \dot{\phi}\widehat{e}_i R_i(\phi), \tag{13.124}$$

where $e_i \in \mathbb{R}^3$ is the unit vector with a one in position i and zero everywhere else.

Write a general rotation matrix as $R = R_{i_3}(\phi_3)R_{i_2}(\phi_2)R_{i_1}(\phi_1)$ so that

$$\begin{aligned} \dot{R} &= \dot{R}_{i_3} R_{i_2} R_{i_1} + R_{i_3} \dot{R}_{i_2} R_{i_1} + R_{i_3} R_{i_2} \dot{R}_{i_1} \\ &= \left[\dot{R}_{i_3} R_{i_3}^T\right] R_{i_3} R_{i_2} R_{i_1} + \left[R_{i_3}\left(\dot{R}_{i_2} R_{i_2}^T\right) R_{i_3}^T\right] R_{i_3} R_{i_2} R_{i_1} \\ &\quad + \left[R_{i_3} R_{i_2}\left(\dot{R}_{i_1} R_{i_1}^T\right) R_{i_2}^T R_{i_3}^T\right] R_{i_3} R_{i_2} R_{i_1} \\ &= \left[\dot{\phi}_3 \widehat{e}_{i_3}\right] R + \left[\dot{\phi}_2 R_{i_3} \widehat{e}_{i_2} R_{i_3}^T\right] R + \left[\dot{\phi}_1 R_{i_3} R_{i_2} \widehat{e}_{i_1} R_{i_2}^T R_{i_3}^T\right] R, \end{aligned} \tag{13.125}$$

and the antisymmetric matrix $\widehat{\omega}$ is extracted as

$$\widehat{\omega} = \left(\dot{\phi}_3 \widehat{e}_{i_3}\right) + \left(\dot{\phi}_2 R_{i_3} \widehat{e}_{i_2} R_{i_3}^T\right) + \left(\dot{\phi}_3 R_{i_3} R_{i_2} \widehat{e}_{i_1} R_{i_2}^T R_{i_3}^T\right). \tag{13.126}$$

Using Lemma 13.5 again, the angular velocity vector ω reduces to

$$\omega = \dot{\phi}_3 e_{i_3} + \dot{\phi}_2 R_{i_3} e_{i_2} + \dot{\phi}_1 R_{i_3} R_{i_2} e_{i_1}. \tag{13.127}$$

Rewriting this as a matrix equation, $\omega = T\dot{\alpha}$, using $\dot{\alpha} = (\dot{\phi}_1, \dot{\phi}_2, \dot{\phi}_3)^T$, the columns of T are easily identified

$$T_{\bullet 1} = e_{i_3}, \quad T_{\bullet 2} = R_{i_3} e_{i_2}, \quad \text{and } T_{\bullet 3} = R_{i_3} R_{i_2} e_{i_1}. \tag{13.128}$$

As an example, consider the 3-1-3 convention used in Goldstein [105] for instance, where $i_1 = i_3 = 3, i_2 = 1$, and the angles are denoted ϕ, θ and ψ respectively. This yields

$$T = \begin{bmatrix} 0 & \cos(\psi) & \sin(\psi)\sin(\theta) \\ 0 & \sin(\psi) & -\cos(\psi)\sin(\theta) \\ 1 & 0 & \cos(\theta) \end{bmatrix}. \tag{13.129}$$

Matrix T is degenerate when $\theta = 0$ as it reduces to

$$T|_{\theta=0} = \begin{bmatrix} 0 & \cos(\psi) & 0 \\ 0 & \sin(\psi) & 0 \\ 1 & 0 & 1 \end{bmatrix}, \tag{13.130}$$

which has two identical columns. This means that $\dot{\alpha}$ cannot be recovered using $\dot{\alpha} = T^{-1}\omega$ to integrate the angles as function of angular velocities. All the 24 Euler angle conventions suffer from this type of singularity, though at different locations. The reason is that the group of rigid rotations $SO(3)$ is a manifold which is curved, i.e., it is not isomorphic to $\mathbb{R}^3$, and therefore, there is no global three-dimensional chart for it.

Given one of the Euler angle conventions of Section 13.11, expressed in the general for $R(\phi, \theta, \psi) = R_{i_3}(\psi) R_{i_2}(\theta) R_{i_1}(\phi)$, there is a corresponding quaternion $q(\phi, \theta, \psi) = q_{i_3}(\psi) \star q_{i_2}(\theta) \star q_{i_1}(\phi)$, and it is therefore possible to recover the angles for an arbitrary convention but this is not pursued further here.

13.12 End notes

A brief analysis of the quaternion representation of $SO(3)$ is found in Haug [118] but this book is long since out of print. In addition, it concentrates on the analysis of the rotation factors $\mathcal{E}(q), \mathcal{G}(q)$ and the formulae $R(q) = \mathcal{E}(q)\mathcal{G}^T(q)$, $\dot{q} = (1/2)E^T(q)\omega$, and $\dot{q} = (1/2)G^T\omega'$, specifically in order to use the variational principle on the rigid body Lagrangian. The complete representation of the quaternion algebra with the $Q(q)$ and $P(q)$ matrices is not provided.

Tasora [263] is well aware of all this algebra when exploring quaternion-based formulations of kinematic constraints, as is also done in Chapter 14, but it seems that his reference on the topic are course notes in Italian, according to his reference list.

Goldstein [105] does provide a long chapter on rigid body kinematics covering several alternative representations. However, his comments on quaternions are limited to the snide remark: "The connossieur [*sic*] of somewhat musty mathematics will recognize in [...] a representation of Q as a matrix *quaternion*, ...". He proceeds with a representation relying on the Cayley-Klein parameters organized in 2×2 complex matrices, instead of the explicit 4×4 *real* matrices used herein. In addition, he only provides formulae connecting the rate of change of the Euler angles to the angular velocities but he does not provide the similar computation for the Cayley-Klein parameters, though he does provide the matrix representation (13.77).

Euler-Rodrigues parameters are well covered by Angeles and Kecskeméthy [10], but these parameters, even though there is four of them, much like the quaternions, have singular points in their representation. In fact, the relationship is

$$r_0 = \pm\sqrt{\frac{1+q_s}{2}}, \qquad r_v = \frac{1}{2r_0}q_v. \tag{13.131}$$

There does not seem to be a specific advantage to this, especially since there is a problem when q_s is near -1, since $r_0 \to 0$ then, and r_v becomes ill-behaved.

In the graphics literature, one often sees the equation

$$\dot{q} = \frac{1}{2}\omega q, \tag{13.132}$$

without any comment as to what is meant by ω or the product ωq, except that both ω and q are quaternions. The exact formulae (13.119) and (13.120) with complete definition of the matrices $\mathcal{E}(q)$ are $\mathcal{G}(q)$ are found in lecture notes here and there but one has to dig.

The relation $\dot{q} = (1/2)\mathcal{E}(q)^T\omega$ is in fact a description of the tangent bundle TQ for $Q = \mathbb{H}$. Indeed, for arbitrary $\omega \in \mathbb{R}^3$, the velocity $\dot{q}$ defined in (13.119) makes $\dot{q}$ tangent to $\mathbb{H}$ at the point $q \in \{p \in \mathbb{H} \mid \|p\| = 1$.

The identities developed in this chapter are very useful in the analysis of rigid body rotational constraints. Some examples are provided in Chapter 14.

It is hard to say which of the results of this chapter are new and which are not. Since no literature could be found other than the short section in Haug's book [118], everything was derived from scratch and that process required the derivation of the various lemmata. The final expressions for the rotation matrix found in Section 13.6 and the angular velocity in Section 13.10 are well known. The specific representation of the quaternion algebra with the $\mathcal{Q}(q)$ and $\mathcal{P}(q)$ matrices is obviously known to Tasora [263] but the results above were derived before any knowledge of that paper and textbooks or articles were of little help. In fact, all formulas presented in this chapter were derived out of necessity because the literature was insufficient.

The explicit evaluation of analytic functions of quaternions is probably related to the Baker-Campbell-Hausdorff formula and the calculus of Lie derivatives [112] but none of these techniques were used in the derivation. Novel or not, what is certain is that a collection of the results and identities needed for the analysis

of rigid body kinematics is not easily found. This motivated the inclusion of the present chapter.

14 Rigid Bodies III: Constraint Kinematics

As discussed in Section 3.14 and in Chapter 4, kinematic constraints are restrictions on the motion of a mechanical system which account geometrically for the net effect of physics occuring at time and length scales which are not essential to the analysis.

Several mechanical assemblies can be realized to attach bodies together. These are known as joints. Though a complete mathematical description of the usual joints is not provided here, several rotational joints are analyzed in details using the quaternion algebra of Chapter 13. After motivating a quaternion-based strategy and introducing the main concepts in Section 14.1, a full rotational lock joint is described in Section 14.2. The reason for this is that individual constraint equations can be removed to produce joints which do have some rotational degrees of freedom. This is done in Section 14.3 where a hinge or revolute joint is defined, in Section 14.4 where a Hooke's or universal or Cartan joint is defined, and in Section 14.5 where a homokinetic joint is defined. All these representations use the same quaternion-based formulation. An alternative representation of constraints is discussed in Section 14.6 and general remarks are provided in Section 14.7.

14.1 Quaternion-based rotational constraints

Assuming a decoupling between the translational and rotational degrees of freedom connecting two rigid bodies (or two frames in general), there are several constraints which can be expressed directly in terms of the quaternion which expresses the relative rotation between the two frames.

The framework described below is very general and can account for certain types of rheonomic constraints in which attachment points are moving, for instance, though this is not done here.

The advantage of the quaternion representation is that it is free of certain singularities as explained in Section 14.6.

First express the relative quaternion between bodies two bodies labeled $1, 2$, in terms of the rotational coordinates of each rigid body and an *joint attachment* frame which may or may not be fixed in the body frame. In the case where it is not fixed, it is assumed that the motion of the attachment frame is fully specified kinematically, i.e., the angular velocities are known for any given time.

Let $q^{(1)}, q^{(2)} \in \mathbb{H}$ and $x^{(1)}, x^{(2)} \in \mathbb{R}^3$ represent the quaternions of body $1, 2$, respectively, relative to an inertial frame of reference. Let $p^{(1)}, p^{(2)} \in \mathbb{H}$, and

$y^{(1)}, y^{(2)} \in \mathbb{R}^3$ describe the orientation and position of attachment frames of body $1, 2$, respectively, expressed in the frame of reference of body $1, 2$, respectively. With our conventions, the quaternion $q^{(1)} \star p^{(1)}$ corresponds to transformation from the attachment frame of body 1 to the inertial frame. Therefore, the transformation which goes from body 1 to body 2 is given by the expression:

$$\begin{aligned} x &= x^{(2)} + R(q^{(2)})(y^{(2)} - x^{(1)}) - R(q^{(1)}y^{(1)}), \\ q &= (q^{(1)} \star p^{(1)})^{\dagger} \star (q^{(2)} \star p^{(2)}) = p^{\dagger\,(1)} \star q^{\dagger\,(1)} \star q^{(2)} \star p^{(2)}. \end{aligned} \tag{14.1}$$

In addition, we write the quaternion velocities $r^{(i)}, s^{(i)} \in \mathbb{H}$ so that $\dot{q}^{(i)} = \frac{1}{2} q^{(i)} \star r^{(i)}$, and $\dot{p}^{(i)} = \frac{1}{2} p^{(i)} \star s^{(i)}$. We will project the equations down to the more familiar 3D angular velocity vectors later on.

Now, any holonomic rotational constraint can be expressed as some function $g : \mathbb{H} \times T \mapsto \mathbb{R}^m, g(q,t) = 0$. Note that since $\|q\| = 1$, q describes 3 degrees of freedom and so $0 < m \leq 3$. To find the Jacobian of $g(q,t)$ in terms of the generalized coordinates $q^{(1)}, q^{(2)}$, it suffices to find $J^{(1)}, J^{(2)}$ such that $\dot{q} = J^{(1)}\omega^{(1)} + J^{(2)}\omega^{(2)}$, where $\dot{q}^{(i)} = \frac{1}{2}\mathcal{E}(q^{(i)})\omega^{(i)}, \omega^{(i)} \in \mathbb{R}^3$. For more complicated expressions, we use the chain rule to get:

$$\begin{aligned} \frac{dg(q,t)}{dt} &= \frac{\partial g}{\partial q}\left(J^{(1)}\omega^{(1)} + J^{(2)}\omega^{(2)}\right) + \frac{\partial g}{\partial t} \\ &= G^{(1)}\omega^{(1)} + G^{(2)}\omega^{(2)} + \frac{\partial g}{\partial t}, \end{aligned} \tag{14.2}$$

so that $G^{(i)} = GJ^{(i)}$ where $G = \dfrac{\partial g}{\partial q}$. Note that $q \in \mathbb{H}$ is treated strictly as a four-dimensional vector here and all quaternion operations are understood as matrix-vector operations in $\mathbb{R}^4$.

Lemma 14.1. *Given two quaternions $q, p \in \mathbb{H}$, with inertial frame angular velocities $r, s \in \mathbb{H}$, respectively, the product*

$$u = q^{\dagger} \star p = (q^{(1)} \star f^{(1)}) \star (q^{(2)} \star f^{(2)}), \tag{14.3}$$

has the following time derivative:

$$\dot{u} = \frac{1}{2}\mathcal{Q}^T(q)\mathcal{P}(p)s - \frac{1}{2}\mathcal{Q}^T(q)\mathcal{P}(p)r. \tag{14.4}$$

Proof. Using quaternion algebra and the chain rule, noting first that $\dot{q}^{\dagger} = \frac{1}{2}(r \star q)^{\dagger} = -\frac{1}{2}q^{\dagger} \star r$, since $r^{\dagger} = -r$ as we showed before in (13.115). Therefore, we have:

$$\begin{aligned} \dot{u} &= q^{\dagger} \star \dot{p} + \dot{q}^{\dagger} \star p \\ &= \frac{1}{2}q^{\dagger} \star s \star p - \frac{1}{2}q^{\dagger} \star r \star p. \end{aligned} \tag{14.5}$$

Now, using the matrix representation of Section 13.5 for the quaternion product, select the left ordered product (13.51) for q factors and the right ordered product (13.52) for factors involving p yields the result. □

For a more general case, we introduce the notion of *attachment frames* on each body. These frames have orientation quaternions $f^{(i)} \in \mathbb{H}$ in the frame of body i, respectively, and they have a prescribed angular velocity, i.e., $2v^{(i)}(t) = \dot{f}^{(i)} f^{\dagger\,(i)}$ is given. Now, body i has orientation quaternion $q^{(i)}$ with quaternion angular velocity $2r^{(i)} = \dot{q}^{(i)} q^{\dagger\,(i)}$. The quaternion which relates the frame on body 2 to that on body i is given simply by:

$$p = (q^{(1)} \star f^{(1)})^{\dagger} \star (q^{(2)} \star f^{(2)}) = f^{\dagger\,(1)} \star q^{\dagger\,(1)} \star q^{(2)} \star f^{(2)}. \tag{14.6}$$

The time derivative $\dot{p}$ can then be expressed as a function of $v^{(i)}, r^{(i)}$ using the following:

Lemma 14.2. *Given four unit quaternions $q^{(i)}, f^{(i)} \in \mathbb{H}, i = \{1, 2\}$ which have angular quaternion velocity $2r^{(i)} = \dot{q}^{(i)} \star q^{\dagger\,(i)}$ and $2v^{(i)} = \dot{f}^{(i)} \star f^{\dagger\,(i)}$, respectively. Write $p^{(1)} = q^{(1)} \star f^{(1)}$ and $p^{(2)} = q^{(2)} \star f^{(2)}$ and put $q = p^{\dagger\,(1)} \star p^{(2)}$. Then, $\dot{q}$ is given by:*

$$\begin{aligned} \dot{q} = &-\frac{1}{2}\mathcal{P}(u)\mathcal{Q}^T(f^{(1)})\mathcal{P}(f^{(1)})v^{(1)} - \frac{1}{2}\mathcal{Q}^T(p^{(1)})\mathcal{P}(p^{(2)})r^{(1)} \\ &+ \frac{1}{2}\mathcal{Q}^T(p^{(1)})\mathcal{P}(p^{(2)})r^{(2)} + \frac{1}{2}\mathcal{Q}(q)\mathcal{Q}^T(f^{(2)})\mathcal{P}(f^{(2)})v^{(2)}, \end{aligned} \tag{14.7}$$

or using vector angular velocity vectors defined as $\omega^{(i)} = {r_v}^{(i)}$ and $\omega^{(0,i)} = {v_v}^{(i)}$, and using the previous definition for $R(q)$ in (13.73)*:*

$$\begin{aligned} \dot{q} = &-\frac{1}{2}\mathcal{E}^T(q)R^T(f^{(1)})\omega^{(0,1)} - \frac{1}{2}\mathcal{Q}^T(p^{(1)})\mathcal{E}^T(p^{(2)})\omega^{(1)} \\ &+ \frac{1}{2}\mathcal{Q}^T(p^{(1)})\mathcal{E}^T(p^{(2)})\omega^{(2)} + \frac{1}{2}\mathcal{G}^T(q)R^T(f^{(2)})\omega^{(0,2)}, \end{aligned} \tag{14.8}$$

and finally, the same relation can be expressed using the body frame angular velocities $\omega'^{(i)} = R(q^{(i)})\omega^{(i)}$ and $\omega'^{(0,i)} = R(f^{(i)})\omega^{(0,i)}$:

$$\begin{aligned} \dot{q} = &-\frac{1}{2}\mathcal{E}^T(q)\omega'^{(0,1)} - \frac{1}{2}\mathcal{E}^T(q)R^T(f^{(1)})\omega'^{(1)} \\ &+ \frac{1}{2}\mathcal{G}^T(q)\omega'^{(0,2)} + \frac{1}{2}\mathcal{G}^T(q)R^T(f^{(2)})\omega'^{(2)}. \end{aligned} \tag{14.9}$$

Proof. Starting from $q = f^{\dagger\,(1)} \star q^{\dagger\,(1)} \star q^{(2)} \star f^{(2)}$, the product rule yields four terms. The first one reads: $\dot{f}^{\dagger\,(1)} \star q^{\dagger\,(1)} \star q^{(2)} \star f^{(2)}$. Using the definition of $v^{(1)}$ and the fact that $f^{\dagger\,(1)} f^{(1)} = 1$, this term becomes:

$$\dot{f}^{\dagger\,(1)} \star q^{\dagger\,(1)} \star q^{(2)} \star f^{(2)} = -\frac{1}{2} f^{\dagger\,(1)} \star v^{(i)} \star f^{(1)} \star q, \tag{14.10}$$

and using matrices P, Q of (2.29), this yields the first term of (14.7). Similar simple algebraic manipulations apply to each of the four terms and the result follows. □

It is interesting to note here that the natural frame of reference for the angular velocity of attachment frames is *not* the world coordinates but the frame itself.

We now proceed to apply this simple result to compute Jacobians of non-trivial kinematic constraints for rigid bodies.

14.2 A full kinematic control constraint

Using the result of the previous section, we can now specify a constraint equation for a fully kinematically controlled constraint between two bodies. Using the result of Lemma 14.2, the 3-dimensional constraint: $g(q,t) = \mathcal{G}(p)q = 0$, where q is defined as in (14.3), and $p \in \mathbb{H}, \|p\| = 1$, is a time-dependent unit quaternion—the desired relative orientation between the frame attached on body 1 and that attached on body 2—which is given. If the condition $g(q,t) = 0$ is maintained, Lemma 13.22 guarantee that $q(t) = \pm p(t)$. However, since $R(q) = R(-q)$, this identity guarantees that the relative rotation between bodies 1 and 2 is in fact $R(p(t))$ as required. Some simple algebra yields the following definition of the constraint Jacobian:

$$\begin{aligned}
\dot{g}(q,t) &= G^{(1)}\omega^{(1)} + G^{(2)}\omega^{(2)} + s(t), \\
G^{(1)} &= -\frac{1}{2}\mathcal{G}(p^{(1)} \star p)\mathcal{E}^T(p^{(2)}), \\
G^{(2)} &= -G^{(1)} = \frac{1}{2}\mathcal{G}(p^{(1)} \star p)\mathcal{E}^T(p^{(2)}), \\
s(t) &= -\frac{1}{2}\mathcal{G}(p)\mathcal{E}^T(q)R^T(f^{(1)})\omega^{(0,1)}, \\
&\quad + \frac{1}{2}\mathcal{G}(p)\mathcal{G}^T(q)R^T(f^{(2)})\omega^{(0,2)} - \frac{1}{2}\mathcal{G}(q)\mathcal{E}^T(p)\omega^{(r)},
\end{aligned} \tag{14.11}$$

where $\omega^{(0,i)}, i \in \{1,2\}$, are the angular velocity vectors of the attachment frames, and $\omega^{(r)}$ is the angular velocity vector of the relative quaternion. All velocities are expressed in absolute coordinates.

Tasora [263] has derived a similar formula but his result is far more complicated than the one given above. In addition, we clearly see that the attachment angular velocities should be specified in the reference frame of the attachment to save computations. Finally, observe that (14.11) is a set of three linear equations in $\omega^{(i)}$ which are guaranteed to result in $R(q) = R(p)$ if $g(q,t) = 0$, as per Lemma 13.22. Tasora arbitrarily selects three equations and it is not clear that this choice can work when there is significant constraint violation.

With the current representation, that according to the result of Lemma 13.27, the constraint Jacobian blocks have determinant $\det(G^{(i)})$ is proportional to $p^{(2)} \cdot \left(p^{(1)} \star p\right)$. Near the point where the constraint is satisfied, this determinant is ± 1 and the blocks have full row rank.

14.3 Quaternion representation of a hinge constraint

A hinge is a mechanism which only allows rotations about an axis $n \in \mathbb{R}^3$ which is fixed in the constraint frame. From (13.81) a rotation by θ about a fixed axis is given by the quaternion:

$$q_n(\theta) = \begin{bmatrix} \cos(\theta/2) \\ \sin(\theta/2)n \end{bmatrix}. \tag{14.12}$$

If we have vectors $a, b \in \mathbb{R}^3$, constant in the frame of the hinge and such that (a, b, c) forms a right handed orthonormal basis of $\mathbb{R}^3$, then, the constraint can be expressed as

$$\begin{aligned} a^T q_v &= 0, \\ b^T q_v &= 0. \end{aligned} \tag{14.13}$$

Theorem 14.3. *Given a unit quaternion $q \in \mathbb{H}, \|q\| = 1$, and a right handed orthonormal basis of $a, b, c \in \mathbb{R}^3$, then, q describes a rotation about c if and only if the following two conditions hold:*

$$\begin{aligned} a^T q_v &= 0, \\ b^T q_v &= 0. \end{aligned} \tag{14.14}$$

Proof. For the "if" part, since $q_v \in \mathbb{R}^3$, we can always write $q_v = \beta_a a + \beta_b b + \beta_c c$. From the conditions of (14.14), $\beta_a = \beta_b = 0$. Now, since $\|q\|^2 = 1$, we have $q_s^2 + \beta_c^2 = 1$ so we can find some angle $\phi \in [0, 4\pi]$ so that $q_s = \cos(\phi/2), \beta = \sin(\phi/2)$. The rotation described by this quaternion is given by (13.81) and is a rotation by ϕ about c.

For the "only if" part, any rotation matrix $R \in SO(3)$, can be written as $R = (2q_s^2 - 1) + 2q_v {q_v}^T + 2q_s \widehat{q}_v$, where q_v is the axis of rotation. It follows that if $q_v = c$, then, the conditions of (14.14) hold. □

It is interesting to note that a more usual definition is stated in terms of the rotation matrix directly as:

$$\begin{aligned} a^T R(q) c &= 0, \\ b^T R(q) c &= 0. \end{aligned} \tag{14.15}$$

To make the argument simple, take c along the z axis. If $R \in SO(3)$ is a rotation by ϕ about z, then

$$R = R_z(\phi) = \begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{14.16}$$

The conditions of (14.15) correspond to $R_{13} = 0 = R_{23}$. However, consider $R = R_z(\phi) R_x(\pi)$:

$$\begin{aligned} R = R_z(\phi) R_x(\pi) &= \begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \\ &= \begin{bmatrix} \cos(\phi) & \sin(\phi) & 0 \\ \sin(\phi) & -\cos(\phi) & 0 \\ 0 & 0 & -1 \end{bmatrix}. \end{aligned} \tag{14.17}$$

And for this case, we also have $R_{13} = 0 = R_{23}$, even though the matrix R does not correspond to a rotation about z.

The Jacobian matrix for the hinge joint is thus the constant projection

$$\begin{aligned} P_{ab} &= \begin{bmatrix} a^T \\ b^T \end{bmatrix} \\ G^{(1)}_{\text{hinge}} &= P_{ab} G^{(1)} \\ G^{(2)}_{\text{hinge}} &= P_{ab} G^{(2)}. \end{aligned} \tag{14.18}$$

14.4 Quaternion representation of a Hooke's joint

The Hooke's joint a device often used in transmissions. It allows to connect two shafts at some moderate angle so that one shaft drives the other. A Hooke's joint is also known as U-joints, universal joints, Cardan joint, or Hardy-Spicer joint. Hooke's invented this mechanism in 1676 using ideas pioneered by Cardan in 1545. Henry Ford called it the universal joint when he decided to use it in his car designs.

The idea is to transfer the rotational motion of a shaft that is free to rotate about some axis $n^{(1)} \in \mathbb{R}^3$, to a secondary shaft that is free to rotate about another axis $n^{(2)} \in \mathbb{R}^3$. The plane defined by the span of the vectors $n^{(1)}, n^{(2)}$ is invariant in the relative reference frame.

To construct a mathematical description of this type of constraint, we first observe that a universal joint is constructed with two perpendicular hinges which are rigidly attached together, each hinge being connected to a rigid body, namely, one of the two shafts. Therefore, the relative transformation between body 1 and body 2 is a sequence of two rotations about axes $u, v \in \mathbb{R}^3$, where u is fixed in body 1 and so is v in body 2. The perpendicularity condition states that at any time, $u^{(1)} \cdot v^{(2)} = 0$—using the notation $x \cdot y = x^T y$ here to avoid too many superscripts—where $u^{(1)} = R^{(1)} u$, and $v^{(2)} = R^{(2)} v$, and $R^{(i)}$ is the rotation matrix of body i.

The orthogonality condition now reads: $u^T R^{T\,(1)} R^{(2)} v = 0$. Write $R(q) = R^{T\,(1)} R^{(2)}$ for some unit quaternion $q \in \mathbb{H}$, so that

$$R(q) = (2q_s^2 - 1) I_3 + 2 q_v q_v{}^T - 2 q_s \widehat{q}_v. \tag{14.19}$$

Since orthogonality holds for $q = 1$, we find that this is equivalent to $(u^T q_v)(v^T q_v) + q_s u^T \widehat{q}_v v = 0$. After a small amount of algebra, we find the following quadratic form representation

$$\frac{1}{2} q^T A(u, v) q = \frac{1}{2} \begin{bmatrix} q_s & q_v{}^T \end{bmatrix} \begin{bmatrix} 0 & -\frac{1}{2} u^T \widehat{v}, \\ \frac{1}{2} \widehat{v} u & u v^T \end{bmatrix} \begin{bmatrix} q_s \\ q_v \end{bmatrix} = 0. \tag{14.20}$$

We take this as the constraint definition and we then have the following Jacobians

in terms of the product quaternion q and rigid body velocities $\omega^{(i)}$:

$$\begin{aligned}
\dot{g}(q,t) &= q^T A(u,v)\dot{q} = G^{(1)}\omega^{(1)} + G^{(2)}\omega^{(2)} + s(t),\\
G^{(1)} &= -\frac{1}{2}q^T A(u,v)\mathcal{G}(q)R^T(p^{(1)}),\\
G^{(2)} &= \frac{1}{2}q^T A(u,v)\mathcal{E}(q)R^T(p^{(2)}),\\
s(t) &= \frac{1}{2}p^T A(u,v)\left(\mathcal{G}(p)\mathcal{E}^T(q)\omega_p{}^{(1)} - \mathcal{G}(p)\mathcal{G}^T(q)\omega_p{}^{(2)}\right),
\end{aligned} \tag{14.21}$$

where $p^{(i)}$ denotes the quaternions of the attachment points, and $\omega_p{}^{(i)}$ their velocity vectors.

We now provide a detailed description of the kinematics allowed by the universal joint by considering two shafts which are connected together with a universal joint. The first shaft lays along the x axis and is connected at one end by a driver which sets $\omega^{(1)} = const$, and $\omega^{(1)}$ is parallel to the x axis, which we write as $n^{(1)}$. This first rod is attached to the universal joint assembly by a connection pin along $u^{(1)}$. This pin is fixed in body 1 and we take it to be along the z axis originally. Since body 1 rotates along the x axis, we have

$$u^{(1)}(t) = R_x(\phi)u^{(1)}(0) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} u^{(1)}(0) = \begin{bmatrix} 0 \\ -\sin(\phi) \\ \cos(\phi) \end{bmatrix}. \tag{14.22}$$

The universal joint mechanism contains a second connection pin, $v^{(2)}$ which has fixed orientation in body 2 and is orthogonal to $u^{(1)}$. We take this to lie along the y axis in the natural frame of reference of the device. Since the mechanism is free to rotate about $u^{(1)}$, we have

$$v^{(2)} = R_x(\phi)R_u(\alpha)v^{(2)}(0) = R_x(\phi)\begin{bmatrix} -\sin(\alpha) \\ \cos(\alpha) \\ 0 \end{bmatrix} = \begin{bmatrix} -\sin(\alpha) \\ \cos(\phi)\cos(\alpha) \\ \sin(\phi)\sin(\alpha) \end{bmatrix}. \tag{14.23}$$

Likewise, if the shaft of the second body is along the vector $n^{(2)}$, it is free to rotate about the $v^{(2)}$ vector the $n^{(2)}$ axis is then

$$\begin{aligned}
n^{(2)} &= R_x(\phi)R_u(\alpha)R_v(\beta)n^{(2)}(0) = R_x(\phi)R_u(\alpha)\begin{bmatrix} \cos(\beta) \\ 0 \\ \sin(\beta) \end{bmatrix}\\
&= R_x(\phi)\begin{bmatrix} \cos(\alpha)\cos(\beta) \\ \sin(\alpha)\cos(\beta) \\ \sin(\beta) \end{bmatrix} = \begin{bmatrix} \cos(\alpha)\cos(\beta) \\ \cos(\phi)\sin(\alpha)\cos(\beta) - \sin(\phi)\sin(\beta) \\ \sin(\phi)\sin(\alpha)\cos(\beta) + \cos(\phi)\sin(\beta) \end{bmatrix}.
\end{aligned} \tag{14.24}$$

This can be kept in the xy plane if we can set the last equation to zero, and this

is possible with following substitution

$$\begin{aligned} \tau &= \sqrt{\sin^2(\alpha)\cos^2(\beta) + \sin^2(\beta)} = \sqrt{1 - \cos^2(\alpha)\cos^2(\beta)}, \\ \sin(\delta) &= \sin(\beta)/\tau, \\ \cos(\delta) &= \sin(\alpha)\cos(\beta)/\tau. \end{aligned} \tag{14.25}$$

This leads to $\sin(\phi + \delta) = 0$. We can then solve for $\cos(\gamma) = \cos(\alpha)\cos(\beta) = const.$ which leads to $\tau = \sin(\gamma)$ and:

$$n^{(2)} = \begin{bmatrix} \cos(\gamma) \\ \sin(\gamma) \\ 0 \end{bmatrix} = \begin{bmatrix} \cos(\alpha)\cos(\beta) \\ \cos(\phi)\sin(\alpha)\cos(\beta) - \sin(\phi)\sin(\beta) \\ \sin(\phi)\sin(\alpha)\cos(\beta) + \cos(\phi)\sin(\beta) \end{bmatrix}. \tag{14.26}$$

Recall the simplest constraint definition of the Jacobian for the universal joint is: $g(q) = u^{(1)} \cdot v^{(2)} = 0$, leading to the Jacobian

$$\dot{g} = (u^{(1)} \times v^{(2)})^T \omega^{(1)} - (u^{(1)} \times v^{(2)})^T \omega^{(2)} = 0, \tag{14.27}$$

where the notation $x \times y = \widehat{x}y$ is used to simplify the layout. Now, we set everything up so that $\omega^{(i)} = \|\omega^{(i)}\| n^{(i)}$ and therefore, we have:

$$\|\omega^{(2)}\| = \|\omega^{(1)}\| \frac{n^{(1)} \cdot (u^{(1)} \times v^{(2)})}{n^{(2)} \cdot (u^{(1)} \times v^{(2)})} \tag{14.28}$$

Simple algebraic manipulation yield

$$\begin{aligned} n^{(1)} \cdot (u^{(1)} \times v^{(2)}) &= -\cos(\alpha), \\ n^{(2)} \cdot (u^{(1)} \times v^{(2)}) &= -\cos(\alpha)\cos(\gamma) + \sin(\alpha)\cos(\phi)\sin(\gamma). \end{aligned} \tag{14.29}$$

After multiplying both of these by the same factor $\cos(\beta)$ and using the results of (14.25), this is further reduced to

$$\|\omega^{(2)}\| = \|\omega^{(1)}\| \frac{\cos(\gamma)}{1 - \sin^2(\gamma)\sin^2(\phi)}. \tag{14.30}$$

This shows that when the angle between the two shafts is large, significant wobbling occurs.

It is easy to show that the ratio of output to input velocities is bounded by

$$\cos(\gamma) \leq \frac{\|\omega^{(2)}\|}{\|\omega^{(1)}\|} \leq \frac{1}{\cos(\gamma)}, \tag{14.31}$$

and from this, it is easy to check that there is less than 1.5% deviation for angles γ within $\pm 15°$. However, the deviation is more than 20% when γ is near $35°$ and eventually, for $\gamma = 90°$, the mechanism locks.

This is remedied by a homokinetic constraint which is designed precisely so that $\|\omega^{(2)}\| = \|\omega^{(1)}\|$. This is described next.

14.5 Quaternion representation of a homokinetic constraint

As we saw in Section 14.4, the constraint for a universal joint is quadratic in terms of the relative quaternion and in addition to that, the coupling between the angular speeds of the two connected shafts is nonlinear. When the angle between the coupled shafts is within $\pm 15^\circ$, the non-linearity accounts for less than 1 percent but for large angles, universal joints become wobbly.

The solution to this problem is the homokinetic joint and we describe the geometry of this here.

To understand the geometry at work here, consider two shafts which are rotated by angle ϕ about their respective axis and connected together at a common point at an extremity. These two shafts have axes which are aligned along the unit vectors $u^{(i)}$, respectively. The angle between these two vectors is given by $\cos(\theta) = u^{(1)} \cdot u^{(2)}$. Assume that the two shafts lie in the plane defined by vectors $u^{(i)}, v^{(i)}$ at some point in time—this plane can move without affecting the current analysis.

Consider a coordinate system of $\mathbb{R}^3$ with u, v, n as the orthonormal basis. Then, consider the quaternions $q_u(\phi) = (\cos(\phi/2), \sin(\phi/2)u^T)^T$, and $q_n(\phi) = (\cos(\theta/2), \sin(\theta/2)n^T)^T$, which rotate by angle ϕ about axis u and θ about n, respectively. We compute the product $p = q_u^\dagger(\phi) \star q_n(\theta) q_u(\phi)$ in two stages

$$\begin{aligned} t &= q_n(\theta) \star q_u(\phi) \\ &= \begin{bmatrix} \cos(\phi/2)\cos(\theta/2) \\ \cos(\theta/2)\sin(\phi/2)u + \sin(\theta/2)\cos(\phi/2)n + \sin(\theta/2)\sin(\phi/2)v \end{bmatrix}, \end{aligned} \tag{14.32}$$

and finally,

$$\begin{aligned} p(\phi, \theta) &= q_u^\dagger(\phi) \star q_n(\theta) q_u(\phi) \\ &= \begin{bmatrix} \cos(\theta/2) \\ \sin(\theta/2)\cos(\phi)n + \sin(\theta/2)\sin(\phi)v \end{bmatrix}, \end{aligned} \tag{14.33}$$

which shows that for any angles θ, ϕ, the following condition is satisfied $u^T p_v = 0$, defining the homokinetic constraint.

Theorem 14.4. *Given two rigid bodies with quaternions $q^{(1)}, q^{(2)} \in \mathbb{H}$, and angular velocity vectors $\omega^{(1)}, \omega^{(2)} \in \mathbb{R}^3$. Given a unit vector $u \in \mathbb{R}^3$, fixed in the frame of body 1, then, the motion allowed by the constraint $g(q) = u^T q_v = 0$, where $q = q^{\dagger\,(1)} \star q^{(2)}$, consists of:*

1. *Each body is rotated by the same angle ϕ about vector u fixed in body frame;*
2. *Additional rotation by an angle θ about vector n is normal to the plane spanned by $u^{(1)} - u^{(2)}$ if $u^{(1)}, u^{(2)}$ are not collinear, and $\theta = 0$ otherwise;*
3. *Since ϕ is common to both bodies, speed of rotation about u in body frame is same for both bodies and therefore, the connection is homokinetic.*

Proof. We neglect the attachment coordinate frames and assume that the axis of rotation is u in body frame coordinates for both bodies. Using the body coordinate version of the time derivative of the relative quaternion (14.9), dropping irrelevant terms, we have

$$\dot{q} = -\frac{1}{2}\mathcal{E}^T(q)\omega'^{(1)} + \frac{1}{2}\mathcal{G}^T(q)\omega'^{(2)}. \tag{14.34}$$

From this, and using the constraint equation $u^T\dot{q} = 0$, we know the form of q from (14.33). We thus need to know the vectors

$$y = \mathcal{E}(q)\begin{bmatrix}0\\u\end{bmatrix}, \qquad \text{and } z = \mathcal{G}(q)\begin{bmatrix}0\\u\end{bmatrix}. \tag{14.35}$$

A simple computation yields

$$\begin{aligned} y = \mathcal{E}(q)\begin{bmatrix}0\\u\end{bmatrix} &= \cos(\theta/2)u + \sin(\theta/2)\cos(\phi)v - \sin(\theta/2)\sin(\phi)n, \\ z = \mathcal{G}(q)\begin{bmatrix}0\\u\end{bmatrix} &= \cos(\theta/2)u - \sin(\theta/2)\cos(\phi)v + \sin(\theta/2)\sin(\phi)n, \end{aligned} \tag{14.36}$$

and so clearly, if $y^T\omega'^{(1)} = z^T\omega'^{(2)}$, then, the speed of rotation about u in body frame is the same in both bodies and hence, this is a homokinetic connection. This proves the third assertion.

Since the relative quaternion q has the form given in (14.33), the first two parts of the description of the motion follow directly. □

14.6 The direction cosine representation

A more common description of the rotational components of joints is based on a *direction cosine* representation. To do this, consider two bodies labeled 1 and 2 as previously. Define an attachment point on each body with the quadruple $\bar{y}^{(i)}, \bar{a}^{(i)}, \bar{b}^{(i)}$, and $\bar{c}^{(i)}$, so that $\bar{y}^{(i)}$ is the body-frame location of the attachment, and vectors $\bar{a}^{(i)}, \bar{b}^{(i)}$, and $\bar{c}^{(i)}$ form a right handed, orthonormal basis. Write $d^{(i)} = R(q^{(i)})\bar{d}^{(i)}, d = a, b, c$, or y, for the world-frame representation of any of the vectors in the quadruple, where $q^{(i)}$ is the orientation quaternion for body with index $i \in \{1, 2\}$.

The direction cosines are the six coefficients ${w^{(1)}}^T w'^{(2)}$, where w and w' is any pair of choice of a, b or c. Constraints can be built from these in an easy fashion. An important case is a *dot-1* constraint, such as, for instance

$${a^{(1)}}^T b^{(2)} = 0, \tag{14.37}$$

which keeps vectors $a^{(1)}$ and $b^{(2)}$ orthogonal. The constraint (14.37) describes a Hooke's joint in the most natural way. The Jacobians for such constraints are

easily evaluated from

$$\begin{aligned} \dot{a}^{(i)} &= \frac{\mathrm{d}}{\mathrm{d}t} R(q^{(i)}) \bar{a}^{(i)} = \widehat{\omega^{(i)}} R(q^{(i)}) \bar{a}^{(i)} \\ &= -\widehat{a^{(i)}} \omega^{(i)}, \end{aligned} \tag{14.38}$$

where (13.118) was used to evaluate $\dot{R}$, and account taken for $\bar{a}^{(i)}$ being constant. Using the product rule and simple manipulations involving Lemma 13.3, the Jacobian of the dot-1 constraint is revealed

$$\frac{\mathrm{d}}{\mathrm{d}t} \left(a^{(1)^T} b^{(2)} \right) = a^{(1)^T} \widehat{b^{(2)}} \omega^{(1)} - a^{(1)^T} \widehat{b^{(2)}} \omega^{(2)}, \tag{14.39}$$

The issue here is that when constraint violation is large enough so that $a^{(1)}$ becomes nearly parallel to $b^{(2)}$, the Jacobian of (14.39) vanishes.

In addition, a hinge joint can be represented by the pair of constraints

$$\begin{aligned} c^{(1)^T} a^{(2)} &= 0 \\ c^{(1)^T} b^{(2)} &= 0, \end{aligned} \tag{14.40}$$

but this representation does not distinguish between the collinear $c^{(1)} = c^{(2)}$ and the antilinear $c^{(1)} = -c^{(2)}$, as the constraint definition of Section 14.3.

A full set of constraint definitions is direction cosine representation is found in Haug [118].

14.7 End notes

A complete library of useful joints for simulating complicated machines is now in planning stages and will exploit the ideas presented here. The framework described here can in fact be extended to cover n-body constraints, not just pairwise ones.

It will be interesting to see if the representation can be used for the modeling of bio-mechanical joints such as the knee, the wrist and the shoulder. These have complicated restrictions in rotational range as well as couplings between rotational and translational parts which are not easily described in the direction cosine formulation.

Tasora [263] takes credit for first publishing this quaternion constraint framework, though the present derivation extends his work somewhat.

Serban and Haug [250] also provided several constraint definitions in terms of quaternion algebra, using the techniques developed in Haug [118]. This is not entirely equivalent to the present construction, however.

Alternative formulations of joint constraints are found in the robotics literature, based mostly on the *opposite* strategy, namely, the direct modeling of the degrees of freedom, as opposed to the modeling of the restrictions. This analysis is found originally in Denavit and Hartenberg [73], and more recent developments are found in Jain [139] and Orin and Schrader [219]. These are usually covered

in textbooks on robotics [9]. An alternative mathematical framework is also described in Kslicyn [159], as recommended by the American Mathematical Society (AMS) reviewer. A hybrid uniting the two techniques would be interesting, indeed.

15 Rigid Bodies IV: Gyroscopic Forces and Variational Steppers

As seen in Chapter 12, the mass matrix of rigid bodies is configuration dependent. The extra forcing terms appearing in the equations of motion due to this dependency are called *gyroscopic*, since gyroscopes provide the simplest and most vivid illustration of their effects. Gyroscopic forces are quadratic in the generalized velocities which makes them large and fast changing even at moderate speeds. But the worst part is that rigid body motion has a fundamental instability when the three principal inertiae have different values: rotation about the middle axis is unstable. For a body rotating about the middle principal axis, a small disturbance produces exponential increases in the angular velocities around the other two axes, as shown in Marsden and Ratiu [193]. Numerically, this generates instabilities which have been observed repeatedly. Strategies taken to remove these instabilities include high order geometrically accurate integrators or modification of the gyroscopic forces creating instability, or a variety of other, non-physical tricks. Low order variational methods provide a good way for solving the free rigid body problem—this was in fact the application for the variational integrator of Moser and Veselov [208]—but these techniques are difficult to apply in the context of constrained systems, especially when dry frictional contacts are considered.

The present chapter offers an approximation strategy based on the variational method which amounts to a modification of the mass matrix to include stabilizing terms. This leaves the form of the regularized stepping equations of Section 4.5 and Section 10.11.4 unchanged after substituting the new mass matrix and adding correction terms to the forces. This approximation is shown to be unconditionally stable and to be accurate when the velocities are moderate.

After introducing the form of the gyroscopic forces in Section 15.1, the Lagrangian of the system is constructed in Section 15.2 using the quaternion representation of Chapter 13, and the equations of motion are derived. This result is used to reconstruct the classical Euler equations for the rigid body in Section 15.3. The classical theorem on the stability of the rigid body is then presented in Section 15.4, followed by a direct discretization of Euler's equations in Section 15.5, which also contains an ad hoc stabilization scheme. The variational time discretization and the novel approximation scheme are then covered in Section 15.6. After quickly reviewing the equations of motion of the heavy Lagrange top (the gyroscope) in Section 15.7, the results of numerical experiments

on freely rotating rigid bodies as well as slow and fast Lagrange tops are presented in Section 15.8. A short review of other techniques and general comments are found in Section 15.9.

15.1 Introduction

The mass matrix of a rigid body is configuration dependent so that $M : Q \mapsto \mathbb{R}^{3\times 3}$ is a matrix-valued function of the system's coordinates. This introduces extra forcing terms which have been reported under a variety of names but which are properly called *gyroscopic* terms.

Consider the simple Lagrangian with only the kinetic energy term

$$\mathcal{L}(q,\dot{q}) = \frac{1}{2}\dot{q}^T M(q)\dot{q}, \tag{15.1}$$

where $M(q) : Q \mapsto \mathbb{R}^{n\times n}$ is a smooth matrix-valued function of q. Applying the Euler Lagrange equations on this Lagrangian produces

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathcal{L}}{\partial \dot{q}^T} - \frac{\partial \mathcal{L}}{\partial q^T} &= M(q)\ddot{q} + \dot{M}\dot{q} - \frac{1}{2}\dot{q}^T\left(\frac{\partial M}{\partial q^T}\right)\dot{q} \\ &= M(q)\ddot{q} + K(q,\dot{q})\dot{q} = 0,\end{aligned} \tag{15.2}$$

defining the matrix-valued function $K(q,\dot{q})$ as

$$K(q,\dot{q})_{ij} = \sum_k \frac{\partial M(q)_{ij}}{\partial q_k}\dot{q}_k - \frac{1}{2}\sum_k \frac{\partial (M(q))_{kj}}{\partial q_i}\dot{q}_k. \tag{15.3}$$

It is immediately clear from (15.2) that the term $K(q,\dot{q})\dot{q}$ is at least quadratic in $\dot{q}$. For the rigid body case at least, as shown in Section 15.2, matrix $K(q,\dot{q})$ is antisymmetric. For this reason, any force term of the form $A(q,\dot{q})\dot{q}$ with antisymmetric matrix $A(q,\dot{q})$ is called a gyroscopic term.

15.2 Lagrangian form of the equations of motion

As seen in Chapter 12, the translational and rotational terms for the kinetic energy of rigid bodies are decoupled. It is thus sufficient to concentrate on the rotational terms. Consider thus a rigid body with center of mass fixed at the origin and with quaternion coordinates $q \in \mathbb{H}$. Let the inertia tensor be the positive diagonal matrix $\mathcal{I}_0 = \mathsf{diag}(I_1, I_2, I_3), I_i > 0$.

There are several derivations of the rigid body equations in the literature but these are seldom based on the Lagrangian defined in terms of the quaternion coordinates. This derivation is constructed here in a systematic way in order to produce a variational integrator.

Defining the rotation matrix $R(q)$ as in (13.74) so that a vector x' fixed in the body frame transforms to $x = R(q)x'$ in the inertial frame of reference, and using

the inertial frame angular velocity defined in (13.119), namely, $\omega = 2\mathcal{E}(q)\dot{q}$, then, the kinetic energy of the free rigid body which was defined in (12.11) reduces to

$$\begin{aligned} T(q,\dot{q}) &= \frac{1}{2}\omega^T\mathcal{I}\omega = \frac{1}{2}\omega^T R^T\mathcal{I}_0 R\omega \\ &= 2\dot{q}^T\mathcal{G}^T(q)\mathcal{I}_0\mathcal{G}(q)\dot{q} \\ &= 2q^T\mathcal{G}^T(\dot{q})\mathcal{I}_0\mathcal{G}(\dot{q})q, \end{aligned} \tag{15.4}$$

where the identity $\mathcal{G}(q)p = -\mathcal{G}(p)q$ of Lemma 13.26 was used in the penultimate line and Lemma 13.20 in the last. It is clear that the kinetic energy $T(q,\dot{q}) = T(\dot{q},q)$ is symmetric in the two arguments q, $\dot{q}$, that it is a quadratic form in either q or $\dot{q}$, and that it is a homogeneous function of degree 2. To form the Lagrangian for the free rigid body, we still need to introduce the unit constraint $\|q\| = 1$ which leads to

$$\begin{aligned} \mathcal{L}(q,\dot{q},\lambda) &= 2\dot{q}^T\mathcal{G}^T(q)\mathcal{I}_0\mathcal{G}(q)\dot{q} + 2\lambda(\|q\|^2 - 1) \\ &= 2q^T\mathcal{G}^T(\dot{q})\mathcal{I}_0\mathcal{G}(\dot{q})q + 2\lambda(\|q\|^2 - 1), \end{aligned} \tag{15.5}$$

where the factor of 2 in front of the Lagrange multiplier λ is chosen for convenience.

An alternative representation of this Lagrangian is constructed using only the 4×4 matrices $P(q), Q(q)$ introduced in (13.51) and (13.52), and the augmentedinertia tensorr defined as

$$\mathbb{I}_0 = \begin{bmatrix} 0 & 0 \\ 0 & \mathcal{I}_0 \end{bmatrix}, \ \mathbb{I} = P(q)Q^T(q)\mathbb{I}_0 Q(q)P^T(q), \tag{15.6}$$

where $\mathbb{I}_0$ and $\mathbb{I}$ refer to the augmented inertia tensor in the body and inertial frame, respectively.

Taking consideration of the identity $q^\dagger \star \dot{q} = -\dot{q}^\dagger \star q$ for unit quaternions, derived in Section 13.10, which translates to $Q^T(q)\dot{q} = -Q^T(\dot{q})q$ in terms of the matrix algebra representation, an alternative form for the Lagrangian is thus

$$\begin{aligned} \mathcal{L}(q,\dot{q},\lambda) &= 2\dot{q}^T Q(q)\mathbb{I}_0 Q^T(q)\dot{q} + 2\lambda(\|q\|^2 - 1) \\ &= 2q^T Q(\dot{q})\mathbb{I}_0 Q^T(\dot{q})q + 2\lambda(\|q\|^2 - 1). \end{aligned} \tag{15.7}$$

It is the augmented form of (15.7) which is most useful for our present purposes and which is most easily manipulated.

To compute the Euler-Lagrange equations of motion, we first note the identities

$$\begin{aligned} \frac{\partial T(q,\dot{q})}{\partial \dot{q}^T} &= 4Q(q)\mathbb{I}_0 Q^T(q)\dot{q}, \\ \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial T(q,\dot{q})}{\partial \dot{q}^T} &= 4Q(q)\mathbb{I}_0 Q^T(q)\ddot{q} + 4Q(\dot{q})\mathbb{I}_0 Q^T(q)\dot{q} + 4Q(q)\mathbb{I}_0 Q^T(\dot{q})\dot{q} \\ &= 4Q(q)\mathbb{I}_0 Q^T(q)\ddot{q} + 4Q(\dot{q})\mathbb{I}_0 Q^T(q)\dot{q}, \\ \frac{\partial T(q,\dot{q})}{\partial q^T} &= 4Q(\dot{q})\mathbb{I}_0 Q^T(\dot{q})q = -4Q(\dot{q})\mathbb{I}_0 Q^T(q)\dot{q}, \end{aligned} \tag{15.8}$$

where use was made of the following facts

$$\begin{aligned} Q^T(p)p &= \begin{bmatrix} p^T p \\ 0 \end{bmatrix}, \\ \mathbb{I}_0 Q^T(p)p &= \begin{bmatrix} 0 & 0 \\ 0 & \mathcal{I}_0 \end{bmatrix} \begin{bmatrix} p^T p \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \end{aligned} \tag{15.9}$$

for any vector $p \in \mathbb{R}^4$ and 3×3 matrix $\mathcal{I}_0$.

After collecting all the terms and substituting in the Euler-Lagrange equations (15.5), the following DAE of motion is found

$$\begin{aligned} Q(q)\mathbb{I}_0 Q^T(q)\ddot{q} + 2Q(\dot{q})\mathbb{I}_0 Q^T(q)\dot{q} + \lambda q &= 0 \\ \|q\|^2 - 1 &= 0. \end{aligned} \tag{15.10}$$

Observe that using the definition of $Q(p) = \begin{bmatrix} q & \mathcal{G}^T(q) \end{bmatrix}$ from (13.62) and the zero pattern of $\mathbb{I}_0$, we can reduce (15.10) down to an equivalent form which we use in the next section

$$\begin{aligned} \mathcal{G}^T(q)\mathcal{I}_0\mathcal{G}(q)\ddot{q} + 2G^T(\dot{q})\mathcal{I}_0\mathcal{G}(q)\dot{q} + \lambda q &= 0 \\ \|q\|^2 - 1 &= 0. \end{aligned} \tag{15.11}$$

Multiplying the first equation in (15.10) by q^T, using the identity derived in (15.9), reusing the previously observed fact that $q^T Q(\dot{q}) = -\dot{q}^T Q(q)$, and using the constraint $\|q\| = 1$ leads to the conclusion

$$\lambda = T(q, \dot{q}), \tag{15.12}$$

which is a constant along the trajectory, if there are no applied torques.

Interestingly, the 4×4 matrix multiplying the $\ddot{q}$ term, $M = \mathcal{G}^T(q)\mathcal{I}_0\mathcal{G}(q)$ is singular since $q^T M q = 0$. In fact, it only has rank 3. In addition, the time dependence of the kinetic energy term on q as well as $\dot{q}$ produced the extra term $2\mathcal{G}^T(\dot{q})\mathcal{I}_0\mathcal{G}(q)\dot{q}$. This term is responsible for all gyroscopic effects observed in rigid bodies and, being nonlinear, it introduces significant difficulties for stable integration.

Curiously, I could not find the derivation above in literature, though derivations based directly on the rotation matrices themselves are found in [178, 208] and several other places.

A derivation based on the rotation matrix R itself involves the nine unknown dynamic coefficients r_{ij}, the three components of the angular velocity vector ω, but also, 6 additional constraints imposing orthonormality on the matrix R, not even including the handedness constraint $\det(R) = +1$.

15.3 Euler's equations

The most common form of the equations of motion for the free rigid body are now derived. Introduce first the body fixed angular velocity vector as was done

in (13.120) as

$$\begin{aligned}\omega' = R^T\omega = \mathcal{G}(q)\mathcal{E}^T(q)\omega = 2\mathcal{G}(q)\mathcal{E}^T(q)\mathcal{E}(q)\dot{q}\\ = 2\mathcal{G}(q)\dot{q},\end{aligned} \tag{15.13}$$

and compute the time derivative as follows

$$\dot{\omega}' = 2\mathcal{G}(q)\ddot{q} + 2\dot{\mathcal{G}}(\dot{q})\dot{q} = 2\mathcal{G}(q)\ddot{q}, \tag{15.14}$$

which holds due to the linear dependence of $G(q)$ on its argument q and from using Corollary 13.21.

Now, multiplying the first line of (15.11) on the left with $\mathcal{G}(q)$, using the identity $\|q\| = 1$, and substituting for ω', and $\dot{\omega}'$, the equations of motion (15.11) become:

$$\mathcal{I}_0\dot{\omega}' + 2\mathcal{G}(q)\dot{\mathcal{G}}^T(\dot{q})\mathcal{I}_0\mathcal{G}(q)\dot{q} = 0. \tag{15.15}$$

Using Corollary 13.21 again on the last term, and then applying the result of Lemma 13.25 and the definition of ω' from (13.120) to obtain:

$$\mathcal{I}_0\dot{\omega}' + \widehat{\omega}'\mathcal{I}_0\omega' = 0, \tag{15.16}$$

which are known as Euler's equations for the rigid body which are common to textbooks as [105, 22, 193] and many more sources.

Either restarting the derivation from the DAE (15.11) and substituting for the inertial frame angular velocity, $\omega = 2\mathcal{E}(q)\dot{q}$, or by transforming the result (15.16) directly, an equivalent inertial frame formulation is found

$$\mathcal{I}\dot{\omega} + \widehat{\omega}\mathcal{I}\omega = 0, \tag{15.17}$$

but in this case, the inertia tensor $\mathcal{I} = R^T\mathcal{I}_0R$ is time dependent. This last equation is in fact usually written as

$$\dot{p}_\phi = f_\phi, \qquad p_\phi = \mathcal{I}\omega, \tag{15.18}$$

where $p_\phi \in \mathbb{R}^3$ is the angular momentum vector and $f_\phi \in \mathbb{R}^3$ are the applied *torques* (also known as moments, especially in the engineering literature).

For the free rigid body case when $f_\phi = 0$, the equations of motion (15.16) or (15.17) can be integrated in terms of Jacobi elliptic functions, as described in details in [176], and, at least in large part, in [193].

It is striking to realize that the body frame equations (15.16) can be integrated to produce the angular velocity vector, $\omega'(t)$, without having to compute the rotation matrix at all.

Observe also the second term in (15.16) which has the form $K(\omega')\omega'$ where matrix K is *antisymmetric*, $K(\omega') = -K^T(\omega')$ for all $\omega' \in \mathbb{R}^3$. Such antisymmetric forcing terms of the form $f = K(q,\dot{q})\dot{q}$ in equations of motion are called gyroscopic, as observed in the introduction. They are special as they do no work on the system. Indeed, the work done by the force f over a time interval is the time integral of $f^T\dot{q}$, but since $f^T\dot{q} = \dot{q}^TK(q,\dot{q})\dot{q} = 0$, the contribution vanishes.

Euler's equations, either in the body frame or the inertial frame, can be coupled with the corresponding quaternionic velocity (13.120) or (13.119), respectively, to recover the full set of differential equations, namely the fully explicit body frame system

$$\begin{aligned} \mathcal{I}_0\dot{\omega}' + \widehat{\omega}'\mathcal{I}_0\omega' &= f'_\phi \\ \dot{q} &= \frac{1}{2}\mathcal{G}^T(q)\omega', \end{aligned} \tag{15.19}$$

or the semi-implicit inertial frame version

$$\begin{aligned} \mathcal{I}\dot{\omega} + \widehat{\omega}\mathcal{I}\omega &= f_\phi \\ \dot{q} &= \frac{1}{2}\mathcal{E}^T(q)\omega. \end{aligned} \tag{15.20}$$

where f'_ϕ and f_ϕ are the applied torques in the body and inertial frames, respectively.

15.4 Analytic stability of the free rigid body

For a rigid body rotating freely in the absence of an applied torque, there is a surprising analytic result that rotations about the intermediate principal axis, the one with inertia I_2 with $I_1 < I_2 < I_3$ in the notation described previously, is unstable. This means that for a rigid body rotating about its intermediate principal axis, any small perturbation of this motion will quickly diverge from it. This is proven in [193], Section 15.9. The theorem is stated here for reference.

Theorem 15.1 (Rigid Body Stability Theorem). *In the motion of the free rigid body, rotation around the long and short axes is (Lyapunov) stable and around the middle axis is unstable.*

The stability theorem relies on Hamiltonian formulation and an application of the Energy-Casimir Method which are beyond the present scope and the proof is thus omitted.

15.5 Discretizing Euler's equations directly

Euler's equations of motion (15.17) can be discretized directly but, unsurprisingly given the result of Theorem 17.1, they exhibit a fundamental instability.

This is easily seen by using the body frame equations of motion (15.17) and explicitly discretizing to first order yields

$$h^{-1}\mathcal{I}_0(\omega'_{k+1} - \omega'_k) = -\widehat{\omega}'_k\mathcal{I}_0\omega'_k + \tau_k = \widehat{L}'_k\omega'_k + \tau_k, \tag{15.21}$$

where τ_k is the applied torque evaluated at step k, and $L'_k = \mathcal{I}_0\omega'_k$ is the angular momentum vector in the body frame at time step k. A simple rearrangement yields the quasilinear form

$$\mathcal{I}_0\omega_{k+1} = \left(\mathcal{I}_0 + h\widehat{\omega}'_k\mathcal{I}_0\right)\omega_k + h\tau_k, \tag{15.22}$$

which clearly exhibits a constant mass matrix, $\mathcal{I}_0$. Ignoring the applied torque and performing elementary algebraic manipulation yields the recurrence formula for the kinetic energy

$$E_{k+1} = \frac{1}{2}\omega'^T_{k+1}\mathcal{I}_0\omega'_{k+1} = E_k + h^2\omega'^T_k H_k\omega'_k \geq E_k, \tag{15.23}$$

with the definition

$$H_k = -\mathcal{I}_0\widehat{\omega}'_k\mathcal{I}_0^{-1}\widehat{\omega}_k\mathcal{I}_0. \tag{15.24}$$

Matrix H_k is shown to be symmetric and positive semi-definite in the next paragraph and thus the inequality sign in (15.23) is alarming. The energy increases inexorably unless the condition $\widehat{\omega}_k\mathcal{I}_0\omega = 0$, holds, in other words, when ω is aligned along a principal axis, or when $\mathcal{I}_0$ is a multiple of the identity, $\mathcal{I}_0 = \mu I_3$.

