Projected and Partitioned Runge–Kutta Methods for Nonsmooth Mechanical Systems with Frictional Contact and Impacts

Von der Fakultät Konstruktions-, Produktions- und Fahrzeugtechnik
der Universität Stuttgart zur Erlangung der Würde eines
Doktor-Ingenieurs (Dr.-Ing.) genehmigte Abhandlung

Von
Jonas Breuling (geb. Harsch)
aus Stuttgart

Hauptberichter:	Prof. Dr. ir. habil. Remco I. Leine
Mitberichter:	Prof. Dr. sc. Simon R. Eugster
	Dr. habil. Vincent Acary
Tag der mündlichen Prüfung:	18.10.2024

Institut für Nichtlineare Mechanik
der Universität Stuttgart

2024

To Carlo and Olivia,
with love and gratitude.

Preface

This thesis was written during my time at the Institute of Nonlinear Mechanics at the University of Stuttgart. I would like to express my deepest gratitude to Prof. Dr. Remco I. Leine for providing me the opportunity to pursue my doctorate under his supervision and for granting me the freedom to explore my scientific interests. From my early bachelor's studies, you sparked my fascination for nonsmooth mechanics, particularly through your captivating and engaging lecture on the specialization subject, "Nonlinear Mechanics", which was unmatched in entertainment value. In my subsequent master's studies, your unique course on nonsmooth mechanics left a lasting impact, and I continue to reference it today. Thank you for your invaluable advice and your often challenging yet insightful questions, which have been crucial to my growth. I eagerly look forward to our future collaboration as an "Akademischer Rat" at your institute and deeply appreciate the trust you have placed in me.

A special note of thanks goes to Prof. Dr. Simon R. Eugster, my first contact at the Institute of Nonlinear Mechanics. Over eight years ago, your lecture on "Dynamics of Mechanical Systems" laid the foundation for my mechanics education and ignited my desire to delve deeper into the field. Although an email spam filter almost prevented any future collaboration, I was fortunate to begin my bachelor thesis with you the following semester. You introduced me to nonlinear beam theories and their finite element discretization, a subject that has occupied me not only during my master's thesis but throughout much of my time at the institute. Your mentorship provided me with a first-class education, and you were always available for deep and enriching discussions. I fondly recall our many conference trips, where late-night conversations often led to exciting new ideas. I hope these discussions will continue in the future. Thank you for the unforgettable time in Stuttgart and your unconditional support, which finally convinced me to write my doctoral thesis on nonsmooth time integration methods.

I am also immensely grateful to my second co-examiner, Dr. habil. Vincent Acary, for his interest in my research and for reviewing my thesis. Our numerous discussions – whether on nonsmooth mechanics, mechanical history, or the beauty of the French Alps during conferences – have been intellectually stimulating. I look forward to continuing our scientific collaboration and eagerly anticipate our next meeting at a future conference.

My sincere thanks extend to all the current and former members of the Institute of Nonlinear Mechanics. I have shared many wonderful years with you in Stuttgart, and several of you have directly contributed to the quality of this thesis through your feedback. Special thanks to Fabia Bayer and Yassine Karoui for your insightful suggestions on the mathematical preliminaries, and to Dr. Simon Sailer, Dr. Giuseppe Capobianco, and Lisa Eberhardt for our countless discussions on mechanics and Python bootcamps. Giuseppe, I am especially grateful for your valuable comments on my dissertation and for being my go-to resource on nonsmooth mechanics. Even though some of you are no longer in Stuttgart, you remain cherished colleagues and friends. I hope to see you all at the institute's Christmas party in the years to come.

Lastly, my deepest gratitude is reserved for my family, and especially my wife, Olivia. You have stood by me through all the ups and downs of this journey, offering unwavering love, understanding, and support. Without you, none of this would have been possible. I'm thrilled for the new journey we're about to start as three.