To see that H_k is symmetric and positive semidefinite, consider a real symmetric positive definite 3×3 matrix D and a real vector $x \in \mathbb{R}^3$, and set $H = -\widehat{x}D\widehat{x}$. Now, consider an arbitrary vector $y \in \mathbb{R}^3$ and compute the scalar $\sigma = y^T Hy$. Setting $z = \widehat{x}y, z \in \mathbb{R}^3$, produces $z^T = y^T\widehat{x}^T = -y^T\widehat{x}$ using the antisymmetry property of $\widehat{x}$. It then follows that

$$\sigma = y^T Hy = -y^T\widehat{x}D\widehat{x}y = z^T Dz \geq 0, \tag{15.25}$$

since D is positive definite, and since $\widehat{x}y = 0$ whenever $y = \lambda x$ for some scalar λ. Matrix H_k is thus positive semi-definite after substituting $D = \mathcal{I}_0$ and $x = \omega'_k$. Additionally, the situation $\sigma = 0$ in (15.25) occurs only when $\widehat{x}y = 0$ which implies parallelism between x and y, or $y = \alpha x$, as per Lemma 13.2. Now, given that $L'_k = \mathcal{I}_0\omega'_k$, parallelism between L'_k and ω'_k occurs when $\mathcal{I}_0 = \mu I_3$, or when ω'_k is aligned on a principal axis, or when ω'_k is aligned along two principal axes with the same inertia.

A better result can be obtained by using a semi-implicit discretization. Indeed, by choosing different values of time for the two instances of ω' on the right hand side of (15.21), we have

$$h^{-1}\mathcal{I}_0(\omega'_{k+1} - \omega'_k) = \widehat{L}'_k\omega'_{k+1} + \tau_k. \tag{15.26}$$

If all the ω'_{k+1} terms are collected on the left hand side, we get a modified mass matrix of the form

$$\left(\mathcal{I}_0 - \widehat{L}'_k\right), \tag{15.27}$$

horrific for being non-symmetric, but positive definite just the same. A symmetric pseudo mass matrix can be recovered though as now shown.

Going back to (15.26) and expanding the ω'_{k+1} on the right hand side up to second order in the time step h yields

$$\begin{aligned}
\mathcal{I}_0\omega'_{k+1} &= \mathcal{I}_0\omega'_k + h\widehat{L}'_k\omega'_{k+1} + h\tau_k \\
&= \mathcal{I}_0\omega'_k + h\widehat{L}'_k\left(\omega'_k + h\mathcal{I}_0^{-1}\widehat{L}'_k\omega'_{k+1} + h\mathcal{I}_0^{-1}\tau_k\right) + h\tau_k \\
&= \mathcal{I}_0\omega'_k + h\widehat{L}'_k\omega'_k + h^2\widehat{L}'_k\mathcal{I}_0^{-1}\widehat{L}'_k\omega'_{k+1} + h\left(I_3 + h\widehat{L}'_k\mathcal{I}_0^{-1}\right)\tau_k,
\end{aligned} \tag{15.28}$$

which can be rearranged by collecting the $k+1$ terms on the left hand side as

$$\left(\mathcal{I}_0 - h^2 \widehat{L}'_k \mathcal{I}_0^{-1} \widehat{L}'_k\right) \omega'_{k+1} = \left(\mathcal{I}_0 + h\widehat{L}'_k\right) \omega'_k + h\left(I_3 + h\widehat{L}'_k \mathcal{I}_0^{-1}\right) \tau_k. \tag{15.29}$$

The modified inertia tensor appearing on the left hand side of (15.29) is due to Anitescu and Potra [18] and was analyzed in a previous report [169] of mine. It is a bona fide mass matrix, being symmetric and positive definite. Indeed, matrices of the form $-\widehat{x}D\widehat{x}$ are symmetric and positive semi-definite for a symmetric and positive definite 3×3 matrix D as follows from (15.25).

Using Lemma 13.15 to compute the explicit inverse of the modified mass matrix $(I_3 - h^2 \widehat{L}'_k \mathcal{I}_0^{-1} \widehat{L}'_k \mathcal{I}_0^{-1})\mathcal{I}_0$, the energy changes from step to step can be computed explicitly in the absence of applied torque. Noting that matrices of the form $\widehat{x}D\widehat{x}$ are negative semi-definite, the energy satisfies the monotonic non-increasing recurrence

$$E_{k+1} = E_k + \frac{h^3}{2(1 + h^2\kappa_k)} {\omega'}_k^T \widehat{L}'_k \mathcal{I}_0^{-1} \widehat{L}'_k \omega'_k \leq E_k, \tag{15.30}$$

with the following definition for the scalar $\kappa > 0$:

$$\kappa_k = \frac{L'_k \mathcal{I}_0 L'_k}{\det(\mathcal{I}_0)}. \tag{15.31}$$

It is possible to perform a stability analysis near the fixed points of both the explicit stepping formula of (15.22) and the semi-implicit form (15.29), at least in the absence of applied torques. Indeed, using Lemma 13.15 to compute the explicit inverse of the modified inertia matrices, the stepping formulae can be written as in the quasilinear form

$$x_{k+1} = \left(I_3 + \zeta_k B_k + \eta_k B_k^2\right) x_k, \tag{15.32}$$

with time dependent scalars ζ_k, η_k and the special matrix $B_k = \widehat{x}\mathcal{I}_0^{-1}$. Linearizing this expression near a fixed point and computing the eigenvalues for the different cases, yields

Theorem 15.2. *The discrete motion of a free rigid body computed by applying the one-step recurrence* (15.29)*, subject to zero torques, $\tau_k = 0, k = 1, 2, \ldots$, has the following stability properties:*

1. *For the case where the three principal inertiae are equal, $I_1 = I_2 = I_3$, the two steppers are equivalent and all fixed points are stable;*
2. *The explicit stepper of* (15.22) *has only unstable fixed points;*
3. *The semi-implicit stepper of* (15.29) *has at least one stable fixed point for rotations about the principal axis with the* largest inertia*, I_1, where $I_1 \geq I_2 \geq I_3$.*

The proof is straight forward but tedious. It is found in my report [169].

This last conclusion of Theorem 15.2 is in striking contrast with Theorem 17.1. Thus, the semi-implicit stepper of (15.29) is stable but it destroys qualitative

aspects of the physics. The question remains as to whether it is possible to construct a better linear approximation to the rigid body stepping equations. Fortunately, the answer is affirmative as we now proceed to demonstrate.

15.6 Variational discretization

To discretize the Lagrangian defined in (15.5) and apply the variational method, introduce approximations of q and $\dot{q}$ as follows

$$q(kh) \approx \frac{1}{2}(q_k + q_{k-1}), \qquad \dot{q}(kh) \approx \frac{1}{h}(q_k - q_{k-1}). \tag{15.33}$$

Now, the important expression in (15.5) is the product $\mathcal{G}(q)\dot{q}$ and this evaluates to

$$\begin{aligned} \omega_k' &= 2\mathcal{G}(q)\dot{q} = 2\frac{1}{2h}\mathcal{G}\left(q_k + q_{k-1}\right)\left(q_k - q_{k-1}\right) \\ &= 2\frac{1}{2h}\left(\mathcal{G}(q_k)q_k - \mathcal{G}(q_k)q_{k-1} + \mathcal{G}(q_{k-1})q_k - \mathcal{G}(q_{k-1})q_{k-1}\right) \\ &= \frac{2}{h}\mathcal{G}(q_{k-1})q_k = -\frac{2}{h}\mathcal{G}(q_k)q_{k-1}, \end{aligned} \tag{15.34}$$

and thus, the discretized Lagrangian, ignoring constraint and potential terms, reads

$$\mathbb{L}_d(q_k, q_{k-1}, h) = \frac{2}{h}q_{k-1}^T\mathcal{G}^T(q_k)\mathcal{I}_0\mathcal{G}(q_k)q_{k-1} = \frac{2}{h}q_k^T\mathcal{G}^T(q_{k-1})\mathcal{I}_0\mathcal{G}(q_{k-1})q_k. \tag{15.35}$$

Curiously, the final formula remains the same whichever choice is made for q, as long as $\dot{q}$ is defined with $h^{-1}(q_k - q_{k-1})$.

Performing the same discretization on the factors $Q^T(q)\dot{q}$ appearing in the augmented formulation of the Lagrangian (15.10) produces

$$\mathbb{L}_d(q_k, q_{k-1}, h) = \frac{2}{h}q_{k-1}^TQ(q_k)\mathbb{I}_0Q^T(q_k)q_{k-1} = \frac{2}{h}q_k^TQ(q_{k-1})\mathbb{I}_0Q^T(q_{k-1})q_k, \tag{15.36}$$

where the last equality holds because the augmented inertia tensor, $\mathbb{I}_0$ defined in (15.6), has a zero first row and first column. In fact, relabeling $x = q_{k-1}$ and $y = q_k$ reveals

$$\mathbb{L}_d(x, y, h) = \mathbb{L}_d(y, x, h), \tag{15.37}$$

which means that it is possible to switch the arguments in the differential operators since

$$D_1\mathbb{L}_d(x, y, h) = \frac{\partial\mathbb{L}_d(x, y, h)}{\partial x} = D_2\mathbb{L}_d(y, x, h) = \frac{\partial\mathbb{L}_d(y, x, h)}{\partial x}. \tag{15.38}$$

Finally, note that $\mathbb{L}_d(x, y, h)$ is homogeneous of degree two in both arguments so that

$$\begin{aligned} &x^TD_1^T\mathbb{L}_d(x, y, h) = x^TD_2^T\mathbb{L}_d(y, x, h) = 2\mathbb{L}_d(x, y, h), \quad \text{and} \\ &y^TD_2^T\mathbb{L}_d(x, y, h) = y^TD_1^T\mathbb{L}_d(y, x, h) = 2\mathbb{L}_d(x, y, h). \end{aligned} \tag{15.39}$$

These relations are independent of whether (15.35) or (15.36) is used.

The two expressions given in (15.36) and in (15.35) are mathematically identical but it is sometimes easier to work with one or the other form, given the context, which is why both are provided.

Applying the discrete variational principle to the expression (15.35) after including the constraint $\|q\|^2 - 1 = 0$ produces the following discrete Euler-Lagrange equations

$$\begin{aligned} \frac{4}{h}\left(\mathcal{G}^T(q_{k+1})\mathcal{I}_0\mathcal{G}(q_{k+1})q_k + \mathcal{G}^T(q_{k-1})\mathcal{I}_0\mathcal{G}(q_{k-1})q_k\right) + 4h\lambda q_k &= 0 \\ \|q_{k+1}\|^2 - 1 &= 0. \end{aligned} \tag{15.40}$$

This last stepping equation in (15.40) is merely *quadratic* in q_{k+1} and can be solved quickly using Newton-Raphson iterations, though the Jacobian for these iterations is not particularly enlightening. In addition, multiplying (15.40) on the left with q_k^T and using the homogeneity property of $\mathbb{L}_d(q_k, q_{k-1}, h)$ noted in (15.39), the following identity is revealed

$$\lambda = -\frac{1}{h}\left(\mathbb{L}_d(q_k, q_{k-1}, h) + \mathbb{L}_d(q_{k+1}, q_k, h)\right). \tag{15.41}$$

It should be possible to make a good guess for $h\lambda$ as the negative of twice the kinetic energy of the last step, for instance, and this should speed up convergence. However, a simple transformation can be performed to discover that (15.40) is in fact an eigenvalue-eigenvector problem as now demonstrated.

Start by extracting the discrete stepping equations from the augmented form of the Lagrangian (15.36) to get another form for the discrete Euler-Lagrange equations

$$\begin{aligned} \frac{4}{h}\left(Q(q_{k+1})\mathbb{I}_0Q^T(q_{k+1})q_k + Q(q_{k-1})\mathbb{I}_0Q^T(q_{k-1})q_k\right) + 4h\lambda q_k &= 0 \\ \|q_{k+1}\|^2 - 1 &= 0. \end{aligned} \tag{15.42}$$

Before turning this into a linear eigenvalue and eigenvector problem, note the suspicious recurrence of terms of the form $Q^T(q_{k-1})q_k$ and we define accordingly

$$r_k = Q^T(q_{k-1})q_k, \quad \text{or, equivalently, } q_k = Q(q_{k-1})r_k, \tag{15.43}$$

which is equivalent to the quaternion formula

$$q_k = q_{k-1} \star r_k. \tag{15.44}$$

Given that each q_k and q_{k-1} have unit magnitude, it follows that so does $\|r_k\| = 1$. Also, the stepping defined by either (15.43) or the equivalent form (15.44) clearly preserves unitarity of the variable, a subtle detail missing from the definition of the discretization formulae (15.33). It appears therefore beneficiary to formulate the stepping equations directly in terms of the discrete quaternion velocity $r_{k+1} = Q^T(q_k)q_{k+1}$.

The reformulation in terms of the normalized quaternionic velocity does enjoy formal theory, as shown in [51], which could perhaps have shortened the present derivation, but at the cost of having to use more complicated mathematics.

Let us now define a relation between this unit quaternion r_k and the body frame angular velocity vector ω'_k defined in (15.34)

$$r_k = Q^T(q_{k-1})q_k = \begin{bmatrix}(r_k)_s\\(r_k)_v\end{bmatrix} = \begin{bmatrix}\cos(\frac{h}{2}\|\omega'_k\|)\\ \frac{1}{\|\omega'_k\|}\sin(\frac{h}{2}\|\omega'_k\|)\omega'_k\end{bmatrix}, \tag{15.45}$$

which fits both the discretization definition of (15.33) in the limit of $h \to 0$, as well as the unit requirement for r_k for any value of $h > 0$. As seen below, this unit angular velocity quaternion, r_k, allows for an unconditionally stable second order one step integration scheme for the free rigid body which is much cheaper computationally than the Moser Veselov scheme [208] or the Leimkuhler and Reich scheme [178] based on RATTLE. Suitable approximations also yield efficient and stable discretizations in terms of ω'_k itself if needed, using only three equations as opposed to five for the case of the full quaternion formulation.

To obtain a linear equation in r_{k+1}, multiply the first line in (15.42) from the left with the matrix $CQ^T(q_{k+1})$, using the 4×4 quaternion complex conjugation matrix $C = \mathrm{diag}(1,-1,-1,-1)$ defined previously in (13.54), and use the unitary constraint on q_{k+1} so that $Q^T(q_{k+1})Q(q_{k+1}) = \|q_{k+1}\|^2 I_4 = I_4$. Introducing the definitions

$$\begin{aligned} u &= Q(q_{k-1})\mathbb{I}_0 Q^T(q_{k-1})q_k = \mathcal{G}^T(q_{k-1})\mathcal{I}_0\mathcal{G}(q_{k-1})q_k, u \in \mathbb{R}^4,\\ z &= Q^T(u)q_k, (\text{ or, in quaternion algebra, } z = u^\dagger \star q_k)\\ r_{k+1} &= Q^T(q_k)q_{k+1} = CQ^T(q_{k+1})q_k, \text{ and}\\ r_k &= Q^T(q_{k-1})q_k = CQ^T(q_k)q_{k-1}, \end{aligned} \tag{15.46}$$

and performing some algebraic manipulation involving identities derived in section 13.5, noting in particular that $C\mathbb{I}_0 C = \mathbb{I}_0$ and that $Q^T(q)p = CQ^T(p)q$, the stepping equations take the form

$$\begin{aligned} \big(\mathbb{I}_0 + Q(z)\big)\, r_{k+1} - \nu r_{k+1} &= 0\\ \|r_{k+1}\|^2 - 1 &= 0,\\ q_{k+1} &= Q(q_k)r_{k+1}, \end{aligned} \tag{15.47}$$

where the Lagrange multiplier $\nu = -h^2\lambda$ was redefined from λ to absorb the $-h^2$ factor.

It is now clear that system (15.47) is in fact an eigenvalue eigenvector problem of the form $Ax = \mu x$ with

$$A = \mathbb{I}_0 + Q(z), z \in \mathbb{R}^4, \tag{15.48}$$

and $x = r_{k+1}$. Such an eigenvalue problem can be solved to machine precision in finite time. In fact, a simple minded implementation of Newton-Raphson iterations to solve the nonlinear system defined by the first two lines of (15.47), using

the nonlinear solver `fsolve` in Octave, usually required only one computation of the Jacobian matrix

$$J = \mathbb{I}_0 + Q(z) + \nu I_4, \tag{15.49}$$

on average, and four or five residual evaluations. In the absence of external torques, as noted previously in (15.41), the value of ν is easy to compute from the kinetic energy and since we are essentially using the implicit midpoint rule on a quadratic system, we expect the energy to be very nearly constant and thus ν as well.

The main issue here is that neither the nonlinear representation (15.40) nor the eigenvalue problem representation (15.47) of the stepping equations have a form which can be easily combined with the rest of the problem formulation including constraints. Indeed, for this case, we hope to have an equation of the form $Mv_{k+1} = d(v_k, h)$, where M is a square symmetric positive definite mass matrix, v_{k+1} is the velocity we are trying to compute, and $d(v, h)$ is some vector which depends on the current state of the system.

To construct an approximation with the correct format, start with the definition of r_k in (15.45) and notice that for small enough time step h, or more precisely, for small enough $h\|\omega_k'\|$,

$$\begin{aligned} (r_k)_s &= 1 - (h\|\omega_k'\|)^2 + O(h^4) = 1 + O(h^2), \\ \|(r_k)_v\| &= \frac{h}{2} + O(h^3). \end{aligned} \tag{15.50}$$

Next, look at the vector u defined in the first equation of (15.46) as $u = Q(q_{k-1})\mathbb{I}_0 r_k$. Introduce the angular momentum quaternion $l_k = \mathbb{I}_0 r_k$ which is clearly pure imaginary so that $l_k^\dagger = -l_k$. From this definition u and z can be rewritten as follows

$$\begin{aligned} u &= q_{k-1} \star l_k, \\ u^\dagger &= l_k^\dagger \star q_{k-1}^\dagger = -l_k \star q_{k-1}^\dagger \\ z &= u^\dagger \star q_k = -l_k \star q_{k-1}^\dagger \star q_k = -l_k \star r_k. \end{aligned} \tag{15.51}$$

It is now clear that we have very few variables to deal with, namely, the known quaternions z, r_k, $l_k = \mathbb{I}_0 r_k$, the unknown quaternion r_{k+1}, and Lagrange multiplier ν. To simplify the notation of what follows, relabel these variables as listed below

$$\begin{aligned} z &\longleftarrow z, \\ y &\longleftarrow r_k \\ l &\longleftarrow l_k = \mathbb{I}_0 r_k \\ x &\longleftarrow r_{k+1} \\ \nu &\longleftarrow \nu, \end{aligned} \tag{15.52}$$

which means that the unknowns are now x and ν. In terms of these variables,

now, the stepping equations read

$$\begin{aligned} l &= \mathbb{I}_0 y, \\ z &= -l \star y, \\ \big(\mathbb{I}_0 + Q(z)\big)\, x &= \nu x \text{ solve for } x \text{ and } \nu \\ q_{k+1} &= y \star x, \quad \text{update the quaternion} \\ r_{k+1} &= x, \quad \text{update the quaternionic velocity.} \end{aligned} \tag{15.53}$$

The loop is looped by starting back at (15.52).

What remains to do now is to approximate the eigenvalue eigenvector problem at the heart of (15.53) with a simple linear system solve. To this purpose, start by evaluating the scalar and vector components of the vector z, using definition (15.45) and approximations $(1/\|w\|)\sin(h\|w\|/2)x \approx (h/2)w, w \in \mathbb{R}^3$ to produce

$$\begin{aligned} z = -l \star y &= \begin{bmatrix} l_v^T y_v \\ -y_s l_v - \widehat{l}_v y_v \end{bmatrix} \\ &= \begin{bmatrix} \frac{h^2}{2} E_k \\ -\frac{h}{2} L'_k - \frac{h^2}{4} \widehat{L}'_k \omega'_k \end{bmatrix}, \end{aligned} \tag{15.54}$$

where the energy is defined as usual with $(1/2){\omega'}_k^T \mathcal{I}_0 \omega'_k$, and $L'_k = \mathcal{I}_0 \omega'_k$ is the body fixed angular momentum vector at time step k.

After multiplying the third line of equation (15.53) on the left with x^T, using the constraint $x^T x = 1$, the fact that $Q^T(z) + Q(z) = z_s I_4$, approximating $x = O(h)$, $x = y + O(h^2)$, and $y = (h/2)\omega'_k + O(h^2)$, the result is

$$\begin{aligned} \nu &= x^T \mathbb{I}_0 x + x^T Q(z) x \\ &= x_v^T \mathcal{I}_0 x_v + z_s \\ &= y_v^T \mathcal{I}_0 y_v + z_s + O(h^3) \\ &= 2z_s + O(h^3) \\ &= h^2 E_k + O(h^3), \end{aligned} \tag{15.55}$$

where $E_k = (1/2){\omega'_k}^T \mathcal{I}_0 \omega'_k$.

Any of the last two lines in (15.55) can be used to approximate ν using only known quantities. Of course, it is possible to revise the first guess within an iterative procedure. The form $\nu = y_v^T \mathcal{I}_0 y_v + z_s$ is used in what follows to be specific and to reduce the number of variables appearing in the equations.

The last approximation in (15.55) is clarified as

$$\begin{aligned} -z_v^T x_v &= \frac{h^2}{4} {L'}_k^T \omega'_{k+1} + \frac{h^2}{4} {\omega'}_{k+1}^T \widehat{L}_k \omega'_k + O(h^3) \\ &= \frac{h^2}{4} {L'}_k^T \omega'_k + \frac{h^2}{4} {\omega'}_k^T \widehat{L}_k \omega'_k + O(h^3) \\ &= \frac{h^2}{4} {L'}_k^T \omega'_k + O(h^3) = \frac{h^2}{2} E_k + O(h^3), \end{aligned} \tag{15.56}$$

since for any vectors $x, y \in \mathbb{R}^3, y^T \widehat{x} y = 0$, as was used in the penultimate step, and since $\omega'_{k+1} = \omega'_k + O(h)$, obviously. Finally, since $x_s = 1 + O(h^2)$, then, the exact value of x_v does not affect the value of ν to $O(h^2)$.

In practice, any of the last three lines of (15.55) can be used as a reasonable approximation which can be corrected using successive substitution.

Next, consider the vector part of the stepping equation (15.53), after referring back to the definition of quaternion products given in Section 13.3 and especially (13.47), we find

$$\mathcal{I}_0 x_v + z_s x_v + x_s z_v + \widehat{z}_v x_v = \nu x_v = 2 z_s x_v + O(h^3), \tag{15.57}$$

which can be reorganized to read

$$(\mathcal{I}_0 - z_s I_3)\, x_v = -x_s z_v - \widehat{z}_v x_v, \tag{15.58}$$

and this has a striking similitude to the semi-implicit discretization of Euler's equations (15.28). Applying the same trick and expanding x_v on the right hand side with the right hand side—feeding its tail to the serpent more or less—and substituting y_s for x_s in view of the fact that these two scalars differ by terms of $O(h^3)$, the stepper now reads

$$\left(D - \widehat{z}_v D^{-1} \widehat{z}_v\right) x_v = -y_s \left(I_3 - \widehat{z}_v D^{-1}\right) z_v, \tag{15.59}$$

after defining the diagonal matrix

$$D = \mathcal{I}_0 - y_v^T \mathcal{I}_0 y_v I_3 = \mathcal{I}_0 - \frac{h^2}{2} E_k I_3 + O(h^3). \tag{15.60}$$

Using Lemma 13.15, the discrete Euler-Lagrange equations can be brought to the explicit quasilinear form

$$\begin{aligned}
z &= -P(y)\mathbb{I}_0 y, \\
D &= \mathcal{I}_0 - y_v^T \mathcal{I}_0 y_v I_3, \\
\kappa &= \frac{z_v^T D z_v}{\det(D)}, \\
\alpha &= \frac{1}{1+\kappa} \\
x_v &= -y_s D^{-1} \left(I_3 - \alpha \widehat{z}_v D^{-1} + \alpha (\widehat{z}_v D^{-1})^2\right) z_v, \\
x_s &= \sqrt{1 - \|x_v\|^2},
\end{aligned} \tag{15.61}$$

which is a complete one-step procedure to compute x from y and $\mathcal{I}_0$. Note how the time step h is conspicuously absent from the system (15.61). In fact, all that matters is the initial discrete angular velocity, and this is of course larger for a larger time step. The last line of (15.61) is a renormalization procedure. Breakdown of the approximation is expected whenever x_s is significantly less than unity, i.e., whenever $\|x_v\|$ approaches unity.

The definition (15.60) for the diagonal 3×3 matrix D clearly marks a *stability boundary*, namely,

$$y_v^T \mathcal{I}_0 y_v < \min(I_1, I_2, I_3), \tag{15.62}$$

where the scalars I_i are the principal inertiae, i.e., $\mathcal{I}_0 = \mathsf{diag}(I_1, I_2, I_3)$.

The only guarantee that D is well defined is that, by construction, the vector part of the unitary quaternion y has magnitude $\|y_v\| = O(h)$ and, therefore, matrix D differs from matrix $\mathcal{I}_0$ by terms of $O(h^2)$.

The form of the matrix multiplying x_v in (15.59) has been studied before in Section 15.5 and is known to be symmetric, positive definite, at least as long as the diagonal 3×3 matrix D remains positive definite.

One could also try to solve (15.59) iteratively, especially with a modified Newton-Raphson method or just sequential substitution. Since all the scalars are less than unity (recall that $\|x\| = 1$), this mapping is expected to be contractive, though this is not proved yet.

In any case, the linear version of (15.59) was implemented in Octave and the results were identical to a nonlinear solver applied to (15.53). When comparing to the computational cost and complexity of the Moser Veselov equations [200], these results are simply great. Results of numerical experiments are presented in Section 15.8 below.

In order to compare the new stepping scheme described in (15.61) with the previously derived semi-implicit scheme (15.29), all the terms in (15.61) are approximated using the formulae for the quaternionic velocities x, y in (15.50) and the quaternion z in (15.54). Neglecting terms of order higher than $O(h^2)$, the approximation is

$$\left(\mathcal{I}_0 - \frac{h^2 E_k}{2} I_3 - \frac{h^2}{4} \widehat{L}'_k \mathcal{I}_0^{-1} \widehat{L}'_k \right) \omega'_{k+1} = \left(1 - \frac{h^2 \|\omega_k\|^2}{2} \right) \left(\mathcal{I}_0 + \frac{h}{2} \widehat{L}'_k \right) \omega'_k. \tag{15.63}$$

Comparing this with the semi-implicit stabilized angular velocity stepper, as defined by (15.29), we find that replacing h for $h/2$ everywhere yields the desired result, except for the shift of the inertia tensor by $(h^2/2)E_k$. Though one might have guessed the $1/2$ factors—given how midpoint rules are practically always better—, the changes to the inertia tensor are genuinely new. Of course, there are more subtle differences over time if one uses the approximate variational stepper (15.61) directly to compute the quaternionic velocity. In fact, the stepping defined by (15.63) does not appear to be stable as shown in Section 15.8.

Next, if any sort of force were acting on the system, derived from a potential, a constraint or otherwise, the extra term hf_k will appear in the discretized Euler-Lagrange equation and this leads to the new definition of the vectors u and z

of (15.46)

$$\begin{aligned}
\tilde{u} &= Q(q_{k-1})\mathbb{I}_0 Q^T(q_{k-1})q_k + \frac{h^2}{4}f_k = u + \frac{h^2}{4}f_k, \\
\tilde{z} &= \tilde{u}^\dagger \star q_k = z + w, \\
w &= \frac{h^2}{4}f_k^\dagger \star q_k = \frac{h^2}{4}\begin{bmatrix} q_k^T f_k \\ -\mathcal{G}(q_k)f_k \end{bmatrix} = \frac{h^2}{4}\begin{bmatrix} q_k^T f_k \\ -\tau_k' \end{bmatrix},
\end{aligned} \tag{15.64}$$

where τ_k' in the last line is recognized as the net torque applied on the body, expressed in the body's fixed frame.

To get an understanding of the scalar term $q_k^T f_k$, go back to the analytic form (15.10) and after adding the force term f to it, multiplying on the left with q^T, and performing routine manipulations, the multiplier λ is found to be

$$\lambda = \frac{1}{4}q^T f + E, \tag{15.65}$$

which is a good indication of what will happen in the discrete case. Also, the order of the new quaternion term is now $w = O(h^2)$, both for the scalar and vector part.

The stepping equation is still an eigenvalue eigenvector problem

$$\big(\mathbb{I}_0 + Q(z) + Q(w)\big)\, x = \nu x. \tag{15.66}$$

The effect of a single term generic term w is first considered in the next paragraphs. The analysis is then extend to cover the special case where the real part of w vanishes, which is the case for constraint forces as shown below.

Proceeding with the real part of (15.66) as done previously for the force free case of (15.54), the eigenvalue ν satisfies

$$\begin{aligned}
\nu &= x_v^T \mathcal{I}_0 x_v + z_s + w_s \\
&= y_v^T \mathcal{I}_0 y_v + z_s + w_s + O(h^3) \\
&= 2z_s + w_s \\
&= h^2 E_k + \frac{h^2}{4} q_k^T f_k + O(h^3),
\end{aligned} \tag{15.67}$$

which is equivalent to the result for the continuous formulation in (15.65).

Next, using the penultimate line in (15.67) to approximate $\nu = 2z_s + w_s$, the vector components evaluate to

$$(\mathcal{I}_0 - z_s I_3)\, x_v = -x_s\,(z_v + w_v) - \widehat{z}_v x_v - \widehat{w}_v x_v, \tag{15.68}$$

and this is where we come to a branch. Since the real part of the torque quaternion w does not matter, assuming we do have an explicit expression for f_k, we can redefine $z \leftarrow z + w$ and reuse the stepping formula of (15.59). However, in case f_k is a constraint force for which there is no explicit expression, we keep w_v and z_v separate. Notice that in (15.68), the only term on the right hand side

which is of order h^2 or less is $-x_s w_v = -y_s w_v + O(h^3)$, which in fact evaluates to $y_s \tau_k'$.

Assuming instead that we have some constraint equations of the form $g(q) = 0$ and the Jacobian G is defined so that $\mathrm{d}g(q)/\,\mathrm{d}t = G(q)\omega' = 2G\mathcal{G}^T(q)\dot{q}$, then, the torque generated by that constraint is

$$\tau_c' = G^T(q)\rho, \tag{15.69}$$

for some set of Lagrange multipliers ρ (since λ is already used for unitarity constraint). Noting now that $w = O(h^2)$ which is one more power of h than the quaternionic momentum z, we can approximate (15.68) as follows

$$\begin{aligned} D &= \left(\mathcal{I}_0 + z_s I_3\right), \\ \tilde{M} &= D - \widehat{z}_v D^{-1} \widehat{z}_v, \\ \tilde{M} x_v &= -y_s \left(I_3 - \widehat{z}_v D^{-1}\right) z_v + y_s G_k^T \rho, \\ x_v &= \sqrt{1 - \|x_v\|^2}, \\ g(q_k \star x) &= 0, \end{aligned} \tag{15.70}$$

which is precisely the form of previously encountered mixed constrained problems, such as the formulation of (4.31), in Chapter 4.

The strength of the stepping formula (15.70) is that it is derived from the discrete variational principle and therefore, it is expected to approximately preserve the symplectic flow. In addition, it has precisely the form needed for insertion within the basic stepping framework in which the main equation reads $M v_{k+1} = f_k$, where M is some block diagonal, symmetric and positive definite mass matrix, v_{k+1} is a form of velocity vector, and f_k is the force term. Therefore, instead of having a complicated nonlinear system of equations for the rotational components of the rigid body motion, we have a simple modification of the mass matrix which can be computed one body at a time, using only known information.

Recovering the angular velocity vector from the quaternionic velocity vector x involves simple trigonometric manipulations which are usually inexpensive on current CPUs. There is perhaps one aspect that is problematic, namely, that in (15.70), we are solving for $x_v = O(h)$ but in the other equations of (4.31), say, we are solving for $v_{k+1} = O(1)$. This scaling might be problematic but since we can easily multiply everything in (15.70) with the time step h and solve for $(1/h)x_v$ instead of x_v itself, this should not be a fundamental issue.

15.7 The gyroscope

A heavy Lagrange top or gyroscope is a fast spinning rigid body with two identical principal inertiae, fixed with a ball joint at some point at distance l along the principal axis which has a different inertia (see Section 12.3 for definitions of principal inertia and principal axes). A schematics is shown in Figure 15.1.

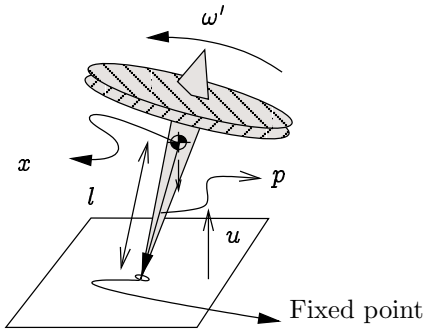

Figure 15.1: Schematics of a Lagrange top

Assuming the principal inertia are ordered with $\iota^{(i)} \geq \iota^{(j)}$ for $i > j$, and $\iota^{(1)} = \iota^{(2)}$, the top is *heavy* then $ml^2 + \iota^{(1)} > \iota^{(3)}$. The reason for this is clarified presently. Define the attachment point $p = -lRu$ where $l > 0$ is a scalar, $R(q)$ is the current rotation matrix of the body, $q \in \mathbb{H}$ is the orientation quaternion, and u is a unit vector in the upward direction in the inertial frame. The constraint specifies that if $x(t) \in \mathbb{R}^3$ is the center of mass of the body, then,

$$x - lR(q)u = 0, \tag{15.71}$$

which defines three bilateral, scleronomic holonomic constraint conditions. The meaning of (15.71) is that the center of mass $x(t)$ is actually a function of the orientation quaternion $q(t) \in \mathbb{H}$ of the body. Using the algebraic identities from Chapter 14, the velocity of the center of mass is

$$\dot{x} = -lR\widehat{u}R^T\omega, \tag{15.72}$$

and thus, the translational kinetic energy depends on the angular velocity as

$$\begin{aligned} &\frac{m}{2}\|\dot{x}\|^2 = \frac{ml^2}{2}\omega^T R(q)(\widehat{u}\widehat{u})R^T(q)\omega = \frac{ml^2}{2}\omega'^T(\widehat{u}\widehat{u})\omega', \qquad \text{and} \\ &\widehat{u}\widehat{u} = \mathsf{diag}(1,1,0). \end{aligned} \tag{15.73}$$

This means that by redefining $\mathcal{I}_0$ to

$$\mathcal{I}_l = \mathsf{diag}(I_1 + ml^2, I_2 + ml^2, I_3), \tag{15.74}$$

the kinetic energy reads $T(q,\dot{q}) = (1/2)\omega'^T\mathcal{I}_l\omega'$. The definition of $\mathcal{I}_l$ in (15.74) is an application of the parallel axis theorem [105] which describes how the inertia tensor changes when measured about a point other than the center of mass. In the present case, the inertia tensor is in fact evaluated about the fixed attachment point.

The potential energy for the gyroscope is due to uniform gravity and that is proportional to the elevation of the center of mass so $U(x) = mgu^Tx$, since vector u is an upward pointing unit. In terms of the orientation quaternion, this becomes the potential

$$\bar{V}(q) = mglu^TR(q)u. \tag{15.75}$$

The goal here is to manipulate this potential energy to a quadratic form in q. Using the expression $R(q) = I_3 + 2q_s\widehat{q}_v + 2\widehat{q}_v\widehat{q}_v$ of (13.75), a degenerate quadratic form is derived

$$\begin{aligned} V(q) &= 2mglq_v^T\widehat{u}_z\widehat{u}_z q_v = -2mglq_v^T P_{xy} q_v \\ &\simeq mglq^T H q, \quad \text{where} \\ P_{xy} &= \mathsf{diag}(1,1,0), \qquad \text{and} \qquad H = \mathsf{diag}(1,-1,-1,1). \end{aligned} \tag{15.76}$$

The equivalence sign $\simeq$ here means that the last expression for $V(q)$ differs from the previous one by a constant, a multiple of $q^T q = 1$ in fact, and such constants do not affect the motion, since the Euler-Lagrange equations only involve *derivatives* of the potential. Thus, the final expression for $V(q)$ in (15.76) differs from $\bar{V}(q)$ in (15.75) by a constant and the potential $V(q)$ can be used to model the uniform gravitational potential.

Therefore, the complete Lagrangian can be expressed in terms of the rotational variables only, taken here to be the quaternions, and is identical to the free body Lagrangian of (15.5) but subject to the potential (15.75). The generalized force (torque in this case) from this potential is found to yield

$$f_\phi = -\frac{\partial V}{\partial q^T} = -2mlgHq, \tag{15.77}$$

in quaternion form. The potential energy changes the DAEs of motion of the free rigid body (15.10) so they become

$$\begin{aligned} Q(q)\mathbb{I}_l Q^T(q)\ddot{q} + 2Q(\dot{q})\mathbb{I}_l Q^T(q)\dot{q} + \left(\lambda I_4 - \frac{mlg}{2}H\right) q &= 0 \\ \|q\|^2 - 1 &= 0, \end{aligned} \tag{15.78}$$

Euler equations (15.16) for their part become

$$\begin{aligned} \mathcal{I}_l\dot{\omega}' + \widehat{\omega}'\mathcal{I}_l\omega' + mlg\mathcal{G}(q)Hq &= 0 \\ \dot{q} - \frac{1}{2}\mathcal{G}^T(q)\omega' &= 0. \end{aligned} \tag{15.79}$$

It is possible to considerably simplify the equations of motion by applying Noether's theorem using what is called *reduction theory*. First, one observes that any rotation of the coordinate systems about the u axis leaves the equations of motion unchanged, and so does a global rotation about the origin. After taking the resulting invariants into accounts, there is only one remaining differential equation and it can be solved using elliptic functions [176] or some simplifications thereof as is done routinely in textbooks [105] to analyze the limit cases, especially the limit of the very fast top when $\omega_3' \to \infty$. In fact, the common textbook analysis entirely neglects the steps shown above to reduce the equations of motion and routinely contradict the assumption that $\iota^{(1)} \leq \iota^{(2)} \leq \iota^{(3)}$ when it comes to discuss the Lagrange top. Also, if the fixed point is not really fixed but merely constrained to a plane for which the action of gravity is a normal, the

translational motion of the contact point in the plane is in fact coupled to the rotational motion of the top.

Reduction theory has not been covered, principally because it requires yet more mathematical tools than have been presented so far, including Lie derivatives and Poisson brackets. An analysis of the Lagrange top based on reduction theory can be found in a paper by Bobenko and Suris [51] though, along with a variational discretization of the reduced problem. This particular discretization is perhaps the most efficient, simplest, and most beautiful way to integrate the equations of motion (reader beware of yet another definition of the inertia tensor to account for the coordinate reduction in this formulation).

The theme in this thesis is that of constrained multibody systems though. It thus makes sense to implement the variational stepper of Section 15.6 and the linearized approximation thereof on the constrained system described at the beginning of this section. The results of this strategy are shown in Section 15.8, with comparisons against the variational discretization of the reduced problem [51] and a standard integrations of the DAEs of motion (15.11) augmented to account for the torque term (15.77) and stabilized to maintain the unit quaternion constraint $\|q\| = 1$ using a stiff penalty term. This is not a generally recommendable strategy but for simple systems, given a standard high order integrator with adaptive time step, the results can be very accurate though the computations are not efficient.

15.8 Numerical experiments

The variational stepping methods defined by (15.47), the approximation thereof defined by (15.59) and the preceding definitions, as well as the stabilized stepper defined in (15.29) were implemented using Octave, using the LAPACK [6] routines to solve the eigenvalue eigenvector problems when needed. For comparison, an implementation of the coupled system of equations in the body frame (15.19), using the `lsode` solver of Octave was realized.

Two cases were simulated as follows

Case 1: $\mathcal{I}_0 = \mathsf{diag}(0.9144, 1.098, 1.66)$, and $\omega'(0) = (0.45549, 0.92625, 0.03476)^T$, shown in Figures 15.2 and 15.3;

Case 2: $\mathcal{I}_0 = \mathsf{diag}(2, 1, 5)$, $\omega'(0) = (1, 20, 2)^T$, shown in Figures 15.4 and 15.5.

Case 1 appears in [200] and is repeated in [163] and illustrates a case of moderate speed. The components of angular velocities are shown in Figure 15.2 and the energy appears in Figure 15.3. On the scale of the plots, it is not possible to distinguish between the variational stepper, the approximated variational one, and the direct ODE integration, but the stabilized stepper of Section 15.5 is not doing so well. On the energy plots in Figure 15.3, the two variational steppers are exhibiting the usual oscillations but are nicely behaved. The stabilized stepper is strongly dissipative, however.

Case 2 illustrates what happens for a much higher angular speed. Note here that in [163], only the first case was considered but the time step was made enormous, with $h \in [1/2, 4]$. For the free rigid body, the homogeneity of the equations of motion means that changing time step is equivalent to scaling the initial angular speed. Thus, in Case 2, the angular velocity trajectories oscillate very quickly as shown in Figure 15.4. Yet, the pure variational and its approximation are holding well in comparison to the direct ODE integration, which uses adaptive time steps. The stabilized stepper however quickly settles to uniform rotations about I_3, as predicted by the stability Theorem 15.2. Meanwhile, the energy for the variational stepper and its approximation oscillate but that of the stabilize stepper decreases until the motion stabilizes to $\omega' = (0, 0, \pm\|\omega'\|)^T$.

Simulation of Lagrange tops were also performed using an ODE formulation, the Bobenko and Suris integrator [51], the variational stepper of Section 15.6 and the linear approximation thereof. In this latter case however, the linearized approximation of the variation stepping equations of do not hold up at high rotational speeds. The case of slow speed, when the top cannot stand upright, is shown in Figures 15.6 and 15.7. Figures 15.8 and 15.9 summarize the results for a faster example which cannot hold upright, yet. Finally, Figures 15.10 and 15.11 summarize results for a top rotating fast enough to hold up. The approximation technique breaks down in this case, though without instability. The full variational technique still works well though. The Bobenko and Suris [51] stepper is more efficient though since it simply computes the elements of tow 3×3 matrices and performs matrix-vector product operations—it is fully explicit! The standard ODE solver of Octave struggles with small time steps in this regime. This interesting problem warrants further investigation.

In all examples involving gyroscopes, the inertia tensor before the application of the parallel axis theorem is $\mathcal{I}_0 = (1, 1, 2)$, the distance between the center of mass and the attachment point is $l = 4$, the magnitude of gravity is $g = 10$, and the initial angle is 10 degrees. The initial condition is $\omega' = (0, 0, \omega_3)$ and ω_3 was chosen as $\omega_3 = 0.1$ for Figure 15.6 and Figure 15.7, $\omega_3 = 2.0$ for Figure 15.8 and Figure 15.9, and $\omega_3 = 20.0$ for Figure 15.10 and Figure 15.11. Using the known estimates [105] for instance, the top is fast and stays upright whenever $\omega_3 > 5$ for the chosen values of inertia.

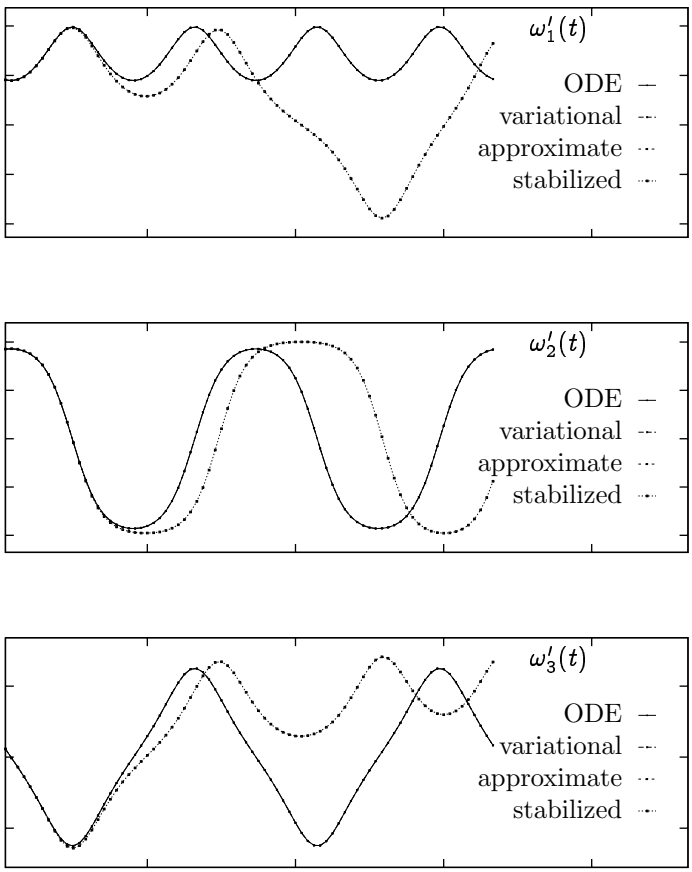

Figure 15.2: Time history of the three components of the angular velocity vector for a rigid body rotating freely in three dimensions at moderate speed, integrated with four different methods.

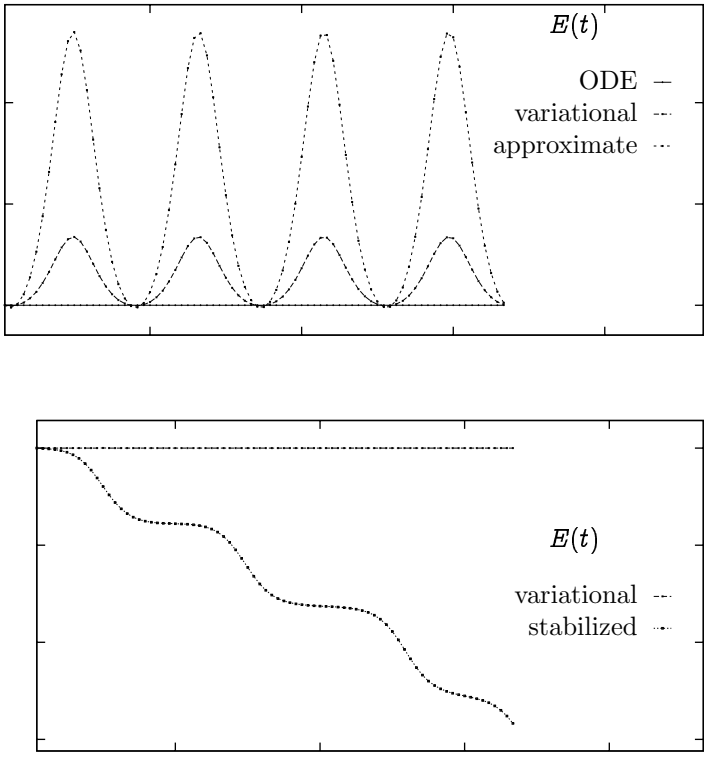

Figure 15.3: The energy of a rigid body rotating freely in three dimensions at moderate speed when simulated with four different methods.

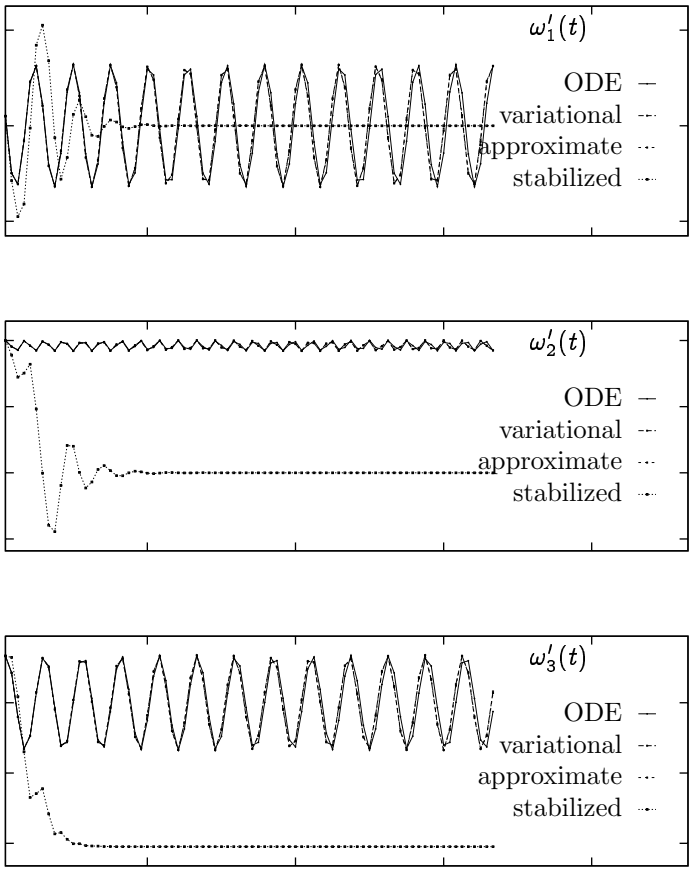

Figure 15.4: Time history of the three components of the angular velocity vector for a rigid body rotating freely in three dimensions at high speed, integrated with four different methods. Notice how the stabilized stepper quickly settles to simple rotation about the axis with the largest inertia.

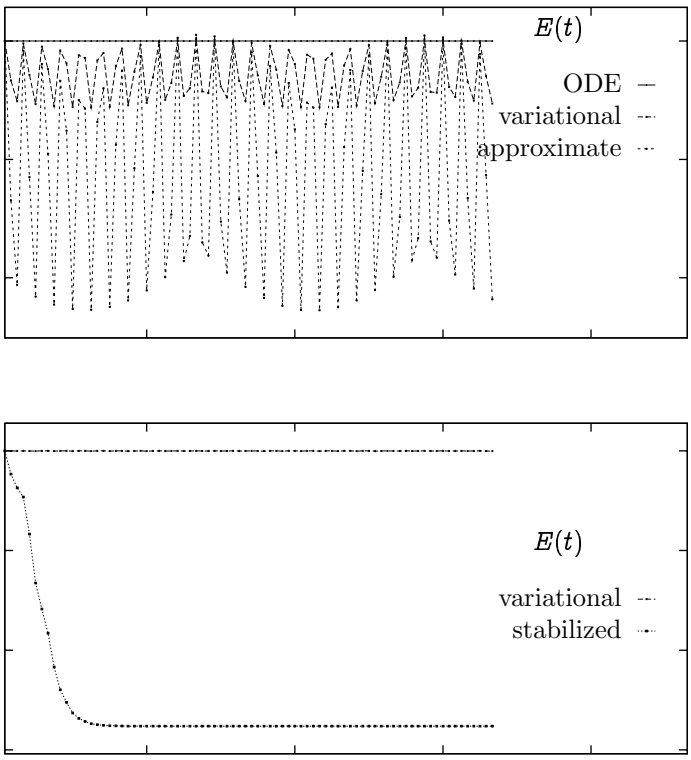

Figure 15.5: The energy of a rigid body rotating freely in three dimensions at moderate speed when simulated with four different methods. Note how the energy of the stabilized stepper decreases quickly until the motion stabilizes to simple rotation about the axis with the largest inertia.

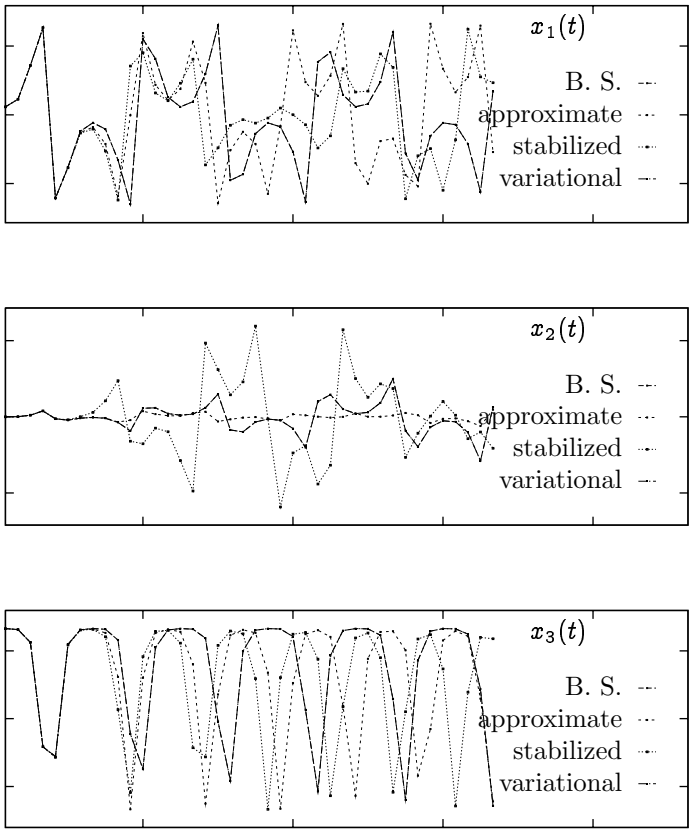

Figure 15.6: The three components of the center of mass of a slow, heavy symmetrical top simulated with four different methods. The rotational speed is too low here to keep the top upright.

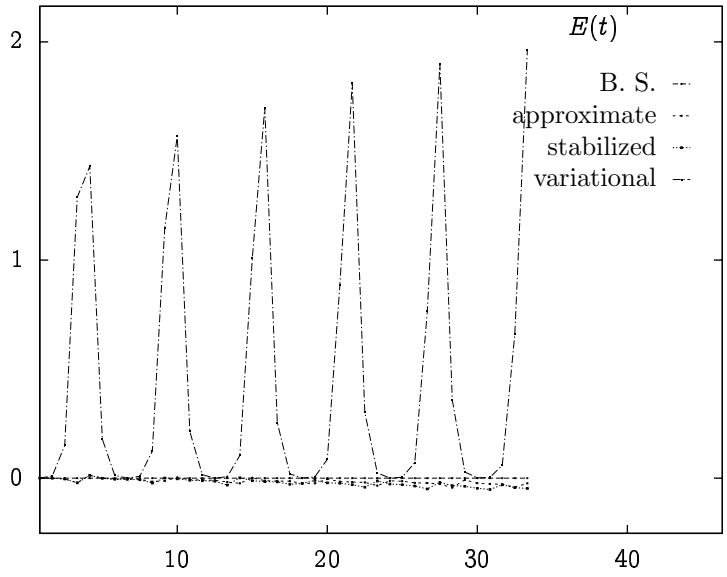

Figure 15.7: Energy of the slow, heavy symmetrical top when simulated with four different integration schemes.

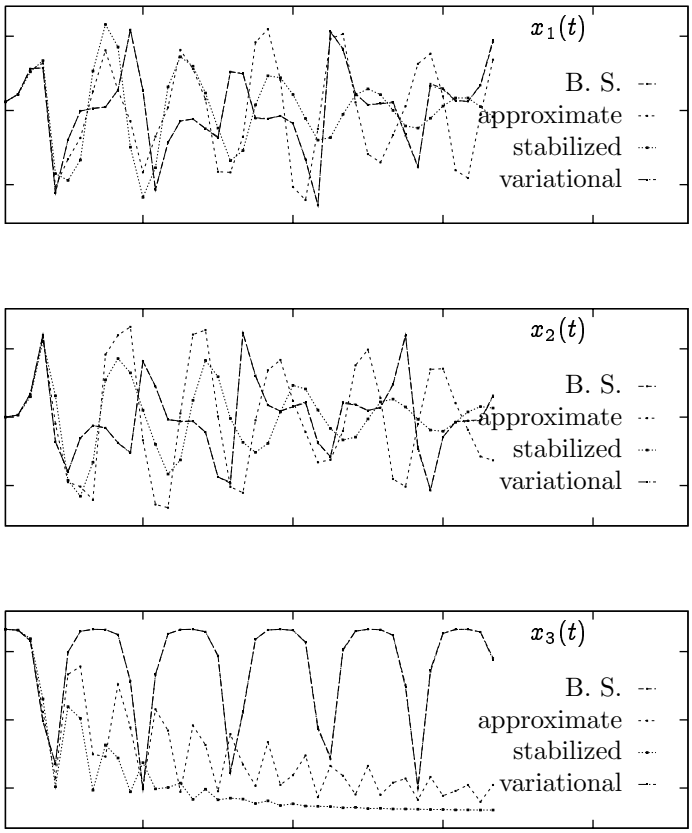

Figure 15.8: The three components of the center of mass of a moderate, heavy symmetrical top simulated with four different methods. The rotational speed is still too low here to keep the top upright, but the approximated stepper still performs well.

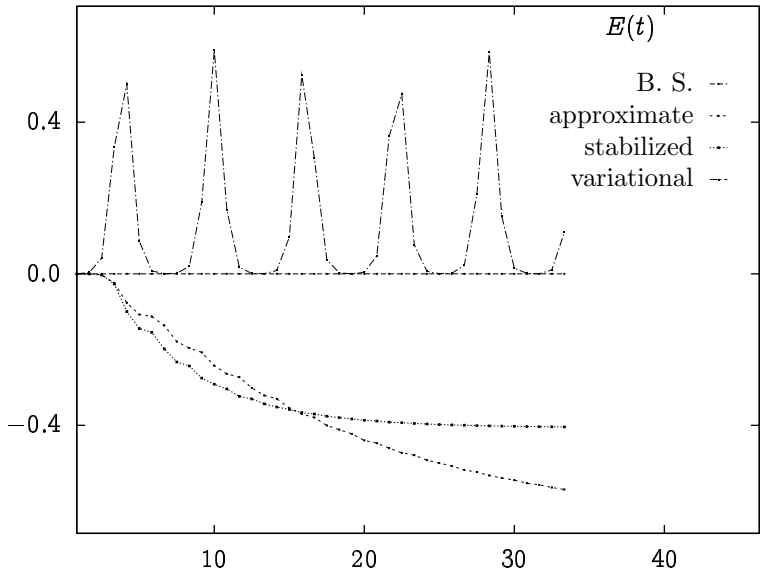

Figure 15.9: Energy of the moderate, heavy symmetrical top when simulated with four different integration schemes.

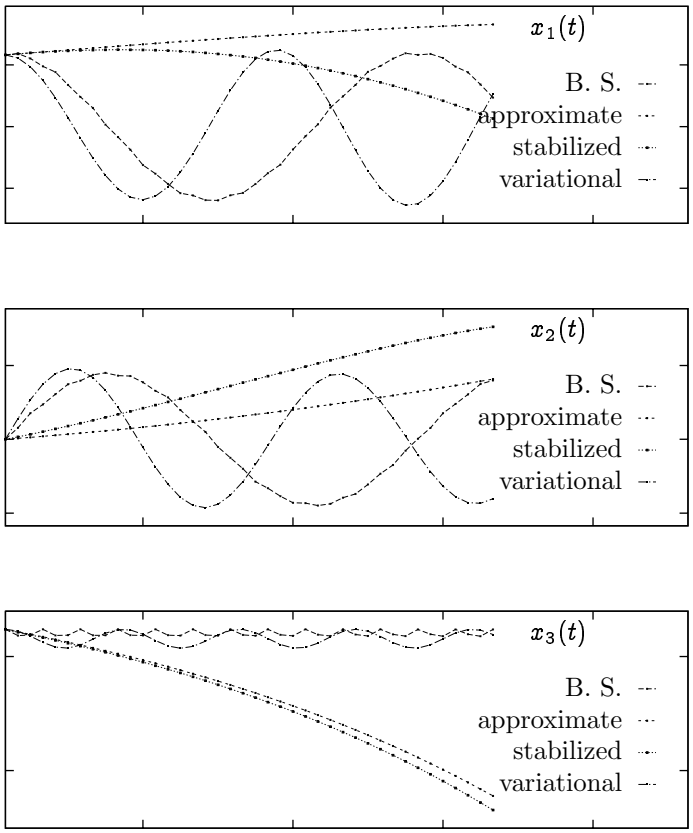

Figure 15.10: The three components of the center of mass of a fast, heavy symmetrical top simulated with four different methods. The rotational speed is high enough here to keep the top upright, but the approximated variational stepper breaks down.