Stuttgart, December 2024 *Jonas Breuling*

Contents

Abstract

This dissertation addresses the numerical time integration of nonsmooth mechanical systems with frictional contact and impacts. Due to the discontinuous nature of their states, these systems are described using measure differential inclusions, which can exhibit inherent stiffness. Therefore, their mathematical formulation draws upon concepts from convex analysis, specifically involving convex sets, normal cone inclusions, and their connections to variational inequalities and (linear) complementarity problems. Furthermore, assembling complex mechanical systems from simpler components requires introducing the concept of perfect bilateral constraints. This process of mathematical modeling typically involves combining flexible and rigid bodies. Therefore, the measure differential inclusions described herein are augmented by discrete impact laws to fully capture experimentally observed physical phenomena.

Given the challenges posed by this modeling approach for numerical time integration methods, this thesis introduces two novel families of implicit Runge-Kutta methods tailored for mechanical systems with frictional contact and impacts. Specifically, these are the *nonsmooth projected stiffly accurate Runge–Kutta methods* and *nonsmooth partitioned Runge–Kutta methods*. This extends the application of implicit Runge–Kutta methods, well-known for their proficiency in solving stiff differential algebraic equations, to measure differential inclusions with mechanical structure. The monograph provides a comprehensive introduction to implicit Runge–Kutta methods for ordinary differential and differential algebraic equations, emphasizing their key features through embedded examples.

The resulting time-discretized equations are given by nonlinear normal cone inclusion problems. While normal cone inclusions elegantly describe set-valued force laws, they are challenging for numerical solution methods. Therefore, this thesis discusses Newton methods and fixed-point iterations for solving such class of problems.

An exclusive collection of numerical experiments demonstrates the outstanding capabilities of the proposed numerical time integration methods to solve nonsmooth mechanical systems. It is showcased that the proposed methods are at least on par with existing approaches and often outperform them.

Zusammenfassung

Diese Dissertation behandelt die numerische Zeitintegration von nichtglatten mechanischen Systemen mit Reibkontakt und Stößen. Aufgrund der unstetigen Natur ihrer Zustände werden diese Systeme mittels Maßdifferentialinklusionen beschrieben, die eine intrinsische Steifigkeit aufweisen können. Daher stützt sich ihre mathematische Formulierung auf Konzepte der konvexen Analysis, insbesondere auf konvexe Mengen, Normalkegelinklusionen und wie diese in Verbindung zu variationellen Ungleichungen und (linearen) Komplementaritätsproblemen stehen. Darüber hinaus erfordert das Zusammensetzen komplexer mechanischer Systeme aus einfacheren Komponenten die Einführung des Konzepts der perfekten bilateralen Zwangsbedingungen. Dieser Prozess der mathematischen Modellierung umfasst typischerweise die Kombination von flexiblen und starren Körpern. Daher werden die hier beschriebenen Maßdifferentialinklusionen durch diskrete Stoßgesetze ergänzt, um experimentell beobachtete physikalische Phänomene vollständig zu erfassen.

Angesichts der Herausforderungen, die dieser Modellierungsansatz für numerische Zeitintegrationsverfahren darstellt, führt diese Dissertation zwei neue Familien an impliziten Runge–Kutta Verfahren ein, die für mechanische Systeme mit Reibkontakt und Stößen maßgeschneidert sind. Konkret handelt es sich um *nichtglatte projizierte steif genaue Runge–Kutta Verfahren* und *nichtglatte partitionierte Runge–Kutta Verfahren*. Dies erweitert die Anwendung der impliziten Runge–Kutta Verfahren, die bekannt für ihre Effizienz bei der Lösung steifer differential-algebraischer Gleichungen sind, auf Maßdifferentialinklusionen mit mechanischer Struktur. Die Monographie bietet eine umfassende Einführung in implizite Runge–Kutta Verfahren für gewöhnliche Differentialgleichungen sowie differential-algebraische Gleichungen und hebt dabei ihre wichtigsten Merkmale durch eingebettete Beispiele hervor.

Die resultierenden zeitdiskretisierten Gleichungen sind durch nichtlineare Normalkegelinklusionsprobleme gegeben. Obwohl Normalkegelinklu-

sionen mengenwertige Kraftgesetze elegant beschreiben können, stellen sie eine Herausforderung für numerische Lösungsmethoden dar. Daher diskutiert diese Dissertation Newton Verfahren und Fixpunktiterationen zur Lösung dieser Probleme.

Eine einzigartige Sammlung von numerischen Beispielen zeigt die herausragenden Fähigkeiten der vorgestellten numerischen Zeitintegrationsverfahren zur Lösung von nichtglatten mechanischen Systemen. Es wird gezeigt, dass die vorgeschlagenen Methoden mindestens auf dem Niveau bestehender Ansätze sind und diese oft übertreffen.

Chapter 1

Introduction

> On cherche en effet des algorithmes permettant de prendre en compte des chocs éventuels, survenant à des instants a priori inconnus: des méthodes "multipas" semblent a priori inadéquates (mais il ne faudrait pas fermer la voie à toute recherche dans ce sens).
>
> —*Jean Jacques Moreau*

This dissertation is concerned with the numerical solution of mechanical systems with frictional contact and impacts. As outlined by a self-contained theory, such systems are elegantly described by possibly stiff measure differential inclusions. To this day, the numerical solution of such systems is a challenging task and is still subject of ongoing research. Thereto, this thesis proposes two novel families of implicit Runge–Kutta methods for the numerical time integration of possibly stiff measure differential inclusions with mechanical structure. In order to circumvent problems with rank deficient Jacobian matrices in semi-smooth Newton methods, arising from redundant contacts of rigid body models, a simple fixed-point iteration method is presented. Finally, selected benchmark examples validate the proposed methods and show their superior performance compared with existing methods.

1.1 Motivation and Research Question

Accurately modeling the dynamics of mechanical systems with frictional contact is of paramount importance across numerous scientific disciplines. Unlike classical smooth dynamical systems, these exhibit sudden changes

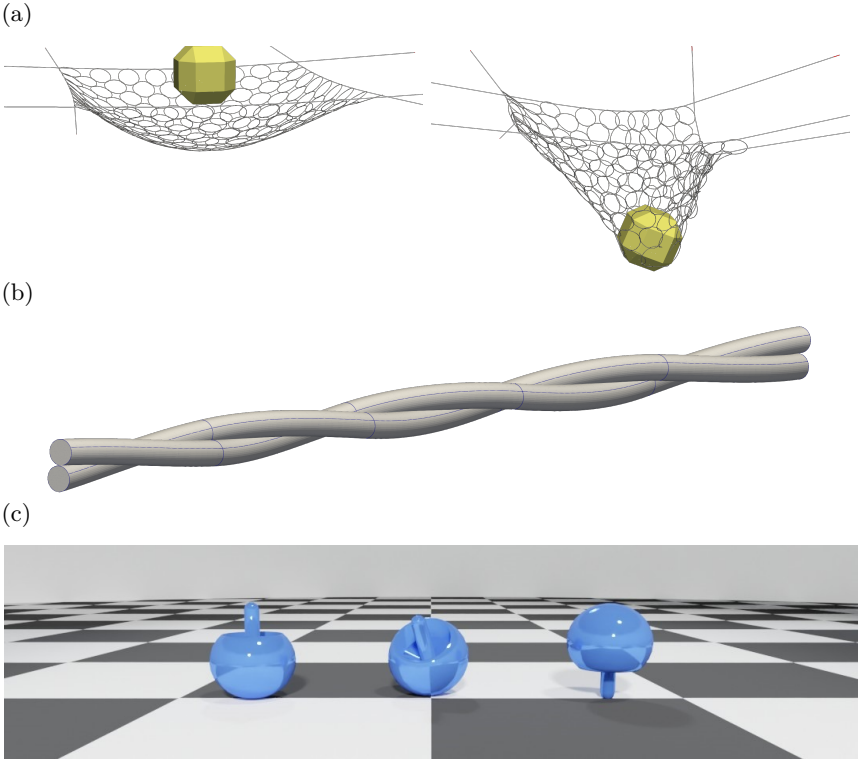

Figure 1.1: Applications of nonsmooth mechanical systems. (a) Initial and deformed configuration of rockfall protection nets [55], (b) Twisted rod assembly and (c) snapshots of the tippetop toy [154].