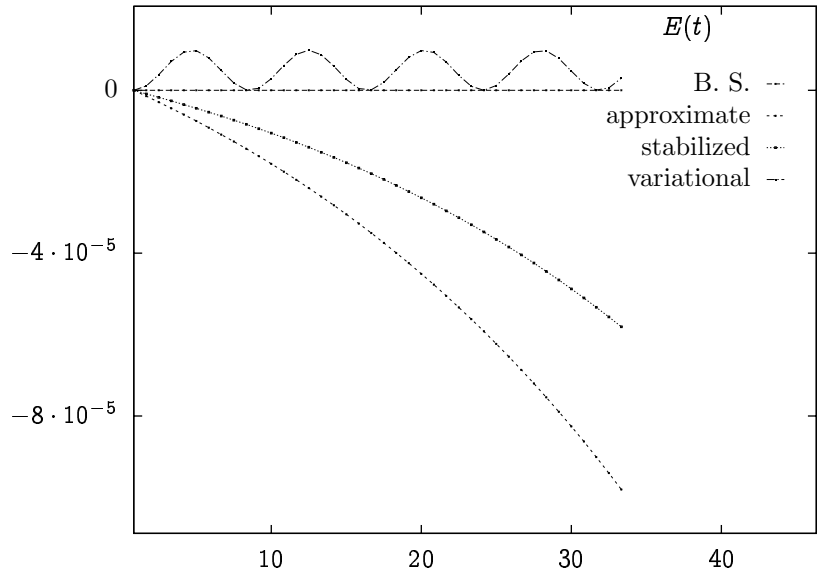

Figure 15.11: Energy of the moderate, heavy symmetrical top when simulated with four different integration schemes.

15.9 End notes

Good methods for integrating the gyroscopic forces of rigid body are routinely needed in molecular dynamics. Except for one overly simple geometric integration technique [81], all existing methods are implicit to some degree [200, 163]. Being implicit is not necessarily a bad thing but the main issue is that it is not clear how to include constraints in the formulation to go from free rigid bodies or gyroscopes even, to full multibody systems subject to kinematic constraints, without destroying the saddle-point structure of the equations of motion. By contrast, mass matrix modification techniques such as those presented in this chapter are explicit, easy to implement, and integrate seamlessly with other multibody methods. The ability to integrate the free rigid body with an explicit method and within what appears to be global error bounds is of significance. Future work will determine how far the linearized approximation of Section 15.6 can go, and what can be done to refine them when they are not delivering good enough estimates.

The results of this section are a first step in the direction of simple integration methods allowing both accurate processing of gyroscopic forces and general constraints, but there is more work to do to refine this.

16 Complementarity I: Definitions and Classical Solution Methods

Complementarity is the mathematics of decision making, when a choice for "this" or "that" must be made. It is rooted in mathematical programming where the minimization of a constrained function naturally lead to complementary alternatives. The present chapter provides the background necessary for solving the discrete Euler-Lagrange equations presented in Chapter 10 where impacts and dry friction are considered.

Background is provided in Section 16.1 by considering the problem of optimizing a one-dimensional function over a bounded domain. Linear complementarity is introduced in Section 16.2, presenting the problem and its solutions in one and two dimensions, before introducing definitions necessary for generalizations to higher-dimensional problems. The nonlinear complementarity problem is defined in Section 16.3 and it allows for a simple solvability analysis presented in Section 16.4. This is based on Leray-Schauder type alternatives and the notion of *exceptional families of elements* introduced recently by Isac [133]. After that, classical solution methods based on Gauss-Jordan pivot operations are introduced in Section 16.5 where they are presented in block matrix form, in contrast with the standard pivotal algebraic format of mathematical programming textbooks. The reason for this is that implementation based on block matrices can use high performance basic linear algebra subroutines (BLAS) libraries, especially the general matrix matrix multiplication (GEMM)-based variants which have top performance [149, 148]. Iterative methods are covered in Section 16.6 and methods based on Newton-Raphson techniques in Section 16.7. The results of numerical experiments are presented in Section 16.8 and general remarks are provided in Section 16.9.

16.1 Introduction and background

The simplest instance of a complementarity problem is the minimization of a real scalar function $f : \mathbb{R} \mapsto \mathbb{R}$ over a *bounded* interval $[a, b] \in \mathbb{R}, a < b$. From calculus, we expect that $f(x)$ is minimal when the sufficient condition $f'(x) = 0$ but this may or may not occur in the interval $[a, b]$. One of three situations may occur, namely,

1. **criticality:** $f'(x^{(i)}) = 0$ for $x^{(i)} \in [a, b], i = 1, 2, \ldots,$

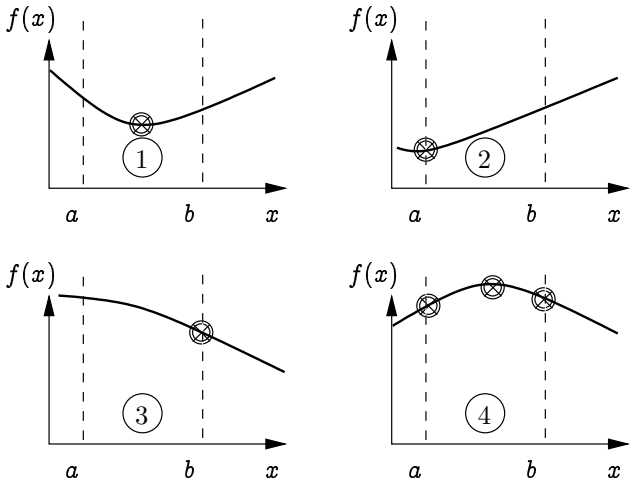

Figure 16.1: A simple constrained optimization problem in one dimension.

2. **monotonic increase:** $f'(x) \geq 0$ for $x \in [a, b]$,

3. **monotonic decrease:** $f'(x) \leq 0$ for $x \in [a, b]$.

Therefore, we look for the minimum point at the lower bound a if the function f is increasing there, namely, if $f'(a) > 0$, at the upper bound b if the function f is decreasing there, namely, if $f'(b) < 0$, and at any critical point $x_i \in [a, b]$ as well. Notice that the bounds a and b are candidates only if the derivative is non-negative at a or non-positive at b. The various cases are illustrated in Figure 16.1. In the first graph with label 1, the function has a unique critical point in the interval $[a, b]$ and is minimal there. The graph with label 2 illustrates the case of a monotonically increasing function which has a minimum at a with $f'(a) \geq 0$. Graph with label 3 illustrates the case of a monotonically decreasing function which has a minimum at b with $f'(b) \leq 0$. Finally, in the graph with label 4, we show a case in which the sufficient conditions listed above are not enough to determine the minimum point and an explicit evaluation is necessary instead. This set of necessary conditions for optimality can be phrased succinctly as a complementarity problem

$$\begin{aligned} f'(z^{(i)}) - w_+^{(i)} + w_-^{(i)} &= 0 \\ 0 \leq z^{(i)} - a \ &\perp \ w_+^{(i)} \geq 0 \\ 0 \leq b - z^{(i)} \ &\perp \ w_-^{(i)} \geq 0, \end{aligned} \tag{16.1}$$

where $z^{(i)}, i = 1, 2, \ldots$, is the ordered set $\{a, x^{(1)}, x^{(2)} \ldots, b\}$, $f'(x^{(i)}) = 0$. At the boundary points where $z^{(i)} = a$ or $z^{(i)} = b$, the derivative may not vanish and so we can decompose it into positive and negative parts as $f'(z^{(i)}) = w_+^{(i)} - w_-^{(i)}, w_\pm^{(i)} \geq 0$.

For the case $f(x) = \frac{1}{2}mx^2 + qx$, where m and q are simple scalars, the constrained minimization problem consists of solving the following mixed linear complementarity problem (MLCP)

$$\begin{bmatrix} m & -1 & 1 \\ 1 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ w_+ \\ w_- \end{bmatrix} + \begin{bmatrix} q \\ -a \\ b \end{bmatrix} = \begin{bmatrix} 0 \\ u \\ v \end{bmatrix} \qquad 0 \leq \begin{bmatrix} u \\ v \end{bmatrix} \perp \begin{bmatrix} w_+ \\ w_- \end{bmatrix} \geq 0. \tag{16.2}$$

This problem can be solved by simple enumeration. Indeed, first set $w_+ = w_- = 0$ and solve for $x = -m/q$. Then check that this is within the bounds $[a, b]$ and if so, the problem is solved. Otherwise, try for $x = a$ and decompose $ma + q = w_+ - w_-$. If this is satisfied for $w_- = 0$, the solution is found. Otherwise, try at $x = b$ and decompose $mb + q = w_+ - w_-$. If $w_+ = 0$, the solution is found.

Solving for x in (16.2) and substituting, we obtain the linear complementarity problem (LCP)

$$m^{-1} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} w_+ \\ w_- \end{bmatrix} + \begin{bmatrix} -m^{-1}q - a \\ m^{-1}q + b \end{bmatrix} = \begin{bmatrix} u \\ v \end{bmatrix} \qquad 0 \leq \begin{bmatrix} u \\ v \end{bmatrix} \perp \begin{bmatrix} w_+ \\ w_- \end{bmatrix} \geq 0. \tag{16.3}$$

This is particularly instructive since it is obvious that the matrix on the left hand side of the first line in (16.3) is *singular* because the second row is a scalar multiple of the first. This shows that we have two redundant equations and it suffices to solve one of them, checking the other one is consistent. For this simple case, one of five alternatives are possible namely:

1. **internal minimum:** $-q/m \in [a, b]$
2. **lower bound minimum:** $-q/m \notin [a, b]$ and $ma + q > 0$
3. **upper bound minimum:** $-q/m \notin [a, b]$ and $mb + q < 0$
4. **lower bound degeneracy:** $-q/m = a$
5. **upper bound degeneracy:** $-q/m = b$.

When this analysis is applied to a general multidimensional constrained optimization problem, the result is the famous Karush-Kuhn-Tucker conditions (KKT) theorem (see for instance [49], Section 3.3, or any other book on nonlinear programming) which is now cited.

Theorem 16.1. *Karush-Kuhn-Tucker Necessary Conditions. Let $x^\star$ be a local minimum of function $f : \mathbb{R}^n \mapsto \mathbb{R}$, subject to the continuously differentiable*

constraints $h : \mathbb{R}^n \mapsto \mathbb{R}^m, h(x) = 0$, *and* $g : \mathbb{R}^n \mapsto \mathbb{R}^r, g_i(x) \geq 0, i = 1, 2, \ldots, r$, *and assume that* $x^\star$ *is regular, then, there exist unique Lagrange multipliers* $\lambda^\star = (\lambda_1^\star, \lambda_2^\star, \ldots, \lambda_m^\star)^T$, *and* $\mu^\star = (\mu^\star, \mu_2^\star, \ldots, \mu_r^\star)^T$, *such that*

$$\begin{aligned} \frac{\partial f(x^\star)}{\partial x^T} - \frac{\partial h(x^\star)}{\partial x^T}\lambda^\star - \frac{\partial g(x^\star)}{\partial x^T}\mu^\star &= 0 \\ h(x^\star) &= 0 \\ 0 \leq g(x^\star) \ \perp \ \mu^\star &\geq 0. \end{aligned} \tag{16.4}$$

Note that a point x is regular if the Jacobian rows $\partial h_i / \partial x, i = 1, 2, \ldots, m$, and $\partial g_j / \partial x$ for each $j \in \{1, 2, \ldots, r\}$ *s.t.* $g_j(x) = 0$ are linearly dependent. The KKT conditions illustrate clearly the connection between constrained nonlinear optimization and the mixed complementarity problem.

16.2 Linear complementarity

Though the MLCP is natural in the context of nonlinear programming, one can reduce such a problem to a pure LCP by performing a Schur complement operation, i.e., solving for the equality conditions and substituting the results back. This leads to the fundamental definition of the linear complementarity problem.

Definition 16.1 (LCP). *Given an* $n \times n$ *real matrix* M *and an* n*-dimensional real vector* q*, the linear complementarity problem* $\mathrm{LCP}(M, q)$ *is that of finding* n*-dimensional real solution vectors* $z \in \mathbb{R}^n$ *such that*

$$\begin{aligned} Mz + q &= w \\ 0 \leq z \ \perp \ w &\geq 0. \end{aligned} \tag{16.5}$$

The set of solution vectors is labeled $\mathrm{SOL}(\mathrm{LCP}(M, q))$. *The* n*-dimensional vector* w *is vector of slack variables.*

The idealization of LCP is very useful to develop and understand solution algorithms. In practice however, it is more common to find mixed problems consisting of coupled linear equations and a mix of unbounded as well as bounded variables. In addition, bounded variables might have an upper, a lower, or both upper and lower bounds. The MLCP was defined in 16.2 and is discussed in[69].

The formal definition of MLCP is now provided.

Definition 16.2 (MLCP). *For an* $n \times n$ *real matrix* M*, an* n*-dimensional real vector* q*, and two* n*-dimensional extended real vectors,* l *and* u*, with* $l_i, u_i \in \mathbb{R} \cup \{\pm\infty\}$ *and* $l_i \leq u_i$*, the problem* $\mathrm{MLCP}(M, q, l, u)$ *is to find solution vectors* z *such that*

$$\begin{aligned} Mz + q &= w_+ - w_- \\ 0 \leq w_+ \perp z - l &\geq 0, \\ 0 \leq w_- \perp u - z &\geq 0. \end{aligned} \tag{16.6}$$

The vectors w_+ *and* w_- *contain the slack variables. The solution set of* $\mathrm{MLCP}(M, q, l, u)$ *is called* $\mathrm{SOL}(\mathrm{MLCP}(M, z, q, l, u))$.

Of course, $\mathrm{MLCP}(M, q, l, u))$ is equivalent to $\mathrm{LCP}(M, q)$ when $l_i = 0$ and $u_i = \infty$.

A slightly different way to view MLCP is to consider the slack variables w_+ and w_- as unknowns and to rewrite (16.6) as

$$\begin{bmatrix} M & -I_n & I_n \\ I_n & 0 & 0 \\ -I_n & 0 & 0 \end{bmatrix} \begin{bmatrix} z \\ w_+ \\ w_- \end{bmatrix} + \begin{bmatrix} q \\ -l \\ u \end{bmatrix} = \begin{bmatrix} 0 \\ r \\ s \end{bmatrix} \qquad (16.7)$$
$$0 \leq r \quad \perp \quad w_+ \geq 0$$
$$0 \leq s \quad \perp \quad w_- \geq 0.$$

This form makes it more clear that a mix of equalities is being dealt with.

It is not very difficult to extend an LCP algorithm to handle mixed problems though, in most cases, the class of matrices which can be solved by the resulting algorithm is much smaller than the original. Because the extensions to MLCP for principal pivot methods are straight forward, they are not provided here. Extensions to the Lemke algorithm 16.5.6 are more difficult and are found in [245, 146].

16.2.1 One-dimensional problems

To fix some ideas, consider the one-dimensional LCP

$$\begin{aligned} mz + q &= w \\ 0 \leq z \perp Mz + q &\geq 0, \end{aligned} \qquad (16.8)$$

where m and q are real scalars.

The following cases are found.

1. $q > 0$. Solution is $z = 0$ and $w = q$;
2. $q < 0, m > 0$. Solution is $z = -q/m$, and $w = 0$;
3. $q < 0, m \leq 0$. There is no solution.

Note that the case $q = 0$ produces the *degenerate solution* $z = w = 0$.

This example already illustrates that the notion of *positivity* of the matrix M in (16.5) plays a central role in complementarity theory. It also illustrates how zero components of vector q leads to *degeneracy* which corresponds to simultaneous vanishing of components of the solution vector z and the slack vector w.

16.2.2 Two-dimensional problems

Now consider (16.5) for the case of a 2×2 real matrix M. Partition M by columns as

$$M = \begin{bmatrix} -u & -v \end{bmatrix}, \tag{16.9}$$

and therefore, the solution vector for (16.5) can be written as

$$q = z_1 u + z_2 v + w_1 e_1 + w_2 e_2, \tag{16.10}$$

where e_i are the basis vector of $\mathbb{R}^2$ with a 1 at position i and 0 elsewhere. Because of the complementarity conditions, this can be further written as

$$q = \sigma x + \tau y,\ \sigma, \tau \geq 0,\ x \in \{u, e_2\}, y \in \{v, e_2\}, \tag{16.11}$$

which means that vector q must lie in one of the four *cones*

$$\begin{aligned} \mathcal{K}_1 &= \{r \in \mathbb{R}^2 \mid r = \sigma u + \tau v, \sigma, \tau \geq 0\}, \\ \mathcal{K}_2 &= \{r \in \mathbb{R}^2 \mid r = \sigma e_1 + \tau v, \sigma, \tau \geq 0\}, \\ \mathcal{K}_3 &= \{r \in \mathbb{R}^2 \mid r = \sigma u + \tau e_2, \sigma, \tau \geq 0\}, \\ \mathcal{K}_4 &= \{r \in \mathbb{R}^2 \mid r = \sigma e_1 + \tau e_2, \sigma, \tau \geq 0\}. \end{aligned} \tag{16.12}$$

Obviously, if vector q is not found in any of these four cones, there is no solution.

For this case, we can perform a complete enumeration of the possible LCPs. There are four quadrants in $\mathbb{R}^2$ and therefore, there are four choices for each column of matrix M, namely, vectors $u = -M_{\bullet 1}$, and $v = M_{\bullet 2}$. This is illustrated in Figure 16.2. Each sub-figure is labeled according to which quadrant vectors u and v belong to. The arcs have increasing radius from the origin spanning the cones $\mathcal{K}_i, i = 1, 2, 3, 4$. In order to be able to solve LCP (M, q) for a given vector q, it must lie in a region of the plane which is spanned by at least one of the arcs. If two arcs cover the same region, as in sub picture III/III for instance, the corresponding LCP has multiple solution vectors z.

The interesting cases are those for which there is a unique solution to (M, q) for any $q \in \mathbb{R}^2$ and this is seen to occur in sub pictures II/III and III/IV. For both of these cases, we find that $u_1 = -m_{11} < 0$, $v_2 = -m_{22} < 0$ and $u_1 v_2 - u_2 v_1 = \det(M) > 0$. For this last identity, observe that $\det(M)$ is proportional to the signed area spanned by vectors u, v. Collecting these results, we find that given a real 2×2 matrix M which has all positive minors, then, LCP (M, q) has a unique solution for any $q \in \mathbb{R}^2$. Real $n \times n$ matrices M which have all positive principal minors are called P matrices and those which have non-negative principal minors are the P_0 matrices. These observations are generalized to n dimensions in the next section.

In geometric terms, observe that in cases II/III and III/IV, we can assign one of the vectors e_1, e_2, u, v uniquely to each of the four quadrants. For case II/III, the assignment is e_1 : IV, e_2 : I, u : II, v : III, and for case III/IV it is: e_1 : I, e_2 : II, u : III, v : IV. This also generalizes to $\mathbb{R}^n$ in which cases P matrices have been shown to have this pigeon hole property.

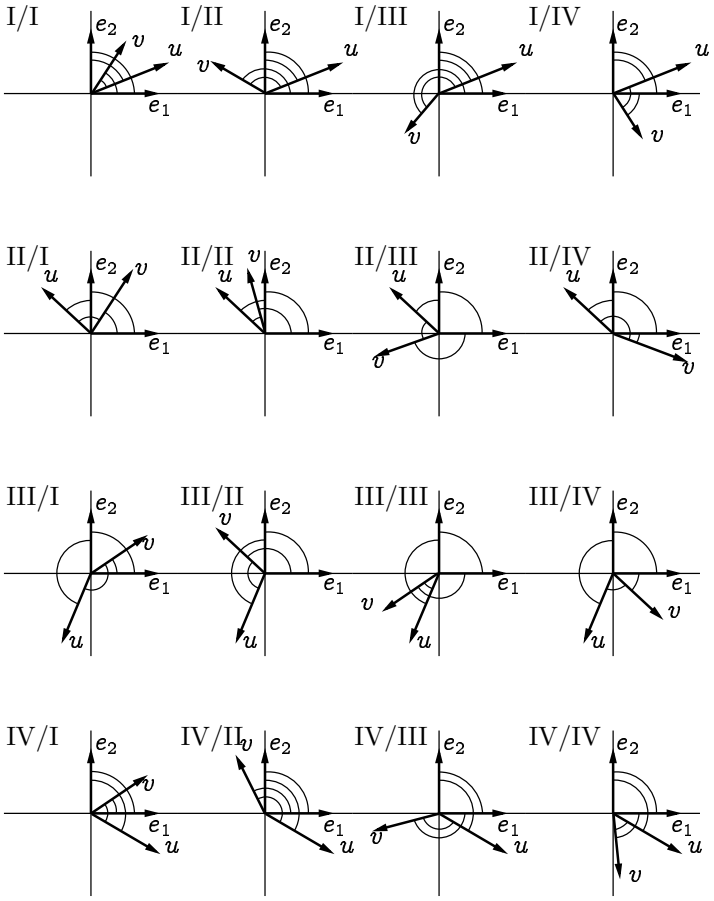

Figure 16.2: The 16 cases of two-dimensional LCP. The arcs shown with increasing radius illustrate the span for the four different potential solution bases, namely, (u, v), (e_1, v), (u, e_2), and (e_1, e_2).

Observe further that the case shown in sub-figure I/I corresponds to a matrix M which has all negative entries. For this case, there is only the trivial solution $z = 0$. So, again, the key to solving LCPs is to identify how *positive* a given matrix is.

To fix ideas even further, consider the following examples taken from [144].

Example 16.1. *For* $\mathsf{LCP}(M, q)$ *with*

$$M = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \; q = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \tag{16.13}$$

the unique solution is $z = (0, 0)^T$, $w = (1, 1)^T$. *This example corresponds to a limit case of either case II/III or case III/IV.*

Example 16.2. *For* $\mathsf{LCP}(M, q)$ *with*

$$M = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \; q = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \tag{16.14}$$

the solution set consists of $z = (\lambda, 1 - \lambda)^T, \lambda \in [0, 1]$, $w = (0, 0)^T$.

Example 16.3. *For* $\mathsf{LCP}(M, q)$ *with*

$$M = \begin{bmatrix} -1 & 0 \\ 1 & 2 \end{bmatrix}, \; q = \begin{bmatrix} -1 \\ 2 \end{bmatrix}, \tag{16.15}$$

there is no solution since for $z_1 \geq 0$, $w_1 = -1 - z_1 \leq 0$ *and therefore, the problem is infeasible.*

Example 16.4. *For* $\mathsf{LCP}(M, q)$ *with*

$$M = \begin{bmatrix} 1 & 0 \\ 1 & -2 \end{bmatrix}, \; q = \begin{bmatrix} 3 \\ -1 \end{bmatrix}, \tag{16.16}$$

the first equation leads to $z_1 + 3 = w_1$ *and since* $z_1 \geq 0$, $w_1 \geq 3$. *However, the second equation* $z_1 - 2z_2 - 1 = 0$ *cannot be solved for* $z_1 = 0$ *which means that both* $z_1 > 0$ *and* $w_1 > 3 > 0$ *and therefore, the problem has no solution.*

16.2.3 Definitions and classes of matrices

The LCP is already defined formally in Definition 16.1 and the MLCP in Definition 16.2, along with the notation for the solution sets. Introduce the *feasible set*, $K(z, w)$, which contains all the potential candidates for solving $\mathsf{LCP}(M, q)$.

Definition 16.3. *Given a square* $n \times n$ *matrix* M *and an* n*-dimensional vector* q, *the feasible set* $K(z, w)$ *of* $\mathsf{LCP}(M, q)$ *is*

$$K(z, w) : \{z, w \in \mathbb{R}^n_+ \mid w = q + Mz\}. \tag{16.17}$$

It follows that $\mathtt{SOL}(\mathtt{LCP}(M,q)) \in K(z,w)$. Of course, for the case where $K(z,w) = \emptyset$, $\mathtt{LCP}(M,q)$ is *infeasible* and has therefore no solution.

Definition 16.4 (Submatrices). *Given a square $n \times n$ matrix M, sets of the form $\alpha \subseteq \{1,2,\ldots,n\}$ are called index sets. Matrices of the form*

$$M_{\alpha\alpha} = (m_{ij}), \quad i,j \in \alpha, \tag{16.18}$$

are called principal submatrices of M, whilst matrices of the form

$$M_{\alpha\beta} = (m_{ij}), \quad i \in \alpha, j \in \beta, \tag{16.19}$$

are submatrices of M.

Definition 16.5 (The cone of matrix M). *Given the $n \times m$ real matrix M, with no restriction on n, m, a cone is a set of the form*

$$\mathcal{K}(M) = \{z \in \mathbb{R}^n \mid z = \sum_{j=1}^{m} \sigma_j M_{\bullet j}, \sigma_j > 0\}. \tag{16.20}$$

In other words, a cone $\mathcal{K}(M)$ is the positive span of the columns of matrix M, sometimes written $\mathcal{K}(M) = \mathtt{pos}(\{M_{\bullet i}\}_{i=1}^{i=n})$. Given $\mathtt{LCP}(M,q)$, we define the 2^n complementarity cones for this problem as:

Definition 16.6 (The augmented cone of matrix M). *Given a square $n \times n$ matrix M, let α be an index set, $\alpha \subseteq \{1,2,\ldots,n\}$ and define β so that $\alpha \cap \beta = \emptyset$ and $\alpha \cup \beta = \{1,2,\ldots,n\}$. There are 2^n different index sets α and for each of these, define the cone*

$$\mathcal{K}_\alpha(M) = \mathcal{K}\left(\begin{bmatrix} I_{\bullet\beta} & -M_{\bullet\alpha} \end{bmatrix}\right), \tag{16.21}$$

in other words, $\mathcal{K}_\alpha(M)$ is the positive span of the negative α columns of M and β columns of the identity.

In view of Def. 16.6, solving $\mathtt{LCP}(M,q)$ consists of identifying the cones such that $q \in \mathcal{K}_\alpha(M)$. From this perspective, once the index set α is identified, one must solve the linear system

$$Bx = q, \tag{16.22}$$

where the columns of B are $B_{\bullet\alpha} = -M_{\bullet\alpha}$, and $B_{\bullet\bar{\alpha}} = I_{\bullet\bar{\alpha}}$. In this representation, the columns of matrix B is a *basis* for the problem. In addition, we say that z_i is a *basic* variable if $i \in \alpha$ and *non-basic* otherwise. Conversely, w_i is a basic variable if z_i is non-basic, and vice versa.

As explained in the previous section, the notion of positivity plays a central role in complementarity theory so we introduce the following classes of matrices.

Definition 16.7. *A positive (semi-)definite $n \times n$ matrix M is such that given any $x \in \mathbb{R}^n$, $x \neq 0$, then*

$$\begin{aligned} &x^T Mx > 0 \text{ for } M \text{ positive definite}, \\ &x^T Mx \geq 0 \text{ for } M \text{ positive semi-definite}. \end{aligned} \tag{16.23}$$

As is well-known, symmetric matrices have real eigenvalues [107]. It follows from this that symmetric, positive semi-definite matrices have non-negative eigenvalues, and that symmetric positive definite matrices have strictly positive eigenvalues.

In addition, we introduce the following class of matrices.

Definition 16.8. *A square $n \times n$ real matrix M may belong to one or several of the following classes.*

Q-matrices: *matrices M such that $\mathrm{LCP}(M, q)$ has a solution for all $q \in \mathbb{R}^n$;*

Q_0-matrices: *matrices M such that $\mathrm{LCP}(M, q)$ has a solution for all $q \in \mathbb{R}^n$ such that $K(z, w) \neq \emptyset$;*

P-matrices: *matrices M such that all principal minors are positive;*

P_0-matrices: *matrices M such that all principal minors are non-negative;*

Copositive matrices: *matrices M such that for all vectors $x \in \mathbb{R}^n_+$, $x^T M x \geq 0$;*

Strict copositive matrices: *matrices M such that for all vectors $x \in \mathbb{R}^n_+$, $x^T M x > 0$;*

Copositive plus matrices: *matrices M which are copositive and such that for any $y \in \mathbb{R}^n$ such that $y^T M y = 0$, then, $(M^T + M)y = 0$.*

For $M \in P$, any principal minor is strictly positive. Any principal submatrix $M_{\alpha\alpha}, \alpha \in \{1, 2, \ldots, n\}$, is thus invertible. When $M \in P_0$, it is possible that one of the submatrices $M_{\alpha\alpha}$ is singular however.

When matrix M is symmetric, the property $M \in P$ is equivalent to M being positive definite. In general, positive definite matrices form a subset of P matrices, and positive semi-definite matrices form a subset of P_0 [144, 210, 69].

Another distinction of P matrices is that $\mathrm{LCP}(M, q)$ has a *unique* solution for any $q \in \mathbb{R}^n$ for $n \times n$ matrices $M \in P$. Geometrically, this means in fact that the 2^n cones $\mathcal{K}_\alpha, \alpha \subseteq \{1, 2, \ldots, n\}$ are pigeon holed, with one per orthant in $\mathbb{R}^n_+$ as demonstrated in [110].

Theorem 16.2. *For a square, real, $n \times n$ matrix M that is a P matrix, $\mathrm{LCP}(M, q)$ has a unique solution for any real vector $q \in \mathbb{R}^n$.*

16.3 Nonlinear complementarity

The generalization to real nonlinear problems starts with a cone $\mathcal{K} \in \mathbb{R}^n$.

Definition 16.9. *A closed convex cone $\mathcal{K}$ is a point set such that*

1. $\mathcal{K} + \mathcal{K} \subseteq \mathcal{K}$
2. $\lambda \mathcal{K} \subseteq \mathcal{K}$, *for all* $\lambda \geq 0$.

If $K \cap (-K) = \{0\}$, cone $\mathcal{K}$ is said to be pointed.

The standard example is the set of nonnegative linear combinations of a set of vectors $x_i \in \mathbb{R}^n, i = 1, 2, \ldots, m$, where $m < n$, so that $\mathcal{K} = \{x \mid x = \sum_j \alpha_j x_j, \alpha_j \geq 0, j = 1, 2, \ldots, m\}$, which is a pointed cone.

For the case of general Hilbert spaces, point sets for which there is an inner product operation $\langle x, y \rangle \in \mathbb{R}$ for any two points $x, y \in H$, the dual cone can be defined as follows.

Definition 16.10 (Dual cone). *The dual $\mathcal{K}^\star$ of a closed convex cone is the set:*

$$\mathcal{K}^\star = \{y \in \mathbb{R}^n \mid \langle y, x \rangle \geq 0, \textit{for all } x \in \mathcal{K}\}. \tag{16.24}$$

This applies to $\mathbb{R}^n$ with the usual inner product $\langle y, x \rangle = y^T x$.

Definition 16.11. *Given real vector function $f : \mathbb{R}^n \mapsto \mathbb{R}^n$, the nonlinear complementarity problem NCP is that of finding solution vectors z*

$$\mathcal{K}^\star \ni f(z) \quad \perp \quad z \in \mathcal{K}, \tag{16.25}$$

which means that $f(z) \in \mathcal{K}^\star$, $z \in \mathcal{K}$ and $z^T f(z) = 0$.

For the simple case $f(z) = Mz + q$, where M is an $n \times n$ matrix, and $\mathcal{K} = \{z \in \mathbb{R}^n \mid z_i \geq 0, i = 1, 2, \ldots, n\}$, this reduces to Definition 16.1 of LCP.

The Coulomb friction problem defined in Section 10.11.2 is an example of a nonlinear complementarity problem (NCP). Of course, the NCP reduces to the LCP when $f(z) = Mz + q$, where M is an $n \times n$ real matrix and $q \in \mathbb{R}^n$ is a real vector.

Both the NCP and the LCP can be formulated as fixed point problems. This allows to invoke the Brouwer theorem to prove the existence of a solution. To see this, introduce the projection $P_\mathcal{K} : \mathbb{R}^n \mapsto \mathcal{K}$. For the standard complementarity problem for instance, the cone $\mathcal{K}$ is defined as $\mathcal{K} = \{z \in \mathbb{R}^n \mid z_i \geq 0, i = 1, 2, \ldots, n\}$ and $P_K(z) = \mathtt{max}(0, z)$, where the $\mathtt{max}$ function is understood component-wise so that if $w = \mathtt{max}(0, z)$, then, $w_i = \mathtt{max}(0, z_i), i = 1, 2, \ldots, n$.

Then, define the following map $\phi : \mathbb{R}^n \mapsto \mathbb{R}^n$, as:

$$\phi(z) = P_\mathcal{K}(z - f(z)). \tag{16.26}$$

For the standard formulation, this amounts to

$$\phi_{\text{std}}(z) = \mathtt{max}(0, z - Mz - q), \tag{16.27}$$

and a fixed point of $z_\star = \phi_{\text{std}}(z_\star)$ thus satisfies $z_i = 0$ when $(Mz + q)_i > 0$ and $z_i > 0$ when $(Mz + q)_j = 0$ as desired.

Thus, the solvability of a given NCP depends on whether or not the corresponding map ϕ has a fixed point. There are alternative ways to define NCP as a fixed point problem but the one provided above suffices for the purpose of the solvability theory, below. Observe also that $\phi_{\text{std}}(z)$ in (16.27) is *semismooth*, meaning that it has jump discontinuities in its derivatives, much like the absolute value function.

16.4 Solvability theory

Without going into the depth of solvability theory for either LCP or NCP, a few important results are stated here. The most important notion is that of *exceptional families of elements* due to Isac, Kalashnikov, and Bulavsky [135, 57, 134] and also to Obuchowska [217]. Consider a real cone $\mathcal{K} \in \mathbb{R}^n$ with dual $\mathcal{K}^\star$, and a continuous function $f : \mathbb{R}^n \mapsto \mathbb{R}^n$. An exceptional family of element is defined as follows

Definition 16.12. *A set of points $\{z^{(r)}\}_{r>0} \in \mathcal{K}$ is an exceptional family of elements if the following conditions hold:*

1. $z^{(r)} \in \mathcal{K}$ *for all* $r > 0$;
2. $\|z^{(r)}\| \to \infty$ *as* $r \to \infty$;
3. *for every* $r > 0$, *there exists a* $\mu^{(r)} > 0$ *such that* $w^{(r)} = \mu^{(r)} z^{(r)} + f(z^{(r)}) \in \mathcal{K}^\star$ *and* $\langle w^{(r)}, z^{(r)} \rangle = 0$;

where $\langle w, z \rangle = w^T z$ is the inner product.

With this definition, a theorem due to Isac [133] establishes existence as stated in Theorem 16.3, reproduced here without proof.

Theorem 16.3 (Solvability of NCP). *Consider a Hilbert space H, a closed convex cone $\mathcal{K} \in H$ and a completely continuous map $f : H \mapsto H$. Either there exists a solution to NCP$(f, \mathcal{K})$ or f has an exceptional family of elements.*

To see how this plays, assume that function f does have an exceptional family of elements. Since this implies that ${w^{(r)}}^T z^{(r)} = 0$, consider the expansion

$$ {w^{(r)}}^T z^{(r)} = \mu^{(r)} \|z^{(r)}\|^2 + {z^{(r)}}^T f(z^{(r)}) = 0, \text{ and } z^{(r)} \in \mathcal{K}. \tag{16.28} $$

Consider now that $f(z) = Mz + q$ where M is a real, square matrix of size $n \times n$ and $q \in \mathbb{R}^n$, so that

$$ {w^{(r)}}^T z^{(r)} = \mu^{(r)} \|z^{(r)}\|^2 + {z^{(r)}}^T M z^{(r)} + q^T z^{(r)}. \tag{16.29} $$

Thus, if matrix M is positive definite with minimum eigenvalue $\lambda_n(M) > 0$, the following bound holds for (16.29)

$$ {w^{(r)}}^T z^{(r)} \geq {z^{(r)}}^T \left((\mu^{(r)} + \lambda_n(M)) z^{(r)} + q \right), \tag{16.30} $$

and all terms in the last parenthesis are strictly positive whenever

$$ z_i^{(r)} > |q_i| / \lambda_n(M). \tag{16.31} $$

Therefore, for positive definite matrices M, LCP(M, q) is solvable.

16.5 Classical solution methods

Though there is ample literature on solution methods for LCP [69, 144, 210], a few classical methods are now demonstrated here in order to fix some ideas and notation. It is interesting to note also that the LCP literature is deeply rooted in the linear programming literature where *pivoting methods* are in systematic usage. However, such pivoting methods do not properly illustrate the matrix operations involved in the implementation of any given algorithm, especially since these operations are almost fundamentally BLASlevel 2 types, i.e., matrix-vector products, or matrix rank-1 update types. The presentation below relies strictly on explicit submatrices.

Consider $\mathsf{LCP}(M, q)$ of size n and some index set $\alpha \subseteq \{1, 2, \ldots, n\}$. The matrix $M_{\alpha\alpha}$ is called a principal submatrix of M and consists of

$$M_{\alpha\alpha} = (m_{ij}), \quad i, j \in \alpha \subseteq \{1, 2, \ldots, n\}. \tag{16.32}$$

From Definition 16.1 for $\mathsf{LCP}(M, q)$, the solution consists of an index set α and the solution of the linear system

$$M_{\alpha\alpha} z_\alpha = -q_\alpha, \tag{16.33}$$

from which the rest of the solution is recovered using complementarity conditions, namely

$$z_\beta = 0, w_\alpha = 0, w_\beta = q_\beta + M_{\beta\alpha} z_\alpha. \tag{16.34}$$

A number of LCP solution methods can be constructed by sequentially considering index sets $\alpha^{(\nu)}, \nu = 1, 2, \ldots$, and solving linear systems involving the corresponding principal submatrices of M.

We consider three of these in what follows to show the variety of ideas which can be applied.

16.5.1 Murty's principal pivot method for LCP

The absolutely simplest solution method is due to Katta Murty [209] and is described also in [210] and [69]. The idea is this. Given $\mathsf{LCP}(M, q)$, pick a candidate index set $\alpha \subseteq \{1, 2, \ldots, n\}$. Solve the principal subsystem

$$M_{\alpha\alpha} z_\alpha = -q_\alpha, \tag{16.35}$$

compute $w_\beta = q_\beta + M_{\beta\alpha} z_\alpha$, and set $w_\alpha = 0, z_\beta = 0$, and $s = w + z$. Vector $s \in \mathbb{R}^n$ is called the *complementarity point*. Now, if we are lucky, vector s is non-negative and we therefore have a solution. Otherwise, there is at least one index j such that $s_j < 0$. There are several choices here and the original idea due to Y. Bard (as quoted in [210]) is to pick the *most infeasible index* and either remove it from set α if $s_j = z_j$ or add it to set α if $s_j = w_j$. However, this can cycle and therefore, the simplest method that actually works is due to K. G. Murty and described in Algorithm 16.5.1.

This algorithm works without cycling for P matrices. The reason is an inductive argument which we now describe.

Algorithm 16.5.1 Murty's principal pivoting algorithm for LCP.

Given $n \times n$ matrix M and n-dimensional vector q
initialize $\alpha = \emptyset$, $\beta = \{1, 2, \ldots, n\} \setminus \alpha$
repeat
  Solve $M_{\alpha\alpha} z_\alpha = -q_\alpha$
  Set $z_\beta = 0$
  Compute $s = z + q + Mz$
  Find $r = \max\{i \mid s_i < 0\}$
  if $r \in \alpha$ **then**
    $\alpha \leftarrow \alpha \setminus \{r\}$
    $\beta \leftarrow \beta \cup \{r\}$
  else
    $\beta \leftarrow \beta \setminus \{r\}$
    $\alpha \leftarrow \alpha \cup \{r\}$
  end if
until $r = \emptyset$

Theorem 16.4. *Given* $\mathrm{LCP}(M, q)$ *with an* $n \times n$ P *matrix, Murty's max index algorithm 16.5.1 computes the unique solution* z *in finite number of steps* $k \leq 2^n$.

Proof. The proof is by induction. Obviously, the algorithm works for $n = 1$ in which case there are only two choices. Assume it works for LCP of size m up to $m = n - 1$. Now, variable n will only be considered once $\mathrm{LCP}(M_{\delta\delta}, q_\delta), \delta = \{1, 2, \ldots, n-1\}$ is solved, since variables $j < n$ are always considered first. This leads to two possibilities.

Case I. Index set $\alpha \subseteq \{1, 2, \ldots, n-1\}$ solves $\mathrm{LCP}(M_{\delta\delta}, q_\delta)$ and $q_n + M_{n\alpha} z_\alpha \geq 0$. The solution of $\mathrm{LCP}(M, q)$ can be constructed directly from z_α. Since matrix $M \in P$, this is the unique solution.

Case II. Index set $\alpha \subseteq \{1, 2, \ldots, n-1\}$ solves $\mathrm{LCP}(M_{\delta\delta}, q_\delta)$ but $q_n + M_{n\alpha} z_\alpha < 0$. With the maximum index rule, this means that index n must be added to the set α. Note also that we cannot encounter Case II if case I completes which means that the index set of the solution must contain index n. Performing a Schur complement on variable z_n, i.e., solving for z_n as a function of z_δ, we find the reduced problem to be $\mathrm{LCP}(\tilde{M}, \tilde{q})$ of dimension $n - 1$ with

$$\begin{aligned} \tilde{M} &= M_{\delta\delta} - m_{nn}^{-1} M_{\delta n} M_{n\delta}, \\ \tilde{q} &= q_\delta - m_{nn}^{-1} M_{\delta n}, \\ \delta &= \{1, 2, \ldots, n-1\}. \end{aligned} \tag{16.36}$$

Now, if M is a P matrix, then, so is the Schur complement $\tilde{M}$ and therefore, $\mathrm{LCP}(\tilde{M}, \tilde{q})$ is solvable by induction.

Note also that all principal subproblems of Algorithm 16.5.1 are solvable since $M \in P$ implies that $\det(M_{\alpha\alpha}) > 0$, hence, $M_{\alpha\alpha}$ is non-singular.

Assuming that solving LCP of size $n - 1$ takes at most 2^{n-1} operations, at most $2 \times 2^{n-1} = 2^n$ operations are needed to solve LCP of size n.

The non-cycling assumption therefore survives induction as well. □

This Murty max index rule has another nice property that it can be *warm started*, i.e., we can start from a non-empty index set $\alpha \neq \emptyset$ which is, hopefully, near the solution in some sense.

Note also that the maximum index rule can be changed to a minimum index rule or a kth index rule. These variations correspond to permutations of the original matrix M and vector q which do not affect the P property or the induction step.

There is an extension to this algorithm which is a *block pivot* rule in which all variables j such that $s_j < 0$ are switched to their respective complementary sets. This essentially corresponds to an undamped Newton method as we show in Section 16.7.1.

As a final remark, note that Murty's maximum index rule does not require matrix M to be positive definite or symmetric, but only requires property P.

16.5.2 Murty's principal pivot method for MLCP

The simplest way to reformulate Murty's principal pivot rule is to consider the MLCP formulation of (16.7) and apply the maximum index rule on that. Of course, no one should attack the augmented matrix in (16.7) with a factorization algorithm. The linear problems to solve still only involve the principal submatrix $M_{\alpha\alpha}$. The trick here is to split the index set into two main groups as before, namely, α, β with $\alpha \cap \beta = \emptyset$ and $\alpha \cup \beta = \{1, 2, \ldots, n\}$. However now, the set β is refined into two parts, namely, δ contains the variables which have reached the upper bound, and γ contains those which are at their lower bound.

Define two complementarity points $s^{(+)}$ and $s^{(-)}$ as

$$\begin{aligned} s^{(+)} &= z - l + w_+ \\ s^{(-)} &= u - z + w_-. \end{aligned} \tag{16.37}$$

All components of $s^{(\pm)}$ will be positive when the problem is solved. The maximum index rule here is to locate the largest index i such that either $s^{(+)} < 0$ or $s^{(-)} < 0$. The pivoting is done as before, namely, if index i is free so $i \in \alpha$, then, if $s^{(+)} < 0$, index i is transferred to δ, and if $s^{(-)} < 0$, then, index i is transferred to γ. Finally, if index i is in either δ or γ, it is transferred to α. All that really changes is the right hand side vector in the main solve operation which contains contributions from the active lower and upper bounds.

Algorithm 16.5.2 Murty's principal pivoting algorithm for MLCP.

Given $n \times n$ matrix M, n-dimensional vector q and n-dimensional vectors of lower and upper bounds, l and u, respectively.
Initialize $\alpha = \emptyset$, $\beta = \{1, 2, \ldots, n\} \setminus \alpha \quad \delta = \gamma = \emptyset$;
repeat
  Solve $M_{\alpha\alpha} z_\alpha = -q_\alpha - M_{\alpha\delta} l_\delta - M_{\alpha\gamma} u_\gamma$;
  Set $z_\delta = l_\delta$;
  Set $z_\gamma = u_\delta$;
  Compute $s^{(+)} = (z - l) + (q + Mz)$;
  Compute $s^{(-)} = (u - l) - (q + Mz)$;
  Find $r = \max\{i \mid s_i^{(+)} < 0 \quad \text{or} \quad s_i^{(-)} < 0\}$;
  if $r \in \alpha$ and $s_i^{(+)} < 0$ **then**
    $\alpha \leftarrow \alpha \setminus \{r\}$;
    $\delta \leftarrow \delta \cup \{r\}$;
  else if $r \in \alpha$ and $s_i^{(-)} < 0$; **then**
    $\alpha \leftarrow \alpha \setminus \{r\}$;
    $\gamma \leftarrow \gamma \cup \{r\}$;
  else
    if $i \in \delta$ **then**
      $\delta \leftarrow \delta \setminus \{r\}$;
    else $\gamma \leftarrow \gamma \setminus \{r\}$;
    end if
    $\alpha \leftarrow \alpha \cup \{r\}$;
  end if
until $r = \emptyset$

16.5.3 The Keller algorithm

Edward L. Keller [157] introduced an interesting and efficient algorithm for solving LCPs associated with P matrices. In fact, Keller specifically considered KKT matrices of quadratic programming (QP) problems which have the form

$$\begin{bmatrix} Q & -G^T \\ G & 0 \end{bmatrix}, \tag{16.38}$$

where Q is $n_1 \times n_1$, symmetric, and positive semi-definite, G is a rectangular $n_2 \times n_1$ matrix of full row rank (thus with $n_2 \leq n_1$), and 0 is an $n_2 \times n_2$ zero block. Such matrices belong to the class P_0 and in general, special care is required to avoid steps involving zero diagonal entries. This is explained in detail in [157] but we avoid these subtleties here and stick to P matrices since for regularized problems, there is never a 0 block in the lower right corner.

The Keller principal pivot algorithm proceeds along a feasible vector z where at each iteration, $z_\alpha \geq 0$ and $z_\beta = 0$. Likewise, $w_\alpha = 0$ but there is no guarantee that w is feasible. Find the index $s = \arg\min_i(w_i)$. If $w_s < 0$, pivot on s but keeping z feasible. Otherwise w is feasible and the solution is found.

In contrast to what was done in the Murty's maximum index rule, care will be taken to chose indices $r \in \{1, 2, \ldots, n\}$ to add or remove from the active set α so as to keep z in the feasible set, i.e., $z_\alpha \geq 0$.

This leads to two cases. The first one is when a single pivot step is taken which adds the variable z_s to the active set z_α and maintains feasibility, and another case where one or several variables z_r have to be removed from the active set in order to maintain feasibility before the variable z_s can be added to the active set.

We consider the simplest case first. Assume for now that z_α is the active basic vector for some step ν. This means that it solves the following system

$$\begin{bmatrix} M_{\alpha\alpha} & M_{\alpha\beta} \\ M_{\beta\alpha} & M_{\beta\beta} \end{bmatrix} \begin{bmatrix} z_\alpha \\ 0 \end{bmatrix} + \begin{bmatrix} q_\alpha \\ q_\beta \end{bmatrix} = \begin{bmatrix} 0 \\ w_\beta \end{bmatrix} \tag{16.39}$$
$$z_\alpha \geq 0,$$

and so $z_\alpha = -M_{\alpha\alpha}^{-1} q_\alpha$. Assume now that variable w_s is the pivot candidate so that $w_s < 0$, and $w_s \leq w_j, j \in \beta$. The rule here is to add variable z_s to the active set to remedy the problem. After reorganizing the variables, the system becomes

$$\begin{bmatrix} M_{\alpha\alpha} & M_{\alpha s} & M_{\alpha\bar{\beta}} \\ M_{s\alpha} & M_{ss} & M_{s\bar{\beta}} \\ M_{\bar{\beta}\alpha} & M_{\bar{\beta}s} & M_{\bar{\beta}\bar{\beta}} \end{bmatrix} \begin{bmatrix} \bar{z}_\alpha \\ \bar{z}_s \\ 0 \end{bmatrix} + \begin{bmatrix} q_\alpha \\ q_s \\ q_{\bar{\beta}} \end{bmatrix} = \begin{bmatrix} 0 \\ \bar{w}_s \\ \bar{w}_{\bar{\beta}} \end{bmatrix}. \tag{16.40}$$

Write the solution $\bar{z}$ of this system as an update of the form $\bar{z}_\alpha = z_\alpha + \theta v_\alpha$. This is done by explicitly writing out the first block equation as follows

$$\begin{aligned} M_{\alpha\alpha}\bar{z}_\alpha + q_\alpha + M_{\alpha s}\bar{z}_s &= 0, \\ \bar{z}_\alpha + M_{\alpha\alpha}^{-1} q_\alpha + M_{\alpha\alpha}^{-1} M_{\alpha s}\bar{z}_s &= 0. \end{aligned} \tag{16.41}$$

Accounting for $M_{\alpha\alpha}z_\alpha + q_\alpha = 0$ then yields

$$\begin{aligned}\bar{z}_\alpha &= z_\alpha - \bar{z}_s M_{\alpha\alpha}^{-1} M_{\alpha s} \\ &= z_\alpha + \theta v_\alpha,\end{aligned} \tag{16.42}$$

which is the desired format by setting $\theta = \bar{z}_s$ and $M_{\alpha\alpha}v_\alpha = -M_{\alpha s}$. Likewise, if we look at the value of $\bar{w}_s$ which we intend to drive to 0, we have

$$\begin{aligned}\rho_s &= M_{s\alpha}v_\alpha + M_{ss} = M_{s\alpha}M_{\alpha\alpha}^{-1}M_{\alpha s} + M_{ss}, \\ \bar{w}_s &= w_s + \theta\rho_s, \\ \bar{w}_{\bar{\beta}} &= w_{\bar{\beta}} + \theta(M_{\bar{\beta}\alpha}v_\alpha + M_{\bar{\beta}s}).\end{aligned} \tag{16.43}$$

Now, since M is a P matrix, we have $\rho_s > 0$. Choosing $\theta_1 = -w_s/\rho_s$ makes $\bar{w}_s = 0$. However, in order to keep $\bar{z}$ feasible, we must set θ to be less than

$$\theta_2 = \min_i \left\{ \frac{-z_i}{v_i} \,\middle|\, i \in \alpha \text{ and } v_i < 0 \right\} = \frac{-z_r}{v_r}. \tag{16.44}$$

Therefore, if $\theta_1 < \theta_2$, we add z_s to the active set and we are done. However, if $\theta_1 > \theta_2$, we need to first remove z_r from the active set before we can drive w_s towards zero. Let us write $\bar{\alpha} = \alpha \setminus r$, $\bar{\bar{\beta}} = \bar{\beta} \cup r$, and $\bar{\bar{z}}_{\bar{\alpha}}$ for updated variables. Note first that we can write

$$\begin{aligned}\bar{z}_\alpha &= \begin{bmatrix}\bar{z}_{\bar{\alpha}} \\ 0\end{bmatrix}, \quad M_{\alpha\alpha} = \begin{bmatrix}M_{\bar{\alpha}\bar{\alpha}} & M_{\bar{\alpha}r} \\ M_{r\bar{\alpha}} & M_{rr}\end{bmatrix}, \text{ and} \\ M_{\alpha\alpha}\bar{z}_\alpha &= \begin{bmatrix}M_{\bar{\alpha}\bar{\alpha}}\bar{z}_{\bar{\alpha}} \\ M_{r\bar{\alpha}}\bar{z}_{\bar{\alpha}}\end{bmatrix} = \begin{bmatrix}-q_{\bar{\alpha}} \\ -q_r\end{bmatrix} + \theta\begin{bmatrix}-M_{\bar{\alpha}s} \\ -M_{rs}\end{bmatrix},\end{aligned} \tag{16.45}$$

from which we can extract the following identity

$$M_{\bar{\alpha}\bar{\alpha}}\bar{z}_{\bar{\alpha}} = -q_{\bar{\alpha}} - \theta M_{\bar{\alpha}s}. \tag{16.46}$$

The new system we are trying to solve is now

$$\begin{bmatrix}M_{\bar{\alpha}\bar{\alpha}} & M_{\bar{\alpha}s} & M_{\bar{\alpha}\bar{\bar{\beta}}} \\ M_{s\bar{\alpha}} & M_{ss} & M_{s\bar{\bar{\beta}}} \\ M_{\bar{\bar{\beta}}\bar{\alpha}} & M_{\bar{\bar{\beta}}s} & M_{\bar{\bar{\beta}}\bar{\bar{\beta}}}\end{bmatrix}\begin{bmatrix}\bar{\bar{z}}_\alpha \\ \theta + \bar{z}_s \\ 0\end{bmatrix} + \begin{bmatrix}q_{\bar{\alpha}} \\ q_s \\ q_{\bar{\bar{\beta}}}\end{bmatrix} = \begin{bmatrix}0 \\ \bar{\bar{w}}_s \\ \bar{\bar{w}}_{\bar{\bar{\beta}}}\end{bmatrix}, \tag{16.47}$$

from which we extract the first block equations

$$\begin{aligned}M_{\bar{\alpha}\bar{\alpha}}\bar{\bar{z}}_{\bar{\alpha}} &= M_{\bar{\alpha}\bar{\alpha}}\bar{z}_{\bar{\alpha}} - \theta M_{\bar{\alpha}s}, \text{ which means that} \\ \bar{\bar{z}}_{\bar{\alpha}} &= \bar{z}_{\bar{\alpha}} + \theta\bar{v}_{\bar{\alpha}}, \text{ where} \\ M_{\bar{\alpha}\bar{\alpha}}\bar{v}_{\bar{\alpha}} &= -M_{\bar{\alpha}s}.\end{aligned} \tag{16.48}$$

This demonstrates how the same updating rules can be applied in either case. The algorithm therefore proceeds as illustrated below in Algorithm 16.5.3. The proof is given in [144] that this process cannot terminate on a ray whenever M is a P matrix. The same is true for $M \in P_0$ but in this case, we need to include 2×2 pivots.

Algorithm 16.5.3 Keller's Algorithm for solving LCPs with P matrices.

Initialize $\alpha = \emptyset, \beta = \{1, 2, \ldots, n\}$
while $\exists\, i$ s.t. $w_i < 0$ **do**
  $s = \arg\min_i \{w_i \mid \text{ s.t. } w_i < 0\}$
  repeat
    Solve: $M_{\alpha\alpha} v_\alpha = -M_{\alpha s}$
    $\rho_s \leftarrow M_{ss} + M_{s\alpha} v_\alpha$
    if $\rho_s > 0$ **then**
      $\theta_1 \leftarrow \frac{-w_s}{\rho_s}$
    else
      $\theta_1 \leftarrow \infty$
    end if
    if $\exists\, v_i < 0$ **then**
      $\theta_2 \leftarrow \min_i \{\frac{-z_i}{v_i} \mid v_i < 0\}$
      $r \leftarrow \arg\min_i \{\frac{-z_i}{v_i} \mid v_i < 0\}$
    else
      $\theta_2 \leftarrow \infty$
    end if
    $\theta = \min(\theta_1, \theta_2)$
    if $\theta = \infty$ **then**
      Termination on a ray: no solution. End.
    else
      $z_\alpha \leftarrow z_\alpha + \theta v_\alpha$
      $w_s \leftarrow w_s + \theta \rho_s$
      $w_\beta \leftarrow w_\beta + \theta(M_{\beta\alpha} v_\alpha + M_{\beta s})$
    end if
    if $\theta = \theta_1$ **then**
      $w_s = 0$, $\alpha \leftarrow \alpha \cup s$, $\beta \leftarrow \beta \setminus s$
    else if $\theta = \theta_2$ **then**
      $z_r = 0$, $\alpha \leftarrow \alpha \setminus r$, $\beta \leftarrow \beta \cup r$
    end if
  until $\theta = \theta_1$
end while

16.5.4 The Keller algorithm for MLCP

Here, we consider MLCPs as per Definition 16.2 so

$$\begin{aligned} Mz + q &= w_+ - w_- \\ 0 \le w_+ \perp z - l &\ge 0, \\ 0 \le w_- \perp u - z &\ge 0, \end{aligned} \tag{16.6'}$$

where the bound vectors l, u are extended reals so that any or all $l_i \le u_i$, might be infinite.

The original Keller algorithm proceeds along feasible z vectors, and pivots on the most infeasible component of the w. As the infeasibility is reduced to zero, the step length is chosen to maintain feasibility on the z vector. This description can be used directly to adapt the algorithm to process MLCPs. The adaptation consists of considering all the new cases introduced by the upper bounds.

The index set is split into four parts $F, T = T_l \bigcup T_u$ such that $F \bigcap T = \emptyset$, $F \bigcup T = \{1, 2, \ldots, n\}$, $T_l \bigcap T_u = \emptyset$, $T_l \bigcup T_u = T$. The set F corresponds to the free variables, whilst the set T corresponds to the tight variables which is split into the tight variables at the lower bound T_l, and the tight variables at the upper bound T_u.

Picking the most infeasible component of w, we compute $s_l = \arg\min_{i \in T_l}\{w_i\}$ and $s_u = \arg\max_{i \in T_u}\{w_i\}$ and we define $s_l = -1$ and $s_u = -1$ if $T_l = \emptyset$ or $T_u = \emptyset$, respectively. Then, we set $s = \arg\min\{w(s_l), -w(s_u)\}$ and $p = 1$ if $s = s_l$, or $p = -1$ if $s = s_u$. The search direction and free step length are set as before

$$\begin{aligned} M_{FF} v_F &= -M_{Fs}, \qquad \text{and} \\ \rho_s &= M_{ss} + M_{sF} v_F. \end{aligned} \tag{16.49}$$

The test to accept the step length now splits into two parts, based on whether $p = \pm 1$. We need to test both for upper and lower bounds and therefore, we compute three potential blocks as follows:

$$\begin{aligned} \text{whenever} \quad p &= 1, \qquad \text{then} \\ \theta_0 &= -w(s)/\rho_s, \\ \theta_1 &= u(s) - z(s), \\ \theta_2 &= \min_{\{i \in F | v_i < 0\}} \{(l_i - z_i)/v_i\}, \\ r_2 &= \arg \min_{\{i \in F | v_i < 0\}} \{(l_i - z_i)/v_i\}, \\ \theta_3 &= \min_{\{i \in F | v_i > 0\}} \{(u_i - z_i)/v_i\}, \\ r_3 &= \arg \min_{\{i \in F | v_i > 0\}} \{(u_i - z_i)/v_i\}, \\ t &= \min\{\theta_0, \theta_1, \theta_2, \theta_3\}. \end{aligned} \tag{16.50}$$

In addition to this,

$$\begin{aligned}
\text{whenever} \quad p &= -1, \qquad \text{then} \\
\theta_0 &= -w(s)/\rho_s, \\
\theta_1 &= l(s) - z(s), \\
\theta_2 &= \max_{\{i \in F | v_i > 0\}} \{(l_i - z_i)/v_i\}, \\
r_2 &= \arg \max_{\{i \in F | v_i > 0\}} \{(l_i - z_i)/v_i\}, \\
\theta_3 &= \max_{\{i \in F | v_i < 0\}} \{(u_i - z_i)/v_i\}, \\
r_3 &= \arg \max_{\{i \in F | v_i < 0\}} \{(u_i - z_i)/v_i\}, \\
t &= \max\{\theta_0, \theta_1, \theta_2, \theta_3\}.
\end{aligned} \tag{16.51}$$

Taking the pivot step, there are now four possibilities:

Driving variable not blocked: infeasibility r reduced to 0;

Driving variable blocked at a bound;

Blocking variable blocked at a lower bound;

Blocking variable blocked at an upper bound.

These correspond to using step lengths $\theta_0, \theta_1, \theta_2$ or θ_3 respectively. In the first two cases, we go back to picking the most infeasible w variable. In the last two, we pivot on r_2 or r_3 respectively.

The algorithm is summarized below in Algorithm 16.5.4, which contains a slight correction to what is shown in [147].

Notice how only the $\min$ function is used for both the $p = 1$ and $p = -1$ case. Noting the clever trick, $\max(x_i) = -\min(-x_i)$, all that is needed is to use $p \cdot \min(px_i)$.

Algorithm 16.5.4 Keller's Algorithm for MLCP

Initialize: $\alpha = \{i \mid l_i = -\infty \quad \text{and} \quad u_i = \infty\}$;
$\beta_u = \{i \mid l_i = -\infty \text{ and } u_i \neq \infty\}$; $\beta_l = \{i \mid l_i \neq -\infty\}$;
$z_{\beta_l} = l_{\beta_l}$; $z_{\beta_u} = l_{\beta_u}$;
Solve: $M_{\alpha\alpha} z_\alpha = -q_\alpha$; $w = Mz + q$;
while $\exists\, i \in \beta_l$ s.t. $w_i < 0$ or $\exists\, i \in \beta_u$ s.t. $w_i > 0$ **do**
  $s_l \leftarrow \arg\min_{i \in \beta_l}\{w_i \mid \text{ s.t. } w_i < 0\}$;
  $s_u \leftarrow \arg\max_{i \in \beta_u}\{w_i \mid \text{ s.t. } w_i > 0\}$;
  $s \leftarrow \arg\min\{w_{s_l}, -w_{s_u}\}$; $p \leftarrow -\operatorname{sgn}(w_s)$;
  repeat
    Solve: $M_{\alpha\alpha} v_\alpha = -M_{\alpha s}$; for v_α
    $\rho_s \leftarrow M_{ss} + M_{s\alpha} v_\alpha$;
    Reset: $\theta_0 = \theta_1 = \theta_2 = \theta_3 = p \cdot \infty$;
    if $\rho_s > 0$ **then**
      $\theta_0 \leftarrow \frac{-w_s}{\rho_s}$;
    end if
    if $p = 1$ **then**
      $\theta_1 = u_s - z_s$;
    else
      $\theta_1 = l_s - z_s$;
    end if
    $\theta_2 \leftarrow \min\{p\frac{l_i - z_i}{v_i} \mid p \cdot v_i < 0\}$; $r_2 \leftarrow \arg\min\{p\frac{l_i - z_i}{v_i} \mid p \cdot v_i < 0\}$;
    $\theta_3 \leftarrow \min\{p\frac{u_i - z_i}{v_i} \mid p \cdot v_i > 0\}$; $r_3 \leftarrow \arg\min\{p\frac{u_i - z_i}{v_i} \mid p \cdot v_i > 0\}$;
    $\theta = p \cdot \min\{p\theta_0, p\theta_1, p\theta_2, p\theta_3\}$;
    if $|\theta| = \infty$ **then**
      Termination on a ray: no solution. End.
    else
      $z_\alpha \leftarrow z_\alpha + \theta v_\alpha$; $w_s \leftarrow w_s + \theta\rho_s$; $w_\beta \leftarrow w_\beta + \theta(M_{\beta\alpha} v_\alpha + M_{\beta s})$;
    end if
    if $\theta = \theta_0$ **then**
      $w_s = 0$; $\alpha \leftarrow \alpha \cup \{s\}$;
      $\beta \leftarrow \beta \setminus \{s\}$; $\beta_l \leftarrow \beta_l \setminus \{s\}$; $\beta_u \leftarrow \beta_u \setminus \{s\}$;
    else if $\theta = \theta_1$ **then**
      if p=1 **then**
        $z_s = u_s$; $\alpha \leftarrow \alpha \setminus \{s\}$; $\beta_u \leftarrow \beta_u \cup \{s\}$; $\beta \leftarrow \beta \cup \{s\}$;
      else
        $z_s = l_s$; $\alpha \leftarrow \alpha \setminus \{s\}$; $\beta_l \leftarrow \beta_l \cup \{s\}$; $\beta \leftarrow \beta \cup \{s\}$;
      end if
    else if $\theta = \theta_2$ **then**
      $z_r = l_r$; $\alpha \leftarrow \alpha \setminus \{r\}$; $\beta \leftarrow \beta \cup \{r\}$; $\beta_l \leftarrow \beta_l \cup \{r\}$;
    else if $\theta = \theta_3$ **then**
      $z_r = u_r$; $\alpha \leftarrow \alpha \setminus \{r\}$; $\beta \leftarrow \beta \cup \{r\}$; $\beta_u \leftarrow \beta_u \cup \{r\}$;
    end if
  until $\theta = \theta_0$;
end while

16.5.5 The Cottle-Dantzig algorithm for LCP

The Cottle-Dantzig algorithm, known as *the* principal pivot method, works similarly to the Keller algorithm but with one important difference: it only considers the first k variables and proceeds by incrementing k to eventually cover the full set, $\{1, 2, \ldots, n\}$. This might sound like an improvement but in fact, the performance for this is inferior to that of the Keller algorithm in most cases. Much like the Keller algorithm, this algorithm can find the unique solution of $\mathsf{LCP}(M, q)$ for $M \in P$, and this includes positive definite matrices. The Cottle-Dantzig can also process problems such that M is positive semi-definite and either compute a solution to $\mathsf{LCP}(M, q)$, or determine that the problem is infeasible [210].

The Cottle-Dantzig algorithm has two main loops. The outer loop considers $w_k, k = 1, 2, \ldots, n$ in sequence, proceeding to $k + 1$ only when a solution to the subproblem $\mathsf{LCP}(M_{\gamma\gamma}, q_\gamma), \gamma = \{1, 2, \ldots, k\}$ has been found. As observed previously in Section 16.5.1, if the computed value $(w_{\gamma_k}^T, w_{k+1})^T$ from $(z_{\gamma_k}^T, 0)^T$, where z_{γ_k} is the solution of $\mathsf{LCP}(M_{\gamma_k\gamma_k}, q_{\gamma_k})$, has $w_{k+1} < 0$, then, it is certain that z_{k+1} must be in the active set for subproblem $k+1$. The algorithm therefore checks that $w_{k+1} \geq 0$ and if not, it tries to add z_{k+1} to the active set without making any other variable negative. If that works, we are done but otherwise, we must either drop some variables from the active set or add new ones. This is done repeatedly as long as we cannot safely add z_{k+1} to the active set, at which point the $k + 1$ cycle is finished. Details are shown below in the pseudo code Algorithm 16.5.5.

Observe here that until the end, the union of the sets $\alpha \cup \beta \neq \{1, \ldots, n\}$ and we only work with the k variables already visited. This is different from the Keller algorithm.

The analysis to extend the Cottle-Dantzig to solve MLCP is very similar to that provided in Section 16.5.4 and is not worth repeating here. The result is not particularly interesting though in view of the numerical experiments of Section 17.6 and a detailed listing is therefore left out.

Algorithm 16.5.5 Cottle Dantzig Principal Pivot Method for LCPs with symmetric positive definite matrices.

Initialize $\alpha = \emptyset, \beta = \emptyset, w = q, z = 0$
for $s = 1 \ldots n$ **do**
  if $w_s > 0$ **then**
    $\beta = \beta \cup \{s\}$
  else
    repeat
      Solve : $M_{\alpha\alpha} v_\alpha = -M_{\alpha s}$
      $\rho_s \leftarrow M_{ss} + M_{s\alpha} v_\alpha$
      $u_\beta \leftarrow M_{\beta\alpha} v_\alpha + M_{\beta s}$
      $\theta_0 \leftarrow -w_s / \rho_s$
      $\theta_1 \leftarrow \min_{i \in \alpha} \{-z_i / v_i \mid v_i < 0\}$
      $r_1 \leftarrow \arg\min_{i \in \alpha} \{-z_i / v_i \mid v_i < 0\}$
      $\theta_2 \leftarrow \min_{i \in \beta} \{-w_i / u_i \mid u_i < 0\}$
      $r_2 \leftarrow \arg\min_{i \in \alpha} \{-w_i / v_i \mid u_i < 0\}$
      $\theta \leftarrow \min\{\theta_0, \theta_1, \theta_2\}$
      $z_\alpha \leftarrow z_\alpha + \theta v_\alpha; \quad z_s \leftarrow z_s + \theta;$
      $w_s \leftarrow w_s + \theta \rho_s, w_\beta = w_\beta + \theta u_\beta$
      if $\theta = \theta_0$ **then**
        $\alpha \leftarrow \alpha \cup \{s\}; \quad \beta \leftarrow \beta \setminus \{s\}$
      else if $\theta = \theta_1$ **then**
        $\alpha \leftarrow \alpha \setminus \{r_1\}; \quad \beta \leftarrow \beta \cup \{r_1\}$
      else
        $\alpha \leftarrow \alpha \cup \{r_2\}; \quad \beta \leftarrow \beta \setminus \{r_2\}$
      end if
    until $\theta = \theta_0$
  end if
end for

16.5.6 The Lemke algorithm

The algorithm of Carlton E. Lemke[179] differs significantly from the previous ones in that it does *not* proceed along principal pivots. In addition, it maintains an almost complementarity condition between vectors z and w at each stage. This is achieved by introducing an artificial variable in the system and the problem is solved when that variable can be removed while keeping feasibility of the remaining z and w vectors.

At any given stage in the algorithm, the following relation holds

$$\begin{aligned} w &= Mz + q + z_0 p \\ 0 \leq z \ &\perp \ w \geq 0,\ z_0 \geq 0, \end{aligned} \tag{16.52}$$

where $z_0 > 0$ is an artificial variable, and the covering vector $p \in \mathbb{R}^n$ is non-negative, and usually taken as $p_i = 1$.

The fact that this is not a principal pivot algorithm actually means that variable z_r is responsible to drive variable w_s towards $w_s = 0$, with $s \neq r$ as we explain shortly.

Now, to start the algorithm, we choose z_0 so that variables z, w, z_0 are all non-negative, feasible, and complementary. The choice is simply

$$z_0 = \max\{-q_r/p_r\},\ z = 0,\ w = q + z_0 p \geq 0,\ w^T z = 0. \tag{16.53}$$

Of course, if $q_i \geq 0$, the LCP admits the trivial solution $z = 0, w = q$, and we are done.

Assuming that $q_r < 0$ for some index r, the next step is to introduce variable z_r in the active set. Since w_r was already forced to zero in the previous stage, introducing variable z_r will make some *other* variable vanish, namely, either w_s for some $s \neq r$, or the artificial z_0 itself, in which case the problem is solved. The idea for the algorithm is to choose the complement of the variable which is driven to zero as the new variable to enter in or drop out of the active set. At some point, assuming there is no cycling, either the variable z_0 will drop out of the active set or some impossible condition will be met meaning that the problem cannot be solved. This is analyzed further below.

To construct the rest of the algorithm, we need to understand the changes in the variables of (16.52) when a variable z_r is added or removed from the active set.

At any given stage ν in the algorithm, let the active variables consist of the set $\beta \subseteq \{1, 2, \ldots, n\}$ as well as the artificial variable z_0. Introduce the three following sets

$$\begin{aligned} &\alpha \subseteq \{1, 2, \ldots, n\}, \text{ with } z_\alpha \geq 0, w_\alpha = 0; \\ &r \in \{1, 2, \ldots, n\}, \text{ the current pivoting variable} \\ &\beta \subseteq \{1, 2, \ldots, n\},\ \alpha \cup \beta \cup \{r\} = \{1, 2, \ldots, n\},\ w_\beta \geq 0, \text{ and } z_\beta = 0. \end{aligned} \tag{16.54}$$

The pivoting variable might be either z_r when as it is added to the active set, or the slack variable w_r as variable z_r is removed from the active set. Both cases are analyzed in what follows.

Consider first that we are introducing variable z_r into the active set. Taking account of the fact that variables z, z_0 and w are kept feasible with respect to (16.52) at all steps, the linear problem to solve is now

$$\begin{bmatrix} p_\alpha & M_{\alpha\alpha} & M_{\alpha r} \\ p_r & M_{r\alpha} & M_{rr} \\ p_\beta & M_{\beta\alpha} & M_{\beta r} \end{bmatrix} \begin{bmatrix} z_0 + \lambda v_0 \\ z_\alpha + \lambda v_\alpha \\ \lambda \end{bmatrix} + \begin{bmatrix} q_\alpha \\ q_r \\ q_\beta \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ w_\beta + \lambda v_\beta \end{bmatrix}. \tag{16.55}$$

These equations are now spelled out as

$$\begin{aligned} (M_{\alpha\alpha} z_\alpha + z_0 p_\alpha + q_\alpha) + \lambda (M_{\alpha\alpha} v_\alpha + v_0 p_\alpha + M_{\alpha r}) &= 0, \\ (M_{r\alpha} z_\alpha + z_0 p_r + q_r) + \lambda (M_{r\alpha} v_\alpha + v_0 p_r + M_{rr}) &= 0, \\ \big(M_{\beta\alpha} z_\alpha + z_0 p_\beta + q_\beta\big) + \lambda \big(M_{\beta\alpha} v_\alpha + v_0 p_\beta + M_{\beta r}\big) &= w_\beta + \lambda v_\beta. \end{aligned} \tag{16.56}$$

Now, during the previous stage, the terms in parenthesis of the first two equations in (16.56) vanished, and for the last equation, we had: $M_{\beta\alpha} z_\alpha + z_0 p_\beta + q_\beta = w_\beta$. Therefore, the vector v must satisfy

$$\begin{bmatrix} p_\alpha & M_{\alpha\alpha} \\ p_r & \mathcal{M}_{r\alpha} \end{bmatrix} \begin{bmatrix} v_0 \\ v_\alpha \end{bmatrix} = - \begin{bmatrix} M_{\alpha r} \\ M_{rr} \end{bmatrix}, \qquad v_\beta = M_{\beta\alpha} v_\alpha + v_0 p_\beta + M_{\beta r}. \tag{16.57}$$

This leads to the updates

$$\begin{aligned} z_\alpha &\leftarrow z_\alpha + \lambda v_\alpha, \\ z_r &\leftarrow \lambda, \text{ and} \\ w_\beta &\leftarrow w_\beta + \lambda v_\beta. \end{aligned} \tag{16.58}$$

Observe also that vector v defines a solution to the homogeneous problem $w = Mz + z_0 p$ using

$$\begin{aligned} z_\alpha = v_\alpha, z_r = 1, z_\beta = 0, z_0 = v_0, \text{ and} \\ w_\alpha = 0, w_r = 0, w_\beta = v_\beta. \end{aligned} \tag{16.59}$$

We now choose a value of $\lambda > 0$ that maintain feasibility of both z and w variables. Either all components of v are nonnegative, in which case λ can take any positive value, so that we have an unbounded ray, or there is a finite maximum value

$$\theta = \max_i \{-(z_i + w_i)/v_i \mid v_i < 0\}, \tag{16.60}$$

where we defined $w_0 = 0$ for convenience.

Now, among the set of indices $\gamma = \{i \mid -(z_i + w_i)/v_i = \theta\}$, we pick at least one index s which is called the blocking variable. This either corresponds to the w_s variable, which is now forced to $w_s = 0$ by the action of z_r, or to the z_s variable which is no longer needed because z_r is taking its place. When more than one variable vanish for a given maximum $\lambda = \theta$, a special tie breaking rule must be applied to avoid cycling, as described in [210] and [69]. These details are not

covered here. In fact, for the regularized matrices considered in Section 10.11.4, there is no cycling possible.

To resolve what to do at the next stage, we need a special rule which is the second ingredient of the Lemke algorithm, namely:

one variable $w_s \mapsto 0$: then, variable z_s is added to the active set at the next stage;

one variable $z_s \mapsto 0$: then, variable z_s is removed from the active set and w_s is maximized at the next stage;

variable $z_{n+1} \mapsto 0$: then, the solution is reached.

multiple variables $z_{s_i} \mapsto 0$ or $w_{s_i} \mapsto 0$: must apply a tie breaking rule (see references [210, 69] for details).

We therefore need to know what happens when variable z_r is set to $z_r = 0$ but variable w_r is set free in (16.55). A calculation similar to what lead to (16.57) shows that we should now solve for

$$\begin{bmatrix} p_\alpha & M_{\alpha\alpha} \\ p_r & \mathcal{M}_{r\alpha} \end{bmatrix} \begin{bmatrix} v_0 \\ v_\alpha \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \tag{16.61}$$

and follow the exact same procedure as before when updating variables z and w. In this case also, the vector v is a solution to the homogeneous problem $w = Mz + z_0 p$, by reversing the values of z_r and w_r in (16.59).

It was already observed that the algorithm fails when it encounters an *unbounded ray*, i.e., when the update vector v in (16.57) or (16.61) has all positive components. After reorganizing the system of equations (16.56), we have

$$Mz + q + z_0 p + \lambda\left(M\bar{z} + \bar{z}_0\right) = w + \lambda\bar{w}, \tag{16.62}$$

where variables $\bar{z}, \bar{z}_0$ and $\bar{w}$ are obtained from vector v computed as per (16.59) or its complement, reversing the values of z_r and w_r. Since complementarity between w and z is maintained at each step, we have: $(w + \lambda\bar{w})^T(z + \lambda\bar{z}) = 0$. Now, since all variables are non-negative, this leads to the four identities

$$\begin{aligned} w^T z = 0,\ w^T\bar{z} = 0, \\ \bar{w}^T z = 0,\ \bar{w}^T\bar{z} = 0, \end{aligned} \tag{16.63}$$

and given that $\bar{w}, \bar{z}$ and $\bar{z}_0$ solve the homogeneous problem, we find that the ray termination condition implies

$$\bar{z}^T M\bar{z} = -\bar{z}_0\bar{z}^T p \leq 0, \tag{16.64}$$

and equality is satisfied if either $\bar{z}$, $\bar{z}_0$, or both vanish.

Therefore, if a matrix M is such that $z^T Mz > 0$ whenever $z > 0$, then, the algorithm cannot terminate on a ray. This is of course true for strict positive matrices defined in 16.8, which include the positive definite matrices. The basic

Lemke algorithm is listed in Algorithm 16.5.6. Note that this is a slightly simplified version which does not protect against cycling. The standard technique to break ties is to perform a *lexicographic* ordering the tied rows of the *current basis* matrix.

Definition 16.13. *Given an n-dimensional vector x, x is lexicographically less than 0, or $x \preceq 0$ when the first non-zero component of x is negative, so that $x_k = 0, k = 1, 2, \ldots, i-1$, and $x_i < 0$. Vector x is lexicographically greater than 0, or $x \succeq 0$ if $-x \preceq$. For two n-dimensional real vectors, x, and y, $x \preceq y$ if $x - y \preceq 0$ and $x \succeq y$ when $x - y \succeq 0$.*

Performing the lexicographic ordering of rows of the inverse is computationally intensive since it becomes necessary to solve for as many columns of the inverse as necessary to determine the lexicographic order of the rows. Ties occur when the problem is degenerate, i.e., when $z_i = w_i = 0$ for some index i at some given stage in the algorithm.

The Lemke algorithm solves many more problems than either Murty's, Keller's, or Cottle-Dantzig's methods. In fact, the class of copositive matrices is quite large, containing P matrices for instance. However, the artificial covering vector is problematic. In addition, it does not appear possible to warm start the algorithm with a guess of the solution index set α. However, as we shall see below, Lemke's algorithm performs very well on average.

When extending the Lemke algorithm to solve MLCPs as well, it is then only guaranteed to solve problems with positive semi-definite matrices [245, 146]. Since it then performs just as well as the Keller Algorithm 16.5.4, as seen in the numerical experiments of Section 16.8, but operating on non-symmetric augmented matrices, the analysis for the extension and the algorithm to cover MLCPs are not included.

Algorithm 16.5.6 Lemke's Algorithm for solving LCPs (without tie breaking).

Given real matrix M of size $n \times n$ and n-dimensional vector q
Choose covering vector $p \in \mathbb{R}^n$ with $p_i \geq 0$ ▷ $p_i = 1$ is the usual choice.
Initialize index sets $\alpha = \emptyset, \beta = \{1, 2, \ldots, n\}$
Find $\lambda = \max\{-q_i/p_i\} = -q_r/p_r$
$z_0 \leftarrow \lambda, w = z_0 p$
repeat
  if $r \in \beta$ **then**
    $b_\alpha = -M_{\alpha r}, b_r = -M_{rr}$
  else
    $b_\alpha = 0, b_r = 1$
  end if
  Solve: $\begin{bmatrix} p_\alpha & M_{\alpha\alpha} \\ p_r & M_{r\alpha} \end{bmatrix} \begin{bmatrix} v_0 \\ v_\alpha \end{bmatrix} = \begin{bmatrix} b_\alpha \\ b_r \end{bmatrix}$
  Compute : $v_\beta = q_\beta + M_{\beta\alpha} v_\alpha + v_0 p_\beta$
  if $v_i \geq 0$ $i = 1, 2, \ldots, n$ **then**
    Termination on an infinite ray: no solution! **return**
  end if
  Find $\lambda = \max\{-(z_i + w_i)/v_i \mid v_i > 0\} = -(z_s + w_s)/v_s$
  Update variables with $z_\alpha \leftarrow z_\alpha + \lambda v_\alpha, w_\beta \leftarrow w_\beta + \lambda v_\beta, z_0 \leftarrow z_0 + \lambda v_0$
  if $r \in \beta$ **then**
    $\beta \leftarrow \beta \setminus \{r\},\ \alpha \leftarrow \alpha \cup \{r\},\ z_r = \lambda$
  else
    $\alpha \leftarrow \alpha \setminus \{r\},\ \beta \leftarrow \beta \cup \{r\},\ w_r = \lambda$
  end if
  $r \leftarrow s$
until $z_0 = 0$

16.6 Iterative methods

Pivoting methods are generally problematic in that it is difficult to exploit the sparsity of the problem, to start from an advanced point which might be close to the solution, or to terminate early with an approximation. In addition, implementations are generally difficult, especially because factorization updates and down-dates are required and these operations are rarely provided in linear algebra software libraries.

By contrast, iterative methods are easy to implement, they can be started from any reasonable guess, and they can be terminated at any point. However, most of them cannot produce high accuracy solutions even when given a lot of time, and the class of problems they can be used on is typically limited to those $\mathtt{LCP}(M, q)$ where M is symmetric and positive definite.

Nevertheless, iterative methods, especially the projected Gauss-Seidel variety, have become dominant in the game physics literature [109, 275, 155, 84] and software libraries whilst direct methods have practically been eradicated from practice.

16.6.1 Projected Gauss-Seidel and SOR

Perhaps the very simplest methods to implement are iterative methods equivalent to the Gauss-Seidel iterations in linear algebra. This will be covered in more details in Section 18.5 but a brief description is provided here.

Consider a real $n \times n$ matrix M and a real n-dimensional vector q. Assume that we have a current candidate solution $z \geq 0$ and we are now looking at component k, i.e.,

$$w_k = q_k + m_{kk} z_k + \sum_{j \neq k} m_{kj} z_j, \tag{16.65}$$

and we consider how the value of z_k would be modified. If we have $w_k > 0$, the complement variable z_k should be set to 0, or relaxed in that direction. Conversely, if $w_k < 0$, the value of variable z_k should be increased. This leads to sequential over-relaxation update rule

$$z_k \leftarrow \max\left(0, \frac{z_k - \alpha w_k}{m_{kk}}\right), \tag{16.66}$$

where $\alpha \in (0, 2)$ [144]. This is detailed in Algorithm 16.6.1.

As observed in [144] and references therein though, the update rules given by (16.65) and (16.66) should be used with caution. The algorithm may not converge at all, for symmetric, positive definite matrix M. A generalization of update formulae (16.65) and (16.66) is given in [210] where convergence criteria are provided. The convergence is linear, at best, as in the case of ordinary Gauss-Seidel, or any other stationary process [158].

equivalent to the

Algorithm 16.6.1 The projected SOR algorithm for solving $\mathrm{LCP}(M, q)$.

Given $n \times n$ matrix M and real vector $q \in \mathbb{R}^n$, initial vector $z \in \mathbb{R}^n_+$, tolerance parameter $\tau > 0$, and integer $\nu_{max} > 1$
$\nu \leftarrow 0$
repeat
  for $i = 1, 2, \ldots, n$ **do**
    $w_i = q_i + \sum_j m_{ij} z_j$
    $z_i \leftarrow \max(0, \frac{z_i - \alpha w_i}{m_{ii}})$
  end for
  $\nu \leftarrow \nu + 1$
until ($w = q + Mz \geq 0$ and $z^T w \leq \tau$) or ($\nu > \nu_{max}$)

16.6.2 Conjugate gradient and other methods

It is of course possible to use conjugate gradient [158, 52, 40] to solve the linear algebra problems in Algorithms 16.5.1, 16.5.1, 16.5.3, or 16.5.5.

16.7 The LCP as a nonlinear problem

It should be clear by now that solving an LCP is not equivalent to a solving a linear system and it should be no surprise that we can formulate it as a nonlinear system of equations. Consider the non-smooth definition of the min function for two variables, x, y

$$\mu(x, y) = \frac{1}{2}\left(x + y - \sqrt{(x-y)^2}\right) = \min(x, y). \tag{16.67}$$

The function μ is known as the Fischer Burmeister [88] function and is continuous but has a discontinuous derivative at $x = y$. Extend this definition to vectors in $\mathbb{R}^n$ componentwise so that $z_i = \mu(x_i, y_i), x, y, z \in \mathbb{R}^n$, and therefore, we can define the LCP as the *nonlinear system of equations*

$$\mu(z, Mz + q) = 0. \tag{16.68}$$

The definition of the function $\mu(x, y)$ can be modified, e.g., by introducing a *smoothing parameter* $c > 0$ so that, for instance,

$$\mu_c(x, y) = \frac{1}{2}\left(x + y - \sqrt{(x-y)^2 + c^2}\right). \tag{16.69}$$

This type of perturbation for $c > 0$ is the root of *smoothed Newton methods* for solving LCP [285, 183].

16.7.1 Block principal pivots and Newton's method

Given any system of non-linear equations, the first reasonable thing to try is Newton-Raphson iterations. This is now applied directly to the Fischer-Burmeister equation definition for $\mathrm{LCP}(M, q)$) of (16.68) for a given $n \times n$ real matrix M and real n-dimensional vector q.

The partial derivatives of the non-smooth function $\mu(x,y)$ are easily computed to be

$$D_1\mu(x,y) = \begin{cases} 1 & \text{when } x < y \\ 0 & \text{when } x > y \\ [0,1] & \text{when } x = y \end{cases} \tag{16.70}$$

$$D_2\mu(x,y) = \begin{cases} 0 & \text{when } x < y \\ 1 & \text{when } x > y \\ [0,1] & \text{when } x = y\ , \end{cases} \tag{16.71}$$

where notions of convex analysis were used for defining the derivative at the point $x = y$, as in [66].

If we ignore the degenerate cases $x = y$ for the moment, we can compute the Jacobian of $\mu(z, Mz+q)$ by introducing disjointed index sets $\alpha, \beta \subset \{1,2,\ldots,n\}$ with $\alpha \cup \beta = \{1,2,\ldots,n\}$ so that $i \in \alpha$ whenever $\mu(z_i, M_{i\bullet}z + q_i) = M_{i\bullet}z + q_i$, and $i \in \beta$ whenever $\mu(z_i, M_{i\bullet}z + q_i) = z_i$, breaking ties arbitrarily. Defining $w = Mz + q$ as previously, a simple computation shows that

$$\begin{aligned} \nabla\mu(z, Mz+q) &= \begin{bmatrix} \frac{\partial\mu(z_\alpha,w_\alpha)}{\partial z_\alpha} & \frac{\partial\mu(z_\alpha,w_\alpha)}{\partial z_\beta} \\ \frac{\partial\mu(z_\beta,w_\beta)}{\partial z_\alpha} & \frac{\partial\mu(z_\beta,w_\beta)}{\partial z_\beta} \end{bmatrix} \\ &= \begin{bmatrix} M_{\alpha\alpha} & M_{\alpha\beta} \\ 0 & I_{\beta\beta} \end{bmatrix}. \end{aligned} \tag{16.72}$$

Now, defining the Newton-Raphson iterations as

$$\mu(z^{(\nu)}, Mz^{(\nu)} + q) + \nabla\mu(z^{(\nu)}, Mz^{(\nu)} + q)\left(z^{(\nu+1)} - z^{(\nu)}\right) = 0, \tag{16.73}$$

and after reorganizing terms and performing a simple computation, the linear system to solve at stage $\nu + 1$ is then

$$\begin{bmatrix} M_{\alpha\alpha} & M_{\alpha\beta} \\ 0 & I_{\beta\beta} \end{bmatrix} \begin{bmatrix} z_\alpha^{(\nu+1)} \\ z_\beta^{(\nu+1)} \end{bmatrix} = \begin{bmatrix} -q_\alpha \\ 0 \end{bmatrix}, \tag{16.74}$$

which, really, is nothing more than the decoupled system

$$\begin{aligned} M_{\alpha\alpha}z_\alpha^{(\nu+1)} &= -q_\alpha, \\ z_\beta^{(\nu+1)} &= 0, \\ w_\alpha^{(\nu+1)} &= 0, \\ w_\beta^{(\nu+1)} &= M_{\beta\alpha}z_\alpha^{(\nu+1)} + q_\beta. \end{aligned} \tag{16.75}$$

So now, consider evaluating the function $\mu(z^{(\nu+1)}, w^{(\nu+1)})$ to split the index sets α, β for the stage $\nu + 2$. According to the assignments listed in (16.75), for any $i \in \alpha$, if $z_i > 0$, $\mu(z_i, w_i) = w_i = 0$ and index i stays in set α. Likewise, for any index $i \in \beta$, if $w_i > 0$, then, index i stays in β. However, if $i \in \alpha$ and $z_i < 0$,

then, that index will appear in set β at the next stage and similarly, any index $i \in \beta$ such that $w_i < 0$ will be transferred to α at the next stage. Therefore, a strict application of the Newton-Raphson method to LCP(M, q) defined by the non-smooth nonlinear Fischer-Burmeister equations $\mu(z, Mz + q) = 0$ leads to a block principal pivot algorithm as described below in Algorithm 16.7.1. This observation is due to Kostreva [162] and though it has been mentioned in passing in the literature [144], it is far from widely known.

Algorithm 16.7.1 Newton-Raphson iterations applied to nonsmooth formulation of LCP.

Given real $n \times n$ real matrix M and n-dimensional real vector q, and integer $\nu_{max} > 1$
Initialize: $\alpha \leftarrow \alpha_0, \beta \leftarrow \{1, 2, \ldots, n\} \setminus \alpha$
repeat
  Solve : $M_{\alpha\alpha} z_\alpha = -q_\alpha$
  Compute : $w_\beta \leftarrow M_{\beta\alpha} z_\alpha + q_\beta$
  Find : $\sigma = \{i \in \alpha \mid z_i < 0\}$
  Find : $\tau = \{i \in \beta \mid w_i < 0\}$
  Set : $\alpha \leftarrow \tau \cup \alpha \setminus \sigma$
  Set : $\beta \leftarrow \sigma \cup \beta \setminus \tau$
  $\nu \leftarrow \nu + 1$
until $\sigma = \tau = \emptyset$ or $\nu > \nu_{max}$

It should be noted here that Newton-Raphson's iterations may not converge to $\mu(z, Mz + q) = 0$ since the function $\mu(z, Mz + q)$ does not have smooth derivatives. What happens then is that Algorithm 16.7.1 *cycles*, and a sequence of index sets $\alpha^{(\nu)}, \alpha^{(\nu+1)}, \ldots, \alpha^{(\nu+p)}, \alpha^{(\nu)}$ repeats ad infinitum.

There are several alternative strategies to force convergence developed in the context of *smoothed* Newton methods [183, 285]. In fact, the entire volume 17 of *Computational Optimization and Applications*, published in 2000, was devoted to such smoothed Newton methods.

16.8 Numerical experiments

The sad news about LCP is that it is in fact an NP-hard problem [69]. In the worst case, one has to enumerate all 2^n complementarity combinations of the columns of matrix $-M$ and the identity I_n, for an n-dimensional LCP(M, q). This can be done with some efficiency perhaps [145], but this is not really practical.

The good news is that *on average*, for well-behaved problems—and no one at this time knows exactly what that means—direct solvers process LCP(M, q) using roughly the same number of Gauss-Jordan pivot operation as is needed to factor matrix M with LU, say.

This said, not all algorithms are created equal. To compare the algorithms listed in the present Chapter, sets of random problems were created, following the

strategy of Alefeld Chen and Potra [3]. Random matrices with known condition number τ and known rank r are generated by creating a diagonal matrix, $D = \mathtt{diag}(d_1, d_2, \ldots, d_r, 0, \ldots, 0)$, where the first entry $d_1 = \sqrt{\tau}$ and the last non-zero entry is $1/\sqrt{\tau}$. Entries in between have value τ^{α_i} where α_i is uniformly distributed in $(-1/2, 1/2)$. This diagonal matrix then has condition number τ and rank r. It is then transformed with random orthogonal matrices generated by performing random Givens rotations [107] on the identity. Only data for symmetric and positive definite matrices is presented below. Once matrix M is obtained in this way, a vector q is generated with a uniform random deviate in a given range. Bounds can be generated in the same way to produce MLCPs.

In addition to random data, different solvers were tested on simulation data involving the stacking of forty identical cylindrical logs. These logs are let to fall under gravity from a separated configuration and they eventually pile up, confined by four immovable posts. The original simulations were performed using the block pivot method of Section 16.7.1 and the problems were saved at each step. These were loaded again in Octave and solved with different solvers.

The performance measurement for the pivot and Newton-type methods is a *frequency* histogram, which logs the fraction of problems solved for a given number of iterations or pivot steps. For the scales to concord between pivoting and Newton-type methods, the number of pivots taken for the former is divided by the problem size. The iteration count of Newton-type methods is not scaled. Results are plotted separately and are not an absolute measure of performance.

For the projected Gauss-Seidel method, the relative error is plotted as a function of the absolute iteration count as is customary. Since projected Gauss-Seidel does solve the problem exactly, no frequency histograms are presented.

Figure 16.3 shows performance data for the projected Gauss-Seidel method applied to random problems of size 100×100 of moderate condition numbers, between 10^3 and 10^6, and moderate range of q vector in $[-100, 100]$. The convergence is linear as expected but not particularly fast either. Also expected is a linear anti-correlation between the convergence rate and the condition number. The real problem with projected Gauss-Seidel is seen in Figure 16.4 however, which illustrates the performance on moderately large stacking contact problems. The iterations simply stagnate in most cases, leaving large residual errors.

Figure 16.5 and Figure 16.6 illustrate performance of the Newton-type algorithms on boxed MLCPs of size 100 and 400 respectively. There is no clear winner here but all solvers do quickly find the solution. What is usually expected from a Newton-Raphson method is convergence within five to ten iterations and this is what is seen here. For larger problems, the method of Zhang and Gao [285] is slightly better than brute force block pivoting. On contact problems, as shown in Figure 16.7, block pivot performs best. However, it was necessary to perturb the problems on the diagonals to lower the condition number down to the 10^3 range and this warrants further research to improve performance without having to strongly regularize the problems first.

Figure 16.8 illustrates the performance of the pivoting methods on LCPs of size 100 with condition number 10^4. The performance is reasonable for all the

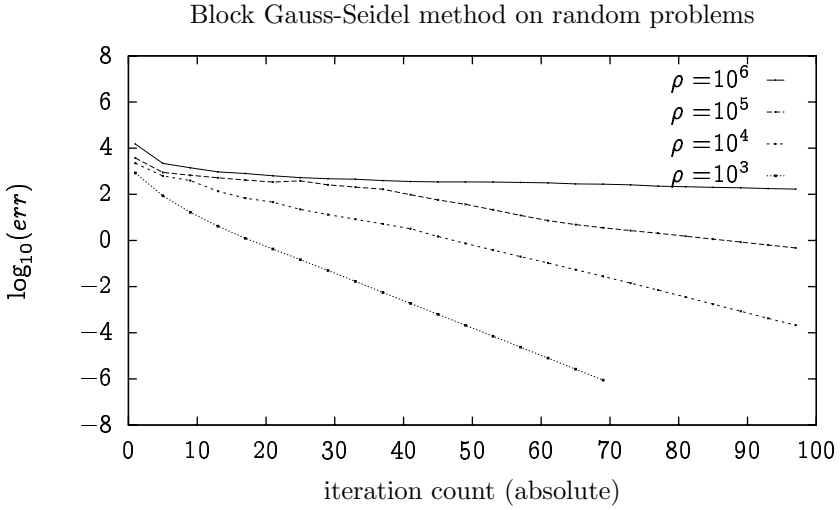

Figure 16.3: Progression of projected Gauss-Seidel iteration on random matrices of size 100×100. Convergence is linear and decreases with increasing condition number.

methods presented but Lemke's and Keller's algorithms are clear winner, taking close to exactly 100 pivot steps for each of these problems. This is the same as matrix factorization though in the present case, the benefit of using GEMM-based algorithms is lost. The performance degrades for Murty's and Cottle-Dantzig's methods on MLCPs though, as seen in Figure 16.9. Murty's method often does not complete after a full 1000 pivot operations and was then stopped, explaining the large frequency count on the graph. Keller's and Lemke's methods hold good, however. The same story is repeated in Figure 16.10 where performance is measured for random problems of size 400. For this case, Murty's method never terminates with less than $10 \cdot 400$ iterations. The Cottle-Dantzig method performs poorly but Keller's and Lemke's methods hit the target in 400 pivots. Performance measurements on simulation data is found in Figure 16.11. Unlike in the case of the Newton method, the matrices were not further perturbed and condition numbers were sometimes as high as 10^8. This is possible thanks to the precision of the linear algebra routines used to solve the subproblems. By contrast, the Newton methods need smoothness to converge, and this is achieved by diagonal perturbation. Lemke's method is slightly more reliable than Keller's here and Cottle-Dantzig is of no particular use. It had to be stopped often after performing $10 \cdot n$ pivot operations on problems of size n as seen in the picture.

The real issue though is that Newton-type methods are stateless and can be started at an advanced solution. Not so of Keller's or Lemke's method. Also, it is possible to optimize the linear algebra for the Newton-type methods and so, they are typically faster. The construction of fast, efficient and robust solvers is an open problem.

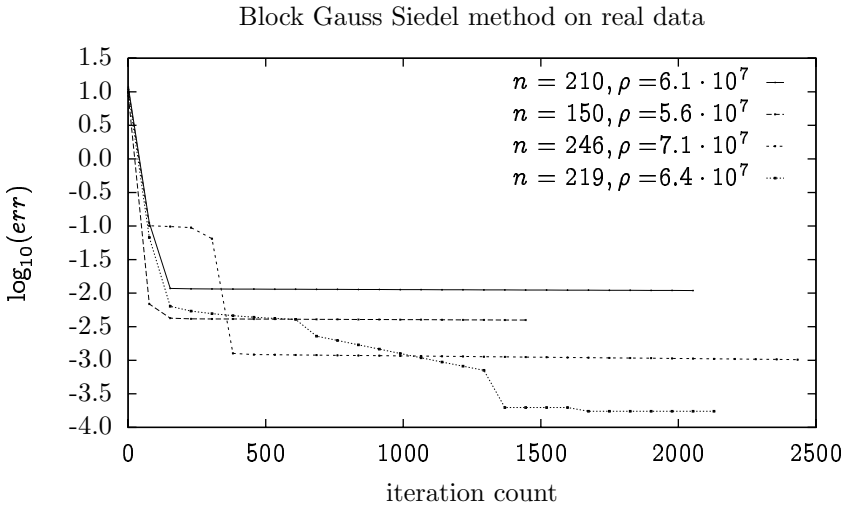

Figure 16.4: Progression of projected Gauss-Seidel iterations on dry frictional contact problems using box friction model.

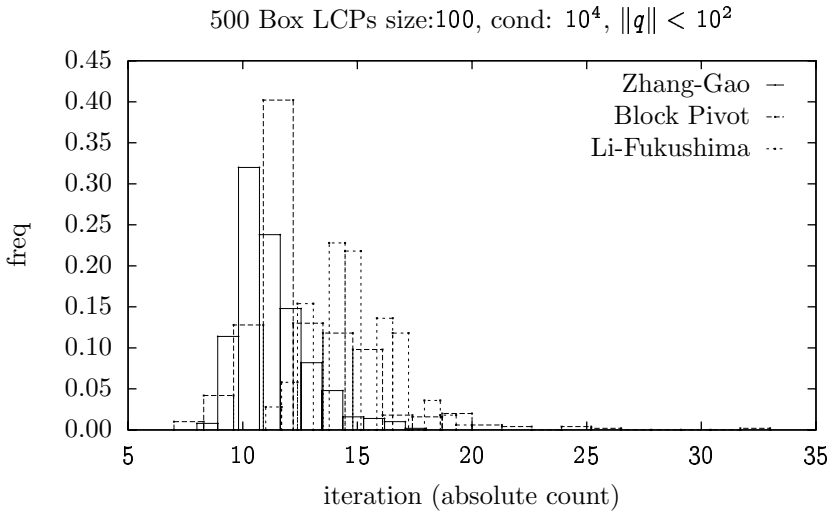

Figure 16.5: Solution frequency for Newton-type solvers on small random boxed MLCPs with moderate condition numbers.

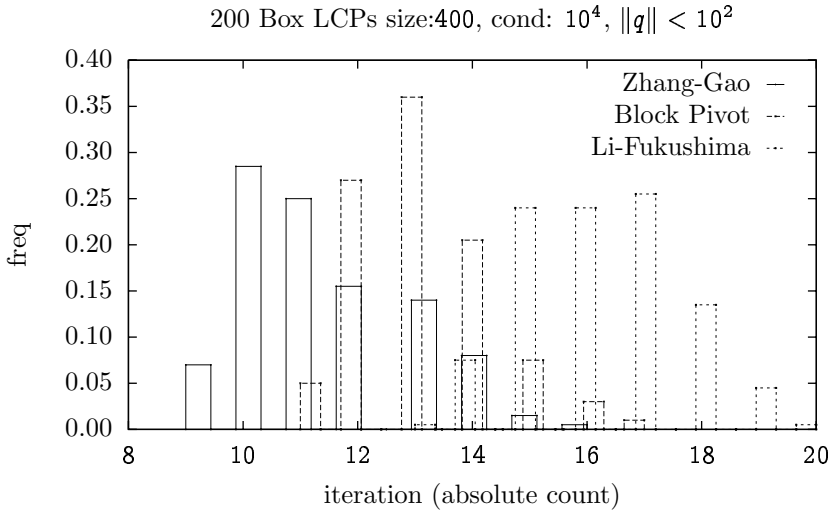

Figure 16.6: Solution frequency for Newton-type solvers on medium sized random boxed MLCPs with moderate condition numbers.

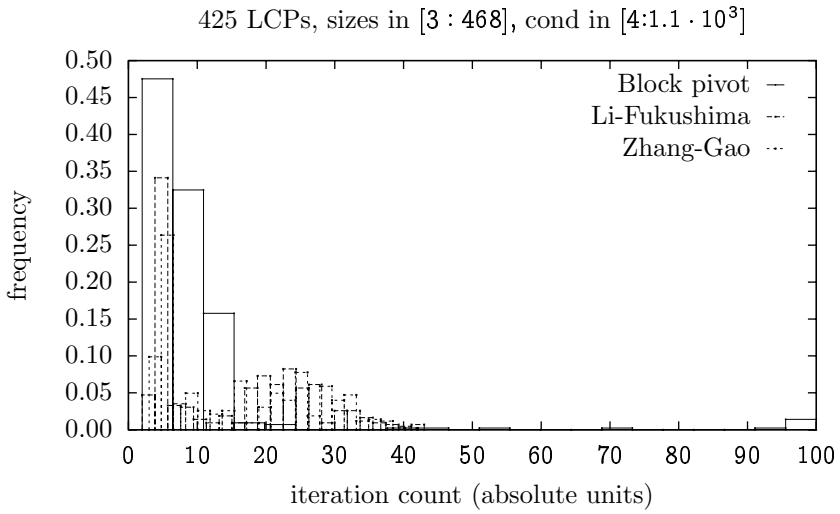

Figure 16.7: Solution frequency of Newton-type solvers on dry frictional contact problems using box friction model.

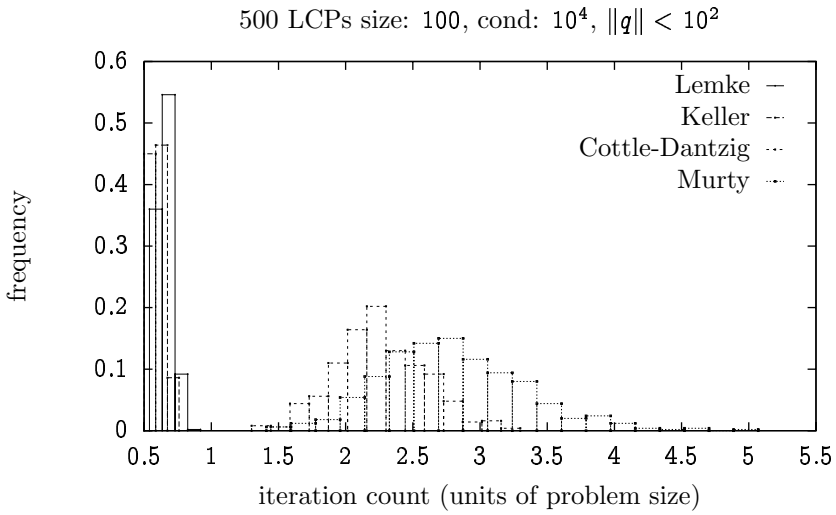

Figure 16.8: Solution frequency of direct solvers on small LCPs with moderate condition numbers.

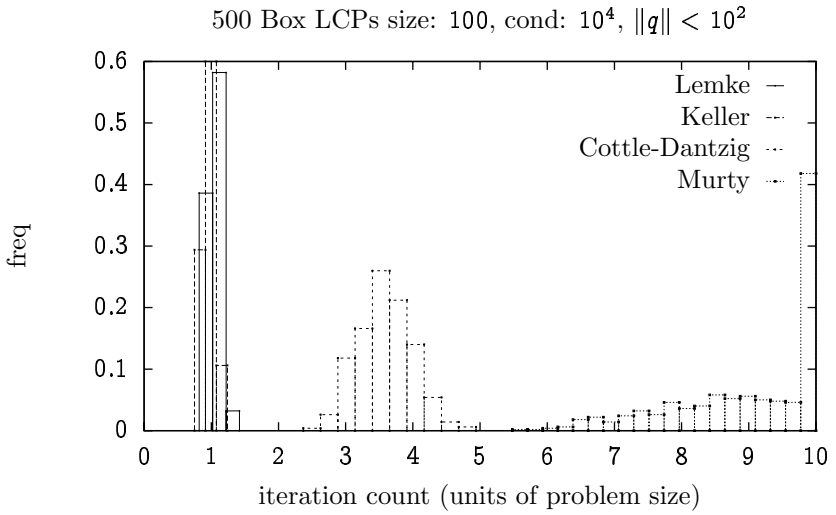

Figure 16.9: Solution frequency of direct solvers on small boxed MLCPs with moderate condition numbers.

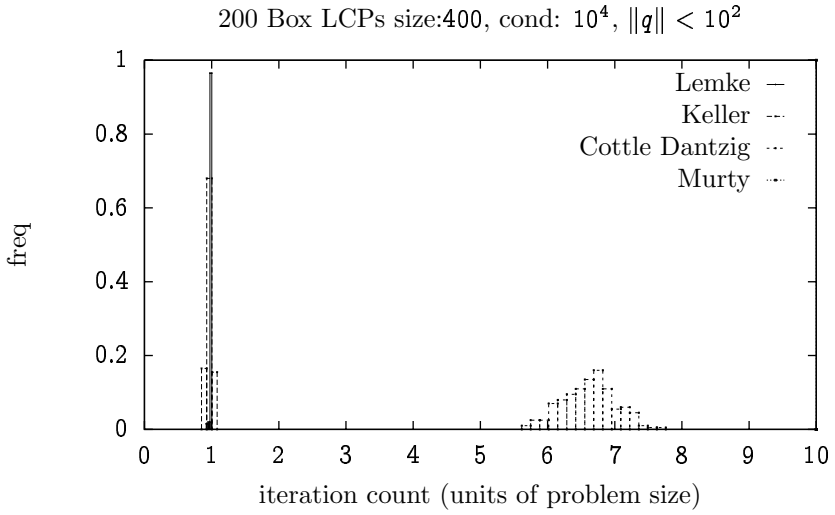

Figure 16.10: Solution frequency of direct solvers on medium sized boxed MLCPs with moderate condition numbers.

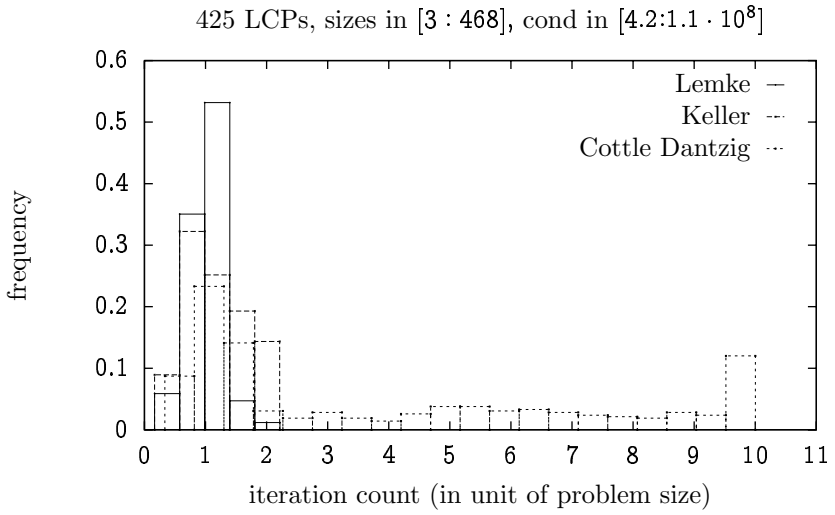

Figure 16.11: Solution frequency of direct solver on dry frictional contact problems using box friction model.

16.9 End notes

The review of LCP algorithms by Júdice [144] is invaluable for plain descriptions of the standard methods. It is also a fast and direct introduction to the topic. The textbook of Murty [210] and the monograph of Cottle, Pang, and Stone [69] provide encyclopedic coverage.

Implementation of complementarity algorithms is frustrating since linear algebra libraries seldom provide factorization update and down-date routines required to avoid repeatedly factoring matrices for each pivot step. Some "poor man's" solutions for this problem are provided in Section 18.7 but this is far from satisfactory. The commercial libraries based on the work of Gil, Golub, Murray, Saunders, and Wright [99, 98, 97], are exceptions to this rule, however.

Reliable and fast algorithms to solve complementarity problems, especially those arising from frictional contact problems, are still missing and their development is part of future work. Direct methods can be very good for moderate sized problems but are far too slow when the problem size increases. Moreover, they cannot be warm-started, except for Murty's method, but this latter is not doing well on average. By contrast, Newton-type iterative methods can be warm started and and can be fast on large problems. However, they appear to need excessive smoothing for convergence.

17 Complementarity II: Splitting and other tricks

The model of dry friction developed in Chapter 10 requires the solution of complementarity problems which, either in their quasi-linear or linear formulation, do not enjoy a problem matrix that is positive definite. In addition, these matrices are neither symmetric nor bisymmetric. The asymmetry arises from the *switching conditions* which determine whether a contact is in static friction or kinetic friction. The present chapter investigates the mathematics of a splitting strategy in which the asymmetry is removed from the matrix and moved to the data vectors. This leads to an iterative strategy which has the potential to converge linearly.

Background and definitions are provided in Section 17.1 and the essential aspects of the set of iterates produced by the splitting are presented in Section 17.2. The geometric features of these sets is further described in Section 17.3 in which it is concluded that the sequence is closed and bounded. The convergence rate is analyzed in Section 17.4 where it is found that contractivity of the norm of the difference between iterates is not guaranteed and thus, convergence is conditional. Application to box friction is described in Section 17.5 and the results of numerical experiments are presented in Section 17.6, followed by remarks in Section 17.7.

17.1 Introduction

It was shown in Section 10.11 that introducing dry friction models in dynamics turns the stepping equations of a mechanical system into a mixed complementarity problem, linear or nonlinear according to the friction model choice. In turn, the form of these complementarity problems is problematic because they are neither symmetric nor bisymmetric as is typical of standard constrained extremal problem. Instead, the linearization of the strict complementarity problem can be written in terms of a square, $n \times n$ matrix H as

$$H = H_0 + U, \tag{17.1}$$

where H_0 is a square $n \times n$ positive definite matrix (not necessarily symmetric) and U is a matrix with non-negative elements. This splitting is discussed in more details in Section 10.11.4. The result developed in the next few sections is not sensitive to the exact form of H_0 or U and we therefore leave precise definitions to the applications sections below.

The LCP to be solved is defined as

$$\begin{aligned} Hz + q &= w \\ 0 \leq z \perp w &\geq 0. \end{aligned} \tag{17.2}$$

Being the sum of a positive definite matrix and a non-negative one, matrix H is copositive as defined in 16.8, the Lemke algorithm 16.5.6 can solve the linear complementarity problem (LCP) defined in (17.2) when protection is taken against cycling.

Consider the splitting of H in a positive semi-definite matrix $H(\tau)$ and a non-negative matrix $\bar{U}(\tau)$ with $H = H(\tau) + \bar{U}(\tau)$, where $H(\tau) = (H_0 + \tau(U - U^T))$ and $\bar{U}(\tau) = \tau U^T + (1-\tau)U$. An iterative scheme based on this splitting is defined as follows

$$\begin{aligned} H(\tau)z^{(\nu+1)} + q^{(\nu+1)} &= w^{(\nu+1)} \\ 0 \leq z^{(\nu+1)} \perp w^{(\nu+1)} &\geq 0 \\ q^{(\nu+1)} &= q + \bar{U}(\tau)z^{(\nu)}. \end{aligned} \tag{17.3}$$

We denote this sequence $\Sigma(\tau)$. It is clear that a fixed point of this sequence is a solution to the original problem. This form of splitting has already appeared in [16].

Also considered is regularization of the original problem in which the scalar $\epsilon > 0$ is added on the diagonal entries of matrix $H(\tau)$. The corresponding sequence is labeled $\Sigma(\tau, \epsilon)$. This corresponds directly to the regularization scheme of Chapter 4. With this diagonal perturbation, the matrix $H(\tau, \epsilon) = H(\tau) + \epsilon I$, where I is the identity matrix of order n, becomes strictly positive definite.

Any square matrix M of size $n \times n$ and n-dimensional vector q defines an LCP denoted by $LCP(M, q)$. The set of all solution vectors $z \in \mathbb{R}^n_+$ of this problem is denoted $\mathrm{SOL}(M, q)$. For a P matrix, this set is a singleton [210, 69].

For any given solution, introduce the index sets $\alpha, \bar{\alpha}$ such that $\alpha \cup \bar{\alpha} = \{1, 2, \ldots, n\}$ and $\alpha \cap \bar{\alpha} = \emptyset$, and such that the solution can be written as:

$$\begin{aligned} z &= \begin{bmatrix} z_\alpha \\ z_{\bar{\alpha}} \end{bmatrix}, \quad z_\alpha = -M_{\alpha\alpha}^{-1} q_\alpha, \quad z_{\bar{\alpha}} = 0, \\ w &= \begin{bmatrix} w_\alpha \\ w_{\bar{\alpha}} \end{bmatrix}, \quad w_\alpha = 0, \qquad\qquad w_{\bar{\alpha}} = q_{\bar{\alpha}} + M_{\bar{\alpha}\alpha} z_\alpha. \end{aligned} \tag{17.4}$$

The vector $y = (y_\alpha^T, y_{\bar{\alpha}}^T)^T$ with $y_\alpha = z_\alpha$, $y_{\bar{\alpha}} = w_{\bar{\alpha}}$ is called the *complementarity point*, and $y \geq 0$ by construction. If $y_i = 0$ for some index i, the complementarity point is said to be degenerate.

The solution set of $LCP(M, q)$ can then be written as

$$\begin{aligned} \mathrm{SOL}(M, q) = \bigcup_\alpha \{ z \in \mathbb{R}^n_+ \mid z_{\bar{\alpha}} = 0, \quad q_\alpha + M_{\alpha\alpha} z_\alpha = 0, \\ q_{\bar{\alpha}} + M_{\bar{\alpha}\alpha} z_\alpha \geq 0 \}. \end{aligned} \tag{17.5}$$

Each subset in the union is labeled $S_\alpha(M, q)$, and some of these are empty.

17.2 Characterization of the iterates

Observe from (17.3) that since both matrix $\bar{U}(\tau)$ and vector $z^{(\nu)}$ are non-negative, the value of $q^{(\nu)}$ can be expressed as $q + \bar{q}$ where $\bar{q} \geq 0$. Therefore, the sequence of iterates $z^{(\nu)}$ is included in the set $\tilde{S}(H(\tau), q) = \{\mathrm{SOL}(H(\tau), q + \bar{q}) \mid \bar{q} \in \mathbb{R}^n_+\}$. This can be split into 2^n subsets corresponding to the index sets α so that $\tilde{S}(H(\tau), q) = \bigcup_\alpha \tilde{S}_\alpha(H(\tau), q)$, with the definition:

$$\begin{aligned} \tilde{S}_\alpha(H, q) = \{z \in \mathbb{R}^n_+ \text{ and } \mid \\ z_{\bar{\alpha}} = 0, q_\alpha + \bar{q}_\alpha + H_{\alpha\alpha} z_\alpha = 0, \text{ and} \\ q_{\bar{\alpha}} + \bar{q}_{\bar{\alpha}} + H_{\bar{\alpha}\alpha} z_\alpha \geq 0 \text{ for } \bar{q} \in \mathbb{R}^n_+\}. \end{aligned} \tag{17.6}$$

In the case where $\epsilon > 0$, $H(\tau, \epsilon)$ is a P matrix and there is a unique vector $z \in \mathbb{R}^n_+$ for any value of $\bar{q}$ which must be found in one of the sets $\tilde{S}_\alpha(H(\tau, \epsilon), q)$ by the pigeon hole principle.

Corresponding to each set $\tilde{S}_\alpha(H, q)$, there is a set $\mathcal{Q}_\alpha(H, q) = \{\bar{q} \in \mathcal{R}^n_+ \mid S_\alpha(H, q + \bar{q}) \neq \emptyset\}$. In the case where H is a P matrix, there is a unique vector $z(q + \bar{q})$ for each $\bar{q} \in \mathbb{R}^n_\alpha$. However, there might be more than one $\bar{q} \in \mathbb{R}^n_+$ for a given $z \in \tilde{S}_\alpha(H, q)$.

We proceed now to prove that for any $\epsilon > 0$, the sequence $\Sigma(\tau, \epsilon)$ in a closed and bounded set and therefore, it is closed and bounded and it must contain a converging subsequence. This is achieved by proving that $\tilde{S}(H(\tau, \epsilon), q)$, which contains the sequence $\Sigma(\tau, \epsilon)$, is closed and bounded.

17.3 Characterization of the sets $S_\alpha(H, q)$ and $\tilde{S}(H, q)$.

The suffixes τ and ϵ are now dropped to simplify notation and to concentrate on general matrices H. First look at the basic properties of the sets $S_\alpha(H, q)$ and $\tilde{S}_\alpha(H, q)$.

Consider an index set α and the set of $\bar{q}$ vectors which lead to solutions $z \in \tilde{S}_\alpha(H, q)$. Any such solution can be written as

$$\begin{aligned} z_\alpha(\bar{q}_\alpha) &= -H^{-1}_{\alpha\alpha}(q_\alpha + \bar{q}_\alpha) = -H^{-1}_{\alpha\alpha} q_\alpha - H^{-1}_{\alpha\alpha} \bar{q}_\alpha \\ &= z_\alpha(0) + \tilde{\bar{q}}_\alpha, \end{aligned} \tag{17.7}$$

with $z(0)_\alpha = -H^{-1}_{\alpha\alpha}$ and $\tilde{\bar{q}}_\alpha = -H^{-1}_{\alpha\alpha} \bar{q}_\alpha$. Note that $z_\alpha(\bar{q})$ only depends on the value of $\bar{q}_\alpha$. Similarly, we have the following formula for $w_{\bar{\alpha}}$:

$$\begin{aligned} w_{\bar{\alpha}}(\bar{q}) &= q_{\bar{\alpha}} + \bar{q}_{\bar{\alpha}} + H_{\bar{\alpha}\alpha} z(\bar{q}_\alpha) \\ &= q_{\bar{\alpha}} + \bar{q}_{\bar{\alpha}} - H_{\bar{\alpha}\alpha} H^{-1}_{\alpha\alpha}(q_\alpha + \bar{q}_\alpha) \\ &= q_{\bar{\alpha}} - H_{\bar{\alpha}\alpha} H^{-1}_{\alpha\alpha} q_\alpha + \bar{q}_{\bar{\alpha}} - H_{\bar{\alpha}\alpha} H^{-1}_{\alpha\alpha} \bar{q}_\alpha \\ &= w_{\bar{\alpha}}(0) + \bar{q}_{\bar{\alpha}} - H_{\bar{\alpha}\alpha} H^{-1}_{\alpha\alpha} \bar{q}_\alpha, \text{ where} \\ w_{\bar{\alpha}}(0) &= q_{\bar{\alpha}} - H_{\bar{\alpha}\alpha} H^{-1}_{\alpha\alpha} q_\alpha. \end{aligned} \tag{17.8}$$

The results in (17.7) and (17.8) can be collected in a single matrix equation as

$$y^{(\alpha)}(\bar{q}) = y^{(\alpha)}(0) + \begin{bmatrix} -H_{\alpha\alpha}^{-1} & 0 \\ -H_{\bar{\alpha}\alpha}H_{\alpha\alpha}^{-1} & I \end{bmatrix} \begin{bmatrix} \bar{q}_\alpha \\ \bar{q}_{\bar{\alpha}} \end{bmatrix} = y^{(\alpha)}(0) + \tilde{H}^{(\alpha)}\bar{q} = y^{(\alpha)}(0) + \tilde{\bar{q}},$$

where

$$y^{(\alpha)}(0) = \begin{bmatrix} z_\alpha(0) \\ w_{\bar{\alpha}}(0) \end{bmatrix}. \tag{17.9}$$

Introduce now a parametric set of vectors $\bar{q} = \lambda\bar{q}^{(1)}$ where $\bar{q}^{(1)} \geq 0$ and $\lambda \geq 0$.