in velocity due to impacts and frictional interactions. This necessitates a specialized mathematical framework known as *nonsmooth mechanics*, see e.g. [5, 30, 72, 113, 128]. The omnipresence of mechanical systems with frictional contact in real world applications ranges from mechanical engineering problems [75] to natural hazard research [117], see Figure 1.1 for some illustrations. Prominent examples are given by legged robots navigating rough terrain [69] and physics engines required for reinforcement learning [121] or as the core of modern game engines [50]. Therein, mostly rigid bodies are interconnected by joints, springs and dash-pot dampers. On the contrary, in the emerging field of soft robotics, the systems are constructed from compliant materials. Hence, their mathematical modeling as well as a subsequent numerical solution has to deal with the description

of such compliant materials. Typically, finite element models are applied for the spatial discretization of the soft matter [9], which introduces a huge number of degrees of freedom, making this approach infeasible to be solved within real-time environments [13]. Alternative formulations rely on the application of reduced one-dimensional rod finite element models that can adequately describe the compliant slender structures [23, 60, 152, 173]. This approach leads to possibly stiff differential equations and introduces again rigidity due to the assumed incompressible cross-sections [25]. The aforementioned stiffnesses pose demanding stability requirements on the numerical time integration of such systems. Similar methods are recently investigated for the simulation and control of legged robots [20, 153], numerical description of waving processes [52, 53, 54], the description of fiber-reinforced materials [143, 165, 166] and the numerical simulation of bio polymers [124, 131]. Another related and very recent research field is interested in the development of new meta-materials that are based on a pantographic microstructure [46, 87, 168] and involve frictional contact interactions of highly slender structures. This small summary of applications can be completed by more classical ones from mechanical engineering. Typically, this involves the simulation of industrial products like electropneumatic or percussion drilling machines [72], understanding the phenomenon of curve squealing of railway wheels [75] and improving the grasping and sensing of robots within manufacturing lines [120].

Since nonsmooth mechanical systems introduce some sort of unilateral constraint, they can be seen as the natural extension of bilaterally constrained mechanical systems [106]. The latter one can elegantly be described by systems of differential algebraic equations (DAE's). Consequently, it is advisable to strive the extension of well-known numerical time integration methods for differential algebraic equations to nonsmooth mechanical systems. Charles William "Bill" Gear is widely accepted as the first mathematician of modern time who turned his attention to the field of differential algebraic equations [162]. In his renowned paper [66] he coined the name *differential algebraic equations*. Ten years later, his academic student Linda Petzold published the famous software DASSL – *a differential/algebraic system solver* [140] that successfully can solve differential algebraic problems of the form $\boldsymbol{F}(t, \boldsymbol{x}, \dot{\boldsymbol{x}}) = \boldsymbol{0}$ with differentiation index one and two. The code implements a variable step-size variable order fixed leading coefficient [28] variant of backward differentiation formulas (BDF), the most prominent representative of linear multi-step methods. Later, the joint work of Ernst Hairer, Christian Lubich and Michel Roche published the groundbreaking monograph on *The Numerical Solution of Differential-Algebraic Equations by Runge–Kutta Methods* [83]. Besides the remarkable

theoretical content of this work, the authors published the new DAE code RADAU5 that uses a three-stage (5th-order) Radau IIA method for the solution of stiff differential algebraic equations up to differentiation index three. Driven by the natural rivalry between multi-step and single-step methods of Runge–Kutta-type, the two codes DASSL and RADAU5 were the main driving force in the development of numerical methods for stiff differential algebraic equations [162]. This rivalry is nicely reflected by the often mentioned quote of Linda Petzold saying that *BDF is so beautiful that it is hard to imagine something else could be better* [82]. Although both multi-step and single-step methods are successfully applied to differential algebraic equations, their extension to nonsmooth mechanical systems involving frictional contact and impacts is a challenging task. As apparent from the given quote at the beginning of this chapter, already Jean Jacques Moreau [127] recognized that the application of multi-step methods to nondifferentiable or nonsmooth systems introduces serious problems due to the dependence of the methods on previously computed states. Moreover, implicit Runge–Kutta methods have superior stability properties compared to higher-order multi-step methods like BDF. Therefore, its application to nonsmooth mechanical systems with frictional contact and impacts is kept as an open task for future investigations and this thesis concentrates on implicit Runge–Kutta methods. Specifically, the research question that has to be answered within this thesis is as follows.

Research question: How can implicit Runge–Kutta methods be extended to stiff measure differential inclusions with mechanical structure?

1.2 Literature Review and Research Objectives

With the above stated research question in mind, this section reviews existing time integration methods for nonsmooth mechanical systems, with particular attention given to their features and drawbacks. By examining these methods, mandatory design rules for developing nonsmooth time integration methods for mechanical systems with frictional contact and impacts can be distilled. These design rules will be stated as the research objectives of this thesis and are subsequently implemented by the proposed families of nonsmooth Runge–Kutta methods.

Time integration methods for nonsmooth mechanical systems are classically distinguished as either being event-driven (see the work of Pfeiffer and Glocker [72, 141]) or event-capturing [5, 30, 112, 113, 128] methods.

Since the former is mainly designed for systems with a small number of events and cannot describe accumulation points, this thesis focuses on the latter approach.

During the late 80s, the most famous time integration method for nonsmooth mechanical system was proposed by Jean Jacques Moreau [127, 128]. The method is known as Moreau's mid-point rule [74, 130], since it performs an explicit mid-point rule for the position update and detects contacts as active if the corresponding mid-point positions violate the unilateral constraint conditions. Subsequently, the impulsive dynamics is discretized by an implicit scheme that combines the unilateral constraint conditions on velocity level with a possibly elastic impact law. Various variants thereof have been developed subsequently, see e.g. [102, 103, 130], which all have in common that the unilateral constraint conditions are only enforced on velocity level. Although a position stabilization approach in the sense of Gear–Gupta–Leimkuhler (GGL) [68] is proposed in [159], closer inspection of the method reveals that the unilateral constraints are considered to be active using the very same explicit prediction of the mid-point positions as already proposed by Moreau. Hence, all the resulting schemes suffer from unilateral constraint drift/contact penetration. Eliminating such penetration problems is a desirable property of nonsmooth time integration methods and will be stated as the first research objective of this thesis.

Objective 1: *Impenetrability* – To develop numerical time integration methods that satisfy unilateral constraints on position level thereby circumventing excessive penetration due to constraint drift.

Alternative methods that enforce the unilateral constraints on position level are proposed by Stewart and Trinkle [167] or Paoli and Schatzman [138, 139]. The former method enforces the unilateral constraints on position level and consequently does not suffer from penetration problems. However, it is not capable of including an elastic impact law, which is a typical drawback of physic simulators used in robotics and computer graphics [19, 59, 120]. Moreover, only faceted friction laws are incorporated, i.e., Coulomb's friction law is only included approximately. The method of Paoli and Schatzman [138, 139] enforces a possibly elastic impact law on position level over two consecutive time-steps. Although the authors in [138, 139] have proven that this is a valid approach for a single frictionless contact, the extension to multiple collisions leads to serious problems violating physics. Due to the often made rigid body assumptions in mechanical models, an

elastic impact law has to be enforced in order to accurately describe the observed physical phenomena. This brings us to the next objective.

Objective 2: *Elastic impacts* – To develop numerical time integration methods that simultaneously enforce unilateral constraints on both position and velocity level, thereby incorporating (partially) elastic impact laws.