Theorem 17.1. *Given $\bar{q}^{(1)} \in \mathbb{R}^n_+$ the set $\{\mathrm{SOL}(H, q + \lambda\bar{q}^{(1)}) \mid \lambda \geq 0\}$ is closed and bounded if H is positive definite.*

Proof. The complementarity points can be written as $y(\lambda) = y(0) + \tilde{\bar{q}}^{(1)}$ and in particular, $z_\alpha(\lambda) = z_\alpha(0) + \lambda\tilde{\bar{q}}_\alpha^{(1)}$. The ray $z_\alpha(\lambda), \lambda \geq 0$ is unbounded only if $\tilde{\bar{q}}^{(1)} \geq 0$. Assume this is the case. Since $\bar{q}_\alpha \geq 0$, we should have $\bar{q}_\alpha^{(1)T}\tilde{\bar{q}}_\alpha^{(1)} \geq 0$. Since H is assumed positive definite, so is $H_{\alpha\alpha}^{-1}$ and we have the following inequalities

$$0 \leq \bar{q}_\alpha^{(1)T}\tilde{\bar{q}}_\alpha^{(1)} = -\bar{q}_\alpha^{(1)T}H_{\alpha\alpha}^{-1}\bar{q}_\alpha^{(1)} \leq 0. \tag{17.10}$$

This can only be true if equality is satisfied on both sides which leads to $\bar{q}_\alpha^{(1)} = 0$ and therefore, $z_\alpha(\lambda) = z_\alpha(0)$. In this case, the solution set is the singleton $z_\alpha(0)$ which is closed and bounded.

Otherwise, if there is at least one negative component in $\tilde{\bar{q}}^{(1)}$, and then, we have

$$\bar{\lambda} = \min_{\tilde{\bar{q}}_i^{(1)} < 0} (y(0)_i / \tilde{\bar{q}}_i^{(1)}) \tag{17.11}$$

and the solution set is: $\{z_\alpha(\lambda) = z_\alpha(0) + \lambda\tilde{\bar{q}}_\alpha \mid \lambda \in [0, \bar{\lambda}]\}$. This is a closed and bounded set as it is a linear mapping of the closed interval $[0, \bar{\lambda}] \in \mathbb{R}$. □

Before proceeding to prove an equivalent result for the multivariate case, we need to establish a number of preliminary results.

Lemma 17.2. *The sets $S_\alpha(H, q)$ are convex for any real matrix H and vector q.*

Proof. Let $z^{(1)}$ and $z^{(2)}$ be distinct elements in the set $S_\alpha(H, q)$. Pick a non-negative scalar $s \in [0, 1]$ and consider the vector $z(s) = sz^{(1)} + (1 - s)z^{(2)}$. This

vector is also in the set since

$$\begin{aligned} q_\alpha + H_{\alpha\alpha} z_\alpha(s) &= q_\alpha + sH_{\alpha\alpha} z_\alpha^{(1)} + (1-s) H_{\alpha\alpha} z_\alpha^{(2)} \\ &= s(q_\alpha + H_{\alpha\alpha} z_\alpha^{(1)}) + (1-s)(q_\alpha + H_{\alpha\alpha} z_\alpha^{(2)}) \\ &= 0 + 0 = 0 \\ &\quad \text{and similarly:} \qquad (17.12) \\ q_{\bar{\alpha}} + H_{\bar{\alpha}\alpha} z_\alpha(s) &= q_{\bar{\alpha}} + sH_{\bar{\alpha}\alpha} z_\alpha^{(1)} + (1-s) H_{\bar{\alpha}\alpha} z_\alpha^{(2)} \\ &= s(q_{\bar{\alpha}} + H_{\bar{\alpha}\alpha} z_\alpha^{(1)}) + (1-s)(q_{\bar{\alpha}} + H_{\bar{\alpha}\alpha} z_\alpha^{(2)}) \\ &\geq 0. \end{aligned}$$

The last inequality holds since both terms in the next to last equation are non-negative. □

We proceed with a similar result for the sets $\tilde{S}_\alpha(H,q)$.

Lemma 17.3. *The set* $\tilde{S}_\alpha = \{z \in \mathbb{R}^n_+ \mid z_{\bar{\alpha}} = 0, q_\alpha + \bar{q}_\alpha + H_{\alpha\alpha} z_\alpha = 0, q_{\bar{\alpha}} + \bar{q}_{\bar{\alpha}} + H_{\bar{\alpha}\alpha} z_\alpha \geq 0, \bar{q} \in \mathbb{R}^n_+\}$ *is convex.*

Proof. Pick $z^{(1)}, z^{(2)} \in \tilde{S}_\alpha$, $s \in [0,1]$, and let $\bar{q}^{(1)}, \bar{q}^{(2)} \in \mathbb{R}^n_+$ be vectors such that $z^{(i)} \in S_\alpha(H, q + \bar{q}^{(i)}), i = 1,2$. Repeating the calculation of Lemma 17.2 produces

$$\begin{aligned} q_\alpha + \bar{q}_\alpha(s) + H_{\alpha\alpha} z_\alpha(s) &= q_\alpha + s(\bar{q}_\alpha^{(1)} + H_{\alpha\alpha} z_\alpha^{(1)}) + (1-s)(\bar{q}_\alpha^{(2)} + H_{\alpha\alpha} z_\alpha^{(2)}) \\ &= s\left(q_\alpha + \bar{q}_\alpha^{(1)} + H_{\alpha\alpha} z_\alpha^{(1)}\right) + (1-s)\left(q_\alpha + \bar{q}_\alpha^{(2)} + H_{\alpha\alpha} z_\alpha^{(2)}\right) \\ = \quad 0 \quad + \quad 0 \quad &= \quad 0, \qquad (17.13) \end{aligned}$$

and similarly,

$$\begin{aligned} q_{\bar{\alpha}} + \bar{q}_{\bar{\alpha}}(s) + H_{\bar{\alpha}\alpha} z_\alpha(s) &= q_{\bar{\alpha}} + s(\bar{q}_{\bar{\alpha}}^{(1)} + H_{\bar{\alpha}\alpha} z_\alpha^{(1)}) + (1-s)(\bar{q}_{\bar{\alpha}}^{(2)} + H_{\bar{\alpha}\alpha} z_\alpha^{(2)}) \\ &= s(q_{\bar{\alpha}} + \bar{q}_{\bar{\alpha}}^{(1)} + H_{\bar{\alpha}\alpha} z_\alpha^{(1)}) + (1-s)(q_{\bar{\alpha}} + \bar{q}_{\bar{\alpha}}^{(2)} + H_{\bar{\alpha}\alpha} z_\alpha^{(2)}) \\ &\geq 0. \qquad (17.14) \end{aligned}$$

The inequality in (17.14) follows since each of the two terms in the penultimate line are non-negative. □

We now go back to the definition of the complementarity points for a given index set α, as defined in (17.9) to prove the following:

Theorem 17.4. *The set* $\tilde{S}(H,q)$ *is a finite union of polytopes and therefore, it is a closed, bounded, and compact subset of* $\mathbb{R}^n_+$ *if the matrix* H *is positive definite.*

Proof. The set $P_\alpha(H,q) = \{\bar{q} \geq 0 \mid y(0) + \tilde{H}^{(\alpha)} \bar{q} \geq 0, \bar{q} \geq 0\}$ is a polyhedron. By the decomposition theorem of polyhedra (see [256], Corollary 7.1b), this can be written as a sum of a polytope $\mathcal{Q}_\alpha(H,q)$ and a cone $\mathcal{C}_\alpha(H,q)$, i.e., $P_\alpha(H,q) = \mathcal{Q}_\alpha(H,q) + \mathcal{C}_\alpha(H,q)$. The cone consists of a set of unbounded rays $\mathcal{C}_\alpha(H,q) =$

$\mathtt{pos}\{\tilde{\bar{q}}^{(i_1)}, \tilde{\bar{q}}^{(i_2)}, \ldots, \tilde{\bar{q}}^{(i_n)}\}$, where $\mathtt{pos}\{\cdot\}$ is the positive span of the given set of vectors. The elements of $S_\alpha(H, q)$ corresponding to the elements $\bar{q} \in P_\alpha(H, q)$ are

$$\tilde{S}_\alpha(H, q) = \{z(\bar{q}^{(1)} + \bar{q}^{(2)}) \mid \bar{q}^{(1)} \in \mathcal{Q}_\alpha(H, q), \bar{q}^{(2)} \in \mathcal{C}_\alpha(H, q)\}. \tag{17.15}$$

Now, $z(\bar{q}^{(1)} + \bar{q}^{(2)}) = z(0) + \tilde{\bar{q}}_\alpha^{(1)} + \tilde{\bar{q}}_\alpha^{(2)}$. From Theorem 17.1, if the matrix H is positive definite, the unbounded ray $\lambda\tilde{\bar{q}}^{(2)} \in \mathcal{C}$ satisfies $\lambda\tilde{\bar{q}}_\alpha^{(2)} = 0$ and therefore, we have

$$\tilde{S}_\alpha(H, q) = \{z(\bar{q}) \mid \bar{q} \in \mathcal{Q}_\alpha(H, q)\}. \tag{17.16}$$

The set $\tilde{S}_\alpha(H, q)$ is a linear mapping of a polytope, and is therefore a polytope for any index set α. Since $\tilde{S}(H, q) = \bigcup_\alpha \tilde{S}_\alpha(H, q)$ is a finite union of polytopes, it is a closed and bounded set. Since polytopes in $\mathbb{R}^n$ are compact sets, so is $\tilde{S}(H, q)$. □

Corollary 17.5. *The sequence of iterates $z^{(\nu+1)} \in \mathsf{SOL}(\bar{H}(\epsilon, \tau), q + \bar{U}(\tau)z^{(\nu)})$ is bounded and has a converging subsequence.*

Proof. The iterates all lie within the space $\tilde{S}(H(\tau, \epsilon), q)$. Since $H(\tau, \epsilon)$ is positive definite for any $\epsilon, \tau > 0$, Theorem 17.4 applies and therefore, the sequence is bounded as it is a subset of $\tilde{S}(H(\tau, \epsilon), q)$. From the Bolzano-Weierstrass theorem, the sequence must have a converging subsequence. □

17.4 Convergence rate

Assuming the index set is fixed between iterates, it is possible to estimate the norm of the difference in $z_\alpha^{(\nu)}$. Expanding the norm of the difference between iterates

$$\begin{aligned} H_{\alpha\alpha}(\tau, \epsilon) z_\alpha^{(\nu+1)} &= -q_\alpha - \bar{U}(\tau) z_\alpha^{(\nu)}, \\ H_{\alpha\alpha}(\tau, \epsilon)(z_\alpha^{(\nu+1)} - z_\alpha^{(\nu)}) &= -\bar{U}_{\alpha\alpha}(\tau)(z_\alpha^{(\nu)} - z_\alpha^{(\nu-1)}), \\ \|z_\alpha^{(\nu+1)} - z_\alpha^{(\nu)}\| &\leq \|H_{\alpha\alpha}^{-1}(\tau, \epsilon)\bar{U}_{\alpha\alpha}(\tau)\| z_\alpha^{(\nu)} - z_\alpha^{(\nu-1)}\|. \end{aligned} \tag{17.17}$$

And therefore, the convergence rate depends on the spectral radius of the iteration matrix $H_{\alpha\alpha}^{-1}(\tau, \epsilon)\bar{U}_{\alpha\alpha}(\tau)$. Unfortunately, little can be said about that. One might expect that when the friction coefficients—the nonzero elements of matrix U—are small enough, the mapping should be a contraction.

17.5 Application to box friction

Box friction is a simple approximation to dry friction. At each contact point i, we have an operator $D^{(i)}$ which projects the velocity vector of the system of rigid bodies v, into the plane tangent to the contact normal, i.e., $D^{(i)}v = v_t^{(i)}$, where v_t is the velocity in the tangent plane. For box friction, we choose $D^{(i)}$ to have two rows only which are perpendicular to each other. In the case of

static friction where there is no sliding, the constraint reads $D^{(i)}v = 0$. This leads to Lagrange multipliers $\beta^{(i)} \in \mathbb{R}^2$ and a tangential contact force given by $F_t^{(i)} = D^{(i)T}\beta^{(i)}$. If the Lagrange multipliers $\beta^{(i)}$ are allowed to take on any value in $\mathbb{R}^n$, the constraint $D^{(i)}v = 0$ will be satisfied. However, if we restrict these multipliers to lie within the simple box: $l^{(i)} \le \beta^{(i)} \le u^{(i)}$, then, the constraint will not be satisfied and we have residuals $D^{(i)}v = \sigma_+^{(i)} - \sigma_-^{(i)}, \sigma_+^{(i)}$, where $\sigma_-^{(i)} \ge 0$, and $\sigma_+^{(i)}, \sigma_-^{(i)}$, are the positive and negative components of the *slip* velocity, respectively.

These equations and conditions can be written as follows

$$\begin{aligned} D^{(i)}v - \sigma_+^{(i)} - \sigma_-^{(i)} &= 0 \\ 0 \le \beta^{(i)} - l^{(i)} \perp \sigma_+^{(i)} &\ge 0 \\ 0 \le -\beta^{(i)} + u^{(i)} \perp \sigma_-^{(i)}&. \end{aligned} \tag{17.18}$$

Introducing the matrices $\tilde{E}$ with block form $\tilde{E}^{(i)} = [1, -1]^T$, and agglomerated in the block diagonal $\tilde{E} = \mathsf{diag}(\tilde{E}^{(1)}, \tilde{E}^{(2)}, \ldots, \tilde{E}^{(m)})$, and $\tilde{F}^{(i)} = \mu^{(i)}[1, 1]^T$, $\tilde{F} = \mathsf{diag}(\tilde{F}^{(1)}, \tilde{F}^{(2)}, \ldots, \tilde{F}^{(m)})$ where m is the number of contact points, and

$$F = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \tilde{F} & 0 & 0 \end{bmatrix}, \tag{17.19}$$

which is a matrix with nonnegative entries. We now have the system matrix

$$H = \begin{bmatrix} Q_1 & B^T & 0 \\ B & Q_2 & -\tilde{E}^T \\ 0 & \tilde{E} & 0 \end{bmatrix} \tag{17.20}$$

and the corresponding mixed LCP is

$$\begin{aligned} \begin{bmatrix} Q_1 & B^T & 0 \\ B & Q_2 & -\tilde{E}^T \\ 0 & \tilde{E} & 0 \end{bmatrix} \begin{bmatrix} \nu \\ \beta \\ \sigma \end{bmatrix} + \begin{bmatrix} q_\nu \\ q_\beta \\ q_\sigma \end{bmatrix} &= \begin{bmatrix} \delta \\ 0 \\ \gamma \end{bmatrix} \\ 0 \le \begin{bmatrix} \nu \\ \sigma \end{bmatrix} \perp \begin{bmatrix} \delta \\ \gamma \end{bmatrix} &\ge 0, \end{aligned} \tag{17.21}$$

where $q_\sigma = [-l^{(1)}, u^{(1)}, -l^{(2)}, u^{(2)}, \ldots, -l^{(m)}, u^{(m)}]^T$. In the case of symmetric bounds, we have $-l^{(i)} = u^{(i)} \ge 0$.

Taking the Schur complement of the variables β, we have the following transformed matrix, q and z vectors

$$\tilde{H} = \begin{bmatrix} Q_1 - B^T Q_2^{-1} B & B^T Q_2^{-1} \tilde{E}^T \\ -\tilde{E} Q_2^{-1} B & \tilde{E} Q_2^{-1} \tilde{E}^T \end{bmatrix}, \qquad \tilde{q} = \begin{bmatrix} q_\nu - B Q_2^{-1} q_\beta \\ q_\sigma - \tilde{E} Q_2^{-1} q_\beta \end{bmatrix}, \qquad \tilde{z} = \begin{bmatrix} \nu \\ \sigma \end{bmatrix}, \tag{17.22}$$

Figure 17.1: The operator splitting scheme applied to the inclined plane problem.

which lead to the LCP

$$\begin{aligned} \tilde{H}\tilde{z} + \tilde{q} &= \tilde{w} \\ 0 \leq \tilde{z} &\perp \tilde{w} \geq 0. \end{aligned} \tag{17.23}$$

In this case, the sequential scheme case updates the lower and upper bounds, $l^{(i)}, u^{(i)}$ such that

$$l^{(i)} = -\mu^{(i)}\nu^{(i)} \qquad u^{(i)} = \mu^{(i)}\nu^{(i)} \tag{17.24}$$

and this is written as $q_\sigma = F_{\sigma\nu}z_\nu$ and we therefore have the sequence

$$z^{s+1} \in \mathrm{SOL}(\tilde{H}(\epsilon), q + Fz^{(s)}). \tag{17.25}$$

Since the matrix $\tilde{H}(\epsilon)$ is positive definite for $\epsilon > 0$, the results of Theorem 17.4 apply and we have a converging subsequence.

17.6 Numerical experiments

A simple test case is to let a rectangular box slide down an inclined plane. For typical geometric interference libraries, as described in [85] for instance, this leads to four contact points, and that is a degenerate problem. What is presented in Figure 17.1 is the effective measured friction coefficient as a function of the tangent of the inclination angle $\tan(\theta)$. The expected curve is a ramp up to the point where $\tan(\theta) = \mu$, where μ is the (static) friction coefficient (no distinction is made here between static and kinetic friction coefficients). For angles θ such that $\tan(\theta) > \mu$, the line should lie flat at $\mu_{\text{eff}} = \mu$. However, it does matter how the iteration sequence is started. Indeed, if the first solve assumes $\mu = 0$, the sequence converges. Assuming $\mu = \infty$ in the first pass however, an increasing number of iteration is needed as the inclined plane is angled toward the vertical. Much more investigation is needed to improve the convergence rate and stabilization of the splitting scheme for friction.

Simulation data is shown Figure 17.2. This corresponds to applying the splitting technique in combination with the block pivot mixed linear complementarity problem (MLCP) solver of Section 16.7.1 with protection against cycling based on hash tables and restart. The example consists of 40 identical cylinders confined within four posts. The cylinders are dropped from a moderate height and are originally separated. They stack as they fall under gravity. This sample illustrates that convergence is not guaranteed and that it depends more on factors other than the condition number. What is most likely for this example is that contact configurations become infeasible due to early approximation errors. Investigations on large realistic data sets are part of planned future work.

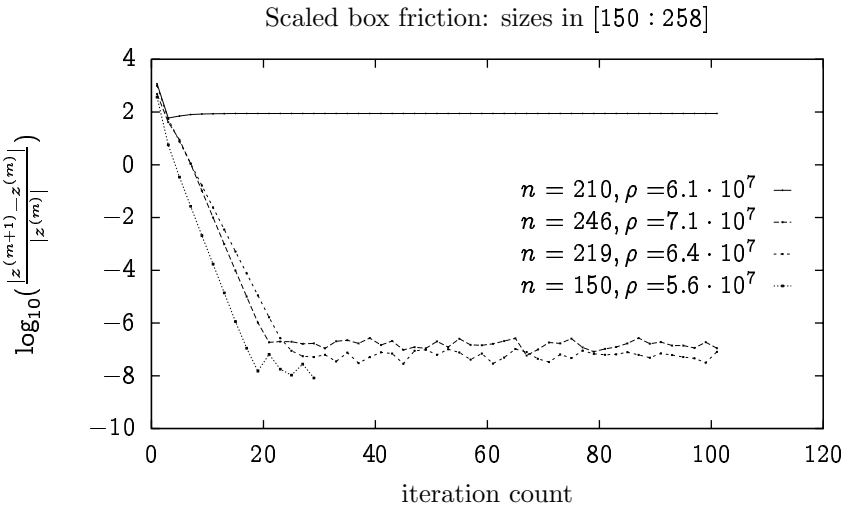

Figure 17.2: Convergence of the splitting scheme to solve dry frictional contact problems for stacking problem involving 40 identical cylinders dropped on top of each other in a pile. Convergence is linear in general but can stagnate with a large residual error.

17.7 End notes

Splitting schemes are common. Lötstedt [191] immediately tried that on his friction formulation, deciding it was better to solve a sequence of QPs than an LCP with a non-symmetric matrix. The splitting discussed in this section is applied in one form or another in the engineering literature [77, 142, 238, 2, 273, 191] to address all sorts for frictional contact problems. But as shown in Section 17.3 and Section 17.4, the sequence of iterates for this splitting is closed and bounded but not necessarily convergent, and so this is not a robust method. In addition, the convergence is at most linear, like a fixed point iteration, which is not very good. What is clear though is that the regularization parameter does improve the situation. This was seen also in an existence proof for quasistatic frictional contacts by Klarbring and Pang [160], for instance. Of course, diagonal perturbations are nothing new but in the formulation of Chapter 4 and Chapter 10, it does correspond to physical models.

A high performance solver for contact problem needs to use some form of splitting for parallelization and so more remains to be done to pick a converging subsequence among the iterates.

18 Solving the Linear Problems

Linear algebra is at the heart of any computational task. This is so important in fact that computer designs are driven by linear algebra benchmarks and no effort is spared to make sure that dense matrix operations, such as the GEMM, are available, fast and efficient, for any given computer platform. This is good news for a developer but there is a rub. Computer designs have become increasingly complicated due to differences in performance characteristics of different subunits such as memory banks and CPUs, to name just two. These discrepancies mean that certain types of operations performed on certain types of data organized in a specific way in memory are *very* fast, but others are orders of magnitude slower. For this reason, great care must be taken in organizing data in a computer program and making sure that those algorithms which are fast on current CPUs can be used on the problems considered.

This motivates the present chapter which looks at the logical algorithms presented in previous chapters from an implementation viewpoint, namely, which data, what format, what operations, how. Section 18 provides a short background and Section 18.2 details the types of operation needed and the corresponding types of format best supporting these. Section 18.3 provides a review of known identities of saddle point problems in order to describe a known sparse factorization strategy in Section 18.4, which works as long as the graph of the matrix is a *rooted tree* without any *closed loops*. Gauss-Seidel iterations are then discussed in Section 18.5 with special emphasis on the matrix structure found in multibody problems. A conjugate gradient method adapted for the regularized problems described in Section 4.5 and using the sparse factorization of Section 18.4 is found in Section 18.6. Simple techniques for factorization updates and down-dates, necessary operations for the implementation of complementarity solvers, are provided in Section 18.7 and this is followed by general observations in Section 18.8.

18.1 Introduction

Numerical linear algebra—or matrix computations—plays a central role in all computations involving strongly coupled components. Performance and accuracy of nontrivial computations are intimately connected to performance and accuracy of the underlying numerical linear algebra algorithms and implementations. This is why an entire chapter is devoted to this topic in the specific context of multibody simulations.

18.2 Special matrices, formats, and operations

The matrices involved in multibody dynamics include the mass matrix M, which is a square $n \times n$ real, symmetric, and positive definite, and the Jacobian matrix G which is rectangular $m \times n$ real matrix, usually with $m \leq n$.

Assumption is made explicitly that matrix M is block diagonal. In fact, the point was made several times in Chapters 3, 4, and 10 that it is preferable to work with the large sparse representations of the augmented system matrix, by contrast with *reduced coordinate* techniques where the effective mass matrix $\tilde{M}$ is made as small as possible and usually very dense and ill-conditioned as well [28]. Therefore, matrix M is assumed to have the block diagonal format

$$M = \mathrm{diag}(M^{(1)}, M^{(2)}, \ldots, M^{(n_b)}), \tag{18.1}$$

and n_b is the number of bodies in the system. It does help to think that the dimensions of each block matrix $M^{(k)}$ are identical but this is not necessary and will not be explicitly assumed in what follows. In general, the blocks $M^{(k)}$ have $n_v^{(k)} \times n_v^{(k)}$ with $\sum_{k=1}^{n_b} n_v^{(k)} = n$. All body based vectors in the system, in particular, the velocity vector $v \in \mathbb{R}^n$ have a conforming partitioning

$$v^T = \left[v^{(1)^T}, v^{(2)^T}, \ldots, v^{(n_b)^T} \right], \tag{18.2}$$

where each block has dimensions n_k, i.e., $v^{(k)} \in \mathbb{R}^{n_v^{(k)}}$. It is be possible to use, say, a quaternion representation for the orientation of a rigid body so that the position vector is $q \in \mathbb{R}^4$, but then, use the angular velocity vector $\omega = \mathcal{E}(q)\dot{q} \in \mathbb{R}^3$, with matrix $\mathcal{E}(q)$ defined as in (13.64), so the dimensions of position and velocity vectors may not match but the grouping is identical. This sort of details is relegated to technicalities of implementations.

In general, we call a *body vector* any vector which has one block entry per body, i.e., any vector that allows the same block partition as the velocity vector.

Note also that most operations involving the mass matrix actually involve its inverse M^{-1}. Given the presumed block diagonal format and the fact that, at least for point particles and rigid bodies, the inverse is easily computed, we arrive at a representation for the inverse mass matrix which involves the elements displayed in Table 18.1.

With this format, it is possible to pack all the mass matrix blocks into a contiguous array of size $\sum_k n_v^{(k)} \cdot n_v^{(k)}$. One can also exploit symmetry properties as well to reduce storage if needed.

Operations involving the mass matrix include the matrix-vector product

$$x \leftarrow \alpha M y + \beta x, \tag{18.3}$$

where $\alpha, \beta \in \mathbb{R}$, and both y and x are real, body vectors. Assuming that both y and x can be traversed sequentially, the only operation needed is the partitioned one

$$x^{(k)} \leftarrow \alpha M^{(k)} y^{(k)} + \beta x^{(k)}, k = 1, 2, \ldots, n_b, \tag{18.4}$$

Description	Symbol	Size of data set
Total dimension of mass matrix	n	single entry
Total number of bodies in this subsystem	n_b	single entry
Dimensionality of velocity vector for each body	$n_v^{(k)}$	one for each body $k = 1, 2, \ldots, n_b$, and $\sum_{k=1}^{n_b} n_v^{(k)} = n$
Block matrix data	$M^{(k)}$	one for each body $k = 1, 2, \ldots, n_b$

Table 18.1: Requirements and dimensions for the system mass matrix.

which is easily done optimally.

There is also a matrix-matrix product operation of the form

$$K \leftarrow \alpha GM + \beta K, \tag{18.5}$$

which is described below once the Jacobian matrix storage format is fully specified.

Jacobian matrices are agglomerated from constraint conditions which are usually grouped. For instance, a ball joint between two bodies introduces three constraint equations which have the same block pattern, whilst a hinge joint introduces five equations. This level of blocking appears as

$$G^T = \begin{bmatrix} G^{(1)^T} & G^{(2)^T} & \ldots & G^{(n_c)^T} \end{bmatrix}, \tag{18.6}$$

where n_c is the number of constraint block, and where each block row $G^{(i)}$ is of size $m_i \times n$, with $\sum_i m_i = m$. The integer n_c is the number of constraints being considered. Each block row $G^{(i)}$ contains only a few non-zero column blocks with indices, $k_1^{(i)}, k_2^{(i)}, \ldots, k_{n_b^{(i)}}^{(i)}$, where $n_b^{(i)}$ is usually 1 or 2. This is the case for most of the common mechanical constraints such as hinge, prismatic joints, and contact constraints. In this format, each column block corresponds to a single physical body, with the same labeling used for the mass matrix.

To parallel the concept of body vectors, we introduce *constraint vectors* which are real vectors, $y \in \mathbb{R}^m$, with the following partitioning

$$y^T = \begin{bmatrix} y^{(1)^T} & y^{(2)^T} & \ldots & y^{(n_c)^T} \end{bmatrix}, y^{(i)} \in \mathbb{R}^{n_c^{(i)}}, \tag{18.7}$$

and we note that these vectors include the constraint velocity, $G\dot{q}$ and the Lagrange multipliers λ among other examples. So now, for each constraint index i, there is a corresponding block $y^{(i)}$ for any constraint vector. In order to randomly access a given constraint vector y, one needs a map between each constraint index i and the address of the first element of $y^{(i)}$.

Jacobians can be decomposed one level further by introducing a *Jacobian block,* $G^{(jk)}$, which is a matrix of dimensions $n_c^{(j)} \times n_v^{(k)}$ associated to one and only

Description	Symbol	Size of data set
Total number of constraints in the subsystem	n_c	single entry
Total number of nonzero Jacobian blocks in the subsystem	n_k	single entry: $n_k = \sum_{j=1}^{n_c} n_b^{(j)}$
Dimensionality (number of rows) for each constraint	$n_c^{(k)}$	one for each constraint $k = 1, 2, \ldots, n_c$ and $\sum_{j=1}^{n_c} n_c^{(j)} = m$
Number of bodies attached to a constraint	$n_b^{(k)}$	one for each constraint $k = 1, 2, \ldots, n_c$
List of bodies attached to each constraint	$l_b^{(k)}$	one for each constraint $k = 1, 2, \ldots, n_c$
List of constraints attached to each body	$l_c^{(j)}$	one for each body $j = 1, 2, \ldots, n_b$
List of constraint blocks attached to each body	$s_c^{(j)}$	one for each body $j = 1, 2, \ldots, n_b$
List of constraint blocks attached to each constraint	$s_b^{(k)}$	one for each constraint $k = 1, 2, \ldots, n_c$
Block Jacobian data	$G^{(i,j)}$	n_k blocks, one for each body (global index j) in each constraint i: each block (i, j) has dimensions $n_c^{(i)} \times n_v^{(j)}$.

Table 18.2: Requirements and dimensions for Jacobian matrices

.

one constraint, j, and one and only one body k. There are n_k such blocks where $n_k = \sum_{j=1}^{n_c} n_b^{(j)}$. These blocks can be stored contiguously in a packed format and to do this, one needs the data described in Table 18.2.

Jacobian matrices define two types of matrix vector products, mapping body vectors and constraint vectors to each other as follows

$$\begin{aligned} y &\leftarrow \alpha G x + \beta y, \quad \text{body to constraint} \\ x &\leftarrow \alpha G^T y + \beta x, \quad \text{constraint to body} \end{aligned} \tag{18.8}$$

where x is a constraint vector, y is a body vector, and $\alpha, \beta \in \mathbb{R}$ are real scalars.

Depending on the data layout and organization, each of these two operations is either a GATHER or a SCATTER type, i.e., the process involves collecting data from different memory locations to update an array sub element—GATHER—, or, conversely, an array sub element is used to compute updates to different memory locations—SCATTER.

There are two types of traversals depending on whether the constraint vector or the body vector is traversed sequentially and these are now described.

Description	Symbol	Size of data set
List of bodies attached to given constraint	$l_b^{(k)}$	one per constraint $k = 1, 2, \ldots, n_c$
List of constraints attached to given body	$l_c^{(j)}$	one per body $j = 1, 2, \ldots, n_b$
Body vector random access index	$\star x^{(j)}$	one table for the subsystem with one entry per body index $j = 1, 2, \ldots, n_b$
Constraint vector random access index	$\star y^{(k)}$	one table for the subsystem with one entry per constraint index $k = 1, 2, \ldots, n_c$

Table 18.3: Minimal requirements for gather and scatter operations

Constraints traversal

If the constraint vector is accessed sequentially and the body vector storage allows random access, then, for each constraint index i, we need to retrieve the b_i indices of the bodies involved, i.e., $b_k^{(i)}, k = 1, 2, \ldots, b_i$ and from this, look up the relevant component $x^{(b)}$ and compute the appropriate block matrix-vector product. This makes the operation $y \leftarrow \alpha Gx + \beta y$ a GATHER type and $x \leftarrow \alpha G^T y + \beta x$ a SCATTER type.

Bodies traversal

Conversely, if body vectors are stored and accessed sequentially and constraint vectors allow random access, then, for each body index b, we need to retrieve the c_b indices of the constraints attached to body b, $c_l^{(b)}, l = 1, 2, \ldots c_b$. For each of these $c \in \{c_l^{(b)} \mid l = 1, 2, \ldots c_b\}$, we identify the corresponding component of the constraint vector $y^{(c)}$ and perform the appropriate matrix-vector update. This makes the operation $y \leftarrow \alpha Gx + \beta y$ a SCATTER type and $x \leftarrow \alpha G^T y + \beta x$ a GATHER type.

The list of requirements for implementing these operations is summarized in Table 18.3 below.

Matrix matrix operations

In terms of matrix-matrix operations, there are three cases involving mass-like matrix M and Jacobian-like matrices G, K, with identical structure

$$\begin{aligned} K &\leftarrow \alpha GM + \beta K \\ S &\leftarrow \alpha KG^T + \beta S \\ T &\leftarrow \alpha K^T G + \beta T. \end{aligned} \tag{18.9}$$

The first of these is very simple to implement since all is needed is to traverse each block $G^{(c,b)}$ of the matrix G sequentially, look up corresponding matrix block $M^{(b)}$, and perform the elementary product operation

$$G^{(c,b)}M^{(b)} \longleftarrow G^{(c,b)}M^{(b)}. \tag{18.10}$$

The best way to implement this is to allocate enough space for two copies of the data contained in G, and store both GM and G.

The second operation described in (18.9) is the construction of the Schur complement S. There is a simple block algorithm for computing this which is optimal and which we now describe.

First observe that matrix S maps constraint vectors to constraint vectors and therefore, it allows a partitioning into blocks $S^{(i,j)}$ of size $m^{(i)} \times m^{(j)}$, where both indices c, d range from $1, 2, \ldots, n_c$. Now, it is clear from the rules of matrix multiplication that

$$S^{(i,j)} = \sum_{k=1}^{n_b} K^{(i,k)} G^{(j,k)^T}, \tag{18.11}$$

where the indices have been permuted in the block representation of G^T. Now, obviously, $G^{(jk)}$ is a zero block if body with index k is not found in constraint j, and likewise with $K^{(i,k)}$. This means that the sum in (18.11) is limited to $k \in \{l_b^{(i)} \mid l = 1, 2, \ldots, n_b^{(i)}\}$. In other works, for most cases, we have only one or two bodies to consider. But note also that if we fix constraint i and body index k, the only non-zero terms in (18.11) come from constraints with index j which are linked to body index k.

Now, therefore, if there is a list $l_b^{(i)}$ for each constraint i containing the indices of the bodies it refers to, as well as a list $l_c^{(k)}$ for each body containing the indices of the constraints it is related to, we can perform the multiplication as described in Algorithm 18.2.1. This is not particularly efficient because of the repeated search operation in the inner loop.

Algorithm 18.2.1 List search based matrix-matrix multiply algorithm

```
initialize all S blocks
for i = 1, 2, ..., n_c do                                  ▷ build S row-wise
    for k = head(l_b^(i)), ..., tail(l^(i)b) do            ▷ Loop over related columns
        for j = head(l_c^(k)), ..., tail(l_c^(k)) do
            Find block G^(i,k) in constraint i
            Update : S^(i,j) ← αK^(i,k) G^(j,k)^T + βS^(i,j)
        end for
    end for
end for
```

One way to avoid that is to build a sorted list of relative blocks for each rigid bodies. This is easily done when the Jacobian blocks are computed with $O(1)$ complexity. The result is the array of sorted lists, $s_c^{(k)}$, containing the

addresses of the blocks $G^{(j,k)}$ in increasing order of constraint index j. The Schur complement algorithm of Algorithm 18.2.2 is now optimal in the sense that no search operation is performed any more, only direct list traversal and kernel computations. The computational cost can be reduced further by changing the inner for-loop to only consider the lower triangle, i.e., only considering indices $u \leq t$.

Algorithm 18.2.2 Block list matrix-matrix multiply algorithm

initialize all S blocks
for $t = 1, 2, \ldots, n_k$ **do** ▷ build S row-wise
  Look up constraint index i for block t
  Look up body index k for block t
  for $u = \text{head}(s_c^{(k)}), \ldots, \text{tail}(s_c^{(k)})$ **do**
    Look up constraint index j for block u
    Update : $S^{(i,j)} \leftarrow \alpha K^{(i,k)} G^{(j,k)^T} + \beta S^{(i,k)}$
  end for
end for

To construct as truly sparse representation of the Schur complement matrix, one must also be careful not to touch unnecessary memory. This is easily done by keeping track of which pair of indices (c, d) has been looked at so far, and clearing block $S^{(i,j)}$ only the first time it is visited.

Of course, eliminating search operations is not enough to make a matrix-matrix multiply algorithm optimal and there is much that could be done to the data layout to optimize performance. However, as it is, the complexity of Algorithm 18.2.2 is $O(n_c \bar{n}_b \bar{n}_c)$, where n_c is the number of constraints, and $\bar{n}_b$ is the average number of bodies per constraint, and $\bar{n}_c$ is the average number of constraints per body. To see this, note that the first loop goes once over each block and the number of blocks is in fact $n_c \bar{n}_b$. Then, for each block, we need to go over the related blocks. Assume that a given block is related to body with index k. There is one and only one related block for each other constraint that body k is involved with. Therefore, the total number of operations is

$$ops = \sum_{k=1}^{n_k} n_c^{(k)} = n_c \bar{n}_b \bar{n}_c, \tag{18.12}$$

where n_k is the total number of blocks. This complexity analysis should dispel the widely spread myth that direct matrix methods for multibody systems have inherent complexity of $O(n^3)$ or $O(m^3)$. However, as we shall see below, the worst case complexity of computing the Schur complement S is $O(m^2)$, unfortunately.

The last form of matrix-matrix product in (18.9) is the inside out version of Algorithm 18.2.2 though no search is necessary. Indeed, the blocks in matrix $T^{(b,a)}$, where b, a are body indices, are computed as

$$T^{(i,j)} = \sum_{k=1}^{n_c} K^{(k,i)^T} G^{(k,j)}. \tag{18.13}$$

This means that the blocks $K^{(k,i)}$ and $G^{(k,j)}$ both come from a single constraint and that we can avoid all search operations by looping over the constraints and then looping over bodies. All we need to assume here is that each constraint j has a sorted list of blocks $s_b^{(j)}$. This leads to Algorithm 18.2.3. Again, the exact sequence of this looping might not be optimal from a performance viewpoint but this is the minimum operation count ignoring memory look ups.

Algorithm 18.2.3 Block matrix-matrix multiply algorithm

```
initialize all T blocks
for t = 1, 2, ..., n_k do                    ▷ Loop directly over the blocks
    Look up body index i of Jacobian block t
    Look up constraint index k of Jacobian block t
    for s = head(s_c^(i)), ..., tail(s_c^(i)) do
        Look up body index j of Jacobian block s
        Update : T^(i,j) ← α K^(k,i)^T G^(k,j) + β T^(i,j)
    end for
end for
```

A good library will provide implementations of these operations, abstracting the data format as much as possible so that it is possible to test alternative implementations and chose those formats offering best performance.

With the formats provided, all the matrix-matrix multiply algorithm can make use of the GEMM-based approach of Kågström, Ling and van Loan [149] to deliver high performance, both during matrix construction and subsequently, during matrix factorization.

18.3 Saddle point identities

The saddle point problem of optimization translates to the linear algebra problems of a special type, namely, involving a matrix of the block form

$$K = \begin{bmatrix} M & -G^T \\ G & C \end{bmatrix}, \tag{18.14}$$

where sub-matrix M is a square $n \times n$ symmetric and positive definite, matrix G is $m \times n$, usually with $m < n$, and matrix C is square $m \times m$ matrix which is symmetric and positive semidefinite. A matrix with the form given in (18.14) is called *bisymmetric*. Dollar, Gould, Shilders and Wathen [74, 75] recently published a comprehensive review of techniques applying to the present problem. A review of methods is found also in Benzi, Golub, and Liesen [48]. The present section only covers the basic techniques.

Consider the two vectors $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$. It is easy to verify that

$$\begin{bmatrix} x^T & y^T \end{bmatrix} \begin{bmatrix} M & -G^T \\ G & C \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = x^T M x + y^T C y \geq y^T C y, \tag{18.15}$$

since we assumed that M was positive definite. Therefore, K is positive definite or semidefinite according to the whether C is positive definite or semi-definite. It is interesting to note that K is not symmetric.

Note also that

$$\begin{bmatrix} M & -G^T \\ G & C \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} M & G^T \\ G & -C \end{bmatrix} \begin{bmatrix} x \\ -y \end{bmatrix}, \tag{18.16}$$

which means that we can use a factorization of the symmetric matrix

$$H = \begin{bmatrix} M & G^T \\ G & -C \end{bmatrix}, \tag{18.17}$$

to solve linear problems of the form $Kz = b$.

Saddle point matrices K satisfy the following identities as is well known (see for instance [52], Chapter III §4 and exercises therein). In what follows, L is a lower triangular matrix with unit diagonal, U is an upper triangular matrix with unit diagonal, and D is a block diagonal matrix. The following two block factorizations are readily verified

$$\begin{aligned} K = L\bar{D}L^{-T} &= \begin{bmatrix} I & 0 \\ GM^{-1} & I \end{bmatrix} \begin{bmatrix} M & 0 \\ 0 & GM^{-1}G^T + C \end{bmatrix} \begin{bmatrix} I & -M^{-1}G^T \\ 0 & I \end{bmatrix} \\ H = LDL^T &= \begin{bmatrix} I & 0 \\ GM^{-1} & I \end{bmatrix} \begin{bmatrix} M & 0 \\ 0 & -GM^{-1}G^T - C \end{bmatrix} \begin{bmatrix} I & M^{-1}G^T \\ 0 & I \end{bmatrix}, \end{aligned} \tag{18.18}$$

using the identity

$$L = \begin{bmatrix} I & 0 \\ GM^{-1} & I \end{bmatrix} \Longrightarrow L^{-1} = \begin{bmatrix} I & 0 \\ -GM^{-1} & I \end{bmatrix} = 2I - L, \tag{18.19}$$

which is easily verified. With these identities in mind, it should be clear that the Schur complement matrix

$$S = GM^{-1}G^T + C, \tag{18.20}$$

plays a central role in the determination of the spectrum and factorization of K. In fact, assuming that the inverse M^{-1} is easy to compute, a factorization of matrix S allows the solution of the linear problem $Kz = b$ using only straight forward matrix-vector operations.

Note also that the spectrum of matrix D is the union of the spectrum of matrices M and S, whereas that of matrix $\bar{D}$ is the union of the spectrum of matrix M and the *negative* spectrum of S. Now, since matrix S is symmetric and at least positive semi-definite, the spectrum of H contains both positive and negative eigenvalues and, therefore, H is indefinite.

Since all block matrices in (18.18) are easily inverted, explicit formulae for the inverses are found to be

$$\begin{aligned} K^{-1} &= L^T \bar{D}^{-1} L^{-1} = \begin{bmatrix} M^{-1} - M^{-1} G^T S^{-1} G M^{-1} & M^{-1} G^T S^{-1} \\ -S^{-1} G M^{-1} & S^{-1} \end{bmatrix} \\ H^{-1} &= L^{-T} D^{-1} L^{-1} = \begin{bmatrix} M^{-1} - M^{-1} G^T S^{-1} G M^{-1} & M^{-1} G^T S^{-1} \\ S^{-1} G M^{-1} & -S^{-1} \end{bmatrix}. \end{aligned} \tag{18.21}$$

These formulae illustrate once more the crucial role played by the Schur complement matrix S. Direct factorization of saddle point matrices, especially in the symmetric indefinite form H has been the subject of much research and software packages such as the MA27 and MA57 codes [79, 80, 78], among others. Iterative techniques with preconditioners are found in Dollar, Gould, Schilders, and Wathen [74], which contains a description of Algorithm 18.6.2, in fact.

There are alternatives to factoring matrix H or K directly, exploiting the overall optimization superstructure but treating matrix G as dense. The monograph of von Schwerin [272] details the list of options. However, the alternative choices involve QR factorization of the Jacobian matrix G as well as *generally dense* matrix-matrix products and that therefore, these choices are not appealing, especially in view of the fact that there are few, if any, sparse QR factorization libraries available at this time [197, 216].

18.4 Sparse factorization

As observed in Section 18.2, the mass matrix M and Jacobian matrix G are both very sparse. Altogether, we expect that the rows of matrix K contain no more than 10 non-zero block in the upper half, no more than 3 blocks on average in the bottom section. More precisely, assuming the average number of bodies per constraint is $\bar{n}_b$, then, the fill fraction of matrix K is

$$\text{fill}(K) = \text{fill}(H) = \frac{n + 2m\bar{n}_b}{(n+m)^2}. \tag{18.22}$$

It is expected that a high performance sparse library designed for handling indefinite matrices, symmetric or otherwise, such as MA27 [80] [79], or SuperLU [72], or the more recent MA57,DUFF:2004:MCS, or even UMFPACK [71], can deliver high performance here. This has not yet been benchmarked accurately, perhaps due to the fact that most interesting large scale multibody problems involve contacts and friction, not just equality constraints. Notably though, UMFPACK does provide factorization update and down-date operations, but not the ones needed to solve LCPs or QPs. The day might come yet. In the meantime, here is how to grow your own.

Given that writing a multifrontal algorithm for processing saddle point matrix H is far from easy, we describe the simplest case of a sparse factorization based

on topological consideration. To do this, we first note the recurrence formula

$$\begin{aligned} H = \begin{bmatrix} H_{11} & H_{21}^T \\ H_{21} & H_{22} \end{bmatrix} &= \begin{bmatrix} I & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} D_{11} & 0 \\ 0 & D_{22} \end{bmatrix} \begin{bmatrix} I & L_{21}^T \\ 0 & L_{22}^T \end{bmatrix} \\ &= \begin{bmatrix} D_{11} & D_{11} L_{21}^T \\ L_{21} D_{11} & L_{21} D_{11} L_{21}^T + L_{22} D_{22} L_{22}^T \end{bmatrix}, \end{aligned} \tag{18.23}$$

which can be used to define the following transformations

$$\begin{aligned} \tilde{H}_{11} &\leftarrow H_{11} = D_{11}, \\ \tilde{H}_{21} &\leftarrow L_{21} = H_{21} D_{11}^{-1}, \\ \tilde{H}_{22} &\leftarrow H_{22} - \tilde{H}_{21} \tilde{H}_{11} \tilde{H}_{21}^T = H_{22} - H_{21} H_{11}^{-1} H_{21}^T, \end{aligned} \tag{18.24}$$

which, applied recursively and in place on matrix H, yields the factors L and D. It is a good idea to store a copy of the date before starting the factorization.

As easily seen from (18.24), neither of the first two transforms $\tilde{H}_{11}$ or $\tilde{H}_{21}$ can introduce new non-zero elements, but the last transform for $\tilde{H}_{22}$ can. Specifically, the term $\tilde{H}_{21} \tilde{H}_{11} \tilde{H}_{21}^T$ has the potential to add new non-zero blocks. However, if the block column H_{21} has only one non-zero block, say H_{k1} then we only have diagonal updates and there are no fill-ins during factorization as we apply (18.24) recursively to complete the computations of the factors L and D.

To analyze this structure, define a graph relationship in matrix H such that on any block column i, the set of indices k of the non-zero block H_{ki} above the diagonal—those with $1 \leq k < i$—are called the *children* of block H_{ii}, i.e., $\{k \mid k < i \text{ and } H_{ki} \neq 0\} = \text{children}(i)$. This relationship is enough to build a graph where the nodes are the block matrices H_{ki} and the edges link the diagonal nodes H_{ii} with their children, H_{ki} such that $k \in \text{children}(k)$. For a symmetric matrix, we have $H_{ik} = H_{ki}^T$ and therefore, if there is a non-zero element $H_{ki} \neq 0$ below the diagonal, i.e., with $k > i$, then, $i \in \text{children}(k)$. Therefore, the case where there is only one such element leads to a graph which is a *rooted tree*, i.e., a graph in which each node has one and only one direct ancestor. This is precisely what is needed to factor matrix $H = LDL^T$ without introducing any fill-ins. This leads to computational work that is directly proportional to the number of non-zero blocks. Such a matrix H with n diagonal blocks whose graph is a rooted tree can have at most $n - 1$ non-zero off-diagonal elements, one in each column.

Given a matrix H, we can first compute the graph by listing the children of every node. If we find that this graph is a rooted tree, we can then relabel the elements so that each node has an index which is less than that of its parent. To do this, we start with an arbitrary node i and relabel it as n. Then, the n_1 children are re-labeled as $n - 1, n - 2, \ldots, n - n_1$. The same process is applied recursively to each child $k \in \{n - 1, n - 2, \ldots, n - n_1\}$. This reordering defines a permutation which can be applied to matrix H after which we can apply formulae (18.24) recursively to obtain the factors without any fill-ins.

Storage can be saved by working in place, directly on the blocks H_{ik} though. Looking at (18.24), it is easy to see that each off-diagonal element $H_{ki}, k > i$ is touched once only, i.e., for each node k, the off-diagonal element $H_{p(k)k} \leftarrow H_{p(k)k}H_{kk}^{-1}$, where $p(k) = \text{parent}(k)$. A diagonal block H_{ii} will be updated once for each of the indices $k \in \text{children}(i)$ as

$$H_{ii} \leftarrow H_{ii} - \sum_{k \in \text{children}(i)} H_{ki}H_{kk}^{-1}H_{ki}^T = H_{ii} - \sum_{k \in \text{children}(i)} \tilde{H}_{ki}\tilde{H}_{kk}\tilde{H}_{ki}^T. \quad (18.25)$$

Obviously, if a child is always processed before it's parent, then, we can use the second equality of the last equation and we only have to work with the updated data. This leads to Algorithm 18.4.1 below.

Algorithm 18.4.1 LDL^T factorization of symmetric matrix H whose graph is a rooted tree and which is stored in variable size blocks H_{ki} for $k \geq i$ (lower triangle). The elements of the tree are labeled so the index of the root is n and for each $k \in \{1, 2, \ldots, n\}$, $k \in \text{children}(i) \Rightarrow k < i$, and $i = \text{parent}(k) = p(k)$.

▷ Walk up the tree starting from the leaf nodes.

for $i = 1, \ldots, n$ **do**
  for $k \in \text{child}(i)$ **do**
    $H_{ii} \leftarrow H_{ii} - H_{ik}H_{kk}H_{ik}^T$
  end for
  $H_{p(i)i} \leftarrow H_{p(i)i}H_{ii}^{-1}$
end for

Algorithms for solving the triangular systems $Lx = b$ and $L^Ty = c$ are easily derived by using the same recursion techniques and they are shown below in Algorithm 18.4.2.

Algorithm 18.4.2 Solution of $Hx = b$ for symmetric block matrix H whose graph is a rooted tree, using factorization $H = LDL^T$ as per Algorithm 18.4.1.

Given matrix $H = H^T$ whose graph is a rooted tree, decomposed in sparse factors $H = LDL^T$ as per Algorithm 18.4.1 and block vector b with n variable size blocks, the following procedure solves for $Hx = b$.

▷ Forward elimination: walk up the tree starting from the leaf nodes.

for $i = 1, \ldots, n$ **do**
  for $k \in \text{child}(i)$ **do**
    $x_k \leftarrow x_k - H_{ki}x_i$
  end for
end for ▷ Backward substitution: walk down the tree starting from the root node.
for $i = n, \ldots, 1$ **do**
  $x_i \leftarrow H_{ii}^{-1}x_i$
  $x_i \leftarrow x_i - H_{ip(i)}x_{p(i)}$
end for

This is easily extended to the case of a bisymmetric matrix K for which we want to compute the lower triangular factors L and diagonal elements $\tilde{D}$ in $K = L\bar{D}U$. We do not need to compute the upper triangular U factor since $U = 2I - L^T$ as is easily verified. As previously, we start from the recursion formula

$$\begin{aligned} K &= \begin{bmatrix} K_{11} & -K_{21}^T \\ K_{21} & K_{22} \end{bmatrix} = \begin{bmatrix} I & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} \bar{D}_{11} & 0 \\ 0 & \bar{D}_{22} \end{bmatrix} \begin{bmatrix} I & -L_{21}^T \\ 0 & U_{22}^T \end{bmatrix} \\ &= \begin{bmatrix} \bar{D}_{11} & -\bar{D}_{11}L_{21}^T \\ L_{21}^T\bar{D}_{11} & L_{22}\bar{D}_{22}U_{22}^T - L_{21}\bar{D}_{11}L_{21}^T \end{bmatrix}, \end{aligned} \tag{18.26}$$

which yields the recurrence relations

$$\begin{aligned} &\tilde{K}_{11} \leftarrow \bar{D}_{11} = K_{11}, \\ &\tilde{K}_{21} \leftarrow L_{21} = K_{21}\bar{D}_{11}^{-1}, \\ &\tilde{K}_{22} \leftarrow K_{22} + \tilde{K}_{21}\tilde{K}_{11}\tilde{K}_{21}^T = K_{22} + K_{21}K_{11}^{-1}K_{21}^T. \end{aligned} \tag{18.27}$$

Obviously, when applying recurrence (18.24), one must keep track of whether the diagonal block is negative definite or positive definite but this requirement is lifted in the application of recurrence (18.27).

The sparse factorization algorithm described in this section was "rediscovered" numerous times and has been described in one form or another in [86, 139, 192, 37, 27] for the multibody context, and at least in [79] for the general sparse matrix context. Note that both [27] and [28] discuss the connection between robotics techniques designed specific methods to handle tree-structured mechanisms such as [86, 184] for instance, and the direct approach based on constraints described here.

Another point to note here is that the exact blocking structure of the original matrix H or K was left unspecified. In both [27] and [37], it is explicitly assumed that the first level of blocking is that of a single rigid body and this obviously creates a problem if one or more of the bodies is anchored to the universe with a constraint. But note however that for a single rigid body with coordinates q subject to an agglomerated constraint $g(q) = 0$, the effective mass matrix can be thought of as

$$\bar{M}^{(i)} = \begin{bmatrix} M^{(i)} & G^{(i)^T} \\ G^{(i)} & -C^{(i)} \end{bmatrix}, \quad \text{or } \tilde{M}^{(i)} = \begin{bmatrix} M^{(i)} & -G^{(i)^T} \\ G^{(i)} & C^{(i)} \end{bmatrix}. \tag{18.28}$$

Note that this changes the recurrence relations (18.24) and (18.27) to the extent that we may no longer assume that diagonal blocks D_{ii} or $\bar{D}_{ii}$ are symmetric and positive (or negative) definite. Nevertheless, this allows a factorization of a broader class of systems.

In fact, if appropriate care is taken to factorize the diagonal blocks suitably, we can use any level of grouping which means that submatrix H_{jj} might be a multibody subsystem in itself.

It should also be clear that any partitioning of the form

$$H = \begin{bmatrix} H_0 & \tilde{G}^T \\ \tilde{G} & \tilde{C} \end{bmatrix}, \tag{18.29}$$

leads to the same form of block factorization as were presented so far in this section provided H_0 is invertible. Obviously, if we choose H_0 to be the submatrix of H which consists of a *maximum spanning tree* of the graph of H, then, H_0 can be factored quickly using the foregoing sparse technique and that can be used to compute the Schur complement H/H_0 which can be factored using a direct or iterative, dense or sparse method. This was described in some detail in [37].

18.5 Gauss-Seidel type iteration methods

In a general, dense linear algebra problem, each and every variable is coupled to each and every other one. Untangling this web of interdependencies and identifying a suitable computational sequence allowing to evaluate each variable as a function of already known data is essentially what any matrix factorization algorithm performs.

The all-at-once character of direct factorization methods and the high computational cost—especially the $O(n^3)$ complexity order—make direct methods less than attractive in general, despite accuracy guarantees.

The fundamental principle behind a large family of iterative methods, of which of Gauss-Seidel is the best known member, is that one might be able to estimate the values of the solution vector, x_i, one at a time. This is expected to work well if the variables are only weakly coupled, a statement that will be made clear shortly.

Consider a square matrix A of dimensions $n \times n$ and the splitting

$$A = L + D + U, \tag{18.30}$$

where L, D, U are all square matrices of size $n \times n$ like A. Matrix L is strictly lower triangular, D is block diagonal, and U is strictly upper triangular. The splitting is such that any non-zero element of A, a_{ij}, say, appears at position (i, j) only once in either L, D, or U.

Consider a square matrix A of dimension $n \times n$ and the linear system of equations $Ax = b$, where $x, b \in \mathbb{R}^n$ are real vectors, and where b is given data. Suppose we already have an estimate $\bar{x}$ for the solution. The residual error for this candidate solution is $\bar{r} = b - A\bar{x}$. Taking the view that the ith component of the solution vector x is mostly responsible for the ith component of the residual error vector, we can force the value of the error to 0 by replacing $\bar{x}_i$ with the solution of

$$\alpha r_i + a_{ii}\bar{x}_i - a_{ii}x_i = 0. \tag{18.31}$$

where $0 < \alpha < 2$ is known as the *relaxation* parameter. For $\alpha < 1$, the iterations are *under-relaxed* and conversely for $\alpha > 1$, *over-relaxed*. The parameter α can be

chosen optimally if the spectrum of matrix A are well-known [52] but in general, the choice of α is a black art. When in doubt, one should set $\alpha = 1$.

One can apply the update rule (18.31) on an arbitrary sequence of indices, and there will likely be debate until the end of time as to what is the "best" sequence to pick, between sequential order, zigzag-, random pics, largest $|x|r_i$ first, etc. The asymptotic convergence always remains linear but of course, in one applies only a few iterations, using the right order is important. The resulting process is detailed in Algorithm 18.5.1 below.

Algorithm 18.5.1 Basic Gauss-Seidel algorithm

Initialize: $x \leftarrow x^{(0)}, r \leftarrow Ax - b, \nu = 0$
repeat
  for $j = 1, 2, \ldots, n$ **do**
    pick index i ▷ can go sequentially or randomly
    $z \leftarrow a_{ii}^{-1} r_i$
    $r \leftarrow r + A_{\bullet i} z$
    $x_i \leftarrow x_i + z$
  end for
  $\nu \leftarrow \nu + 1$
until $\nu > \nu_{max}$ or $\|r\| \leq tol$

The real problem here is that forcing $r_i \mapsto 0$ might increase the value $r_j, j \neq i$. Nevertheless, the process is monotonic for certain classes of matrices, notably so for symmetric positive definite ones, as shown below.

To get a better understanding of what this means, assume that the indices are traversed sequentially so that in Algorithm 18.5.1, $i = j$ in the inner loop. A short analysis of the operations in Algorithm 18.5.1 reveals that the main operation is

$$D_{ii} x_i^{(\nu+1)} = b_j - \left(\sum_{j<i} U_{ij} x_j^{\nu+1} + \sum_{i<j} L_{ij} x_j^{\nu} \right). \tag{18.32}$$

In words, the value of x_i is updated using the newly updated values x_k for $k = 1, 2, \ldots, i-1$, but the old values of x_k for $k = i+1, i+2, \ldots, n$. The consequence is that

$$\begin{aligned} Dx^{(\nu+1)} &= b - Lx^{(\nu+1)} - Ux^{(\nu)}, \text{ and grouping } x^{(\nu+1)} yields \\ (D+L)\, x^{(\nu+1)} &= b - Ux^{(\nu)}, \text{ so that} \\ x^{(\nu+1)} &= Nx^{(\nu)} + d, \text{ where} \\ N &= -(D+L)^{-1} U, \quad \text{and } d = (D+L)^{-1} b. \end{aligned} \tag{18.33}$$

This shows clearly that the Gauss-Seidel process is a stationary iterative process and that it converges if and only if matrix N has spectral radius less than unity

$$\rho(N) < 1. \tag{18.34}$$

It is well-known that for diagonally dominant matrices A, which includes symmetric positive definite matrices, $\rho(N) < 1$. Proofs and examples are found in the book by Kelley [158] on iterative methods.

The rate of convergence of any iterative process is *linear* meaning that the norm of the error is expected to behave as

$$\|r^{(\nu)}\| = O(\alpha^{\nu}), \tag{18.35}$$

where $\alpha < 1$. In practice, this means that each digit of accuracy costs a fixed amount of computational time which is not exactly very good. This said, Gauss-Seidel iterations are *smoothing* in the sense that they spread the error over the components of the residual vector. This is a welcome feature in certain applications [52].

The one consideration retained here for improving on the basic Gauss-Seidel idea is blocking. Assuming that matrix D in the splitting (18.30) is block diagonal, and assuming that each diagonal block D_{ii} is invertible, then, a *block* Gauss-Seidel process can be constructed which generally has a better convergence rate than the original. To see this, consider a random matrix A of size $n \times n$ and estimate the spectral radius of the iteration matrix as a function of the block size k. Obviously, if $k = n$, the spectral radius is 0 and we need just one iteration to compute the solution. Figure 18.1 illustrates what can happen for two different random matrices, plotting the spectral radius as a function of blocking size. The top line on the plot is from a matrix of the form $A = BB^T + \beta I$, where entries of $n \times n$ matrix B are uniform deviates, $b_{ij} \in [0, 1]$, and $\beta = 4$ here. For this case, blocking can move $\rho(A)$ significantly below unity. For the second line on the plot however, the matrix A was generated by first producing a set of eigenvalues λ_i uniformly distributed between $1/\sqrt{c}$ and $\sqrt{c}$, where c is the condition number. Set $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$, and then form the sandwich as $A = Q\Lambda Q^T$ with a random orthonormal rotation matrix Q generated by producing random Givens rotation [107]. This technique is adapted from Alefeld Chen and Potra [3]. For this case, the spectral radius is mostly unaffected by blocking, illustrating that this is not a sure fire method.