Most time integration methods presented in literature assume that the underlying mechanical systems are modeled in such a way that the time derivative of the generalized coordinates coincides with the generalized velocities. Although this is the case for most problems from particle and continuum mechanics, typical flexible multibody dynamics formulations use unit quaternions for the description of finite rotations [56, 86, 151, 155]. In doing so, the governing equations are no longer second-order measure differential inclusions [22] but can be expressed as a system of first-order differential inclusions.

A variant of Moreau's mid-point rule that can deal with such a generalized kinematics and further enforces bilateral constraints on velocity level can be found in [69, 108, 133, 175, 176]. Since this method suffers from both unilateral and bilateral constraint drift it clearly does not respect **Objective 1** and won't be considered subsequently, besides the usage as reference method. Alternative methods that can cope with such an extended kinematics are the nonsmooth RATTLE algorithm [29] and the nonsmooth generalized-alpha method [41]. The latter method requires this kinematic relation to be formulated on acceleration level, involving the costly and cumbersome derivation and computation of additional derivatives. Moreover, the bilateral and unilateral constraint conditions are formulated on position, velocity and acceleration level, increasing the theoretical and numerical work even more dramatically. Since this thesis aims to develop highly generalized time integration methods for simulating nonsmooth mechanical systems with frictional contact and impacts, the next research objective will address generalized kinematic relations.

Objective 3: *Generalized kinematics* – To develop numerical time integration methods that cope with generalized kinematic differential equations and consistently discretize the resulting first-order measure differential inclusions.

The popular time integration method of Stewart and Trinkle [167] as well as different derivatives thereof, are commonly used in physic simulators.

In order to end up with complementarity problems, the key ingredient of their method is the polyhedral approximation of the desired Coulomb friction law. Depending on the number of introduced facets, this gives a better approximation to the isotropic Coulomb's friction law but at the same time increases the number of unknowns dramatically [118, 120]. Moreover, it introduces an approximation bias where frictional effects are stronger in some directions than others [120]. Consequently, depending on the resolution of the approximation, the method becomes increasingly unsuited for large scale problems and motivates the next research objective.

Objective 4: *Set-valued friction* – To develop numerical time integration methods that enforce set-valued spatial Coulomb friction laws without introducing a polyhedral approximation.

Although Moreau's mid-point rule has been continuously improved since its development, see e.g. [102, 103, 130], the resulting methods are at most convergent of order one, even on time intervals without collisions and constant contact conditions (excluding stick-slip transitions). Consequently, the total error decreases linearly with the applied step-size [5]. Using such first-order methods, the numerical solution of nonsmooth mechanical systems with frictional contact poses demanding restrictions on the step-size choice in order to solve the underlying equations with satisfactory accuracy. Up to now, there is no straightforward extension of higher-order time integration methods to nonsmooth mechanical systems. Nonetheless, a few attempts were already made. For frictionless contacts, applicability of extrapolation methods are discussed in [171], while in [2] higher-order event-capturing methods are designed by coupling implicit Runge–Kutta methods with Moreau's mid-point rule. A nonsmooth extension of the classical generalized-alpha method, a sophisticated algorithm for solving smooth structural dynamics problems, was presented in [44]. However, all these higher-order methods have in common that the unilateral constraints (including Newton-type impact laws) are formulated on velocity level only. Hence, they do not obey **Objective 1** and are therefore not considered further.