There is much written about special applications of Gauss-Seidel methods where it is possible to get faster convergence, at least during the first few iterations. It is also possible sometimes to construct a very good starting approximation or to construct a good preconditioner. This is not discussed any further here, however.

For constrained mechanical systems with block diagonal, symmetric and positive definite mass matrix M, and agglomerated constraint Jacobians G of compatible size, the regularized Schur complement is $S_\epsilon = GM^{-1}G^T + \Sigma$, where Σ is a diagonal matrix with positive entries of compatible dimensions. The Gauss-Seidel algorithm can be applied to this and some savings can be made in the computations by exploiting the structure of the system. In particular, it is not necessary to compute the entire matrix S_ϵ but only the diagonal blocks, and

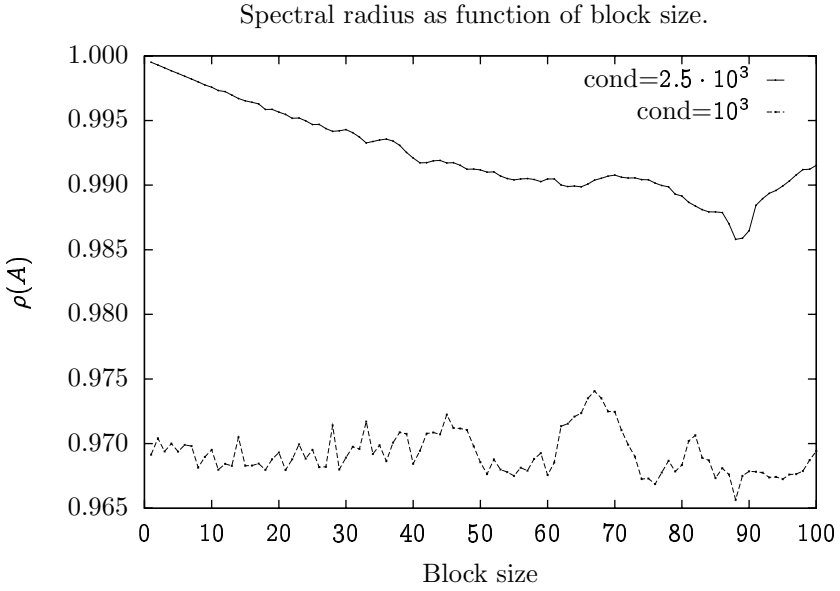

Figure 18.1: Blocking typically—but not necessarily—decreases the spectral radius of the GS iteration matrix, increasing the convergence rate. The plot contains data from two 200×200 random matrices with different initial condition numbers.

these are simply

$$[S_\epsilon]_{ii} = \sum_k G^{(ik)} M^{(k)^{-1}} G^{(ik)^T} + \Sigma_{ii}, \tag{18.36}$$

where the index k labels all the bodies involved with constraint i.

Next, consider that each constraint i, produces a multiplier $\lambda^{(k)}$, and this in turn generates body forces $f_c^{(ki)} = G^{(ik)^T} \lambda^{(i)}$ for each of the bodies involved in constraint i. For each body, the net constraint force is then $\sum_i G^{(ik)^T} \lambda^{(i)}$, and the index i only has to run over the constraints which body k is involved with. Now, the splitting formula for index i reads

$$\begin{aligned} D^{(ii)} \lambda^{(i)} &= b^{(i)} - \sum_{kj} G^{(ik)} M^{(k)^{-1}} G^{(jk)^T} \lambda^{(j)} \\ &= b^{(i)} - \sum_{kj} G^{(ik)} M^{(k)^{-1}} f_c^{(k)}. \end{aligned} \tag{18.37}$$

In the last sum though, the contribution from $\lambda^{(i)}$ is not included. What this means really is that it is possible to keep $\lambda^{(i)}$ for each constraint and then, keep the total constraint force $f_c^{(k)}$ on each body to avoid building the whole matrix S_ϵ and to perform only operations involving the list of bodies involved in a given constraint, as long as the $\lambda^{(i)}$ contribution can be removed from bodies k involved in constraint i. Note also that only the inverse mass matrix is needed

and this is now written $U = \mathtt{diag}(U^{(1)}, U^{(2)}, \ldots, U^{(n_b)})$. Similarly, the matrix product $J^{(k)} = G^{(ik)}U^{(k)}$ can all be precomputed.

The residual error for component i is then

$$r^{(i)} = \Sigma^{(i)}\lambda^{(i)} - b^{(i)} + \sum_k J^{(ik)} f_c^{(k)}, \tag{18.38}$$

where the sum now extends over all constraint forces.

When applying projected Gauss-Seidel on complementarity problems, it is not enough to compute just the *change* in $\lambda^{(i)}$ since there are bounds to satisfy. For this reason, a slightly more complicated version of the algorithm is provided in Algorithm 18.5.2. This version involves storing the net constraint forces on each body but the individual constraint forces associated with each constrained body on each constraint. Several variants are possible and simpler versions can be constructed when the main solve operation in the inner loop is not an MLCP.

It is of course possible to treat each individual constraint equation one at a time, which is the standard Gauss-Seidel procedure, but it is also possible to go the other way around and pack several constraints, with their associated bodies, in a diagonal block when splitting the system matrix S_ϵ. Doing this properly requires *partitioning* the system in suitable groups which are as nearly disconnected. When this is done, the MLCP solver in the inner loop might have an easier time if the smaller problems are not so degenerate as the larger one. In addition, once could choose *different* MLCP solvers for different groups. Results for this type of grouping were presented previously [170] and the results show that this is indeed very advantageous. In addition, the structure of the Gauss-Seidel is parallelizable with little overhead as long as most of the time is spent at the core operation of solving the MLCP. Since multicore CPUs are now common, this is the way to go when speed is needed. For problems that can be split easily, speedups of up to 10 can be observed [170]. This appears to be creditable to the fact that splitting removes degeneracies and ill-conditioning in subproblems but that is not entirely certain. There is the issue of accuracy which has not been properly investigated yet though.

Algorithm 18.5.2 Gauss Seidel Iterations for Multibody Systems

Given block inverse mass matrix $U = \mathsf{diag}(U^{(1)}, U^{(2)}, \ldots, U^{(n_b)})$, Jacobian blocks $G^{(ik)}$ and precomputed $J^{(k)} = G^{(ik)}U^{(k)}$
Given vector b
Given bound vector l, u
Build blocks $D^{(c)} = \sum G^{(ck)} J^{(ck)^T}$ and factorize
Allocate f_b vector holding forces for each body, set to 0
Allocate f_c vectors holding forces for each body k in each constraint, set to 0
repeat
  for $c = 1, \ldots, m$ **do** ▷ Loop over constraints
    for $k = 1, \ldots, k_c$ **do** ▷ Gather from bodies in constraint c
      Find global index b_k of body with index k in constraint c
      $f_{b_k}^{(b)} \leftarrow f_{b_k}^{(b)} - f_{ck}^{(c)}$
      $q \leftarrow b^{(c)} - J^{(cb_k)} f_{b_k}^{(b)}$
    end for
    $\lambda^{(c)} \leftarrow \mathsf{SOL}(\mathsf{MLCP}(D^{(c)}, q, l^{(c)}, u^{(c)}))$
    for $k = 1, \ldots, k_c$ **do** ▷ Scatter new forces to the bodies
      Find global index b_k for body k in constraint c
      $f_{ck}^{(c)} \leftarrow G^{(cb_k)^T} \lambda^{(c)}$
      $f_{b_k}^{(b)} \leftarrow f_{b_k}^{(b)} + f_{ck}^{(c)}$
    end for
  end for
until Error is small enough.

18.6 A preconditioned conjugate gradient method

The classical definition of the saddle point problem uses the symmetric form of H in (18.17), but with $C = 0$. To be specific, write $Hz = a$ where

$$H = \begin{bmatrix} M & G^T \\ G & C \end{bmatrix}, z = \begin{bmatrix} x \\ y \end{bmatrix}, a = \begin{bmatrix} b \\ c \end{bmatrix}, \tag{18.39}$$

The simplest iterative method for solving this is a relaxation process whereby we first make an approximation of y with $y^{(\nu-1)}$, say, and then solve for $Mx^{(\nu)} = b - G^T y^{(\nu-1)}$. This is then used to update the value of the approximation using $y^{(\nu)} = y^{(\nu-1)} + \alpha\left(Gx^{(\nu)} - c\right)$. This will be stationary when the term in parenthesis vanishes, i.e., when we satisfy the constraint equation $Gx^{(\star)} = c$ which solves the system as desired. The choice of α is essential for the convergence rate and it is difficult to estimate the optimal value as it depends on the condition number of the matrix $S = GM^{-1}G^T$. One can update α dynamically as per the gradient method as shown in Braess [52], Section 5.2.

The idea for improve on this is to apply the CG algorithm on the Schur complement matrix $S = GM^{-1}G^T + C$, but without having to compute S, i.e., using only matrix-vector operations involving G, G^T, and M^{-1} and C. Braess [52] does this in Algorithm 5.2 for the case where $C = 0$. To recover the correct modifications to include $C \neq 0$, first consider the case where matrix S is available and perform the conjugate gradient (CG) algorithm to solve for $Sy = c - GM^{-1}b = \tilde{c}$. Vector x can then be recovered with $x = M^{-1}(G^T y + b)$. What is needed is the value of the current gradient $g^{(\nu)}$, the local search direction $d^{(\nu)}$, the step magnitude $\alpha^{(\nu)}$, and the factors $\beta^{(k)}$. This is detailed in Algorithm 18.6.1, in which the S-norm is defined as:

$$\|(\cdot)\|_S^2 = (\cdot)^T S(\cdot). \tag{18.40}$$

Algorithm 18.6.1 Conjugate gradient algorithm

Given: matrix S and vector $\tilde{c}$, $y^{(0)}$ and $\tau > 0$
Initialize: $g^{(1)} = Sy^{(0)} - \tilde{c}$, and $d^{(1)} = -g^{(1)}$
repeat
  $\alpha^{(\nu)} \leftarrow \|g^{(\nu)}\|^2 / \|d^{(\nu)}\|_S^2$
  $y^{(\nu+1)} \leftarrow y^{(\nu)} + \alpha^{(\nu)} d^{(\nu)}$
  $g^{(\nu+1)} \leftarrow g^{(\nu)} + \alpha^{(\nu)} S d^{(\nu)}$
  $\beta^{(\nu)} \leftarrow \|g^{(\nu+1)}\|^2 / \|g^{(\nu)}\|^2$
  $d^{(\nu+1)} \leftarrow -g^{(\nu+1)} + \beta^{(\nu)} d^{(\nu)}$
until $\|g^{(\nu)}\| < \tau$

The aim now is to modify Algorithm 18.6.1 to avoid computing S. To this end, introduce auxiliary vectors so that all that is needed matrix vector multiplication

for G and G^T, as well as solutions of the linear system $M\tilde{x} = \tilde{b}$ for different right hand side vectors $\tilde{b}$ as the algorithm progresses. This latter system is presumed to be easy to solve.

Observe that the gradient of the quadratic form $\frac{1}{2}y^T S y + y^T \tilde{c}$ is

$$g(y) = Sy + \tilde{c}. \tag{18.41}$$

Using $x = M^{-1}(b + G^T y)$, then, the expression for the gradient can be reduced to $g(x, y) = Gx + Cy - c$.

Next, the equation $x = M^{-1}(b + G^T y)$ is evaluated for $x^{(\nu+1)}$ and $y^{(\nu+1)}$, and using the update rule $y^{(\nu+1)} = y^{(\nu)} + \alpha^{(\nu)} d^{(\nu)}$ from Algorithm 18.6.1 above,

$$x^{(\nu+1)} = M^{-1}\left(b + G^T y^{(\nu)}\right) = x^{(\nu)} + \alpha^{(\nu)} M^{-1} G^T d^{(\nu)}. \tag{18.42}$$

This suggests storing the vectors $p^{(\nu)} = G^T d^{(\nu)}$ and $h^{(\nu)} = M^{-1} p^{(\nu)}$. With these, we can easily compute the expression

$$\alpha^{(\nu)} = \|g^{(\nu)}\|^2 / ({h^{(\nu)}}^T p^{(\nu)} + \|y^{(\nu-1)}\|_C^2), \tag{18.43}$$

efficiently as well. Combining these observations, we arrive at Algorithm 18.6.2 listed below, in which all iteration indices have been dropped for clarity. This algorithm requires one solution of a linear system involving matrix M, one matrix-vector multiplication of the form Gx, one of the form $G^T z$, and one of the form Cy, as well as two scalar products with vectors of length n, and one with vectors of length m. Testing the termination condition requires to take the norm of two vectors of length m and two of length n.

Algorithm 18.6.2 Uzawa Algorithm with conjugate directions

Given: $y^{(0)} \in \mathbb{R}^m$, $\tau_x, \tau_y > 0$
Initialize: $y = y^{(0)}$, solve $Mx = b + G^T y^{(0)}$ for x. Set $d = -g = Gx + Cy - c$, $w = g^T g$, and $q = Cy$.
repeat
  $p \leftarrow G^T d$
  $h \leftarrow M^{-1} p$
  $\alpha \leftarrow \frac{w}{p^T h + q^T y}$
  $y \leftarrow y + \alpha d$
  $x \leftarrow x + \alpha h$
  $q \leftarrow Cy$
  $g \leftarrow Gx + q - c$
  $w_1 \leftarrow g^T g$
  $\beta \leftarrow \frac{w}{w_1}$
  $w \leftarrow w_1$
  $d \leftarrow -g + \beta d$
until $\frac{\|\alpha g\|}{\|y\|} < \tau_y$ and $\frac{\|\alpha h\|}{\|x\|} < \tau_x$

Of course, a CG algorithm is not so useful unless it is preconditioned. The choice of matrix M in Algorithm 18.6.2 is limited by the necessity to keep the

matrix S_c positive definite, and the requirement that the action of M^{-1} should be easy to compute. When matrix C vanishes, there is little choice for M. However, if $C \neq 0$ and block diagonal, we can repartition the original system as follows

$$K = \begin{bmatrix} M & -G_{1\bullet}^T & -G_{2\bullet}^T \\ G_{1\bullet} & C_{11} & 0 \\ G_{2\bullet} & 0 & C_{22} \end{bmatrix} = \begin{bmatrix} \tilde{M} & -\tilde{G}^T \\ \tilde{G} & \tilde{C} \end{bmatrix}, \tag{18.44}$$

where we defined

$$G = \begin{bmatrix} G_{1\bullet} \\ G_{2\bullet} \end{bmatrix}, \quad \tilde{G} = \begin{bmatrix} G_{2\bullet} & 0 \end{bmatrix}, \quad \tilde{M} = \begin{bmatrix} M & -G_{1\bullet}^T \\ G_{1\bullet} & C_{11} \end{bmatrix}, \quad \tilde{C} = C_{22}, \tag{18.45}$$

and assumed that $C_{12} = 0, C_{21} = 0$.

Using this partition of K, we can now apply Algorithm 18.6.2 using $\tilde{M}$ instead of M. This amounts to performing CG iterations on the matrix

$$\begin{aligned} \tilde{S} &= \begin{bmatrix} G_{2\bullet} & 0 \end{bmatrix} \begin{bmatrix} M & -G_{1\bullet}^T \\ G_{1\bullet} & C_{11} \end{bmatrix}^{-1} \begin{bmatrix} G_{2\bullet}^T \\ 0 \end{bmatrix} \\ &= G_{2\bullet} \left(M^{-1} - M^{-1} G_{1\bullet}^T \left(G_{1\bullet} M^{-1} G_{1\bullet}^T + C_{11} \right)^{-1} G_{1\bullet} M^{-1} \right) G_{2\bullet}^T. \end{aligned} \tag{18.46}$$

Matrix $\tilde{S}$ is still positive definite because it is the Schur complement of the positive definite block matrix $\tilde{M}$ of (18.45) in the positive definite matrix K of (18.44). Positive definiteness is preserved under the Schur complement operations [69].

This preconditioned solver was developed in 2005 and is the core of the Master's thesis of Dalgård [70] which dealt with cloth simulation. The literature did not appear to cover the case where matrix C was non-zero. But since then, Algorithm 18.6.2 did appear [74], along with a variety of techniques for solving saddle point problems. More investigation is needed to select a suitable technique for multibody systems.

18.7 Factorization updates and down-dates

When solving LCPs, there is a need to solve linear systems of the form

$$S_{\alpha\alpha} z_\alpha = -q_\alpha, \tag{18.47}$$

where S is a square $n \times n$ matrix, $z, q \in \mathbb{R}^n$ are n-dimensional real vectors, and $\alpha \in \{1, 2, \ldots, n\}$ is an index set. The matrix $S_{\alpha\alpha}$ is a principal submatrix of S.

Now, if a factorization of S is known and if the index set α is close to the full set $\{1, 2, \ldots, n\}$, it is possible to update the factorization of S to obtain a factorization of $S_{\alpha\alpha}$ and to solve system (18.47) with a relatively small number of operations. This section is devoted to that problem.

We shall assume in what follows that matrix S is symmetric and positive definite so that it is possible to compute the Cholesky factors $S = LL^T$ without any pivoting operation. This simplifies much of the presentation. In addition, this is one important benefits of the regularization theory presented in Chapter 4. For any given constrained mechanical system, given knowledge of the scaling of the mass matrix, it is possible to estimate the regularization parameters needed to guarantee that the final matrix is positive definite enough to avoid all pivoting operations during factorization. When the regularizations are too small though, one has to resort to pivoting and minimal perturbation as described in Higham [126].

Assume that matrix S is partitioned along the two index sets, α, β, with $\alpha \cup \beta = \{1, 2, 3, \ldots, n\}$ and $\alpha \cap \beta = \emptyset$, so that

$$\begin{aligned} S &= \begin{bmatrix} S_{\alpha\alpha} & S_{\alpha\beta} \\ S_{\beta\alpha} & S_{\beta\beta} \end{bmatrix} \\ S^{-1} = \tilde{S} &= \begin{bmatrix} \tilde{S}_{\alpha\alpha} & \tilde{S}_{\alpha\beta} \\ \tilde{S}_{\beta\alpha} & \tilde{S}_{\beta\beta} \end{bmatrix}, \end{aligned} \tag{18.48}$$

where the notation $\tilde{S} = S^{-1}$ was introduced for clarity. The inverse of the block $S_{\alpha\alpha}$ is denoted with $S_{\alpha\alpha}^{-1}$ and the goal of the following calculation is to derive an expression for $S_{\alpha\alpha}^{-1}$. A simple calculation of the matrix products shows that

$$S\tilde{S} = \begin{bmatrix} S_{\alpha\alpha}\tilde{S}_{\alpha\alpha} + S_{\alpha\beta}\tilde{S}_{\beta\alpha} & S_{\alpha\alpha}\tilde{S}_{\alpha\beta} + S_{\alpha\beta}\tilde{S}_{\beta\beta} \\ S_{\beta\alpha}\tilde{S}_{\alpha\alpha} + S_{\beta\beta}\tilde{S}_{\beta\alpha} & S_{\beta\alpha}\tilde{S}_{\alpha\beta} + S_{\beta\beta}\tilde{S}_{\beta\beta} \end{bmatrix} = \begin{bmatrix} I_{\alpha\alpha} & 0 \\ 0 & I_{\beta\beta} \end{bmatrix}, \tag{18.49}$$

and therefore, looking at the expression on the first row, second column, we can deduce that

$$S_{\alpha\beta} = -S_{\alpha\alpha}\tilde{S}_{\alpha\beta}\tilde{S}_{\beta\beta}^{-1}, \tag{18.50}$$

and therefore, substituting the appropriate terms we arrive at the expression

$$S_{\alpha\alpha}\left(\tilde{S}_{\alpha\alpha} - \tilde{S}_{\alpha\beta}\tilde{S}_{\beta\beta}^{-1}\tilde{S}_{\beta\alpha}\right) = I_{\alpha\alpha}, \tag{18.51}$$

from which we read off the expression for $S_{\alpha\alpha}^{-1}$

$$S_{\alpha\alpha}^{-1} = \tilde{S}_{\alpha\alpha} - \tilde{S}_{\alpha\beta}\tilde{S}_{\beta\beta}^{-1}\tilde{S}_{\beta\alpha}. \tag{18.52}$$

The upshot here is that all the terms and factors on the right of the equation are known once we have factorized the matrix S, with the exception of the factor $\tilde{S}_{\beta\beta}^{-1}$ for which we still need a computational procedure. Hopefully, the index set is almost full so that most of the constraints are active in which case the matrix $\tilde{S}_{\beta\beta}$ is smaller than $S_{\alpha\alpha}$ and therefore, factorization of that matrix is faster. The algorithm we now proceed to develop works with S and its factorization as well as with the matrix $\tilde{S}_{\beta\beta}$. All the factors involved will be computed by repeatedly solving for the system of linear equations

$$Sx = b, \tag{18.53}$$

with a number of different right hand sides.

Observe first that MM^{-1} is the identity matrix and therefore, we can solve for each column i of the matrix M^{-1} by solving the linear system

$$Mx = e_i, \tag{18.54}$$

where e_i is the unit column vector with 1 at position i and zeros everywhere else. We can construct the matrix $S_{\beta\beta}^{-1}$ with Algorithm 18.7.1.

Algorithm 18.7.1 Computing $\tilde{S}_{\beta\beta}$

Initialize $Q_{ij} = 0$
for i in β **do**
  set $e \longleftarrow 0$ and then $e_i \longleftarrow 1$
  Solve for $Sx = e$
  Set $Q_{\beta i} \longleftarrow e_\beta$
end for

The other terms we need in solving for $S_{\alpha\alpha}z_\alpha = -q_\alpha$ are of obtained from the following two sets of equations equations

$$\begin{aligned} \begin{bmatrix} \tilde{S}_{\alpha\alpha} & \tilde{S}_{\alpha\beta} \\ \tilde{S}_{\beta\alpha} & \tilde{S}_{\beta\beta} \end{bmatrix} \begin{bmatrix} q_\alpha \\ 0 \end{bmatrix} &= \begin{bmatrix} \tilde{S}_{\alpha\alpha}q_\alpha \\ \tilde{S}_{\beta\alpha}q_\alpha \end{bmatrix}, \text{ and} \\ \begin{bmatrix} \tilde{S}_{\alpha\alpha} & \tilde{S}_{\alpha\beta} \\ \tilde{S}_{\beta\alpha} & \tilde{S}_{\beta\beta} \end{bmatrix} \begin{bmatrix} 0 \\ q_\beta \end{bmatrix} &= \begin{bmatrix} \tilde{S}_{\alpha\beta}q_\beta \\ \tilde{S}_{\beta\beta}q_\beta \end{bmatrix}. \end{aligned} \tag{18.55}$$

Therefore, solving for the linear system

$$Su = \begin{bmatrix} q_\alpha \\ 0 \end{bmatrix}, \tag{18.56}$$

produces

$$\begin{aligned} u_\alpha &= \tilde{S}_{\alpha\alpha}q_\alpha, \text{ and} \\ u_\beta &= \tilde{S}_{\beta\alpha}q_\alpha. \end{aligned} \tag{18.57}$$

Next, solve the linear system

$$\tilde{S}_{\beta\beta} y_\beta = u_\beta, \tag{18.58}$$

which produces

$$y_\beta = \tilde{S}_{\beta\beta}^{-1} u_\beta = \tilde{S}_{\beta\beta}^{-1} \tilde{S}_{\beta\alpha} q_\alpha. \tag{18.59}$$

Finally, solving the linear system

$$Mx = y, \quad \text{where } y = \begin{bmatrix} 0 \\ y_\beta \end{bmatrix}, \tag{18.60}$$

produces the solution by setting

$$z_\alpha = x_\alpha - u_\alpha = \left(\tilde{S}_{\alpha\beta} \tilde{S}_{\beta\beta}^{-1} \tilde{S}_{\beta\alpha} - \tilde{S}_{\alpha\alpha} \right) q_\alpha = - \left(\tilde{S}_{\alpha\alpha} - \tilde{S}_{\alpha\beta} \tilde{S}_{\beta\beta}^{-1} \tilde{S}_{\beta\alpha} \right) q_\alpha. \tag{18.61}$$

This algorithm is analogous to the standard Sherman-Morrison-Woodbury formula which is spelled out in [107] p. 50, and is the subject of [97], but not identical.

We summarize those results in Algorithm 18.7.2 which solves for $S_{\alpha\alpha} z_\alpha = -q_\alpha$.

Algorithm 18.7.2 Block Down-dating algorithm to compute $S_{\alpha\alpha} z_\alpha = -q_\alpha$

Compute $\tilde{S}_{\beta\beta}$ using previous Algorithm 18.7.1
Factor $\tilde{S}_{\beta\beta} = LL^T$ using Cholesky
Set $b = (q_\alpha^T, 0)^T$ and solve for $Su = b$
Solve for $\tilde{S}_{\beta\beta} y_\beta = u_\beta$
Set $y = (0, y_\beta^T)^T$ and solve for $Sx = y$
Set $z_\alpha = x_\alpha - u_\alpha$ and $z_\beta = 0$

Note that Algorithm 18.7.2 is applicable to pure LCP case but it is easy to extend this to the case where we have both upper and lower bounds as follows. Assume that the set β is now the list of all variables which are either at a lower or an upper bound and therefore, for these variables, the value of the z vector is $z_\beta = \bar{z}_\beta$. The principal subproblem now reads $S_{\alpha\alpha} z_\alpha + S_{\alpha\beta} \bar{z}_\beta + q_\alpha$. We can therefore apply Algorithm 18.7.2 to this by modifying the vector q in the obvious way.

Algorithm 18.7.2 involves two solves with the full matrix S. An alternative computation method allows to skip one of these which we now derive. We start from the principal subproblem equation:

$$S_{\alpha\alpha} z_\alpha = -q_\alpha - S_{\alpha\beta} \bar{z}_\beta. \tag{18.62}$$

The objective is to write z_α in terms of $S^{-1}q$ only, i.e., to avoid all extra solves. We start with multiplying both sides of the equation with $S_{\alpha\alpha}^{-1} = \tilde{S}_{\alpha\alpha} - \tilde{S}_{\alpha\beta}(\tilde{S}_{\beta\beta})^{-1} \tilde{S}_{\beta\alpha}$. Also note that $(S^{-1}q)_\alpha = \tilde{S}_{\alpha\alpha} q_\alpha + \tilde{S}_{\alpha\beta} q_\beta$. Using all these

identities we have:

$$(S_{\alpha\alpha})^{-1} q_\alpha = \tilde{S}_{\alpha\alpha} q_\alpha - \tilde{S}_{\alpha\beta} (\tilde{S}_{\beta\beta})^{-1} \tilde{S}_{\beta\alpha} q_\alpha \tag{18.63}$$

$$= (S^{-1} q)_\alpha - \tilde{S}_{\alpha\beta} q_\beta - \tilde{S}_{\alpha\beta} (\tilde{S}_{\beta\beta})^{-1} \left((S^{-1} q)_\beta - \tilde{S}_{\beta\beta} q_\beta \right) \tag{18.64}$$

$$= (S^{-1} q)_\alpha - \tilde{S}_{\alpha\beta} (\tilde{S}_{\beta\beta})^{-1} (S^{-1} q)_\beta . \tag{18.65}$$

Noting that $S_{\alpha\beta} \bar{z}_\beta = -S_{\alpha\alpha} \tilde{S}_{\alpha\beta} (\tilde{S}_{\beta\beta})^{-1} \bar{z}_\beta$, we have the result:

$$z_\alpha = (S_{\alpha\alpha})^{-1} \left(-q_\alpha - S_{\alpha\beta} \bar{z}_\beta \right) \tag{18.66}$$

$$= -(S^{-1} q)_\alpha + \tilde{S}_{\alpha\beta} (\tilde{S}_{\beta\beta})^{-1} (S^{-1} q)_\beta + (S_{\alpha\alpha})^{-1} S_{\alpha\alpha} \tilde{S}_{\alpha\beta} (\tilde{S}_{\beta\beta})^{-1} \bar{z}_\beta \tag{18.67}$$

$$z_\alpha = -(S^{-1} q)_\alpha + \tilde{S}_{\alpha\beta} (\tilde{S}_{\beta\beta})^{-1} \left((S^{-1} q)_\beta + \bar{z}_\beta \right) . \tag{18.68}$$

We therefore have a new version of Algorithm 18.7.2:

Algorithm 18.7.3 Block Down-dating algorithm to compute $S_{\alpha\alpha} z_\alpha = -q_\alpha - S_{\alpha\beta} \bar{z}_\beta$

Initialize: solve $S z^{(0)} = -q$
Compute $\tilde{S}_{\beta\beta}$ using Algorithm 18.7.1
Factor $\tilde{S}_{\beta\beta} = LL^T$ using Cholesky
Set $u_\beta = -z_\beta^{(0)} + \bar{z}_\beta$
Solve for $\tilde{S}_{\beta\beta} y_\beta = u_\beta$
Set $z_\alpha = z_\alpha^{(0)} + \tilde{S}_{\alpha\beta} y_\beta$ and $z_\beta = \bar{z}_\beta$

18.8 End notes

Presented in this chapter were essential elements for fast and efficient implementation of the linear algebra core of any multibody computational code. This is not the final word, however. To get best possible performance in practice, consideration must be made of specific CPU features. Currently, this also implies introducing parallelism in the early stages to exploit multicore CPUs which have become commodities, and it is none too early to get symmetric multiprocessor (SMP) systems at consumer prices.

The real fun starts *after* the basic implementation of the algorithms listed in this chapter, when trying to get the most out of each clock cycle on the CPU. All block operations—which have not been detailed—can be implemented using the vector instructions now available in the SSE3 extensions of the x86 processor family. This allows to perform various operations either on four-dimensional single precision vectors or on two-dimensional double precision vectors in a single clock cycle. After taking this into account, it becomes necessary to use various tiled storage formats to get maximum performance. Implementing direct methods which exploit sparsity minimally, using skyline storage and a suitable version of the Cholesky factorization [141], say, and optimizing the block operations with

SSE3 instructions, is challenging but feasible. Experience tells that this strategy delivers higher performance than directly using dense algorithms from LAPACK, for instance, because there is usually some sparsity in the Schur complement matrices. The real benefit of growing your own is that it is possible to implement the updating and downdating algorithms efficiently.

Sparsity is ubiquitous in multibody systems. Indeed, a typical constraint equation involves one or two rigid bodies. Ignoring exotic constraints involving very many bodies, as can be used for a cable model [253] for instance, the fill ratio of the matrices of multibody problems is expected to be very near zero. For n bodies with 6 degrees of freedom and m constraint equations each coupling 2 bodies, the matrix dimension is $N = 6n + m$, but the number of non-zero elements is approximately $36n + 4m = O(N)$, and the matrix size is N^2, giving a fill ratio of $O(N^{-1})$.

Therefore, preconditioned iterative sparse methods are expected to deliver the needed high performance. Some good reviews have recently been published on these methods [74, 75, 48] and more work is needed to tailor a fast and robust technique for multibody systems.

19 Conclusion

The discrete principle of least action is a powerful tool. It was used systematically in the present thesis to analyze several different elements which are often incompatible. Good methods for dissipative systems do not necessarily apply to conservative ones, and *vice versa*. Likewise, good strategies for discretizing smooth systems may or may not apply to nonsmooth ones. But the discrete principle of least action dealt adequately with all cases.

Integration methods designed specifically to address DAEs, or stiff problems, or designed to preserve certain invariants, often rely on higher order discretization. Yet again, the discrete least action principle yields satisfactory low order discretization to integrate such problems.

Low order integration methods perform less computational work per step. This does not necessarily make for savings in the computational budget though, since stability can be quickly lost at low order. The remedy is to lower the time step, sometimes drastically. This is why higher order methods are often better. They can take bigger steps and do less work *overall*. But for real-time applications, like for the 100 meter sprinters, *overall* is not what counts. Time is of the essence and *efficiency* is a lesser concern than speed.

The discrete principle of least action does provide a good resolution to this dilemma. The stepping formulae it produces are all symplectic and though this is not a guarantee of unconditional stability, it is a guarantee of *global* stability, whenever local stability requirements are met. Global stability also implies that the trajectories produced by a discrete mechanical integrator *shadow* the exact trajectories of a perturbed physical system [100, 196]. In other words, the computed trajectories are time-discrete samples of the exact solution of a physical system that is not very different from the one being simulated. More stability analysis must be done to understand the ramifications of this and the detailed nature of the perturbed problem. For simple systems such as the gyroscope presented in Chapter 15 however, the perturbed system can be computed exactly [51] and can be shown to differ from the original problem by terms of second order in the time step.

For the context of interactive physics, this shadowing property is invaluable since what is seen on the computer screen is faithful to physics. It is better to provide the correct dynamics of an approximate physical problem than to relinquish on interactivity or on model complexity due to slow execution speed, or to settle for animations that merely imitate physical motion.

The need for raw speed is not limited to interactive simulations though, and science offers many opportunities to apply the discrete mechanical integrators advantageously. Molecular dynamics is one such case. The evaluation of thermo-

dynamical quantities requires the simulation of very large systems with millions of molecules. This can be both slow if the wrong methods are used, and pointless if the simulated dynamics does not exhibit the correct geometric properties. If a strongly stable discrete mechanical integrator is used with a very large time step, one can quickly compute approximate thermodynamic properties. These approximations are the exact properties of a slightly different physical system because of the shadowing properties of discrete mechanical integrators. This might sound sloppy but many properties of gases and liquids are now well understood thanks to simulations executed with the Lennard-Jones potential, which is a coarse approximation. Leimkuhler and Reich [178] provide a good survey of the aims and techniques of molecular dynamics and a deeper analysis of the consequences of using integration methods that are faithful to the physics. The main conclusion to draw is that low order methods, inaccurate as they may be, are useful in scientific work and not simply a good device to deceive the users of interactive simulations.

But even a broadly applicable principle is not a panacea. It is not the intention here to claim that discrete mechanical integrators solve all problems related to numerical integration, not even all those related to integration of mechanical systems. For one, methods with higher order and higher accuracy are desirable and it remains to be seen how difficult it is to construct these for complex mechanical systems, including a mix of constraints and forcing terms.

To recapitulate, it was shown in Chapter 3 how to construct low order discrete mechanical integrators by approximating the integrals of energetic terms, namely, the potential energy, the kinetic energy, and the Rayleigh dissipation functions. Applying the principle of least action for conservative systems, d'Alembert's principle of vanishing virtual work for forced and dissipative systems, or the Fourier inequality for systems with closed boundaries, a stationarity condition yields the discrete Euler-Lagrange equations, which define a three-terms nonlinear recurrence—a discrete time stepping scheme. Discretizations of the different terms can be mixed and matched, each one being chosen after considering stability requirements. As seen in Chapter 10, a variety of nonsmooth forcing terms can be successfully discretized also, provided Fourier's inequality is taken into account.

The stepping equations can be numerically regularized directly, of course. But as shown in Chapter 4, diagonal regularization terms can be introduced by giving finite potential energy to the ghost variables, leading to a semi-implicit discrete time-stepping scheme, and this is in direct correspondence with the Rubin-Ungar theorem on constraint realization. The fast oscillations which correspond to small ghost potential energy can be damped reliably by introducing Rayleigh dissipation functions acting directly on the ghost variables, and discretized in a stable way as well. Such regularization and stabilization terms guarantee that the stepping equations are well-posed and always solvable, even when nonsmooth forcing terms are included, as was shown in Chapter 10, and exhibit strong linear stability as shown in Section 4.6. Because these regularization and stabilization terms appear directly in the Lagrangian, they are in fact physical models and

the parameters can be used, within stability limits, to model physical phenomena such as joint or contact compliance, or a small viscous creep in dry contacts.

The strength of the variational formulation is felt again in the discretization of the free rigid body as done in Chapter 15. In this case, the differential geometry of the problem is accounted for by the discrete variational principle once the correct parametrization is chosen. This leads to corrections to the stepping equations which would have been difficult to guess by other means based on Taylor series expansions. Indeed, this goes back to the roots of the discrete variational principle, in the work of Moser and Veselov [208] which dealt precisely with the free rigid body problem.

Discrete mechanical integrators may not be a silver bullet but they are a worthy addition to numerical analysis, and a life buoy for the physicist eager to perform simulations, but not necessarily willing to invest years in studying the beautiful techniques of numerical analysis. She can just write down the Lagrangian, estimate the time integrals over a short interval h, differentiate to recover the stationarity conditions, and start stepping away.

But there is a rub, of course. The discrete Euler-Lagrange equations are at least linearly implicit for many important cases. This is not only true for all forms of constraints, but also for quickly varying potential functions. The implicit midpoint rule studied in Sections 2.4.3 and 3.7 is a case in point. What happens exactly when these equations are solved only approximately is not known in general. If the symplectic nature of the flow destroyed when either the solution is approximated? Which types of approximations can preserve the geometry? These are open questions. It may be possible to model the effect of an approximation with additional correction terms in the Lagrangian, as was the case for the regularization, but that is not certain. There is more work to do to identify robust and fast numerical numerical methods for solving the Euler-Lagrange, and especially so for the case of dry frictional contact problems.

List of Tables

List of Figures

Glossary

Below are brief technical descriptions of and most of the technical terms and all acronyms used in the main text. The numbers following each item is the section number where there are mainly used in the text.

3D graphics

The computational process producing a digital two-dimensional image, containing color and intensity information for each pixel on a rectangular screen, from a collection of three-dimensional geometric objects represented with vertices edges and triangulated faces with given position and orientation, by performing a projection on a viewing plane from a given view point. The computational process involves translating and rotating all the the objects, selecting the visible objects and portions thereof using view frustum, backface, and occlusion culling, texture mapping, computing lighting and shading effects, projecting into the view plane, and antialiasing, as well as a growing number of additional techniques which can make the rendered scene more realistic. All these operations involve floating point arithmetic and require high performance, usually available on a special purpose processor on a separate acceleration card.

3D model

A mathematical representation of a three-dimensional object suitable to be displayed using standard 3D graphics rendering techniques. Typical 3D models are collection of geometric primitives including points, edges and triangles. Points are known as vertices in 3D graphics because they also carry lighting information for the rendering process, but this is not relevant to the present context. Because of explicit tessellation (decomposition into triangles), 3D models of simple shapes such as spheres and cones, are nonsmooth since their surface normals are discontinuous whenever two triangles meet at an edge.

ACM: Association for Computing Machinery

The first educational and scientific computing society, founded in 1947 and based in the US. The ACM publishes eleven peer reviewed scientific journals and sponsors more than 100 yearly international conferences for special interest groups (see SIG). SIGGRAPH (see entry) is one of them.

Action

Given a mechanical system with configuration space Q, tangent bundle TQ, generalized coordinates and velocities q and $\dot{q}$, respectively, the functional over paths defined over a finite time interval, $\dot{\gamma} : [t_0, t_1] \mapsto TQ$, defined as $S[\dot{\gamma}] = \int_{t_0}^{t_1} \mathrm{d}s\mathcal{L}(q(s), \dot{q}(s))$.

Algebra

An algebra consists of a set $\mathcal{A}$ and two binary operators generally denoted $+$ and $\cdot$, so that $\mathcal{A}$ is closed under the action of the operators. In other words, given any two elements $x, y \in \mathcal{A}$, we have both $x + y \in \mathcal{A}$ and $x \cdot y \in \mathcal{A}$. The addition operator, $+$, is always commutative (Abelian) so that $x + y = y + x$ but the multiplication operator $\cdot$ is not necessarily so. In general, $x \cdot y \neq y \cdot x$. For the case were $x \cdot y = y \cdot x$, the algebra is said to be commutative.

Algorithm

A finite set of well defined instructions whose execution accomplish a given task on a given set of input data in finite time, with a well defined end state.

Algorithmic complexity

An asymptotic estimate of the increase in computational cost of a given algorithm as a function of the size of the problem being solved. This is written as $O(n^p)$ where n is the problem size and p is the complexity order for algorithms which terminate in polynomial time. For that case, if $C(n)$ is the computational cost for solving a problem of size n, the limit $\lim_{n\to\infty} n^{-p}C(n)$ is a constant. Problems whose solution can be verified in polynomial time but which are not known to have a direct polynomial solution algorithm are called NP-hard.

AMS: American Mathematical Society

The AMS was founded in 1888 and is based in Providence, Rhode Island, USA. Its aim is to further mathematical research and scholarship. The AMS publishes eight scholarly journals as well as indexes and reviews of over 1,800 journals. It organizes several international conferences yearly. The MathSciNet online database which provides reviews of millions of articles—and does export to BibTeX—is invaluable.

Analytic mechanics

In contrast with the Newtonian formulation, analytic mechanics is entirely stated in terms of energy which is divided into kinetic and potential. These two types of energy are scalar functions which can be expressed in terms of the kinematic variables, namely, the generalized coordinates $q \in Q$ and the generalized velocities $\dot{q}$, as $T(q, \dot{q})q$ and $V(q)q$, respectively. The equations of motion of Newton's first two laws are recovered using a variational

principle. For conservative systems, one can use Hamilton's principle of least action though for dissipative systems, d'Alembert's principle of virtual work is required. The analytic formulation makes it easier to change to curvilinear coordinate systems and to analyze the various symmetries of a given system.

Angular momentum

For a point particle in three dimensions with position x and momentum p, the angular momentum about a point $y \in \mathbb{R}^3$ is the vector $l(y) = \widehat{x-y}p$. A consequence of Galilean relativity is that $l(y)$ is constant when only central forces directed toward the origin act on the particle. For this case, the particle is confined to move in the plane tangent to $l(0)$. Changing the angular momentum of a point mass requires a force f which is not collinear with the vector $x - y$. This generates a torque $\tau = \widehat{x-y}f$, which is the rate of change of angular momentum, namely, $\dot{l} = \tau$.
For a rigid body, the summation of the angular momentum over the constitutive point masses yields the expression $l = \mathcal{I}\omega$, where $\mathcal{I}$ is the inertia tensor and ω is the angular velocity vector, both expressed in the inertial frame of reference.

Ball joint

A joint such that, starting from either body, the motion of the second is a pure rotation about a point fixed in the body frame of the first. The origin of the rotation is fixed in both bodies. Ball joints are also known as gimbals and ball and socket joints. Practical ball joints never allow the full range of rotations about the common point but the mathematical idealization does because of limitations in their constructions. Because of the global properties of the group of rigid rotations $SO(3)$, physical constructions involving three hinges—corresponding to the three Euler angles—are subject to gimbal lock which is a singular point that can in fact break the device. The description used in robotics suffers from the same singularity. A quaternion representation of this joint is free of singularity, however.

BDF: Backward Difference Formula

A family of integration methods based on backward differences. Specifically, the BDF family approximates the value of the time derivative $\dot{x}$ in the differential equation $F(x, \dot{x}, t) = 0$, with a polynomial interpolant using $p - 1$ previously computed values for a method of order p. BDF methods are generally implicit and have large stability regions.

Bilateral constraint

A general constraint relation which is satisfied as a strict equality. For such a constraint, the corresponding multiplier is generally unrestricted, unless another general constraint is imposed on this multiplier.

Bisymmetric matrix

A matrix M of size $n \times n$ is bisymmetric if it can be split as $M = L + D - L^T$ where D is block diagonal and symmetric, and L is strictly lower diagonal. A bisymmetric matrix is positive definite as long as D is since $x^T M x = x^T D x$, the other terms cancel each other.

BLAS: Basic Linear Algebra Subroutines

A FORTRAN programming interface definition for software libraries of subroutines implementing fundamental dense linear algebra operations. It is divided into three levels of increasing algorithmic complexity. The Level 1 interfaces describe vector-vector and scalar-vector operations, those in Level-2 describe matrix-vector operations, and those in Level-2 describe matrix-matrix operations. The list of operations performed by the BLAS was judiciously chosen to express all the fundamental operations needed in the development of linear algebra algorithms and is a good template to follow when implementing a library of linear algebra operations for special matrix and vector formats.

Body frame

A rigid frame of reference attached to a moving body. A point y' in this frame has inertial coordinates $y = x + Ry'$, where x is the center of mass of the body. If the point y' is fixed in the body frame, then, its velocity is $\dot{y} = \dot{x} + \widehat{\omega} R y'$, where $\dot{x}$ is the velocity of the center of mass and ω is the angular velocity of the body as seen in the inertial frame. The body frame is noninertial.

Box friction

A dry friction model in which the tangential forces are such that they maintain zero tangential velocity in each perpendicular direction as long as the corresponding multiplier is within the box bounds $[-f^{(\max)}, f^{(\max)}]$. Otherwise, the tangential force takes the corresponding maximum value, opposing a constant force to the sliding velocity. Box friction differs from Coulomb friction in that it is not isotropic, each sliding direction being considered independently, and the bounds are independent of the normal force.

Calculus of variations

The study of derivative of functionals, usually defined as integrals of functions $f : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}$, evaluated over paths $x : \mathbb{R} \mapsto \mathbb{R}^n$, so $S[x] = \int_{t_0}^{t_1} \mathrm{d}s f(x(s), \dot{x}(s))$. Variational calculus allows to derive the stationary conditions for $S[x]$, which can then be used to solve for the path $x(t)$.

Cartesian coordinates

Given an orthonormal basis $\{w^{(i)}\}$ of $\mathbb{R}^n$, any n-dimensional vector x can be written as a linear combination $x = \sum_i \alpha_i w^{(i)}$. The coefficients α_i

are called the coordinates of vector x with respect to the basis $\{w^{(i)}\}$. When the basis vectors are chosen so that the components are $w_i^{(i)} = 1$ and $w_j^{(i)} = 0, i \neq j$, the coefficients α_i are the Cartesian coordinates.

Cayley-Hamilton theorem

A real or complex square $n \times n$ matrix A is a root of its own minimum polynomial $r(\lambda)$, so that $r(A) = \prod_{i=1}^{m}(A - \lambda_i I_n) = \sum \alpha_k A^k = 0$, where 0 here means the $n \times n$ zero matrix, and A^k is the kth power of matrix A with the standard definition that $A^0 = I_n$.

Central force

A force between two bodies acting in the line joining the mass centers.

CG: Conjugate Gradient

An algorithm for the solution of linear systems of equations of the form $Ax = b$, where A is a real, square, symmetric and positive definite $n \times n$ matrix, and b is a real n-dimensional vector. The CG method is special because it only requires matrix-vector multiply operations, namely, Ay, for some n-dimensional vector y, as well as n-dimensional inner products of n-dimensional vectors. The CG method can exactly solve a linear system n stages in infinite precision, but it is considered an iterative method with superlinear convergence in finite precision. It is the archetype of Krylov subspace methods.

Characteristic polynomial

Given a real or complex square $n \times n$ matrix A, the characteristic polynomial $p(\lambda)$ is given by the determinant $p(\lambda) = \det(A - \lambda I_n)$, which is a monic polynomial of degree n and which can be written in factored form as $p(\lambda) = \prod_{k=1}^{m}(\lambda - \lambda_k)^{\mu_k}$, where $\lambda_k, k = 1, 2, \ldots, m$, with $m \leq n$, are the m distinct roots of $p(\lambda)$, and the integers $\mu_k \in \mathbb{N}$ are the multiplicities of each root so that $\sum_{k=1}^{m} \mu_k = n$.

Chart

A local coordinate system of a manifold $\mathcal{M}$ in a neighborhood of a point $x \in \mathcal{M}$ which maps points in $\mathbb{R}^n$ to points in $\mathcal{M}$. For a differentiable manifold, the charts are differentiable functions of the Cartesian coordinates. For a smooth manifold, the charts are infinitely differentiable and invertible whenever two of them overlap.

Condition number

Given a matrix A of size $n \times n$, the *condition number* is the ratio of moduli of the largest to the smallest eigenvalue. A singular matrix thus has $\mathrm{cond}(A) = \infty$. The condition number plays an important role in stability analysis of algorithms.

Cone

A cone is a point set $\mathcal{K} \subseteq \mathbb{R}^n$ which is closed under positive linear combinations so that $\alpha x + \beta y \in \mathcal{K}$ for any two elements $x, y \in \mathcal{K}$ and any positive scalars $\alpha, \beta > 0$. The cone is finitely generated if there are m basis elements, $x_i \in \mathcal{K}, i = 1, 2, \ldots, m$, such that any element $y \in \mathcal{K}$ can be represented as a positive linear combination of the x_is, $y = \sum \alpha_i x_i, \alpha_i \geq 0, i = 1, 2, \ldots, m$.

Configuration space

For a mechanical system, the set $\mathcal{Q}$ which can precisely and completely describe the state of the bodies in the system but without regards to the velocities. For systems of unconstrained point masses, this reduces to $\mathcal{Q} = \mathbb{R}^n$ but for a single rigid body in three dimensions, we have $\mathcal{Q} = \mathbb{R}^3 \times SO(3)$. In general, $\mathcal{Q}$ is a differentiable manifold.

Conservative systems

Mechanical systems which conserve energy over time. The study of these systems is central to analytic mechanics.

Constraint

Any restriction on the generalized coordinates q, the generalized velocities $\dot{q}$, any of the generalized forces $f^{(j)}, j = 1, 2, \ldots$ of a mechanical system. Constraints are written as m nonlinear relations $b(q, \dot{q}, f^{(1)}, f^{(2)}, \ldots, t) \geqq 0$, where the inequality sign is understood component-wise, each element function $b^{(i)}$ satisfying either an equality or an inequality relation, the bilateral and unilateral components, respectively. A constraint alters the motion of the system and thus generates constraint forces $f^{(i)}$ for each of the $b^{(i)}$. For each $b^{(i)}$, the constraint force $f^{(i)}$ is the product of a scalar—the constraint multiplier—with a generalized direction vector.

Constraint elimination

A bilateral holonomic kinematic constraint of the form $g(q, t) = 0$ with $g : \mathbb{R} \times \mathcal{Q} \mapsto \mathbb{R}$, restricts the configuration space $\mathcal{Q}$ to a differentiable manifold $\mathcal{M} \subset \mathcal{Q}$ with dimension $\dim(\mathcal{M}) = n - 1$. If the manifold $\mathcal{M} = g^{-1}(0)$ allows for a global chart, one can globally change of coordinates to produce the $n - 1$-dimensional generalized coordinates, $\tilde{q} \in \mathcal{M}$. In general, constraint elimination is always possible locally but, numerically, this requires the use of orthogonal factorization (QR), subspace identification, as well as subspace projections. Each of these operations is generally expensive to perform numerically. For the case of unilateral constraints, one must also determine when they are active, i.e., when they should be satisfied as equality relations, or when they are breaking, i.e., when the relation is a strict inequality.

Constraint multiplier

Given a general constraint relation $b^{(i)} \geqq 0$, the generalized force is the product of a scalar, which is called the constraint multiplier, and a vector. The reason is that the mechanical system has n degrees of freedom before the addition of the constraint and, assuming it is a well posed problem, the equations of motion consist of n differential relations in the n degrees of freedom. Adding a well behaved constraint relation to the system removes one of these degrees of freedom and thus must introduce an additional variable, lest the system of differential equations become overdetermined. If it were overdetermined, this would mean that one of the equations could be eliminated—this last constraint relation for instance—without altering the problem. Otherwise, there are then $n + 1$ differential relations in $n + 1$ unknowns.

Constraint stabilization

A numerical procedure designed to guarantee that constraint violation for each and every general constraint component is kept small. Such a procedure necessarily corresponds to additional forces applied to the system. Such forces are also necessarily oscillatory in nature, since they act to keep the constraint violations near the desired value of 0, and they may or may not be dissipative.

Constraint violation

For a generic constraint component function $b^{(i)}(q, \dot{q}, f^{(1)}, f^{(2)}, \ldots, t)$, any configuration of the mechanical system such that the corresponding equality or inequality relation is not satisfied is said to violate the constraint. When this happens, either the value of the component function $b^{(i)}$ for an equality relation, or its negative part for an inequality relation, respectively, is the numerical value of the constraint violation.

Contact constraint

A general constraint relation which is kinematic and bilateral is called a contact constraint. This is usually written as $c(q, t) \geq 0$, with Jacobian $C = \partial c / \partial q$, with multiplier ν (for normal), and producing the constraint force $C^T \nu$. When scleronomic, contact constraints are ideal. In addition, the multiplier ν satisfies both a non-negative and a complementarity condition so that $0 \leq c \perp \nu \geq 0$. Contact constraints are used to impose nonpenetration conditions between solids. Contact constraints produce discontinuities at such times when the inequality condition becomes satisfied as an equality, i.e., such instants t_0 such that $c(q(t_0 + \epsilon), t_0 + \epsilon) > 0$ for $\epsilon > 0$, but $\lim_{\epsilon \downarrow 0} c(q(t_0 + \epsilon), t_0 + \epsilon) = 0$. Such discontinuities are called impacts.

Convex combinations

A weighted sum of elements such that the weights are all non-negative

and sum up to 1. For a finite set of points $x_i \in \mathcal{A}, i = 1, 2, \ldots, n$, this is combinations of the form $\sum \alpha_i x_i$ where $\alpha_i \geq 0$ for all $i = 1, 2, \ldots, n$.

Convex hull

Given a point set $\mathcal{A}$, the convex hull $\mathsf{co}(\mathcal{A})$ is the set containing all the elements in $\mathcal{A}$ as well as all the possible convex combinations of the members of $\mathcal{A}$.

Coordinate reduction

See constraint elimination in this glossary.

Copositive matrix

An $n \times n$ real matrix A such that for any n-dimensional real non-negative vector x with $x_i \geq 0$, $x^T A x \geq 0$. When $x^T A x > 0$ for any non-negative vector $x \neq 0$, the matrix A is strictly copositive. Positive semi-definite matrices are copositive and positive definite matrices are strictly copositive. As a simple example, if matrix A can be split as $A = M + U$ where M is positive definite and U is nonnegative, then, $x^T A x = x^T M x + x^T U x \geq x^T U x$. This last term is nonnegative since $[U]_{ij} \geq 0$, and $x_i \geq 0$ by assumption.

Copositive plus matrix

An $n \times n$ matrix A that is copositive and such that whenever $x^T M x = 0$ for $x \geqq 0$, then, $(A + A^T)x = 0$. This definition follows directly from the termination criterion analysis of the Lemke algorithm.

Coriolis force

In a noninertial frame which has local angular velocity ω with respect to the inertial frame, any object not acted upon by any force (or negligibly so) and moving with linear velocity $\dot{x}$ experiences an acceleration $\ddot{x} = -\widehat{\omega}\dot{x}$, where ω and x are vectors measured in the noninertial frame. Note the factor of two and the fact that the body must be in motion with respect to the noninertial frame. These spurious forces are thus unrelated to the gyroscopic forces of rigid bodies.

Cottle-Dantzig algorithm

A principal pivot method to solve $\mathsf{LCP}(M, q)$ for given real, symmetric, positive semi-definite $n \times n$ matrix M and real n-dimensional vector q. This algorithm proceeds by solving principal subproblem of increasing dimension n_k, starting with $n_1 = 1$. At each stage k, the solution of the subproblem with size $n_{k+1} = n_k + 1$ starts from the solution of the previous problem of size n_k and uses pivoting operation to remove infeasibility.

Coulomb friction

The model of tangential force at a contact point. For contacting bodies with normal force $\nu > 0$, the tangential force $f^{(tan)}$ acts in the contact plane,

preventing any relative tangential motion as long as $\|f^{(tan)}\| < \mu_s \nu$, where $\mu_s > 0$ is the coefficient of static friction, and depends on the properties of the contacting materials. Sliding occurs when the condition cannot be met, at which point $\|f^{(tan)}\| = \mu_k \nu$ where $\mu_k > 0$ is the coefficient of kinetic friction, and $\|f^{(tan)}\|$ opposes the sliding velocity to maximize dissipation. In general, $\mu_k < \mu_s$ and the coefficient of kinetic friction is weakly dependent on the sliding velocity, so that $\mu_k(0) = \mu_s$ and $\mu_k(\infty)/\mu_s$ is in the range 0.8–0.9. Also, the net tangential force is almost independent of the contacting areas.

D'Alembert's principle

The variational principle stating that virtual work vanishes along the physical trajectory. This is more general than the principle of least action because it does not require the mechanical system to be conservative. Subjecting an otherwise conservative system with generalized coordinates q and action S to polygenic forces f, d'Alembert's principle states that $\delta S + \int f^T \delta q = 0$ over the entire trajectory. The virtual displacements are assumed to satisfy all other constraints. See also the entry for Fourier's inequality.

DAE: Differential-Algebraic Equations

For the differential equation $f(\dot{y}, y, t) = 0$, (see DE) for an n-dimensional vector $y(t)$, if the Jacobian matrix $\partial f/\partial y$ is singular, having only rank $m < n$, then it is a differential algebraic equation. The equations can be split locally into a differential part and an algebraic part by introducing the m-dimensional vector x and the $n - m$-dimensional vector z which satisfy the combined equations $\tilde{f}(\dot{x}, x, z, t) = 0$, the differential part, and $\tilde{g}(x, z) = 0$. The index of the DAE is the number of derivatives of the function $\tilde{g}$ one must apply in order to locally eliminate the z (algebraic) variables from the system. DAEs of index one can be solved with standard ODE methods, and DAEs of index 2 can be solved using special purpose methods. DAEs of index 3 and above, the higher order DAEs, are generally difficult to solve.

DAI: Differential-Algebraic Inequalities

A differential algebraic inequality is a differential inequality of the form $f(\dot{y}, y) \gtreqless 0$, (see DE), where the inequality sign is understood component-wise meaning that each equation is either an inequality or an equality, such that the matrix $\partial f/\partial \dot{y}$ is singular, and where the inequality sign is understood component-wise. Except for linear systems, this type of problem has not been studied widely.

DI: Differential Inclusion

This is much like a differential equation (DE) but here, the simple explicit case reads: $\dot{x} \in f(x)$, i.e., the derivative of function $x(t)$ is considered to

be a set. Differential inclusions occur in the study of nonsmooth problems such as dry Coulomb friction.

Differential form

The integrands of oriented integrals over p-dimensional domains in $\mathbb{R}^n$. Given a p-form ω^p and a p-dimensional surface $S \in \mathbb{R}^n$, the integral $\int_S \omega^p$ is a scalar. Smooth differential p-forms can be differentiated with the *exterior derivative*, to produce a $p+1$-form, e.g., $\eta^{p+1} = \mathbf{d}\omega^p$.

Discrete action

Given a discrete Lagrangian $\mathbb{L}_d(q_0, q_1, h)$, for some configuration manifold Q, the discrete action for fixed time step h is the sum $\mathbb{S}_d(q_0, \ldots, q_N, h) = \sum_{k=0}^{k=N-1} \mathbb{L}_d(q_k, q_{k+1}, h)$, and the discrete action for variable time step h_k is the sum $\mathbb{S}_d(q_0, \ldots, q_N, h_1, \ldots, h_N) = \sum_{k=1}^{k=N-1} \mathbb{L}_d(q_k, q_{k+1}, h_{k+1})$. The discrete principle of least action states that the discrete action is stationary with respect to variations of the arguments, either q_k or q_k and h_k, which satisfy all constraints. For variable time step, the constraint $\sum_k h_k = t_1 - t_0$ must be imposed.

Discrete energy

Given a discrete Lagrangian $\mathbb{L}_d(q_0, q_1, h)$, the discrete energy is the partial derivative $\mathbb{E}_d(q_0, q_1, h_1) = -\partial \mathbb{L}_d(q_0, q_1, h)/\partial h_1$. This can be conserved if using a variable time step. For fixed time step, the discrete energy fluctuates.

Discrete Euler-Lagrange equations

Given a discrete action for fixed or variable time step, the discrete Euler-Lagrange equations are the sufficient conditions for the discrete action to be stationary with respect to variation of the generalized coordinates q_k, subject to constraints. For the fixed time step case, the stationary conditions are $D_1\mathbb{L}_d(q_k, q_{k+1}, h) + D_2\mathbb{L}_d(q_{k-1}, q_k, h) = 0$, and this defines a nonlinear three-term recurrence relation $\Phi_{\mathbb{L}_d} : Q \times Q \mapsto Q \times Q$, where $(q_{k+1}, q_k) = \Phi_{\mathbb{L}_d}(q_k, q_{k-1})$. For the variable time step case, the stationary conditions are subject to an additional constraint, namely, that $\sum_k h_k = t_N - t_0$. The resulting discrete Euler-Lagrange equations then preserve the discrete energy.

Discrete Lagrangian

Given a Lagrangian $\mathcal{L} : TQ \mapsto \mathbb{R}$, defined as a $\mathcal{L}(q, \dot{q})$, for some configuration manifold Q, a discrete Lagrangian is the approximation of the integral $\mathbb{L}_d(q_0, q_1, h) = \int_0^h \mathrm{d}s \mathcal{L}(q(s), \dot{q}(s))$, which is a function of the endpoints q_0, q_1.

Discrete time-stepper

A computational method for a dynamical system producing a state vector x_{k+1} at discrete time $k+1$ as a function of the state vector x_k at time k. It is a numerical representation of the map $\Phi : X \mapsto X$ such that that $x_{k+1} = \Phi(x_k)$, where $x_k \in X$ and X is the configuration space of the dynamical system.

Discrete-variational method

An application of the discrete principle of least action to produce a discrete time stepping scheme.

Discretization error

Numerical errors resulting from using discrete approximations of continuous functions.

Dissipative systems

Nonconservative systems which monotonically dissipate energy over time.

Driver

A driver is a nonholonomic, bilateral, rheonomic constraint imposing a velocity restriction. A hinge driver for instance might impose a fixed hinge velocity which amounts to a relative angular velocity between the jointed bodies.

Dry friction

Any model of the tangential friction forces between contacting bodies which exhibits stiction or static friction.

Dynamic

That which changes in time, without regards to how or why it changes.

Dynamics

The study of causes and effects of forces in a mechanical system.

Effort constraint

General constraint relations which do specifically depend on the generalized forces or the constraint multipliers are called effort constraints. Such constraints are always nonideal and nonholonomic, and they may be either bilateral or unilateral. Unilateral effort constraints produce discontinuities in both constraint forces and in generalized velocities. Effort constraints serve to include constitutive laws which are experimentally known relations, linear or otherwise, between forces and kinematic variables. A typical example is the Coulomb friction cone condition which imposes a restriction between the magnitude of the tangential and normal forces at a contact point. This nomenclature is borrowed from the systems engineering literature.

Eigenvalues and eigenvectors

For a square $n \times n$ matrix A, an n-dimensional vector x and a scalar λ are a right eigenvalue-eigenvector pair if $Ax = \lambda x$. Similarly, y and ν are a left eigenvalue-eigenvector pair when $y^H A = \nu y^H$, where y^H is the Hermitian conjugate of y. A right eigenvector x satisfies the homogeneous linear system $(A - \lambda I_n)x = 0$ and a left eigenvector y satisfies the homogeneous linear problem $y^H(A - \nu I_n) = 0$, which means that both $A - \lambda I_n$ and $A - \nu I_n$ are rank deficient matrices. Therefore, the eigenvalues λ and ν are in any case roots of the minimum polynomial of A. Each eigenvalue of matrix A has at least one right and one left associated eigenvector. In the case where the set of eigenvectors of matrix A span all of $\mathbb{R}^n$, the matrix A is said to be diagonalizable.

Energy

A fundamental principle in physics, energy is not easily defined in terms of other quantities. The change of energy of a mechanical system which moves from $q(t_0)$ to $q(t_1)$ along a path C is the negative of the net work done along this path, so energy can be defined as "the ability to do work", though this is not always consistent. Energy has multiple forms such as kinetic, potential, as well as thermal, and is additive. In classical mechanics, energy is the constant scalar function of the generalized coordinates and velocities which corresponds to time translation symmetry via Noether's theorem in systems which are not explicitly dependent on time. This results in defining energy as the sum of kinetic and potential terms. In classical mechanics, energy can be negative and is only defined up to an arbitrary constant.

Energy momentum preserving integrator

A numerical integration method designed to preserve the energy and momentum when applied to a conservative mechanical system.

Euler angles

Any set of three angles describing successive rotations about the principal coordinate axes in $\mathbb{R}^3$ to generate an arbitrary rotation matrix so that $R = R_{n_3}(\alpha_3)R_{n_2}(\alpha_2)R_{n_1}(\alpha_1)$, where $R_{n_i}(\alpha_i)$ is a rotation by angle α_i about local or global coordinate axis n_i. As long as no two successive axes n_i, n_{i+1} are coincident, this construction can generate all rotation matrices $R \in SO(3)$. There are 24 different anti-clockwise conventions. Computing the angles from a given R is not always unique however because $SO(3)$ is not a flat manifold. The degeneracy means, in particular, that it is not always possible to recover the $\dot{\alpha}_i$s from the angular velocity vector ω. Indeed, the Jacobian mapping ω to the $\dot{\alpha}_i$s is rank deficient at the degenerate points and therefore, not invertible.

Euler-Lagrange equations

Given a scalar function $f(x, \dot{x})$ of the n-dimensional time-dependent vector $x(t)$ and its velocity $\dot{x}(t)$, the Euler-Lagrange equations read

$\frac{\mathrm{d}}{\mathrm{d}t}(\partial f/\partial \dot{x}^T) - \partial f/\partial x^T = 0$, and any solution of these equations is then a stationary point of the functional $S[\dot{\gamma}] = \int_{t_0}^{t_1} \mathrm{d}s f(x(s), \dot{x}(s))$, where $\dot{\gamma}$ is the path $(x(t), \dot{x}(t))$ for $t \in [t_0, t_1]$.

Extended reals

The set of real numbers with the explicit addition of $\pm\infty$, where the usual rules involving infinities are followed, namely, (i) for each $x \in \mathbb{R}$, $x \pm \infty = \pm\infty$, (ii) for each real $x \in \mathbb{R}$, $x/\pm\infty = 0$, (iii) for each non-zero real $x \in \mathbb{R}, x \neq 0$, $x/0 = \mathrm{sgn}(x) \cdot \infty$, (iv) non-zero real $x \in \mathbb{R}, x \neq 0$, $\pm x\infty = \pm\infty$, (v) $0/0$ is undefined, and (vi) $\pm\infty \cdot 0$ is undefined.

Exterior derivative

The generalization of the vector differential operators div, grad and curl, to arbitrary differential forms defined on manifolds. For a scalar function $\theta : \mathbb{R}^n \mapsto \mathbb{R}$, the exterior derivative $\mathbf{d}\theta$ reduces to the familiar gradient.

Force

That which can cause changes in the state of uniform linear motion of physical bodies.

Fourier's inequality

When applying d'Alembert's principle of virtual work, it is assumed that the configuration manifold Q is an open set and thus, the virtual displacements δq can be chosen in any open neighborhood of the configuration variable $q \in Q$ at a given time t. When Q has a closed boundary though, the virtual work may not necessarily vanish and in fact, d'Alembert's principle is replaced by the inequality $\delta S + \int f^T \delta q \leq 0$. This is known as Fourier's inequality and is central to the analysis of any nonsmooth problem. In fact, this inequality is a form of complementarity condition since it holds strictly as an equality whenever q is away from the closed boundary, and strictly as an inequality when q is on a boundary.

Galilean relativity

Under the assumption that time is universal, advancing at the same rate and accessible instantly everywhere, the laws of physics—and thus the result of any physical experiment—are identical in all frames of reference which move at constant linear velocity with respect to each other. The mathematical formulation of the laws of physics are therefore explicitly independent of the origin of the coordinate system, its absolute orientation, and its absolute velocity. Any two reference frames which differ by a translation of the origin, a rotation of the coordinate axes, or a constant linear relative velocity, are rigorously equivalent for the purpose of describing the laws of motion.

The assumption that time is universal breaks down when moving at very

high speed or subjected to very high acceleration as explained in Einstein's special and general relativity theories.

Gather operation

In numerical linear algebra, any operation which involves copying data that is not stored contiguously in memory into a contiguous storage location.

Gauss' principle of least constraint

The postulate that, for a mechanical system with momentum p and generalized forces f subject to general constraint relations, the trajectories satisfying the constraints minimize the scalar function $\|\dot{p} - f\|^2$. Being based on accelerations, Gauss' principle of least constraints is less fundamental than the principle of least action.

Gauss-Seidel

An iterative process that is used to solve linear and nonlinear equations by successively solving simple one-dimensional problems. The convergence is linear.

GEMM: General Matrix Multiply and add

The fundamental numerical linear algebra operation consisting of computing the updated product $C \leftarrow \alpha AB + \beta C$ for given general, real, dense matrices A, B and C of compatible sizes, and α and β are scalars.

Generalized coordinates

For a mechanical system with n degrees of freedoms with configuration space Q, the generalized coordinates $q = (q_1, q_2, \ldots, q_n)^T$ are n-tuples which can represent all the elements in Q. The generalized coordinates may or may not form a vector space. The term generalized coordinates appeared in Hamilton's work (as quoted in [284] and [173] for instance) to contrast with Cartesian coordinates and to specifically include the case of curvilinear coordinates which do not form a vector space, except locally. The Euler-Lagrange equations of motion are invariant under continuous coordinate transforms and are thus valid for any choice of generalized coordinates.

Generalized directional derivative

A generalization of the directional derivative for nonsmooth functions $f : \mathbb{R}^n \mapsto \mathbb{R}^m$ satisfying a Lipschitz condition near some point x, so that $\|f(y) - f(z)\| \leq K\|y - z\|$ whenever y, z are within a sufficiently small neighborhood of x. The generalized directional derivative $f^\circ(x; v)$ in the direction of $v \in \mathbb{R}^n$ is defined as the limit of the supremum of the ratios $(f(x + \epsilon v) - f(y))/\epsilon$ as $\epsilon \to 0$ from above and $y \to x$. This reduces to the one-sided Gâteaux derivative when choosing $y = x$, and is the Gâteaux derivative when ϵ is not restricted to $\epsilon > 0$. This is sometimes referred to as the Clarke derivative after its main author [66].

Generalized forces

For a mechanical system with generalized coordinates $q \in Q$, the generalized forces are the vectors f so that work, the line integral $W = \int f^T \, \mathrm{d}q$, should have the same value as when evaluated in the original Cartesian coordinate system definition. If $\tilde{f}$ is the force vector in Cartesian coordinates, the generalized forces become $f = (\partial q / \partial x)^{-T} \tilde{f}$, For a monogenic force with potential $\tilde{V}(x)$, first express the Cartesian coordinates as functions of the curvilinear ones (this is where the inverse comes from), $x(q)$, and then, compute $V(q(x)) = \tilde{V}(x(q))$. The generalized forces can then be computed as $f = -\partial V(q) / \partial q^T$, as usual. A similar argument applies to dissipative forces derived from Rayleigh functions. For other cases, the inverse Jacobian is always needed.

Generalized gradient

For a nonsmooth function $f : \mathbb{R}^n \mapsto \mathbb{R}^m$ which is piecewise continuous, the generalized gradient $\partial f(x)$ is the convex hull of the set of all the gradients of different continuous branches of f in a small neighborhood of the point x, in the limit where the neighborhood vanishes. For instance, the generalized gradient of the absolute value function $|x| = \sqrt{x^2}$ is the set $[-1, 1]$.

Generalized momentum

In analytic mechanics, given a Lagrangian $\mathcal{L}(q, \dot{q})$, the generalized momentum is the derivative with respect to the generalized velocity $p = \partial \mathcal{L}(q, \dot{q}) / \partial \dot{q}^T$. This definition reduces to the Newtonian mechanics, namely, $p = m\dot{x}(t)$, for the case of point particles in Cartesian coordinates but generalizes to curvilinear coordinate systems as well.

Generalized velocities

Given a configuration space Q for a mechanical system with generalized positions q. If $q(t)$ is a permissible trajectory so that $q(t) \in Q$ for all times t, then, the time derivatives $\dot{q}$ are the generalized velocities. The distinction is made to correctly compute the velocities in a curvilinear system. Assuming for instance that $Q = \mathbb{R}^n$, and choosing a curvilinear coordinate system $q(x)$, the generalized velocities then become $\dot{q} = (\partial q / \partial x)\dot{x}$, and the term in parenthesis is the Jacobian of the coordinate transformation.

Ghost particles

In a mechanical system, a point particle which has nonpositive mass. Ghosts are only present in the analytic formulation of mechanics when constraints are introduced. Their coordinates are the values of the constraint multipliers and they usually have exactly zero mass. The framework of analytic mechanics allows to treat ghost particles systematically along with all other physical bodies using just one universal set of rules. An isolated ghost particle with generalized coordinate q and finite negative mass $-m$ subject to potential function $-V(q)$ has the same mechanics as that of a point particle

of mass m subject to the potential $V(q)$ but it satisfies a "maximum action principle". The term ghost particle is also used in high energy physics when constraints are imposed on fields, which is analogous to this case.

Gravity

A central force of attraction between any two physical bodies whose potential is inversely proportional to the distance and directly proportional to the product of the masses: $V = G_u m_1 m_2 \|x^{(1)} - x^{(2)}\|^{-1}$, where G_u is the universal gravitational constant. Near the surface of the Earth, which is a large distance from the center, neglecting the reaction force on the Earth itself, this potential reduces to the simple form $V = m a_g u^T x$ for a point particle with mass m and position x, where a_g is the constant acceleration of gravity and u is a unit vector pointing upwards.

Group

A group consists of a set $\mathcal{A}$, a unit element $e \in \mathcal{A}$, and a binary operation $\cdot$ acting on elements of $\mathcal{A}$, so that: (i) $\mathcal{A}$ is closed under the action of $\cdot$ so that for $x, y \in \mathcal{A}$, $x \cdot y \in \mathcal{A}$, (ii) each element $x \in \mathcal{A}$ is invariant when operated on by e so $e \cdot x = x \cdot e = x$, (iii) for each element $x \in \mathcal{A}$ there exists an inverse $x^{-1} \in \mathcal{A}$ such that $x \cdot x^{-1} = x^{-1} \cdot x = e$. When the binary operation commutes so that $x \cdot y = y \cdot x$ for all pairs of elements $x, y \in \mathcal{A}$, the group is Abelian, and non-Abelian otherwise.

Gyroscope

A gyroscope is a device consisting of an axially symmetric rigid body attached by a ball joint to a point on the symmetry axis and rotating at very high speed about the symmetry axis. When the rotation speed is high enough, the body keeps a fixed orientation in space which is useful in the navigation of aircrafts and missiles. The mathematical representation of this problem is called the Lagrange top.

Gyroscopic forces

The extra force terms arising from a configuration dependent mass matrix $M(q)$ when evaluating the Euler-Lagrange equations of a given mechanical system. For a rigid body, this is $\widehat{\omega}\mathcal{I}\omega$ in the inertial frame where $\mathcal{I}$ is the inertia tensor and ω is the angular velocity vector as seen from the inertial frame, and to $\widehat{\omega'}\mathcal{I}_0\omega'$ in the body frame, where $\mathcal{I}_0$ is the constant inertia tensor and ω' is the angular velocity evaluated in the body frame. Gyroscopic forces are workless as easily seen in this case since $\omega^T \widehat{\omega} M(q)\omega = 0$. By extension, any force term that is workless is called a gyroscopic force. Rigid body gyroscopic forces are also called non-inertial forces. They are not Coriolis forces, however, since the gyroscopic forces do not vanish in any frame of reference.

Hinge joint

A joint such that, starting from either body, the motion of the second is a pure rotation about an axis both fixed in the body frame of the first. Mechanical joints are made by firmly attaching a peg on a body and drilling a hole in the second. The common axis shared by the peg and the hole is the hinge axis. Each body then has an axis which is fixed in both orientation and position. The effect of the constraint is to maintain colinearity between these two axes as well as preventing translation along the rotational axis. Hinge joints are also known as revolute joints. Mathematical definitions of hinge joints must be carefully designed lest they do not distinguish between colinearity and *anti*-colinearity. A quaternion based definition does distinguish the two cases.

Holonomic constraint

A general constraint relation which is satisfied as a strict equality and which depends only on the generalized coordinates and time. Holonomic constraints are denoted as the restriction $g(q,t) = 0$ and the corresponding multiplier is written as λ and they generate generalized constraint forces as $G^T\lambda$ where the matrix $G = \partial g/\partial q$ is called the Jacobian of the constraint. Holonomic constraints are always bilateral and ideal when they are scleronomic. They can sometimes be eliminated if the structure of the manifold $\mathcal{M}$ they impose on the configuration space Q is simple enough.

Homogeneous function

A function $f(x)$ such that $f(\alpha x) = \alpha^\kappa f(x)$ is homogeneous of degree κ. For a function of multiple arguments $g(x^{(1)}, x^{(2)}, \ldots, x^{(m)})$, the function is homogeneous of degree κ_i in the ith argument when the identity $g(x^{(1)}, x^{(2)}, \ldots, \alpha x^{(i)}, \ldots x^{(m)}) = \alpha^{\kappa_i} g(x^{(1)}, x^{(2)}, \ldots, x^{(i)}, \ldots x^{(m)})$ is satisfied. From Euler's homogeneous function theorem, we have $(\partial f(x)\partial x)x = \kappa f(x)$, and similarly for homogeneous functions of multiple arguments.

Homokinetic joint

A joint such that, starting from the first, the motion of the second body is a rotation about a point fixed in the body frame of the first but without rotation about the polar axis. This means that if we attach both bodies with separate hinge joints to the world, the hinge angle of the first and second bodies will be identical. This is a superior rotational transmission device when compared to a Hook joint but is more difficult to build and more fragile.

Hook joint

A joint such that, starting from either body, the motion of the second is a rotation about a point fixed in the first body frame but restricted to maintain a perpendicularity condition between an axis fixed in the body

frame of the first and an axis fixed in the body frame of the second. The physical construction of a Hook joint consists of a pair of crossed axis soldered together, and each axis then mounted on a hinge in one of the two bodies. The resulting mechanism allows to transmit rotation about an axis fixed in the first body to the second, without requiring the two rotation axes be aligned. However, the rotational velocity of the second body about its rotational axis varies with the net rotational angle.

Ideal constraints

General restrictions on a mechanical system whose constraint forces do not change the energy of the mechanical systems, i.e., those whose mechanical action is workless. When the constraint force does work on the system, dissipative or otherwise, the constraint is called nonideal, as in the specific case of *effort constraints*.

Impact

Any physical process which produces a finite impulse k for an arbitrary small time interval, or at least, a time interval which is much smaller than the resolution of a measurement apparatus or the time step of a simulation. Impacts serve to decouple the time scales in a simulation or an analysis of the motion, resolving the action of the impulsive forces separately from all others whose impulses vanish over the chosen time interval.

Impulse

The time integral of a force f over an interval $[t_0, t_1]$ is an impulse, $k = \int_{t_0}^{t_1} f$. Given Newton's second law written as $\dot{p} = f$, where p is the momentum, the impulse is thus equal to the net change in momentum between the times t_0 and t_1. Though the definition of impulse makes no reference to the size of the interval $[t_0, t_1]$, it is customary to use the term mostly for the case of an impact. Indeed, if a force is very large but only over a very short interval of time, so that the limit $\lim_{\epsilon \downarrow 0} \int_t^{t+\epsilon} \mathrm{d}s f = k$, the force is called impulsive and it produces a discontinuity in the velocity since we still have $p(t+\epsilon) - p(t) = k$, which means that, assuming constant mass m, the velocity $\dot{x}$ has a jump discontinuity at t.

Inertia

That which restrains the action of a force in changing the quantity of motion. For a point particle, this is the mass and for a rigid body, the inertia tensor. Inertia is unrelated to dissipation.

Inertia tensor

A positive definite matrix describing the rotational inertia of a rigid body. A rigid body in a state of constant rotation about a given axis opposes a different inertia to torques applied along any of the three coordinate axes. Given there are three possible independent axes of rotation, there is a total

of six combinations to consider. These are collected in a symmetric matrix $\mathcal{I}$ in the world frame and is a rank 2 tensor. The inertia tensor depends on the distribution of mass within the rigid body and can be evaluated in the body frame as the constant matrix $\mathcal{I}_0$. Since $\mathcal{I}_0$ is symmetric and positive definite, it can be diagonalized with an orthonormal coordinate transformation. When this is chosen so the diagonal values are sorted in decreasing order, the coordinate axes of this frame are called the principal axes and the corresponding three inertia value are the principal inertiae.

Inertial frame

Any frame of reference in which the three laws of Newtonian mechanics can be verified. In particular, any frame in which there is no observable Coriolis force. Any two inertial frames of reference differ by a translation, a rotation, and a constant relative linear velocity, but rigorously share the same universal time, both in absolute value and in the rate of flow. The relationship between two inertial frames is called a Galilean transformation and these form a group. The independence of the laws of physics under Galilean transformations is called the Galilean principle of relativity. Since all inertial frames are equivalent, any such is called "the inertial frame".

Interactive physics

A computer simulation of a physical system which responds in real-time with minimal delays to user inputs, and is used to drive multisensory devices such as, for instance, real-time 3D graphics displays, motion platforms, haptic devices, and sound systems.

Interactive simulations

Any simulation of a system, physical or otherwise, designed to respond to user inputs with visual or other sensory cues without significant time delay.

Joint

A joint is a bilateral, kinematic, ideal, scleronomic constraint relation, which involves either two bodies or a single body. In the latter case, the body is said to be constrained to the world or the inertial frame. Mathematical joints are idealization of mechanical devices designed to connect various moving parts. Systems composed only of bodies connected with joints allow a particularly simple description of the motion based on kinematic analysis, the topology of the interconnection graph, and constraint elimination. Such techniques are known as articulated body or composite body methods.

Joint angles

The coordinates describing the relative motion of two bodies connected by a joint. Since the relative motion of two bodies is the manifold $\mathbb{R}^3 SO(3)$, allowing arbitrary relative translations and rotations, a joint restricts this

relative motion to a manifold $\mathcal{M} \in \mathbb{R}^3 SO(3)$ of dimension $m = \dim(\mathcal{M}) < 6$. After choosing a coordinate chart on this manifold, the relative motion is described by the coordinates of that chart. Since the manifold $\mathcal{M}$ might not be flat, the general term joint angles is preferred. This terminology comes from robotics where the most important joint is a hinge, since they are easy to build and since most motors naturally produce rotational motion.

Keller's algorithm

A principal pivot algorithm to solve $\mathrm{LCP}(M, q)$ for given, real, $n \times n$ P-matrix M and n-dimensional vector q. The algorithm proceeds by maintaining a feasible candidate solution at all stages and pivoting on the most negative slack variables.

Kinematic

That which is related to motion.

Kinematic constraint

A general constraint relation which does not depend on the generalized forces of a mechanical system is called a kinematic constraint. It imposes a restriction only on the kinematic variables, namely, the generalized coordinates q, the generalized velocities $\dot{q}$, and the time t.

Kinematic variables

In a mechanical system, those variables which describe the state of the motion. This qualifier specifically excludes all forces, masses, and inertia.

Kinematics

The study of the geometry of motion but without regards to what causes it. Kinematics is the part of mechanics which excludes dynamics.

Kinetic energy

The net work needed to bring a physical body from rest to a given value of generalized velocity $\dot{q}$. For a point particle of mass m with Cartesian coordinates x and velocity $\dot{x}$, this is the line integral $\int_C f^T \, \mathrm{d}x$. Using Newton's second law of motion, $f = m\ddot{x}$, and noting that $\ddot{x} \, \mathrm{d}x$ is the total derivative of $(1/2)\|\dot{x}\|^2$, the line integral is computed to be $(m/2)\|\dot{x}(t)\|^2$, and is independent of the path of integration C. This definition is then used additively to construct the kinetic energy of more general physical bodies after resolving them as aggregates of point particles. Kinetic energy of a general system is written $T(q, \dot{q})$ and it takes either the form $(1/2)\dot{q}^T M(q)\dot{q}$ for a configuration dependent mass matrix $M(q)$, or $(1/2)\dot{q}^T M \dot{q}$ for a constant mass matrix M.

Kinetic friction

Kinetic friction is a force acting in the tangential plane of contacting bodies subject to dry friction. It acts directly against the sliding velocity but

is (largely) independent of the magnitude of same. The magnitude of the kinetic friction force is well approximated by $\mu_k \nu$ where $\mu_k > 0$ the coefficient of kinetic friction, and ν is the magnitude of the normal contact force. Generally, μ_k is smaller than the static friction coefficient μ_s for the same materials by 10–20%. It varies quickly with the relative contact speed near zero but quickly levels out. Kinetic friction acts in the same direction as viscous friction would but in sharp contrast to the later, it is very nearly independent on the relative sliding speed but, instead, linearly dependent on the magnitude of the normal contact force, i.e., the magnitude of the force pushing the bodies against each other.

KKT: Karush-Kuhn-Tucker conditions

The KKT conditions are the necessary and sufficient complementarity conditions which are met at an optimal point of a nonlinear program consisting of minimizing a continuously differentiable function $f(x)$ subject to continuously differentiable constraints $h(x) = 0$ and $g(x) \geq 0$. For linear systems, the KKT conditions lead to a linear problem with a saddle-point matrix. For this reason, saddle-point matrices are also called KKT matrices by some authors.

Lagrange function

The difference between kinetic and potential energy expressed in generalized coordinates. The most general form is a function $\mathcal{L}(q, \dot{q}, t)$, which includes possible time dependence, but most common model used is the time independent form $\mathcal{L}(q, \dot{q}) = T(q, \dot{q}) - V(q)$. When holonomic constraints are considered, the augmented Lagrangian is modified to include terms of the form $\lambda^T g(q, t)$ where λ is a Lagrange multiplier (a ghost variable) and $g(q, t) = 0$ is the rheonomic kinematic holonomic constraint to enforce. The augmented Lagrangian is then a proper Lagrangian function of the extended arguments $q, \dot{q}, \lambda, \dot{\lambda}$.

Lagrange multiplier

Given a function $f : \mathbb{R}^n \mapsto \mathbb{R}$ which is to be minimized (or maximized) subject to the restriction $g(x) = 0$ for $g : \mathbb{R}^n \mapsto \mathbb{R}^m$, the Lagrange multiplier is the vector of proportionality constant in the equality $\mathrm{d}f/\,\mathrm{d}x = \lambda^T \,\mathrm{d}g/\,\mathrm{d}x$ which is satisfied at any stationary point of the restriction of f on the manifold $g(x) = 0$.

Lagrange top

A symmetric rigid body of mass m designed so that the three principal inertiae are $I_1 = I_2 < I_3$. It is attached to a point lying on the symmetry axis at some distance l from the center of mass. The distance l is chosen so that the principal inertiae about the fixed point which are $I_1 + ml^2, I_1 + ml^2, I_3$, satisfy $I_1 = I_2 > I3$. The Lagrange top problem is to start the rigid body with a small angle θ from the vertical, rotating at high speed about the

symmetry axis but being at rest otherwise, and to compute the resulting motion. When subjected to a constant downward gravitational force, the heavy symmetric top exhibits simultaneous precession and nutation. Precession is a slow rotational motion about the constant vertical axis aligned with the gravity force, and nutation is an oscillatory motion of the position of the center of mass along the same vertical axis.

LCP: Linear Complementarity Problem

Given a square $n \times n$ matrix M and an n-dimensional real vector q find n-dimensional real vectors z and w such that $Mz + q = w$ subject to the condition that $0 \leq z \perp w \geq 0$, i.e., $z_i, w_i \geq 0$, and $z_i w_i = 0$ for all $i = 1, 2, \ldots, n$. Computing the solution of an LCP is NP-hard in general though the average computational complexity is similar to that of matrix factorization.

Least action principle

In mechanics, the principle stating that the action functional—the time integral of the Lagrange function—has a stationary value with respect to first order infinitesimal variations compatible with imposed constraints, when evaluated on the physically realized trajectory. This variational principle yields the equations of motion of a mechanical system *via* the calculus of variations.

Lemke algorithm

A pivoting algorithm to solve $\mathsf{LCP}(M, q)$ for a given $n \times n$ real matrix M and real n-dimensional vector q. The Lemke algorithm processes more LCPs than any other solution algorithm, either computing a solution z, w or determining that no solution exists, since the only condition for the algorithm to terminate with a conclusive answer is that matrix M be copositive. The performance of the Lemke algorithm is also quite good on average, often terminating with n pivot operations. However, the pivot operations used in the algorithm are not principal pivots and this destroys any symmetry that might be present in M. In addition, the algorithm cannot be restarted from an approximate solution.

Lie algebra

Given an n-dimensional Lie group G, consider the infinitesimal elements $\phi_\epsilon^{(i)} = \mathsf{id} + \epsilon\xi_i$. For small enough ϵ, these elements are linear injections $\mathcal{M} \mapsto \mathcal{M}$ and can be represented by matrices. To first order in ϵ, the composition of two maps $\phi_\epsilon^{(i)}, \phi_\epsilon^{(j)}$ is the linear combination $\mathsf{id} + \epsilon(\xi_i + \xi_j)$. By continuity, the difference $\phi_\epsilon^{(i)} \circ \phi_\epsilon^{(j)} x - \phi_\epsilon^{(j)} \circ \phi_\epsilon^{(i)} x$ must be an infinitesimal of order $O(\epsilon)$ and from bijectivity, that is also a map in G. Therefore, the operators ξ_i form an algebra with binary operations "+" and the Lie bracket $[\xi_i, \xi_j] = \xi_i \xi_j - \xi_j \xi_i = c_{ijk} \xi_k$, where the c_{ijk} coefficients are the structure constants. This is the associated Lie algebra $\mathfrak{g}$ of a Lie group G.

Lie group

A Lie group G is such that the set $\mathcal{A}$ (see entry for Group) is a differentiable, n-dimensional manifold of smooth maps $\phi : \mathcal{M} \mapsto \mathcal{M}$, defined over some manifold $\mathcal{M}$, and such that the binary operation $\cdot$ is the map composition operator $\circ$ so that for $\psi, \phi \in \mathcal{A}$, $\psi \circ \phi \in \mathcal{A}$. In other words, for any $x \in \mathcal{M}$, the result of $\psi \cdot \phi$ on x is $\psi(\phi(x))$. By the assumed group property, these maps are invertible and therefore bijective. By the manifold property, these maps can be expanded in a n-dimensional basis of operators, $\xi_i, i = 1, 2, \ldots, n$, at least in a small ball of radius ϵ of the identity operator defined as $\mathrm{id} : \mathcal{M} \mapsto \mathcal{M}$, $\mathrm{id}\, x = x$ for all x. These infinitesimal generators form a Lie algebra.

LU: General Matrix Factorization

Given a square $n \times n$ matrix A, the LU factorization (also known as PLU) is the computation of a permutation matrix P, a unit lower triangular, matrix L, and an upper triangular matrix U, all of size $n \times n$, such that $A = PLU$. This is achieved using Gauss-Jordan pivot operations in which multiples of a given row are added to others to annihilate non-zero entries in a given column. Permutation of rows and columns—pivoting—is necessary to avoid zero or small divisors when performing the pivots and to maintain numerical stability. LU factorization has complexity $O(n^3)$ and is stable when using *complete pivoting*, searching through the entire matrix at each stage to identify the best pivot. Once the factors L and U are computed, one can easily solve for x in $Ax = b$ with direct operations with L and U.

Manifold

A point set $\mathcal{M}$ that is locally isomorphic to $\mathbb{R}^n$ in an open neighborhood $\mathcal{B}(x)$ of all points $x \in \mathcal{M}$, where n, which is the same for all points, is the dimension $n = \dim(\mathcal{M})$. A manifold allows a chart mapping points in $\mathbb{R}^n$ to points in $\mathcal{B}(x)$, establishing a curvilinear coordinate system. Collection of charts allow to represent any point $x \in \mathcal{M}$ as a function of vectors $y \in \mathbb{R}^n$, each centered at some origin $y^{(i)}$, $i = 1, 2, \ldots$. Only differentiable manifolds are considered, namely, those whose charts are all differentiable functions of their Cartesian coordinates. A simple example of a manifold is the set of points on the unit circle in two dimensions, and the charts would produce the polar angle ϕ, perhaps using either the tangent or cotangent function in different sectors to avoid the singularities of these on the coordinate axes.

Mass

A positive scalar $m > 0$ associated with each physical body in a mechanical system, which is both the inertia of the body with respect to forces applied at the center of mass as well as the parameter m used to compute the gravitational potential energy. No distinction is made between inertial and gravitational mass.

Mass matrix

For a mechanical system with n-dimensional configuration space $\mathcal{Q}$ with generalized coordinates q, generalized velocities $\dot{q}$, Lagrangian $\mathcal{L}(q, \dot{q})$, and momentum $p = \partial\mathcal{L}(q, \dot{q})$, the (possibly configuration dependent) $n \times n$ matrix M such that $p = M(q)\dot{q}$. Cases where the momentum is not linear in the generalized coordinates are not considered.

Mechanics

The description of the causes of the motion of physical bodies. The term mechanics by itself is often taken to apply specifically to the study of the motion of point masses, and various qualifiers are then added to cover the study of other physical bodies such as continuum mechanics, fluid mechanics, or rigid body mechanics. By extension, the analysis of ordinary differential equations, a fundamental element of mechanics, is often called mechanics as well. The denomination "classical mechanics" contrasts with quantum mechanics and relativistic mechanics, which study the special case of very small and very fast moving objects, respectively.

Minimum polynomial

Given a real or complex square $n \times n$ matrix A with characteristic polynomial $p(\lambda) = \prod_{k=1}^{m}(\lambda - \lambda_k)^{\mu_k} = \det(A - \lambda I_n)$, the minimum polynomial of A, $r(\lambda)$, is the monic polynomial with the same roots as $p(\lambda)$, but where all the multiplicities have been removed, namely, $r(\lambda) = \prod_{k=1}^{m}(\lambda - \lambda_k)$. This can be expanded to $r(\lambda) = \sum_{k=0}^{m} \alpha_k \lambda^k$ where $\alpha_k \in \mathbb{C}$, $\alpha_m = 1$.

MLCP: Mixed Linear Complementarity Problem

Given a real $n \times n$ square matrix M a real vector q of dimension n, and extended real vectors l, u of dimension n so that $u_i, l_i \in \mathbb{R} + \{\pm\infty\}$, the MLCP is the problem of finding real n-dimensional vectors z, w_+, w_-, with non-negative components so that:

$$\begin{aligned} Mz + q &= w_+ - w_- \\ 0 \le z - l \quad &\perp \quad w_+ \ge 0 \\ 0 \le u - z \quad &\perp \quad w_- \ge 0, \end{aligned}$$

where the perpendicularity sign is understood componentwise as in the case of the definition of the LCP.

Momentum

In Newtonian mechanics, which specifically considers point masses in three spatial dimensions, momentum is defined as the product of mass and velocity. In Cartesian coordinates, for a point mass with position $x \in \mathbb{R}^3$ and mass m, this is $m\dot{x}$, often written as mv where $v = \dot{x}$. In analytic mechanics, momentum is defined as $p = \partial\mathcal{L}(q, \dot{q})/\partial\dot{q}^T$. The two definitions are identical for point masses but the analytic definition extends to curvilinear coordinates as well.

Monogenic forces

In a mechanical system with configuration manifold Q and generalized coordinates q, any generalized force f which can be written as the gradient of a scalar function $V(q)$ as $f = -\partial V(q)/\partial q^T$. The condition for a force to be monogenic is that the work done by the force along any closed path C vanishes, i.e., that $\oint_C f^T \, \mathrm{d}q = 0$ for any contour C. Monogenic forces are also called conservative or potential forces. For mechanical systems subject only to monogenic forces, Hamilton's principle of least action is a sufficient and necessary variational condition for the motion to satisfy Newton's three laws of motion.

Multibody

A mechanical system composed of several physical bodies which are kinematically constrained to each other. This contrasts with many-body systems which are composed of very many physical bodies interacting only *via* forces.

Murty's principal pivot method

A principal pivot method to solve $\mathsf{LCP}(M, q)$ for given, real, $n \times n$ P-matrix M, and real vector q. The method proceed by performing a principal pivot operation on the infeasible variable with the smallest index at each stage. This is perhaps the simplest method to code, and unlike most other principal pivot methods, it is not restricted to symmetric positive definite or semi-definite matrices. It can also be started at an arbitrary candidate solution and extended to solve problems with P_0-matrices. However, the average performance on random problems is disappointing, performing up to n^2 pivots to solve a problem of size n.

NCP: Nonlinear Complementarity Problem

Given an n-dimensional mapping $f : \mathbb{R}^n \to \mathbb{R}^n$, find real n-dimensional vectors z and w, such that $f(z) = w$, subject to the condition that $0 \leq z \perp w \geq 0$, i.e., $z_i, w_i \geq 0$, and $z_i w_i = 0$ for all $i = 1, 2, \ldots, n$.

Newton-Raphson method

A numerical method to solve nonlinear systems of equations $f_i(x) = 0, i = 1, 2, \ldots, n$, where $f_i : \mathbb{R}^n \mapsto \mathbb{R}$, and $x \in \mathbb{R}^n$. The method proceeds by solving a succession of linear systems, $F^{(k)}(x^{(k+1)} - x^{(k)}) = -f(x^{(k)})$, where k is the iteration index, and $[F^{(k)}]_{ij} = \partial f_i(x^{(k)})/\partial x_j$ is the Jacobian matrix. For the modified Newton method, Jacobian F is not updated at every stage k. A variant of this method can also be used to solve unconstrained minimization problems defined with a scalar function $\theta : \mathbb{R}^n \mapsto \mathbb{R}$, and this formulation applied to $\|f(x)\|^2$ guarantees convergence of the method when a suitable line search strategy is used. See [158] for instance.

Newtonian mechanics

The specific description of motion and its causes provided by Newton in the *Principia* [215] according to a series of axioms and three fundamental laws. The axioms define bodies as point masses, mass as proportional to weight, inertia as the resistance to change of motion, momentum (the quantity of motion) as the product of mass and velocity, forces as the actions which cause change in the quantity of motion, and central (centripetal) forces which act directly in along the line separating two bodies. The laws are as follows (a) "Every body continues in its state of rest, or of uniform motion in a right line, unless it is compelled to change that state by forces impressed upon it" (b) " The change of motion is proportional to the motive force impressed; and is made in the direction of the right line in which that force is impressed" (c) "To every action there is always opposed an equal reaction: or, the mutual actions of two bodies upon each other are always equal, and directed to contrary parts." Newton mechanics is to be contrasted with analytic mechanics which produces the same laws of motion but starting from a different principle.

Noether's theorem

In physics, the theorem stating that any given single-parameter Lie group acting on the kinematic variables of a mechanical system which leaves the Lagrange function unchanged by its action corresponds directly to a scalar function of the same coordinates which is constant along the physical trajectory. The theorem also provides a procedure based on the corresponding Lie algebra to identify the said function.

Nonconservative systems

Mechanical systems which either dissipate energy through various friction mechanisms, or gain energy through drivers and other rheonomic constraints.

Nonholonomic constraint

A general constraint relations which can definitely not be eliminated because it does not correspond to the restriction of the configuration space Q to a differentiable manifold $\mathcal{M} \subset Q$ is called nonholonomic. Unilateral constraints are always nonholonomic. A kinematic bilateral constraints of the general form $a(q, \dot{q}, t) = 0$ is nonholonomic if it cannot be expressed as the time derivative of a holonomic bilateral constraint of the form $g(q, t) = 0$, which means that the differential equation $a(q, \dot{q}, t) = 0$ is not an exact differential.

The term nonholonomic is specifically used to distinguish bilateral kinematic constraints which are nonholonomic, namely, constraints of the form $a(q, \dot{q}, t) = 0$, since other types of nonholonomic constraints have their own specific qualifiers.

Noninertial frame

Any frame of reference in which at least one of the three laws of Newtonian mechanics do not hold. Usually, the first law is clearly violated by the appearance of a Coriolis acceleration. This can be measured using a pendulum for instance since the plane spanned by the supporting rod and the velocity vector will then rotate about the axis of action of the gravity force. The Earth itself is a noninertial frame as demonstrated by Foucault's famous pendulum.

Nonsmooth

A function with discontinuities.

Numerical error

The difference between an exact mathematical solution of a problem and the approximate solution computed by a finite precision algorithm. The sources of numerical error include discretization and roundoff from finite precision arithmetic.

Numerical integration

Any algorithmic process designed to compute an approximate numerical solution of a differential equation.

Numerical stability

A finite precision algorithm designed to solve a given mathematical problem is numerically stable when a small change in input data leads to a corresponding small change in output data. For a linear problem, the ratio of these errors often contains the *condition number* of the matrix, which may be large. A useful refinement of the concept is that of *backward stability*. Backward stable algorithms produce the exact solution of a problem with slightly different input data from the given one.

ODE: Ordinary Differential Equation

An ordinary differential equation is a differential equation $f(x^{(m)}, x^{(m-1)}, \ldots, \dot{x}, x, t) = 0$ such that the matrix $\partial f / \partial x^{(m)}$ is nonsingular. In particular, an explicit ODE is an equation which can be put into the form $x^{(m)} = \bar{f}(x^{(m-1)}, x^{(m-2)}, \ldots, \dot{x}, x, t)$. Applying order reduction on this as was done for differential equations (DE), this now reads $\dot{y} = \tilde{f}(y, t)$. The case where $\partial \tilde{f} / \partial t = 0$ is called an autonomous explicit ODE. Any ODE can be reduced to an equivalent ODE by adding the variable t to the vector y.

P-matrix

An $n \times n$ real matrix A such that all principal minors are strictly positive. The class of P-matrices includes the positive definite matrices as well as the strictly copositive matrices.

P_0-matrix

An $n \times n$ real matrix A such that all principal minors are non-negative.

Painlevé paradox

For a mechanical system subject to Coulomb friction, there exists certain configurations for which there is no solution to the differential equations of motion and certain other configurations allowing multiple solutions.

Path

For any topological space $\mathcal{A}$, a path is a smooth map $\phi : [t_0, t_1] \mapsto \mathcal{A}$.

Penalty force

Any force designed to enforce a given general constraint relation on the variables of a mechanical system. Penalty forces vanish when the constraint relation is satisfied and increase quickly when it becomes violated. Penalty forces often introduce highly oscillatory dynamics.

Pendulum

A point particle of mass m attached by a light rigid rod to a fixed point on a ceiling, falling under the action of gravity. The oscillations period of a pendulum are independent of the mass of the point particle and, for small period, this is $2\pi\sqrt{l/g}$, where g is the acceleration due to gravity and l is the length of the rod. By the conservation of angular momentum in this case, the pendulum motion is confined to a plane spanned by the direction of the thin supporting rod and the initial velocity vector. When reduced to two dimensions, this is perhaps the simplest possible nonlinearly constrained system and is a benchmark for all constrained integration schemes.

Pfaffian form

A vanishing differential form written as $\omega = A(x)\,\mathbf{d}x = 0$.

Phase space

For a mechanical system, this is the space of all possible configurations and velocities. If the configuration space is Q, the phase space is the tangent bundle TQ. For simple one-dimensional systems, this is the Cartesian space with coordinates x and $\dot{x}$.

Poincaré lemma

The generalization to arbitrary differential forms defined on manifolds of the well-known vector differential identities stating that $\mathtt{div}(\mathtt{grad}(f)) = 0$ and that $\mathtt{grad}(\mathtt{div}\, g) = 0$ for any twice-differentiable scalar function $f : \mathbb{R}^3 \mapsto \mathbb{R}$ and vector function $g : \mathbb{R}^3 \mapsto \mathbb{R}^3$, and $\mathtt{grad}$ and $\mathtt{div}$ are the familiar gradient and divergence operators, respectively. These latter identities are easily verified assuming equality of mixed partial derivatives of the function f and the component functions of g. The Poincaré lemma states that

$\mathbf{d}^2\theta = 0$ where θ is any twice differentiable differential form, defined on an arbitrary manifold $\mathcal{M}$.

Point particle

A physical body with no geometric extent. A point particle has scalar mass m and position $x \in \mathbb{R}^n$, where $n = 1, 2$ or 3. Though this might appear as a gross idealization, the motion of a rigid body does decouple into translational motion and rotational motion. The translational motion is rigorously equivalent to that of a point particle with the mass of the entire body, which means that in studying the motion of the Earth Sun system for instance, one can use point particles to represent both objects. Paradoxically, in the study of molecular dynamics, molecules are often modeled as multibodies since the internal degrees of freedom thus modeled bear consequences on the bulk physical properties.

Polygenic forces

In a mechanical system, any generalized force which is not monogenic. For a mechanical system subject to polygenic forces, d'Alembert's principle of vanishing virtual work is a necessary and sufficient variational condition for the motion to satisfy Newton's three laws of motion.

Positive definite matrix

A real $n \times n$ square matrix A such that $x^T A x > 0$ for all non-zero real n-dimensional vectors x. Positive definite matrices need not be symmetric. They are copositive plus and P-matrices as well.

Positive semi-definite matrix

A real $n \times n$ square matrix A such that $x^T A x \geq 0$ for all real n-dimensional vectors x. Positive semidefinite matrices need not be symmetric. They are copositive and P_0-matrices as well.

Potential energy

A real function of the generalized coordinates q, $V : Q \mapsto \mathbb{R}$ which produces generalized forces according $-\partial V / \partial q^T$. Conversely, given any generalized force $f : Q \mapsto \mathbb{R}^n$, where $n = \dim(Q)$, such that $\oint_C f^T(q)\, \mathrm{d}q = 0$ for any closed path C, then, the function $V = \int^q f^T(x)\, \mathrm{d}x$ is a potential energy, up to an arbitrary constant. Potential energy can be negative or positive. It is possible in principle that a potential function depends on the generalized velocities $\dot{q}$ as well, as is the case for the vector potential of magnetic forces, but this is not considered here.

Prismatic joint

A joint such that, starting from either body, the motion of the second is a pure translation along an axis fixed in the body frame of the first. Specifically, no rotation of the second body about the translational axis is allowed.

Much like for the hinge joint, the translational axis is fixed in both body frames and the constraint maintains the colinearity condition. Prismatic joint definitions can suffer from a singularity if the constraint definition preventing rotation about the axis of translation are defined in terms of the translation vector, which is common. Once again, this singularity is lifted if the rotational constraints are imposed using quaternions.

Pullback

Given a smooth map between two manifolds $\phi : \mathcal{M} \mapsto \mathcal{N}$, the pullback $\phi^\star$ is the (locally) linear map transforming one-forms in $\mathcal{M}$ back to the corresponding one-forms in $\mathcal{M}$, evaluated at matching points. It is the local inverse of the pushforward and in the context of curvilinear coordinate systems, it is precisely the Jacobian matrix of the (locally) inverse coordinate transform.

Pushforward

Given a smooth map between two manifolds $\phi : \mathcal{M} \mapsto \mathcal{N}$, the pushforward $\phi_\star$ is the (locally) linear map transforming one-forms in $\mathcal{M}$ to the corresponding one-forms in $\mathcal{N}$, evaluated at matching points. In the context of curvilinear coordinate systems, the pushforward $\phi_\star$ is precisely the local Jacobian matrix of the coordinate transformation.

QP: Quadratic Programming

Given a real, square $n \times n$ matrix Q a real n-dimensional vector q, a real $m \times n$ matrix A and a real m-dimensional vector b, the quadratic programming problem consists of finding the n-dimensional, real, non-negative vector x, with $x_i \geq 0$ for all $i = 1, 2, \ldots, n$, which minimizes the function $f(x) = \frac{1}{2} x^T Q x + b^T x$ subject to the m conditions that $Ax - b = 0$. The necessary and sufficient conditions met at the solution are known as the linear Karush-Kuhn-Tucker conditions.

QR: Orthogonal Factorization

Given an $m \times n$ matrix A with $m \geq n$, the QR factorization is the computation of a real orthonormal $m \times m$ matrix Q, with $Q^T Q = I_m$, and a real, $m \times n$, upper trapezoidal matrix R. Implementations of the QR factorization usually include partial pivoting making it very reliable. The QR algorithm can also be extended to be rank revealing. In that last application, it is usually less computationally expensive than SVD, though generally less reliable in rank determination.

Quaternions

Elements of the ring $\mathbb{H} = \{q = a\mathtt{h} + b\mathtt{i} + c\mathtt{j} + d\mathtt{k} \mid a, b, c, d \in \mathbb{R}\}$ where $\mathtt{h}$ is the multiplicative unit and the elements $\mathtt{i}, \mathtt{j}, \mathtt{k}$ satisfy the anticommuting multiplication rules $\mathtt{i}^2 = \mathtt{j}^2 = \mathtt{k}^2 = -\mathtt{h}$, $\mathtt{ij} = -\mathtt{ji} = \mathtt{k}$, $\mathtt{jk} = -\mathtt{kj} = \mathtt{i}$, and $\mathtt{ki} = -\mathtt{ik} = \mathtt{j}$. These are extended distributively to define the multiplication

$p \star p$ for any pair $p, q \in \mathbb{H}$. The non-zero $q \in \mathbb{H}$ have a unique inverse, q^{-1}, making $\mathbb{H}$ a division ring. Defining the norm $\|q\|^2 = a^2 + b^2 + c^2 + d^2$, the elements $q \in \mathbb{H}$ with $\|q\| = 1$ form a subgroup under multiplication which is isomorphic to $SO(3)$, and can thus be used to parametrize it. The quaternion parametrization of $SO(3)$ is free of singularity, unlike the Euler angle representation. The cost is to have to use four variables and a constraint $\|x\|q = 1$ instead of only three variables.

Ray

A ray is the one-dimensional cone $x + \alpha y, \alpha \geq 0$, where x and y are given n-dimensional vectors.

Rayleigh function

In a mechanical system with configuration manifold Q, generalized and velocities q and $\dot{q}$, a scalar function $\mathfrak{R}(q, \dot{q}, t)$ producing a polygenic generalized force f according to the rule $f = -\partial \mathfrak{R}(q, \dot{q}, t)/\partial \dot{q}^T$. Rayleigh functions which are homogeneous of degree κ in the variables $\dot{q}$ dissipate energy at the rate $\mathrm{d}E/\mathrm{d}t = -\kappa \mathfrak{R}(q, \dot{q}, t)$, making energy decay until $\mathfrak{R}(q, \dot{q}, t) = 0$. If a Rayleigh function is convex and bounded below by 0, the mechanical system moves in such a way as to minimize its value. The archetype of a Rayleigh function is $(\gamma/2)\dot{q}^T D\dot{q}$ where $\gamma > 0$ and D is an $n \times n$ constant, real, symmetric, positive definite matrix. This produces a viscous dissipative force $f = -\gamma D\dot{q}$ which ultimately vanishes when $D\dot{q} = 0$.

Regularization

The process of constructing a well-posed problem by adding small perturbation terms to an ill-posed one. When the perturbation terms are judiciously chosen, the solution of the well posed problem differs by a small amount, proportional to the size of the perturbation, from the original, ill-posed problem. In best cases, the solution—there might be more than one—to the ill-posed problem is recovered by setting the regularization parameter to zero in the computed answer, i.e., deleting all terms which are proportional to the regularization parameter.

Rheonomic constraint

A general constraint relation which depends explicitly on time. Rheonomic constraints are generally nonideal as their action alters the energy of the system. This does not preclude their elimination, however, for the specific case of bilateral holonomic rheonomic constraints.

Rigid body

A physical body with finite extent in which the constituent parts are all at fixed distances with respect to each other and, in consequence, at fixed position with respect to the center of mass of the body. Rigid bodies are idealizations of real physical bodies which are not easily deformed. The

motion of a rigid body decouples exactly into translational and rotational components, unlike the case of an elastic body where a small coupling remains. The configuration space of a rigid body is $Q = \mathbb{R}^3 \times SO(3)$.
In addition to the physical properties, a rigid body is considered to be impenetrable and this imposes kinematic restrictions on the relative motion of any two such.

Rubin and Ungar theorem

For a mechanical system subject to strong penalty forces designed to enforce a holonomic constraint $g(q) = 0$, the trajectories and velocities converge uniformly to those of the corresponding constrained system as the strength of the penalty forces is increase to infinity, though the penalty forces themselves fail to converge except in the weak sense. Weak convergence is established with respect to time integrals of inner products, implying that time averages do converge.

Saddle-point matrices

A matrix H of size $n + m \times n + m$ with the special block form $H = \begin{bmatrix} A & -B^T \\ B & C \end{bmatrix}$, where block matrix A is of size $n \times n$, is symmetric, and positive semi-definite, block matrix B is of size $m \times n$, and block matrix C is of size $m \times m$ and usually symmetric, and positive definite. Saddle-point matrices arise naturally in constrained optimization problems. Stationary points for these are saddle-points of an objective function. In this case, $C = 0$. The matrix is then usually written as $H = \begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix}$, which is symmetric but indefinite. The C block block matrix is usually a regularization added for stability. Saddle-point matrices can be factored with several optimized sparse packages such as UMFPACK [71], SuperLU [72], and MA57 [78]. There is a large array of iterative methods for these as well [74, 75, 48]. See also the entry for KKT conditions as saddle point matrices are sometimes called KKT matrices.

Scatter operation

In numerical linear algebra, any operation which involves copying data from contiguous memory locations to non-contiguous locations.

Scleronomic constraint

A general constraint relation which is not explicitly dependent on time is called a scleronomic constraint. If all constraints are scleronomic, the mechanical system is called scleronomic.

SIAM: Society for Industrial and Applied Mathematics

Founded in Philadelphia, USA, in 1951, SIAM exists to ensure the strongest interactions between mathematics and other scientific and technological

communities through membership activities, publication of journals and books, and conferences. SIAM publishes 14 peer-reviewed journals in applied mathematics as well as a book series.

SIGGRAPH: Special Interest Group on Graphics

A SIG of the ACM devoted to the promotion and dissemination of information on computer graphics and interactive techniques.

Simple harmonic oscillator

A mechanical system consisting of a one-dimensional point mass with coordinate x and mass m, subject to the force $f = -kx$ where $k > 0$ is a constant. The point mass exhibits sinusoidal motion about the origin with angular frequency $\omega = \sqrt{k/m}$ and period $T = 2\pi\sqrt{m/k}$. This is the simplest possible linear dynamical system and is a reference test case for any numerical integration method.

Slack variable

Given an inequality $ax + b \geq 0$, the slack variable s is what must be added to obtain an equality, i.e,. $ax + b - s = 0$, with $s \geq 0$.

Slider crank

A mechanical system designed to convert rotational motion from a driver into translational motion. This is constructed using two rods attached together with a hinge at their extremities. The free extremity of the first rod is then attached to a powered hinge fixed to the inertial frame whose axis is collinear with the connecting hinge between the rods. The free extremity of the second rod is then connected to the inertial frame with a prismatic joint whose axis is perpendicular to that of the three hinge axes. By driving the fixed hinge at constant rotational speed, the extremity of the second rod moves to and fro along the prismatic axis. This simple mechanical system always exhibit a constraint singularity.

Sliding velocity

The relative velocity between two contacting rigid bodies projected in the tangential plane of contact.

SMP: Symmetric Multiprocessor

A computer system containing multiple CPUs which share a memory bank via a common memory bus.

$SO(3)$

The special orthogonal group in $\mathbb{R}^3$, containing all 3×3 real matrices R such that $R^T R = I_3$, and such that $\det R = +1$. This latter property is what makes it "special". The group $SO(3)$ is a Lie group of dimension three which is differentiable, connected, but not simply connected. Viewed

as a manifold, $SO(3)$ is not flat and thus, there no possibility to map all the elements in $SO(3)$ to elements in $\mathbb{R}^3$ with a single chart without singularities.

Splitting

Any partitioning or additive decomposition of a mathematical problem. This is usually chosen so that each subproblem is easier to solve and so the couplings between them are small in some sense.

Spring

A force model producing a restoring force linearly proportional to the displacement from a reference configuration. For two point masses with positions $x^{(1)}$ and $x^{(2)}$, respectively, define the reference configuration so that $\|x^{(1)} - x^{(2)}\| = l_0$. The spring force on body one is then $f^{(1)} = -k(l-l_0)n^{(1,2)}$, where $n^{(1,2)}$ is the unit vector in the direction of $x^{(1)}-x^{(2)}$, and $f^{(2)} = -f^{(1)}$, from Newton's third law. In one dimension, this is the simple harmonic oscillator but in higher dimensions, springs are in fact nonlinear. In the general case, a spring is defined by introducing a nonlinear displacement function $g(q)$ of the generalized coordinates such that $g(q) = 0$ is a reference configuration. The generalized spring force is then $f = -k(\partial g/\partial q^T)g(q)$. The potential energy of a spring is simply $(1/2)g^2(q)$.

Spring and damper

A simple dissipative force model consisting of a spring and a viscous damping force that is proportional to the velocity of the deviation from the reference configuration. For the case of the simple harmonic oscillator, the damping force is simple $-b\dot{x}$, opposing the velocity linearly, leading to an exponential decay of the oscillation amplitude. In the general case of a spring defined with a displacement function $g(q)$, the damping force is given by $-k(\partial g/\partial q^T)\dot{q}$.

Static friction

For two contacting bodies with zero relative velocity in the tangential plane of contact, the absence of relative tangential acceleration in response to applied tangential forces up to a finite threshold. Consider for instance a block of mass m resting on an inclined plane making angle θ with the horizontal, subject to the downward action of gravity force of magnitude mg, with $g > 0$. The gravity force resolved in the contact plane is $mg\sin\theta$, and should accelerate the block downward, even if there was a viscous friction force of $-\gamma\dot{x}^{(\text{tang})}$, where $\dot{x}^{(\text{tang})}$ is the tangential velocity. But the block maintains zero tangential and acceleration as long as $\tan\theta < \mu_s$, where $\mu_s > 0$ is called the coefficient of static friction. Static friction is not limited to the Coulomb model however. See box friction.

Stationary conditions

The necessary conditions for a scalar function $\theta(x)$ with general constraint relations to have vanishing directional derivative in any direction allowed by the constraint. A function satisfying the stationary conditions at a point $x^\star$ might have a minimum, maximum, or an inflection point at $x^\star$.

Stiction

An abbreviation of static friction.

SVD: Singular Value Decomposition

Given a real $m \times n$ matrix A, the SVD is a factorization of the form $A = U\Sigma V^T$ where U and V are $m \times m$ and $n \times n$ real, orthonormal matrices, respectively, and Σ is real $m \times n$ diagonal matrix with non-increasing non-negative elements: $\Sigma = \mathtt{diag}(\sigma_1, \sigma_2, \ldots, \sigma_p), p = \mathtt{min}(m, n), \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$. The SVD reveals the real rank of a matrix since that is equal to the number of non-zero singular values σ_i. The SVD is generally much more computationally expensive than other factorizations but is the most robust and reliable method for computing the numerical rank of an arbitrary matrix.

Symmetry

For a mechanical system, any group of transformations acting on the configuration space Q or the tangent bundle TQ, or on time itself, which leaves the equations of motion unchanged. Symmetries are thus labeled by their groups. For instance, Galilean relativity implies that mechanical systems are unaffected by translation of the origin of the coordinate system and thus, they have translational symmetry. Each symmetry group corresponds to time invariant conserved quantities, called conserved currents or invariants of motion, according to Noether's theorem which is best formulated in the analytic formulation of mechanics. By extension, any quantity which is invariant under the action of a group of transformations G is said to be symmetric under G.

Symplectic flow

A differentiable mapping of time, $\phi_t : \mathcal{M} \mapsto \mathcal{M}$, which leaves a closed non-degenerate differential two-form invariant. A differential form ω is closed if it has an identically vanishing exterior derivative $\mathbf{d}\omega = 0$ and it is non-degenerate if the matrix representation in terms of the basis two-forms has full rank everywhere. The significance in physics is that the trajectories of Lagrangian systems are in fact symplectic flows and thus, they preserve certain oriented integrals. The non-degenerate two form in this case is $\omega = \mathbf{d}\big((\partial\mathcal{L}/\partial\dot{q})\,\mathbf{d}q\big)$.

Symplectic integrator

A numerical integration method designed to preserve the canonical two-form when applied to a physical system.

Tangent bundle

Given an n-dimensional manifold $\mathcal{M}$ with elements x, the tangent bundle $T\mathcal{M}$ is the disjointed union of the tangent spaces of all points $q \in \mathcal{M}$. For each point x, the tangent space T_x is a linear vector space of dimension n.

Tangential plane

For two bodies in contact at a point, the surfaces defining the boundary of these come into contact. When the bodies are defined by smooth enclosing surfaces, the respective surface normals at the contact point coincide and thus uniquely define the tangential plane of contact, tangential plane for short. For the non-smooth surface case, assuming the contact is at a discontinuity point of either or both surfaces, the normal cones must overlap and this defines an overlapping cone of tangent planes. Since nonsmooth surfaces are not analyzed in great detail, only the simple definition of the tangent plane is used.

Torque

Generically, torque is what causes the angular momentum to change. For a rigid body, any force applied away from the center of mass generates torque according to $\tau = \hat{z} f$, where f is the applied force, $z = y - x$, y is the point where the force is applied, and x is the center of mass, all quantities expressed in the inertial frame of reference.

Unilateral constraint

A general constraint relation which is to be satisfied as an inequality relation, which is usually chosen arbitrarily as the "$\geq$" relation. As a consequence of the inequality relation, the intensity of the generalized force produced by a unilateral constraint also satisfies an inequality condition.

Variational calculus

The study of the derivatives of functionals, usually defined as integrals over finite paths $\dot{\gamma} \in [t_0, t_1] \times \mathbb{R}^n$, of a scalar function $f : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}$, so $S[\dot{\gamma}] = \int_{t_0}^{t_1} \mathrm{d}s f(x(s), \dot{x}(s))$. Variational calculus allows to derive the stationary conditions for $S[\dot{\gamma}]$ which can then be used to solve for the optimal path, for instance.

Variational method

Any numerical method which is derived by requiring that a certain function of the unknowns be a stationary point at the solution.

VE: Virtual Environment

An interactive computer system with multiple input devices controlled by a user and multiple sensory output devices driven by a program. The driving

program includes a limited representation of reality—the virtual world—consisting of virtual objects having with simulated properties. Sensory properties of the virtual objects are emulated by driving graphics, audio, motion and haptic rendering devices from the main program. These emulations are controlled in part by numerical simulations of corresponding physical processes and in part by the user. VEs are interactive systems and are designed to respond quickly to changes in user inputs. VEs vary in the degree of immersion, realism, and fidelity, but they are generally constructed to provide a faithful representation of specific aspects of the real world. They are used for training, design, forensics, presentation, and artistic purposes.

Virtual displacements

Given a mechanical system with configuration space Q, generalized coordinates q, and a physical trajectory $(q(t), \dot{q}(t))$, the virtual displacements $\delta q(t)$ are infinitesimal variations of $q(t)$ restricted so the modified trajectory $q(t) + \delta q(t)$ satisfies all the constraints of the mechanical system. Because the virtual displacements satisfy all constraints and act specifically at a given time t, the operators $\mathrm{d}/\,\mathrm{d}t$ and δ commute and so, the generalized velocities are displaced infinitesimally to $\dot{q} + \delta\dot{q} = \dot{q} + \frac{\mathrm{d}}{\mathrm{d}t}\delta q$.

Virtual work

The work done by virtual displacements $\delta q(t)$ along the physical trajectory $q(t)$ for a mechanical system with configuration space Q and generalized coordinates q.

Viscous friction

A polygenic, dissipative force vanishing for zero velocity. The archetype is a generalized force of the form $-\gamma D(q)\dot{q}$, corresponding to the Rayleigh function $\mathfrak{R}(q, \dot{q}) = (\gamma/2)\dot{q}^T D(q)\dot{q}$. The parameter γ is the viscous drag. Viscous forces which vary with higher powers of the the velocity can be constructed as well. This is similar to the damping term in the spring and damper force model. Because viscous forces vanish in zero velocity, they cannot be used to correctly model dry friction in general, and Coulomb friction in particular. Viscous friction is also very different from kinetic friction since in the latter case, the magnitude is very nearly independent of the velocity. Viscous friction is an accurate model of the tangential friction force between lubricated surfaces.

Wedge product

The antisymmetric exterior product of differential forms. For elementary one-forms $\mathbf{d}x^{(i)}, \mathbf{d}x^{(j)}$, the wedge product is the two-form written $\omega^2 = \mathbf{d}x^{(i)} \wedge \mathbf{d}x^{(j)}$. The wedge product is antisymmetric so that, in particular, $\mathbf{d}x^{(i)} \wedge \mathbf{d}x^{(j)} = -\,\mathbf{d}x^{(j)} \wedge \mathbf{d}x^{(i)}$. It is also linear and associative.

Work

For a mechanical system with configuration space Q, generalized coordinates q, given a generalized force vector f, work is the line integral $W = \int_C f^T \, dq$ along a given path C. With this sign convention, W is the work done by the force on the mechanical system, so that $-W$ is the work done by the system on whatever system is producing the force f. Work has units of energy and generalized forces and coordinates are defined so that this is true whatever coordinate system is chosen. For a general non-conservative force, work is path dependent. Conservative forces are path independent.

Zeno point

For a mechanical system subject to impact discontinuities, a Zeno point is an instant t which cannot be crossed because it is preceded by infinitely many impact events in any neighborhood $t - \epsilon$, where $\epsilon > 0$. For instance, a super-ball dropped on a plane exhibits what appears to be a Zeno point as it rebounds faster and faster with smaller and smaller amplitude as it slowly looses energy at each impact. However, the ball eventually looses so much energy that it does not even leave the plane. Numerical methods designed to handle impacts must carefully avoid these apparent Zeno points by introducing sufficient thresholds lest they become very inefficient.

Bibliography

[1] A. M. Al-Fahed, G. E. Stravroulakis, and P. D. Panagiotopoulos, *Hard and soft fingered robot grippers. The linear complementarity approach*, Z angew. Math. Mech., 71 (1991), pp. 257–265.

[2] P. Alart, M. Barboteu, and F. Lebon, *Solution of frictional contact problems by an EBE preconditioner*, Computational Mechanics, 20 (1997), pp. 370–378.

[3] G. E. Alefeld, X. Chen, and F. A. Potra, *Numerical validation of solutions of linear complementarity problems*, Numerische Mathematik, 83 (1999), pp. 1–24. Also available as preprint as report-102.ps from ftp.math.uiowa.edu.

[4] S. L. Altmann, *Hamilton Rodrigues and the quaternion scandal*, Mathematics Magazine, 62 (1989), pp. 291–308.

[5] H. C. Andersen, *RATTLE: A velocity version of the SHAKE algorithm for molecular dynamics calculations*, J. Comp. Phys., 52 (1983), pp. 24–34.

[6] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. D. Croz, A. Greenbaum, S. Hammerling, A. McKenney, S. Ostrouchov, and D. Sorenson, *LAPACK Users' Guide*, SIAM Publ., Philadelphia, second ed., 1995.

[7] L.-E. Andersson, *Existence results for quasistatic contact problems with Coulomb friction*, Appl. Math. Optim., 42 (2000), pp. 169–202.

[8] L.-E. Andersson and A. Klarbring, *A review of the theory of static and quasi-static frictional contact problems in elasticity*, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 359 (2001), pp. 2519–2539. 10.1098/rsta.2001.0908.

[9] J. Angeles, *Fundamentals of Robotic Mechanical Systems: Theory, Methods and Algorithms*, Mechanical engineering series, Springer, New York, NY, 2 ed., 2002.

[10] J. Angeles and A. Kecskeméthy, *Fundamentals of Rigid-Body Mechanics*, in ICMS Courses and Lectures [11], July 1995, ch. 2, pp. 11–46.

[11] ——, eds., *Kinematics and Dynamics of Multi-Body Systems*, vol. 360 of ICMS Courses and Lectures, Springer-Verlag, July 1995.

[12] J. Angeles and S. Lee, *The formulation of dynamical equations of holonomic mechanical systems using a natural orthogonal complement*, ASME Trans. J. of Applied Mechanics, 55 (1988), pp. 243–244.

[13] J. Angeles and O. Ma, *Dynamics of nonholonomic mechanical systems using natural orthogonal complements*, ASME Trans. J. of Applied Mechanics, 58 (1991), pp. 238–243.

[14] M. Anitescu, *Modeling Rigid Multi Body Dynamics with Contact and Friction*, PhD thesis, University of Iowa, Iowa City, IO, Aug. 1997.

[15] M. Anitescu and G. D. Hart, *A constraint-stabilized time-stepping approach for rigid multibody dynamics with joints, contact and friction*, International Journal for Numerical Methods in Engineering, 60 (2004), pp. 2335–2371.

[16] ——, *A fixed-point iteration approach for multibody dynamics with contact and small friction*, Math. Program., 101 (2004), pp. 3–32.

[17] M. Anitescu and F. A. Potra, *Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems*, Nonlinear Dynamics, 14 (1997), pp. 231–247.

[18] M. Anitescu and F. A. Potra, *A time-stepping method for stiff multibody dynamics with contact and friction*, Internat. J. Numer. Methods Engrg., 55 (2002), pp. 753–784.

[19] M. Anitescu, F. A. Potra, and D. E. Stewart, *Time-stepping for three-dimensional rigid body dynamics*, Computer Methods in Applied Mechanics and Engineering, 177 (1999), pp. 183–197.

[20] C. Arévalo, C. Führer, and G. Söderlind, *Stabilized multistep methods for index* 2 *Euler-Lagrange DAEs*, BIT, 36 (1996), pp. 1–13.

[21] G. Arfken, *Mathematical Methods for Physicists*, Academic Press, New York, third ed., 1985.

[22] V. I. Arnol′d, *Mathematical Methods of Classical Mechanics*, vol. 60 of Graduate Texts in Mathematics, Springer-Verlag, New York, second ed., 1989. Translated from the Russian by K. Vogtmann and A. Weinstein.

[23] V. I. Arnold, V. V. Kozlov, and A. I. Neishtadt, *Mathematical aspects of classical and celestial mechanics*, Springer-Verlag, Berlin, 1997. Translated from the 1985 Russian original by A. Iacob, Reprint of the original English edition from the series Encyclopaedia of Mathematical Sciences [*Dynamical systems. III*, Encyclopaedia Math. Sci., 3, Springer, Berlin, 1993; MR1292465 (95d:58043a)].

[24] U. Ascher, H. Chin, L. Petzold, and S. Reich, *Stabilization of constrained mechanical systems with DAEs and invariant manifolds*, Mech. Struct. & Mach., 23 (1995), pp. 135–158.

[25] U. Ascher and P. Lin, *Sequential regularization methods for higher index DAEs with constraint singularities: The linear index-2 case*, SIAM J. Numer. Anal., 33 (1996), pp. 1921–1940.

[26] ——, *Sequential regularization methods for nonlinear higher index DAEs*, SIAM J. Scient. Comput, 18 (1997), pp. 160–181.

[27] ——, *Sequential regularization methods for simulating mechanical systems with many closed loops.*, SIAM J. Scient. Comput., 21 (1999), pp. 1244–1262.

[28] U. Ascher, D. Pai, and B. Cloutier, *Forward dynamics, elimination methods, and formulation stiffness in robot simulation*, Int. J. Robotics Res., 16 (1997), pp. 749–758.

[29] U. M. Ascher and L. R. Petzold, *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*, SIAM Publ., Philadelphia, 1998.

[30] U. M. Ascher and S. Reich, *On some difficulties in integrating highly oscillatory Hamiltonian systems*, in Computational molecular dynamics: challenges, methods, ideas (Berlin, 1997), vol. 4 of Lect. Notes Comput. Sci. Eng., Springer, Berlin, 1999, pp. 281–296.

[31] D. Baraff, *Analytical methods for dynamic simulation of non-penetrating rigid bodies*, Computer Graphics, 23 (1989), pp. 223–232.

[32] ——, *Curved surfaces and coherence for non-penetrating rigid body simulation*, Computer Graphics, 24 (1990), pp. 19–28.

[33] ——, *Coping with friction for non-penetrating rigid body simulation*, Computer Graphics, 25 (1991), pp. 31–40.

[34] ——, *Issues in computing contact forces for non-penetrating rigid bodies*, Algorithmica, 10 (1993), pp. 292–353.

[35] D. Baraff, *Fast contact force computation for nonpenetrating rigid bodies*, Computer Graphics, 28 (1994), pp. 23–42.

[36] D. Baraff, *Interactive simulation of solid rigid bodies*, IEEE Comp. Grahics App., 15 (1995), pp. 63–75.

[37] D. Baraff, *Linear-time dynamics using Lagrange multipliers*, Computer Graphics, 30 (1996), pp. 137–146.

[38] D. Baraff and A. Witkin, *Dynamic simulation of non-penetrating flexible bodies*, Computer Graphics, 26 (1992), pp. 303–308.

[39] ——, *Large steps in cloth simulation*, in SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques, New York, NY, USA, 1998, ACM Press, pp. 43–54.

[40] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. V. der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd Edition*, SIAM, Philadelphia, PA, 1994.

[41] A. Barrlund, *Constrained least squares methods for linear time varying DAE systems*, Numer. Math., 60 (1991), pp. 145–161.

[42] ——, *Comparing stability properties of 3 methods in DAEs or ODEs with invariants*, Bit, 35 (1995), pp. 1–18.

[43] A. Barrlund and B. Kågström, *Analytical and numerical-solutions to higher index linear variable-coefficient DAE systems*, J. Computational Appl. Mathematics, 31 (1990), pp. 305–330.

[44] R. Barzel, *Physically-based modeling for computer graphics: a structured approach*, Academic, Boston, 1992.

[45] R. Barzel and A. H. Barr, *A modeling system based on dynamic constraints*, Computer Graphics, 22 (1988), pp. 179–187.

[46] R. Barzel, J. F. Hughes, and D. N. Wood, *Plausible motion simulation for computer graphics animation*, in Proceedings of the Eurographics workshop on Computer animation and simulation '96, New York, NY, USA, 1996, Springer-Verlag New York, Inc., pp. 183–197.

[47] J. Baumgarte, *Stabilization of constraints and integrals of motion in dynamical systems*, Computer Methods in Applied Mechanics and Engineering, 1 (1972), pp. 1–16.

[48] M. Benzi, G. H. Golub, and J. Liesen, *Numerical solution of saddle point problems*, Acta Numer., 14 (2005), pp. 1–137.

[49] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, second ed., 1999.

[50] A. M. Bloch, P. S. Krishnaprasad, J. E. Marsden, and R. M. Murray, *Nonholonomic mechanical systems with symmetry*, Arch. Rational Mech. Anal., 136 (1996), pp. 21–99.

[51] A. I. Bobenko and Y. B. Suris, *Discrete time Lagrangian mechanics on Lie groups, with an application to the Lagrange top*, Comm. Math. Phys., 204 (1999), pp. 147–188.

[52] D. Braess, *Finite Elements: Theory, Fast Solvers, and Applications in Solid Mechanics*, Cambridge University Press, Cambridge, U. K., 1997.

[53] V. Brasey, *A half-explicit Runge-Kutta method of order 5 for solving constrained mechanical systems*, Computing, 48 (1992), pp. 191–201.

[54] K. E. Brenan, S. L. Campbell, and L. R. Petzold, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, SIAM Publ., Philadelphia, 1996.

[55] V. N. Brendelev, *On the realization of constraints in nonholonomic mechanics*, J. Appl. Math. Mech., 45 (1981), pp. 481–487.

[56] B. Brogliato, *Nonsmooth Mechanics: Models, Dynamics and Control*, Communication and Control Engineering, Springer-Verlag, Berlin, Heidelberg, New York, London, Paris, Tokyo, Hong Kong, second ed., 1999.

[57] V. A. Bulavsky, G. Isac, and V. V. Kalashnikov, *Application of topological degree theory to complementarity problems*, in Multilevel optimization: algorithms and applications, vol. 20 of Nonconvex Optim. Appl., Kluwer Acad. Publ., Dordrecht, 1998, pp. 333–358.

[58] F. Cameron, *A class of low order DIRK methods for a class of DAEs*, Appl. Numer. Math., 31 (1999), pp. 1–16.

[59] ——, *Low-order Runge-Kutta methods for differential-algebraic equations*, vol. 281 of Julkaisuja / Tampereen teknillinen korkeakoulu, 0356-4940, Tampere University of Technology, Tampere, 1999.

[60] D. Chandler and B. J. Berne, *Comment on the role of constraints on the conformational structure of n-butane in liquid solvents*, J. Chem. Phys., 71 (1979), pp. 5386–5387.

[61] H. S. Chin, *Stabilization Methods for Simulations of Constrained Multibody Dynamics*, PhD thesis, Department of Mathematics, University of British Columbia, 1995.

[62] P. W. Christensen, *Algorithms for Frictional Contact*, PhD thesis, Linköping University, S-581 83 Linköping, Sweden, 1997.

[63] P. W. Christensen, A. Klarbring, J. S. Pang, and N. Strömberg, *Formulation and comparison of algorithms for frictional contact problems*, Internat. J. Numer. Methods Engrg., 42 (1998), pp. 145–173.

[64] P. W. Christensen and J.-S. Pang, *Frictional contact algorithms based on semismooth Newton methods*, in Reformulation: nonsmooth, piecewise smooth, semismooth and smoothing methods (Lausanne, 1997), vol. 22 of Appl. Optim., Kluwer Acad. Publ., Dordrecht, 1999, pp. 81–116.

[65] S. Chung and E. J. Haug, *Real-time simulation of multibody dynamics on shared-memory multiprocessors*, J. Dynamic Systems Measurement Control-transactions Asme, 115 (1993), pp. 627–637.

[66] F. H. Clarke, *Optimization and Nonsmooth Analysis*, Classics in applied mathematics, SIAM Publ., Philadelphia, 1990.

[67] J. Cortés, *Energy conserving nonholonomic integrators*, Discrete Contin. Dyn. Syst., (2003), pp. 189–199. Dynamical systems and differential equations (Wilmington, NC, 2002).

[68] J. Cortés and S. Martínez, *Non-holonomic integrators*, Nonlinearity, 14 (2001), pp. 1365–1392.

[69] R. W. Cottle, J.-S. Pang, and R. E. Stone, *The Linear Complementarity Problem*, Computer Science and Scientific Computing, Academic Press, New York, 1992.

[70] M. Dalgård, *A rigid body model for real-time simulation of cloth*, Master's thesis, Engineering Physics, Umeå University, Umeå, Sweden, Sept. 2005.

[71] T. A. Davis, *Algorithm 832: UMFPACK — an unsymmetric-pattern multifrontal method*, ACM Transactions on Mathematical Software, 30 (2004), pp. 196–199.

[72] J. W. Demmel, S. C. Eisenstat, J. R. Gilbert, X. S. Li, and J. W. H. Liu, *A supernodal approach to sparse partial pivoting*, SIAM J. Matrix Analysis and Applications, 20 (1999), pp. 720–755. superlu.

[73] J. Denavit and R. S. Hartenberg, *A kinematic notation for lower-pair mechanisms based on matrices*, J. Appl. Mech., 22 (1955), pp. 215–221.

[74] H. S. Dollar, N. I. M. Gould, W. H. A. Schilders, and A. J. Wathen, *Implicit-factorization preconditioning and iterative solvers for regularized saddle-point systems*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 170–189 (electronic).

[75] H. S. Dollar and A. J. Wathen, *Approximate factorization constraint preconditioners for saddle-point matrices*, SIAM J. Sci. Comput., 27 (2006), pp. 1555–1572 (electronic).

[76] Z. Dostál, *Box constrained quadratic programming with proportioning and projections*, SIAM J. Opt., 7 (1997), pp. 871–887.

[77] Z. Dostál, J. Haslinger, and R. Kučera, *Implementation of the fixed point method in contact problems with Coulomb friction based on a dual splitting type technique.*, J. of Comp. and Appl. Maht., 140 (2002), pp. 245–256.

[78] I. S. Duff, *MA57—a code for the solution of sparse symmetric definite and indefinite systems*, ACM Trans. Math. Softw., 30 (2004), pp. 118–144.

[79] I. S. Duff, A. M. Erisman, and J. K. Reid, *Direct Methods for Sparse Matrices*, Numerical Mathematics and Scientific Computation, Clarendon Press, Oxford, 1986.

[80] I. S. Duff and J. K. Reid, *The multifrontal solution of indefinite sparse symmetric linear systems*, ACM Transactions on Mathematical Software, 9 (1983), pp. 302–325.

[81] A. Dullweber, B. Leimkuhler, and R. McLachlan, *Symplectic splitting methods for rigid body molecular dynamics*, Journal of Chemical Physics, 107 (1997), pp. 5840–5851.

[82] K. L. Dunlap and M. M. Kostreva, *Solving more linear complementarity problems with Murty's Bard-type algorithm*, Journal of Optimization Theory and Applications, 77 (1993), pp. 497–522.

[83] P. E. Dupont and S. P. Yamajako, *Stability of frictional contact in constrained rigid-body dynamics*, IEEE Transactions on Robotics and Automation, 13 (1997), pp. 230–236.

[84] K. Erleben, *Stable, Robust, and Versatile Multibody Dynamics Animation*, PhD thesis, Department of Computer Science, University of Copenhagen, Nov. 2004.

[85] K. Erleben, J. Sporring, K. Henriksen, and H. Dohlman, *Physics-Based Animation*, Charles River Media, Boston, 2005.

[86] R. Featherstone, *Robot Dynamics Algorithms*, Kluwer Academic Publishers Group, Dordrecht, The Netherlands, 1987.

[87] R. C. Fetecau, J. E. Marsden, M. Ortiz, and M. West, *Nonsmooth Lagrangian mechanics and variational collision integrators*, SIAM J. Appl. Dyn. Syst., 2 (2003), pp. 381–416.

[88] A. Fischer, *A special Newton-type optimization method*, Optimization, 24 (1992), pp. 269–284.

[89] M. Fixman, *Classical Statistical Mechanics of Constraints: A Theorem and Application to Polymers*, Proc. Nat. Acad. Sci., 71 (1974), pp. 3050–3053.

[90] H. Flanders, *Differential Forms with Applications to the Physical Sciences*, Dover Publications, New York, 1989.

[91] M. R. Flannery, *The enigma of nonholonomic constraints*, Amer. J. Phys., 73 (2005), pp. 265–272.

[92] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes, *Computer Graphics: Principles and Practice*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd ed., 1990.

[93] J. S. Freeman, G. Watson, Y. E. Papelis, and T. C. Lin, *The Iowa driving simulator: an implementation and application overview*, SAE Trans., 104 (1995), pp. 113–122.

[94] C. Führer and B. J. Leimkuhler, *Numerical solution of differential-algebraic equations for constrained mechanical motion*, Numerische Mathematik, 59 (1991), pp. 55–69.

[95] C. W. Gear, *Towards explicit methods for differential algebraic equations*, BIT, 46 (2006), pp. 505–514.

[96] C. W. Gear, G. K. Gupta, and B. Leimkuhler, *Automatic integration of Euler-Lagrange equations with constraints*, in Proceedings of the international conference on computational and applied mathematics (Leuven, 1984), vol. 12/13, 1985, pp. 77–90.

[97] P. E. Gill, G. H. Golub, W. Murray, and M. A. Saunders, *Methods for modifying matrix factorizations*, Mathematics of Computation, 28 (1974), pp. 505–535.

[98] P. E. Gill, W. Murray, S. M. Picken, and M. H. Wright, *The design and structure of a fortran program library for optimization*, ACM Trans. Math. Softw., 5 (1979), pp. 259–283.

[99] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, *Procedures for optimization problems with a mixture of bounds and general linear constraints*, ACM Trans. Math. Softw., 10 (1984), pp. 282–298.

[100] R. E. Gillilan and K. R. Wilson, *Shadowing, rare events, and rubber bands - a variational Verlet algorithm for molecular-dynamics*, J. Chem. Phys., 97 (1992), pp. 1757–1772.

[101] R. Glowinski, L. Shiau, Y. M. Kuo, and G. Nasser, *The numerical simulation of friction constrained motions. I. One degree of freedom models*, Appl. Math. Lett., 17 (2004), pp. 801–807.

[102] R. Glowinski, L. Shiau, Y. M. Kuo, and G. Nasser, *The numerical simulation of friction constrained motions. II. Multiple degrees of freedom models*, Appl. Math. Lett., 18 (2005), pp. 1108–1115.

[103] ——, *On the numerical simulation of friction constrained motions*, Nonlinearity, 19 (2006), pp. 195–216.

[104] M. Göckeler and T. Schücker, *Differential Geometry, Gauge Theories, and Gravity*, Cambridge monographs on mathematical physics, Cambridge University Press, Cambridge, 1987.

[105] H. Goldstein, *Classical Mechanics*, Addison-Wesley, Reading, MA, USA, second ed., 1980.

[106] H. H. Goldstine, *A History of the Calculus of Variations from the 17th through the 19th Century*, vol. 5 of Studies in the history of mathematics and physical sciences, Springer-Verlag, New York, 1980.

[107] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins Press, Baltimore, third ed., 1996.

[108] S. S. Grigoryan, *Resolution of the dry friction paradox—the Painlevé paradox*, Dokl. Akad. Nauk, 379 (2001), pp. 54–58.

[109] E. Guendelman, R. Bridson, and ronald Fedkiw, *Nonconvex rigid bodies with stacking*, in Proceedings of the ACM SIGGRAPH 2003, J. Hart, ed., vol. 22, ACM Transactions on Graphics, July 2003, pp. 871–878.

[110] G. J. Habetler and M. M. Kostreva, *Sets of generalized complementarity problems and P-matrices*, J. of Mathematics of Operations Research, 5 (1980), pp. 280–284.

[111] E. Hairer, C. Lubich, and M. Roche, *The Numerical Solution of Differential-Algebraic Systems by Runge-Kutta Methods*, no. 1409 in Lecture Notes in Mathematics, Springer-Verlag, 1989.

[112] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration*, vol. 31 of Spring Series in Computational Mathematics, Springer-Verlag, Berlin, Heidelberg, New York, London, Paris, Tokyo, Hong Kong, 2001.

[113] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations I: Nonstiff Problems*, vol. 8 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, Heidelberg, New York, London, Paris, Tokyo, Hong Kong, second revised edition ed., 1991.

[114] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II: Stiff and Differential Algebraic Problems*, vol. 14 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, Heidelberg, New York, London, Paris, Tokyo, Hong Kong, second revised edition ed., 1996.

[115] W. R. Hamilton, *On a general method in dynamics; by which the study of the motions of all free systems of attracting or repelling points is reduced to the search and differentiation of one central relation, or characteristic function*, Philosophical Transactions of the Royal Society of London (1776-1886), 124 (1834), pp. 247–308.

[116] ——, *Second essay on a general method in dynamics*, Philosophical Transactions of the Royal Society of London (1776-1886), 125 (1835), pp. 95–144.

[117] R. Hassani, P. Hild, and I. Ionescu, *Sufficient conditions of non-uniqueness for the Coulomb friction problem*, Math. Methods Appl. Sci., 27 (2004), pp. 47–67.

[118] E. J. Haug, *Computer Aided Kinematics and Dynamics of Mechanical Systems, vol 1; Basic Methods*, Allyn and Bacon, 1989.

[119] ——, *Feasibility and conceptual design of National Advanced Driving Simulator*, Tech. Rep. DOT HS 807 596, US Department of Transportation, National Highway Traffic Safety Administration, Mar. 1990.

[120] E. J. Haug, D. Negrut, and C. Engstler, *Implicit Runge-Kutta integration of the equations of multibody dynamics in descriptor form*, Mechanics Structures Machines, 27 (1999), pp. 337–364.

[121] E. J. Haug, D. Negrut, and M. Iancu, *Implicit integration of the equations of multibody dynamics*, in NATO ASI on Computational Methods in Mechanisms, Sofia, Bulgaria, June 1997, NATO ASI, pp. 141–156.

[122] ——, *A state-space-based implicit integration algorithm for differential-algebraic equations of multibody dynamics*, Mechanics Structures Machines, 25 (1997), pp. 311–334.

[123] E. J. Haug, S. C. Wu, and S. M. Yang, *Dynamics of mechanical systems with Coulomb-friction, stiction, impact and constraint addition deletion I: theory*, Mechanism Machine Theory, 21 (1986), pp. 401–406.

[124] S. Hennessy, J. Wishart, D. Whitelock, R. Deaney, R. Brawn, L. la Velle, A. McFarlane, K. Ruthven, and M. Winterbottom, *Pedagogical approaches for technology-integrated science teaching*, Computers & Education, 48 (2007), pp. 137–152.

[125] N. J. Higham, *Computing a nearest symmetric positive semidefinite matrix*, Linear Algebra Appl., 103 (1988), pp. 103–118.

[126] ——, *Accuracy and Stability of Numerical Algorithms*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, second ed., 2002.

[127] M. Hiller, *Dynamics of Multiloop Kinematic Chains*, in Angeles and Kecskeméthy [11], July 1995, ch. 5, pp. 167–216.

[128] ——, *Multiloop Kinematic Chains*, in Angeles and Kecskeméthy [11], July 1995, ch. 4, pp. 75–166.

[129] A. C. Hindmarsh, *LSODE and LSODI, two new initial value ordinary differential equation solvers*, ACM-SIGNUM Newsletter, 15 (1980), pp. 10–11.

[130] M. Hocke, R. Rühle, and M. Otter, *A software environment for analysis and design of multibody systems*, in Schiehlen [246], pp. 67–86.

[131] H. Y. Huang and C. Y. Gau, *Modelling and designing a low-cost high-fidelity mobile crane simulator*, Int. J. Human-computer Studies, 58 (2003), pp. 151–176.

[132] K. Huang, *Statistical Mechanics*, John Wiley & Sons, New York, London, Sydney, second ed., 1987.

[133] G. Isac, *Leray-Schauder type alternatives and the solvability of complementarity problems*, Topol. Methods Nonlinear Anal., 18 (2001), pp. 191–204.

[134] G. Isac, *Complementarity problems and variational inequalities. A unified approach of solvability by an implicit Leray-Schauder type alternative*, J. Global Optim., 31 (2005), pp. 405–420.

[135] G. Isac, V. Bulavski, and V. Kalashnikov, *Exceptional families, topological degree and complementarity problems*, J. Global Optim., 10 (1997), pp. 207–225.

[136] J. D. Jackson, *Classical Electrodynamics*, John Wiley & Sons Inc., New York, second ed., 1975.

[137] N. Jacobson, *Lie Algebras*, Dover Publications, New York, 1979.

[138] ——, *Basic Algebra I, 2nd ed.*, Freeman, 1985.

[139] A. Jain, *Unified formulation of dynamics of SErial rigid multibody systems*, J. Guidance, Control and Dynamics, 14 (1991), pp. 531–542.

[140] T. Jakobsen, *Advanced character physics*, in Proceedings of the Game Developers Conference, 2001.

[141] A. Jennings and J. J. McKeown, *Matrix Computation*, John Wiley and Sons, Chichester, West Sussex and New York, 2nd ed., 1992.

[142] F. Jourdan, P. Alart, and M. Jean, *A Gauss-Seidel like algorithm to solve frictional contact problems*, Computer Methods in Applied Mechanics and Engineering, 155 (1998), pp. 31–47.

[143] F. Jourdan, M. Jean, and P. Alart, *An alternative method between implicit and explicit schemes devoted to frictional contact problems in deep drawing simulation*, J. of Materials Processing Technology, 80-81 (1998), pp. 257–262.

[144] J. J. Júdice, *Algorithms for linear complementarity problems*, in Algorithms for Continuous Optimization, E. Spedicato, ed., vol. 434 of NATO ASI Series C, Mathematical and Physical Sciences, Advanced Study Institute, NATO, Kluwer Academic Publishers, 1994, pp. 435–475.

[145] J. J. Júdice and A. M. Faustino, *An experimental investigation of enumerative methods for the linear complementarity problem*, Comput. Oper. Res., 15 (1988), pp. 417–426.

[146] J. J. Júdice, J. Machado, and A. Faustino, *An extension of the Lemke's method for the solution of a generalized linear complementarity problem*, in System Modelling and Optimization, P. Kall, ed., vol. 180 of Lecture Notes in Control and Information Sciences, Springer-Verlag, 1992, pp. 221–230. Proc., 15th IFIP TC7 Conference, Zurich, September 1991.

[147] J. J. Júdice and F. M. Pires, *Direct methods for convex quadratic programs subject to box constraints*, Investigação Operacional, 9 (1989), pp. 23–56.

[148] B. Kågström, P. Ling, and C. Van Loan, *Algorithm 784: GEMM-based level 3 BLAS: Portability and optimization issues*, ACM Transactions on Math. Software, 24 (1998), pp. 303–316.

[149] ——, *GEMM-based level 3 BLAS: High-performance model implementations and performance evaluation benchmark*, ACM Transactions on Math. Software, 24 (1998), pp. 268–302.

[150] C. Kane, J. E. Marsden, and M. Ortiz, *Symplectic-energy-momentum preserving variational integrators*, J. Math. Phys., 40 (1999), pp. 3353–3371.

[151] C. Kane, J. E. Marsden, M. Ortiz, and M. West, *Variational integrators and the Newmark algorithm for conservative and dissipative mechanical systems*, International Journal for Numerical Methods in Engineering, 49 (2000), pp. 1295–1325.

[152] C. Kane, E. A. Repetto, M. Oritz, and J. E. Marsden, *Finite element analysis of nonsmooth contact*, Computer Methods in Applied Mechanics and Engineering, 180 (1999), pp. 1–26.

[153] A. V. Karapetian, *On realizing nonholonomic constraints by viscous friction forces and Celtic stones stability*, J. Appl. Math. Mech., 45 (1981), pp. 42–51.

[154] D. C. Karnopp, D. L. Margolis, and R. C. Rosenberg, *System Dynamics: A Unified Approach*, John Wiley & Sons, New York, London, Sydney, second ed., 1990.

[155] D. M. Kaufman, T. Edmunds, and D. K. Pai, *Fast frictional dynamics for rigid bodies*, ACM Trans. Graph., 24 (2005), pp. 946–956.

[156] A. Kecskeméthy, *Object-Oriented Modelling of Mechanical Systems*, in Angeles and Kecskeméthy [11], July 1995, ch. 6, pp. 217–276.

[157] E. Keller, *The general quadratic optimization problem*, Mathematical Programming, 5 (1973), pp. 311–337.

[158] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, vol. 16 of SIAM Frontiers, SIAM Publ., Philadelphia, 1995.

[159] S. G. Kislicyn, *A tensor method in the theory of spatial mechanisms*, Trudy Sem. Teorii Mašin i Mehanizmov, 14 (1954), pp. 51–75.

[160] A. Klarbring and J.-S. Pang, *Existence of solutions to discrete semicoercive frictional contact problems*, SIAM J. Optim., 8 (1998), pp. 414–442 (electronic).

[161] A. Klarbring and J.-S. Pang, *The discrete steady sliding problem*, ZAMM Z. Angew. Math. Mech., 79 (1999), pp. 75–90.

[162] M. M. Kostreva, *Direct Algorithms for Complementarity Problems*, PhD thesis, Rensselaer Polytechnic Institute, Troy, New York, June 1976.

[163] P. Krysl and L. Endres, *Explicit Newmark/Verlet algorithm for time integration of the rotational dynamics of rigid bodies*, Internat. J. Numer. Methods Engrg., 62 (2005), pp. 2154–2177.

[164] J. Kuhl, D. Evans, Y. Papelis, R. Romano, and G. Watson, *The Iowa driving simulator - an immersive research environment*, Computer, 28 (1995), pp. 35–41.

[165] P. Kunkel, V. Mehrmann, W. Rath, and J. Weickert, *A new software package for linear differential-algebraic equations*, SIAM J. Sci. Comput., 18 (1997), pp. 115–138. Dedicated to C. William Gear on the occasion of his 60th birthday.

[166] A. J. Kurdila and F. J. Narcowich, *Sufficient conditions for penalty formulation methods in analytical dynamics*, Computational Mechanics, 12 (1993), pp. 81–96.

[167] C. Lacoursière, *Splitting methods for dry frictional contact problems in rigid multibody systems: Preliminary performance results*, in Conference Proceedings from SIGRAD2003, November 20–21, 2003, Umeå Unversity, Umeå, Sweden, M. Ollila, ed., SIGRAD, Nov. 2003, pp. 11–16.

[168] ——, *A regularized time stepper for multibody systems*, Tech. Rep. UMINF 06.04 ISSN-0348-0542, Dept. of Computing Science, Umeå University, Mar. 2006.

[169] ——, *Stabilizing gyroscopic forces in rigid multibody simulations*, Tech. Rep. UMINF 06.05 ISSN-0348-0542, Dept. of Computing Science, Umeå University, Mar. 2006.

[170] C. Lacoursière, *A parallel block iterative method for interactive contacting rigid multibody simulations on multicore PCs*, in PARA06 State of the Art in Scientific and Parallel Computing, Lecture Notes in Computer Science, Springer-Verlag, June 2007, p. xx. to appear in 2007.

[171] J. L. Lagrange, *Mécanique Analytique, Tome Premier*, Albert Blanchard, 1965. Édition complète réunissant les notes de la Troisième édition revue, corrigée et annotée par Joseph Bertrand, et de la Quatrième édition publiée sous la direction de Gaston Darboux, du texte original publié par l'Académie royale des Sciences le 27 février 1788.

[172] ——, *Mécanique Analytique, Tome Second*, Albert Blanchard, 1965. Édition complète réunissant les notes de la Troisième édition revue, corrigée et annotée par Joseph Bertrand, et de la Quatrième édition publiée sous la direction de Gaston Darboux, du texte original publié par l'Académie royale des Sciences le 27 février 1788.

[173] C. Lanczos, *The Variational Pinciples of Mechanics*, Dover Publications, New York, fourth ed., 1986.

[174] L. Landau and E. Lifchitz, *Mécanique*, no. 1 in Physique Théorique, Éditions Mir, 2, Pervi Rijski péréoulok, Moscou, I-110, GSP, U. R. S. S., 1982. Traduit du russe par Claude Ligny.

[175] L. D. Landau and E. M. Lifsic, *Fluid Mechanics*, vol. 6 of Course of theoretical physics, 99-0117585-2, Elsevier Butterworth Heinemann, Burlington, 2.repr. with corr. ed., 2004.

[176] D. F. Lawden, *Elliptic Functions and Applications*, vol. 80 of Applied Mathematical Sciences, Springer-Verlag, 1989.

[177] R. A. Layton, *Principles of Analytical System Dynamics*, Mechanical Engineering Series, Springer-Verlag, Berlin, Heidelberg, New York, London, Paris, Tokyo, Hong Kong, 1998.

[178] B. Leimkuhler and S. Reich, *Simulating Hamiltonian Dynamics*, vol. 14 of Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge, 2004.

[179] C. E. Lemke, *Bimatrix equilibrium points and mathematical programming*, Management Science, 11 (1965), pp. 681–689.

[180] A. Lew, J. E. Marsden, M. Ortiz, and M. West, *Asynchronous variational integrators*, Arch. Ration. Mech. Anal., 167 (2003), pp. 85–146.

[181] A. J. Lew, *Variational Time Integrators in Computational Solid Mechanics*, PhD thesis, California Institute of Technology, 2003.

[182] A. D. Lewis and R. M. Murray, *Variational principles for constrained systems: theory and experiment*, Internat. J. Non-Linear Mech., 30 (1995), pp. 793–815.

[183] D. Li and M. Fukushima, *Smoothing Newton and quasi-Newton methods for mixed complementarity problems*, Comp. Opt. and Appl., 17 (2000), pp. 203–230.

[184] K. W. Lilly, *Efficient Dynamic Simulation of Robotic Mechanisms*, vol. 203 of Kluwer International Series in Engineering and computer science, Kluwer Academic Publishers Group, Dordrecht, The Netherlands, 1993.

[185] P. Lin, *A sequential regularization method for time-dependent incompressible Navier-Stokes equations*, SIAM J. Numer. Anal., 34 (1997), pp. 1051–1071.

[186] C. Lobry, T. Sari, and S. Touhami, *On Tykhonov's theorem for convergence of solutions of slow and fast systems*, Electron. J. Differential Equations, 1998 (1998), pp. 1–22. tykhonov tykonov.

[187] A. Loock and E. Schömer, *A virtual environment for interactive assembly simulation: From rigid bodies to deformable cables*. `citeseer.ist.psu.edu/loock01virtual.html`.

[188] H. A. Lorentz, A. Einstein, H. Minkowski, and H. Weyl, *The Principle of Relativity: A Collection of Original Memoirs of the Special and General Theory of Relativity*, Dover Publications, 1952.

[189] P. Lötstedt, *Coulomb friction in two-dimensional rigid body systems*, Z angew. Math. Mech., 61 (1981), pp. 605–615.

[190] ——, *Mechanical systems of rigid bodies subject to unilateral constraints*, SIAM J. of Applied Math., 42 (1982), pp. 281–296.

[191] ——, *Numerical simulation of time-dependent contact and friction problems in rigid body mechanics*, SIAM J. Sci. Stat. Comput., 5 (1984), pp. 370–393.

[192] C. Lubich, C. Engstler, U. Nowak, and U. Pöhle, *Numerical integration of constrained mechanical systems using MEXX*, Mech. Struct. & Mach., 23 (1995), pp. 473–497.

[193] J. E. Marsden and T. S. Ratiu, *Introduction to Mechanics and Symmetry: A Basic Exposition of Classical Mechanical Systems*, no. 17 in Texts in Applied Mathematics, Springer-Verlag, 1999.

[194] J. E. Marsden, J. Scheurle, and J. M. Wendlandt, *Visualization of orbits and pattern evocation for the double spherical pendulum*, in ICIAM 95 (Hamburg, 1995), vol. 87 of Math. Res., Akademie Verlag, Berlin, 1996, pp. 213–232.

[195] J. E. Marsden and A. J. Tromba, *Vector Calculus*, W. H. Freeman, New York, third ed., 1988.

[196] J. E. Marsden and M. West, *Discrete mechanics and variational integrators*, Acta Numerica, 10 (2001), pp. 357–514.

[197] P. Matstoms, *Sparse `QR` factorization with applications to linear least squares problems*, Linköping Studies in Science and Technology. Dissertations, 337, Linköping University Department of Mathematics, Linköping, 1994.

[198] S. Mazur and S. Ulam, *Sur les transformations isométriques d'espaces vectoriels normés.*, C. R. Acad. Sci. Paris, 194 (1932), pp. 946–948.

[199] R. I. McLachlan, G. Reinout, and W. Quispel, *Splitting methods*, Acta Numerica, 11 (2002), pp. 341–434.

[200] R. I. McLachlan and A. Zanna, *The discrete Moser-Veselov algorithm for the free rigid body, revisited*, Found. Comput. Math., 5 (2005), pp. 87–123.

[201] N. Melin, *Real time simulation of deformable objects*, Master's thesis, Department of Physics, Umeå University, Umeå, Sweden, Jan. 2006.

[202] V. J. Milenkovic and H. Schmidl, *Optimization-based animation*, in SIGGRAPH 2001 Conference Proceedings, August 12–17, 2001, Los Angeles, CA, Computer Graphics Proceedings, Annual Conference Series, 2001, pp. 37–46.

[203] B. Mirtich, *Timewarp rigid body simulation*, in Proceedings of the 27th annual Conference on Computer Graphics and Interactive Techniques, ACM Press/Addison-Wesley Publishing Co., 2000, pp. 193–200.

[204] B. Mirtich and J. Canny, *Impulse based simulation of rigid bodies*, in Symposium on Interactive 3D Graphics, New York, 1995, ACM, ACM Press, pp. 181–188.

[205] B. V. Mirtich, *Impulse-Based Dynamic Simulation of Rigid Body Systems*, PhD thesis, University of California at Berkeley, Berkeley, CA, USA, 1996.

[206] K. Modin and C. Führer, *Time-step adaptivity in variational integrators with application to contact problems*, ZAMM Z. Angew. Math. Mech., 86 (2006), pp. 785–794.

[207] J. J. Moré, B. S. Garbow, and K. E. Hillstrom, *User guide for MINPACK-1*, Tech. Rep. ANL-80-74, Argonne National Laboratory, Argonne, IL, USA, Aug. 1980.

[208] J. Moser and A. P. Veselov, *Discrete versions of some classical integrable systems and factorization of matrix polynomials*, Comm. Math. Phys., 139 (1991), pp. 217–243.

[209] K. G. Murty, *Note on a Bard-type scheme for solving the complementarity problems*, Opsearch, 11 (1974), pp. 123–130.

[210] K. G. Murty and F.-T. Yu, *Linear Complementarity, Linear and Nonlinear Programming*, Self-published: internet edition, Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, http://www-personal.engin.umich.edu/~murty/book/LCPbook/index.html, 1997.

[211] D. Negrut, E. J. Haug, and H. C. German, *An implicit Runge-Kutta method for integration of differential algebraic equations of multibody dynamics*, Multibody System Dynamics, 9 (2003), pp. 121–142.

[212] D. Negrut, E. J. Haug, and M. Iancu, *Variable step implicit numerical integration of stiff multibody systems*, in NATO ASI on Computational Methods in Mechanisms, Sofia, Bulgaria, June 1997, NATO ASI, pp. 157–166.

[213] D. Negrut, A. Sandu, E. J. Haug, F. A. Potra, and C. Sandu, *A Rosenbrock-Nystrom state space implicit approach for the dynamic analysis of mechanical systems: II - Method and numerical examples*, Proc. Institution Mechanical Engineers Part K-journal Multi-body Dynamics, 217 (2003), pp. 273–281.

[214] N. M. Newmark, *A method of computation for structural dynamics*, ASCE J. of the Engineering Mechanics Division, (1959), pp. 67–94.

[215] I. Newton, *Philosophiea Naturalis Principia Mathematica*, Imprimatur S. Pepys, Reg, Soc. Praeses, Londini, julii 5 1686. Translated to English by Andrew Motte in 1729, revised and edited by Florian Cajori, published as "*Sir Isaac Newton's* Mathematical Principles of Natural Philosohpy and his System of the World, Volume One: The Motion of Bodies", by University of California press, Berkeley, Los Angeles, London, 1964.

[216] E. G. Ng and B. W. Peyton, *Some results on structure prediction in sparse* `QR` *factorization*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 443–459.

[217] W. T. Obuchowska, *Exceptional families and existence results for nonlinear complementarity problem*, J. of Global Optimization, 19 (2001), pp. 183–198.

[218] S. Oh, J. Ahn, and K. Wo, *Low damped cloth simulation*, Visual Comput., 22 (2006), pp. 70–79.

[219] D. E. Orin and W. W. Schrader, *Efficient computation of jacobian for robot manipulator*, Int. J. Robotics Res., 3 (1984), pp. 66–75.

[220] C. O'Sullivan, J. Dingliana, T. Giang, and M. K. Kaiser, *Evaluating the visual fidelity of physically based animations*, ACM Trans. Graph., 22 (2003), pp. 527–536.

[221] M. Otter, M. Hocke, A. Daberkow, and G. Leister, *An object-oriented data model for multibody systems*, in Schiehlen [246], pp. 2–48.

[222] D. K. Pai, *Strands: Interactive simulation of thin solids using cosserat models*, in Eurographics, 2002.

[223] P. Painlevé, *Sur les lois du frottement de glissement*, Comptes Rendus Acad. Sci. Paris, 121 (1895), pp. 112–115.

[224] P. D. Panagiotopoulos and C. Glocker, *Analytical mechanics. Addendum I: inequality constraints with elastic impacts. The convex case*, ZAMM Z. Angew. Math. Mech., 78 (1998), pp. 219–229.

[225] A. Pandolfi, C. Kane, J. E. Marsden, and M. Ortiz, *Time-discretized variational formulation of non-smooth frictional contact*, International Journal for Numerical Methods in Engineering, 53 (2002), pp. 1801–1829.

[226] J.-S. Pang, *Newton's method for B-differentiable equations*, Math. Oper. Res., 15 (1990), pp. 311–341.

[227] J.-S. Pang and D. E. Stewart, *A unified approach to discrete frictional contact problems*, Internat. J. Engrg. Sci., 37 (1999), pp. 1747–1768.

[228] J.-S. Pang and J. C. Trinkle, *Complementarity formulations and existence of solutions of dynamics multi-rigid-body contact problems with Coulomb friction*, Journal of Mathematical Computing, 73 (1996), p. 199.

[229] J.-S. Pang, J. C. Trinkle, and G. Lo, *A complementarity approach to a quasistatic multi-rigid-body contact problem*, Computational Optimization and Applications, 5 (1994), pp. 139–155.

[230] T. W. Park and E. J. Haug, *A hybrid numerical-integration method for machine dynamic simulation*, J. Mechanisms Transmissions Automation In Design-transactions Asme, 108 (1986), pp. 211–216.

[231] B. N. J. Persson, *Sliding friction*, NanoScience and Technology, Springer-Verlag, Berlin, 1998. Physical principles and applications.

[232] F. Pfeiffer and C. Glocker, *Multibody Dynamics with Unilateral Contacts*, Wiley Series in Nonlinear Science, John Wiley & Sons, New York, London, Sydney, 1996.

[233] A. B. K. Rao, S. K. Saha, and P. V. M. Rao, *Dynamics modelling of hexaslides using the decoupled natural orthogonal complement matrices*, Multibody Syst. Dyn., 15 (2006), pp. 159–180.

[234] M. H. Refaat and S. A. Meguid, *On the elastic solution of frictional contact problems using variational inequalities*, Int. J. Mech. Sci., 36 (1994), pp. 329–342.

[235] ——, *On the modeling of frictional contact problems using variational inequalities*, Finite Elements in Analysis and Design, 19 (1995), pp. 89–101.

[236] ——, *A novel finite element approach to frictional contact problems*, International Journal for Numerical Methods in Engineering, 39 (1996), pp. 3889–3902.

[237] ——, *Updated Langrangian formulation of contact problems using variational inequalities*, International Journal for Numerical Methods in Engineering, 40 (1997), pp. 2975–2993.

[238] ——, *A new strategy for the solution of frictional contact problems*, International Journal for Numerical Methods in Engineering, 43 (1998), pp. 1053–1068.

[239] J. M. Rolfe and K. J. Staples, eds., *Flight Simulation*, Cambridge University press, 1986. Reprinted in 2004.

[240] C. W. Rowley and J. E. Marsden, *Variational integrators for degenerate Lagrangians, with application to point vortices*, in Decision and Control, 2002, Proceedings of the 41st IEEE Conference, vol. 2, Dec. 2002, pp. 1521–1527.

[241] H. Rubin and P. Ungar, *Motion under a strong constraining force*, Communications on Pure and Applied Mathematics, X (1957), pp. 65–87.

[242] J.-P. Ryckaert, G. Ciccotti, and H. J. C. Bendersen, *Numerical integration of the Cartesian equations of motion of a system with constraints: Molecular dynamics of n-alkanes*, J. Comp. Phys., 23 (1977), pp. 327–341. shake.

[243] S. K. Saha, *Dynamics of serial multibody systems using the decoupled natural orthogonal complemetnt matrices*, ASME J. Appl. Mech., 66 (1999), pp. 986–996.

[244] A. Sandu, D. Negrut, E. J. Haug, F. A. Potra, and C. Sandu, *A Rosenbrock-Nystrom state space implicit approach for the dynamic analysis of mechanical systems: I - Theoretical formulation*, Proc. Institution Mechanical Engineers Part K-journal Multi-body Dynamics, 217 (2003), pp. 263–271.

[245] R. W. H. Sargent, *An effcient implementation of the Lemke algorithm and its extension to deal with upper and lower bounds*, Mathematical Programming Study, 7 (1978), pp. 36–54.

[246] W. Schiehlen, ed., *Advanced Multibody Dynamics: simulation and software tools*, vol. 20 of Solid Mechanics and its Applications, Kluwer Academic Publishers Group, Dordrecht, The Netherlands, 1993.

[247] C. Schlier and A. Seiter, *High-order symplectic integration: an assessment*, Computer Phys. Comm., 130 (2000), pp. 176–189.

[248] B. F. Schutz, *Geometrical methods of mathematical physics*, Cambridge U. P., Cambridge, 1980.

[249] ——, *A first course in general relativity*, Cambridge U. P., Cambridge, 1985.

[250] R. Serban and E. J. Haug, *Kinematic and kinetic derivatives in multibody system analysis*, Mechanics Structures Machines, 26 (1998), pp. 145–173.

[251] ——, *Globally independent coordinates for real-time vehicle simulation*, J. Mechanical Design, 122 (2000), pp. 575–582.

[252] R. Serban, D. Negrut, E. J. Haug, and F. A. Potra, *A topology-based approach for exploiting sparsity in multibody dynamics in Cartesian formulation*, Mech. Struct. & Mach., 25 (1997), pp. 379–396.

[253] M. Servin and C. Lacoursière, *Massless cable for real-time simulation*, Accepted for publication in Computer Graphics Forum, 26 (2006). To appear in issue 2 or 4.

[254] M. Servin, C. Lacoursière, and N. Melin, *Interactive simulation of elastic deformable materials*, in Proceedings of SIGRAD Conference 2006 in Skövde, Sweden, Linköping University Electronic Press, Linköping, 2006, pp. 22–32.

[255] L. Shiau and R. Glowinski, *Operator splitting method for friction constrained dynamical systems*, Discrete Contin. Dyn. Syst., (2005), pp. 806–815.

[256] A. Shrijver, *Theory of Linear and Integer Programming*, Wiley-Intersicence Series in Discrete Mathematics, John Wiley & Sons, New York, London, Sydney, 1986.

[257] S. Srinivasan, D. P. Mital, and S. Haque, *A quantitative analysis of the effectiveness of laparoscopy and endoscopy virtual reality simulators*, Computers & Electrical Engineering, 32 (2006), pp. 283–298.

[258] D. Stewart, *A high accuracy method for solving ODEs with discontinuous right-hand side*, Numerische Mathematik, 58 (1990), pp. 299–328.

[259] D. E. Stewart, *Existence of solutions to rigid body dynamics and the Painlevé paradoxes*, C. R. Acad. Sci. Paris, 325, Série I (1997), pp. 689–693.

[260] ——, *Time-stepping methods and the mathematics of rigid body dynamics*, in Proceedings of the first International Symposium on Impact and Friction of Solids, Structures and Intelligent Machines: in memoriam P.D. Panagiotopoulos (1950-1998), Ottawa Congress Centre, Ottawa, Canada, 27-30 June 1998, A. A. Guran and P. D. Panagiotopoulos, eds., vol. 14 of Series on stability, vibration and control of systems. Series B, Singapore, 2000, World Scientific, pp. 183–222.

[261] 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, 39 (1996), pp. 2673–2691.

[262] ——, *Dynamics, friction, and complementarity problems*, in Complementarity and Variational Problems: State of the Art, M. C. Ferris and J.-S. Pang, eds., Philadelphia, 1997, SIAM Publ., pp. 425–439.

[263] A. Tasora and P. Righettini, *Application of the quaternion algebra to the efficient computation of jacobians for holonomic rheonomic-constraints*, in Proc. of EUROMECH Colloquium on Advances in Computational Multibody Dynamics, Lisbon, Portugal, Sept. 20–23, J. A. C. Ambrósio and W. O. Schielen, eds., IDMEC/IST Euromech Colloquium 404, 1999, pp. 75–92.

[264] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer, *Elastically deformable models*, in Proceedings of the 14th annual conference on Computer Graphics and Interactive Techniques, ACM Press, 1987, pp. 205–214.

[265] J. Thierry-Mieg, *Geometrical reinterpretation of Faddeev-Popov ghost particles and BRS transformations*, J. Math. Phys., 21 (1980), pp. 2834–2838.

[266] J. C. Trinkle, J. S. Pang, S. Sudarsky, and G. Lo, *On dynamic multi-rigid-body contact problems with Coulomb friction*, Z angew. Math. Mech., 77 (1997), pp. 267–279.

[267] F. F. Tsai and E. J. Haug, *Real-time multibody system dynamic simulation I: A modified recursive formulation and topological analysis*, Mechanics Structures Machines, 19 (1991), pp. 99–127.

[268] ——, *Real-time multibody system dynamic simulation II: A parallel algorithm and numerical results*, Mechanics Structures Machines, 19 (1991), pp. 129–162.

[269] J. A. Tzitzouris, *Numerical Resolution of Frictional Multi-Rigid-Body Systems via Fully Immplicit Time-Stepping and Nonlinear Complementarity*, PhD thesis, Johns Hopkins University, 2001.

[270] J. Väisälä, *A proof of the Mazur-Ulam theorem*, Am. Math. Monthly, 110 (2003), pp. 634–636.

[271] L. Verlet, *Computer "experiments" on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules.*, Phys. Rev., 159 (1967), pp. 98–103.

[272] R. von Schwerin, *MultiBody System SIMulation*, no. 7 in Lecture Notes in Computational Science and Engineering, Springer-Verlag, Berlin, Heidelberg, New York, London, Paris, Tokyo, Hong Kong, 1999.

[273] S. Š and S. Saigal, *Frictional contact formulation using quadratic programming*, Computational Mechanics, 15 (1994), pp. 173–187.

[274] X. Wang, E. J. Haug, and W. Pan, *Implicit numerical integration for design sensitivity analysis of rigid multibody systems*, Mechanics Based Design Structures Machines, 33 (2005), pp. 1–30.

[275] R. Weinstein, J. Teran, and R. Fedkiw, *Dynamic simulation of articulated rigid bodies with contact and collision*, IEEE TVCG, 12 (2006), pp. 365–374.

[276] J. M. Wendlandt, *Pattern evocation and energy-momentum integration of the double spherical pendulum*, Master's thesis, Dept. of Mathematics, UC Berkeley, 1997.

[277] J. M. Wendlandt and J. E. Marsden, *Mechanical integrators derived from a discrete variational principle*, Phys. D, 106 (1997), pp. 223–246.

[278] D. Włodarczyk, *The Multi-Body Contact Problem with Friction*, december, Dept. of Mathematical Sciences, Clemson University, 2005.

[279] Q. Wu, M. X. Zhu, and N. S. V. Rao, *System design for on-line distributed computational visualization and steering*, Technologies For E-learning Digital Entertainment, Proc., 3942 (2006), pp. 1121–1130.

[280] S. C. Wu, S. M. Yang, and E. J. Haug, *Dynamics of mechanical systems with Coulomb-friction, stiction, impact and constraint addition deletion II: planar systems*, Mechanism Machine Theory, 21 (1986), pp. 407–416.

[281] ——, *Dynamics of mechanical systems with Coulomb-friction, stiction, impact and constraint addition deletion III: Spatial systems*, Mechanism Machine Theory, 21 (1986), pp. 417–425.

[282] W. S. Yoo and E. J. Haug, *Dynamics of articulated structures 1: Theory*, J. Struct. Mechanics, 14 (1986), pp. 105–126.

[283] ——, *Dynamics of articulated structures 2: Computer implementation and applications*, J. Struct. Mechanics, 14 (1986), pp. 177–189.

[284] W. Yourgrau and S. Mandelstam, *Variational Principles in Dynamics and Quantum Theory*, Dover Publications, New York, 1979.

[285] L. Zhang and Z. Gao, *Quadratic one-step smoothing Newton method for P_0 LCP without strict complementarity*, Appl. Math. and Comp., 140 (2003), pp. 367–379.

Index

Colophon

Typesetting was done using Donald E. Knuth's TEX formatting engine and Leslie Lamport's LATEX 2ε system. AMS-LATEX font and macro packages were used extensively for typesetting the math. The layout was done using the `scrbook` document class from KOMA-Script package. The ten points version of the Concrete Roman font of Donald E. Knuth was used with the Euler math font from the American Mathematical Society. This is available in the LATEX 2ε package `ccfonts`.

In addition, over 36 more LATEX 2ε packages were used, notably including the `glossary` package by Nicola Talbot, and several utilities from David Carlisle. The `makeindex` program was used to compile the glossary, the notation index, and the main index. The `glossary` package provided automatic acronym expansion as well. The final printable file was produced by first converting the `dvi`-file to PostScript with `dvips`, and then using `ps2pdf`, which is based on GNU-ghostscript, to produce the PDF that was printed.

GNU-`emacs` with `viper` mode was used to edit everything. In conjunction with the `AUC-TeX` and `RefTeX` packages, this was extremely effective for writing the TEX source. Do not write a TEX document without this type of tools, and study them carefully! Spell-checking was done with `aspell` and various other checks were made using the `lacheck`, `chtex`, and `style-check` utilities. Indexing was done using the facilities provided by `RefTeX`.

The bibliography database was created with BIBTEX and entries were taken from the MathSciNet database of the AMS (those entries have MR codes in them) when possible. In other cases, they were collected from the Association for Computing Machinery (ACM) portal, or from various electronic journals. Only in last resort they entered manually, and this hopefully minimized the number of errors. In all cases, the `bibtools` utilities were used to convert records to BIBTEX format and then cleaned up using Nelson Beebe's collection of utilities and `emacs` editing modes. When possible, the entries for the books were collected using Beebe's `cattobib` tool using the Z39.50 protocol to query national libraries across the world over the Internet. The bibliography was typeset using the Society for Industrial and Applied Mathematics (SIAM) bibliography style. All this automation made the handling and processing of my 800+ records manageable—though Bosse still found errors when proof reading.

All simple simulations were performed with Octave and plots were handled by `gnuplot`. Legacy simulation data was stored in custom designed `hdf5` formats, allowing for easy loading in Octave for post-processing.

Pictures were generated using `xfig` and this was then processed by `transfig` to generate suitable combinations of encapsulated PostScript files and LaTeX `eepic` macros for inclusion in the main document, so the fonts of the pictures would match that of the main text. Inkscape is superior to `xfig` and was considered to make nicer looking diagrams but a problem with handling of text elements made that impossible. Now that the thesis is complete, there is time to work on the Inkscape code to make it work like `xfig` and `transfig` so the figures in the next project will look nicer still.

Countless global edits and adjustments to the `gnuplot` output were performed using combinations of `Perl` and `bash` shell scripts, relying in turn on countless POSIX utilities and filters. Every erg spent in learning about regular expressions and the various tools paid back thousandfolds. All plots and simulations were generated from driver scripts, and these were driven by the `make` utility to guarantee systematic updating of the final results whenever a script had been changed (and this happened often). This means that all the data provided in the figures is 100% reproducible. In addition, every piece of text, software, and raw data, was under revision control using SVN.

Small changes had to be made to several of the software utilities listed above to make them do what was needed, and patches were gladly submitted when useful to others. This was possible because the source code and the authors were responsive to questions, suggestions, and bug reports. If you did not guess it yet, the operating system was a patched up Ubuntu distribution of the GNU/Linux system.