A remedy to the above described problem is given by the simultaneous enforcement of unilateral constraints at position and velocity levels as proposed within the nonsmooth generalized-alpha method [31]. More generally, the authors applied a stabilization of the constraint drift in the sense of Gear–Gupta–Leimkuhler (GGL) [68], see also [3, 159]. Based on these investigations, a variety of derivatives of the original nonsmooth generalized-alpha method [44] were developed, e.g. [32, 47]. Specifically,

in [41, 48, 65], these methods have been extended to cope with frictional contact and a more general kinematic relation between generalized positions and velocities. But theses methods either require the evaluation of the constraint accelerations of both unilateral and bilateral constraints [41], use a decoupled (simplified) solution of the resulting discrete equations [47], or enforce friction on position level [65], which is at least a debatable physical concept. The method of Capobianco et al. [41] requires an elaborated active-set strategy for the solution of the arising nonlinear system of equations. Moreover, cumbersome Jacobians of the constraints, as well as the constraint velocities and accelerations are required that at the same time increase the computational complexity significantly. The methods proposed in [48, 65] simplify the underlying mechanics, which results in first-order convergence during phases with persistent contact states [65]. Hence, these methods are not DAE-consistent, i.e., for persistent frictional contact, which can be described by differential algebraic equations (DAE's) with constant index, these methods do not collapse to existing higher-order DAE methods. As also concluded by the exhaustive review of [106], this is a desirable property of nonsmooth time integration methods and thus will be considered as the next research objective.

Objective 5: *Higher-order DAE-consistency* – To develop numerical time integration methods that collapse into existing higher-order methods for differential algebraic equations (DAE's) in scenarios with constant contact conditions that can alternatively be described by DAE's, without the need of acceleration level constraints.

As emphasized in Section 1.1, applications in structural mechanics, soft robotics and game/physics engines require stable and robust numerical time integration methods due to their reliance on stiff differential equations or stiff differential inclusions. Moreau's mid-point rule and variants thereof lead to serious step-size restrictions when applied to stiff problems. Similarly, the proposed nonsmooth RATTLE algorithm [29] and the family of nonsmooth Lobatto IIIA-IIIB methods [42] are not well-suited for such problems. In the context of structural dynamics, the methods of Newmark [132], Hilber, Hughes and Taylor [89], as well as the generalized-alpha method [45] are omnipresent due to the ability of introducing user-defined numerical dissipation. The resulting numerical methods are well-suited for solving stiff differential equations, but have two major disadvantages. Their extension to non-trivial kinematic differential equations is not straightforward and only very limited investigations are found in literature [33, 41]. Moreover, their application to differential algebraic equations requires

the choice of very specific initial conditions [16] to prevent the methods from introducing parasitic oscillations and order reduction problems. The problem is even more serious if variable step-size implementations should be considered. After each change of the step-size, the involved parameters have to be adapted in order to guarantee the method's second-order convergence [33, 101].

Typical applications in nonsmooth mechanics often assume rigid bodies. When numerous such bodies interact, it is common to encounter redundant contacts. That means, the contact forces are non-unique, due to the assumed rigidity. Similar problems occur in structural mechanics applications when rod finite elements are used. Therein, the cross-sections are typically considered to be rigid, which in turn leads to redundant contacts. Both above mentioned topics are summarized in the subsequent research objective.

Objective 6: *Stability and robustness* – To develop stable numerical time integration methods with optionally user-defined numerical dissipation for stiff mechanical systems involving redundant contacts.

As already mentioned in the previous section, sophisticated physic engines are mandatory ingredients for a variety of different applications and research fields. A high-level comparison between popular physic engines like Bullet [50], MuJoCo [174], ODE [164], DART [109] and PhysX [134] is given in [58]. The authors came to the results that *none of the engines being compared was uniformly better than all others*. Depending on the specific application *each engine performs best on the type of system it was designed and optimized for*. This showcases the aid of a more low-level comparison of the involved theories and methods. A comparison between possible contact and friction modeling techniques, which are the key ingredients of every physic simulator, is given in [92]. This formed the basis of the very recent investigation of LeLidec et al. [118]. Therein, the authors claim that *some approximations or algorithms commonly used in robotics can severely widen the reality gap*. Moreover, they *may induce nonphysical artifacts, thus widening the reality gap, leading to unrealistic behaviors when the simulator is later used for practical robotics applications*. The authors clearly stress the importance of a common basis of benchmark examples and low-level tests in order to guarantee that proposed physic simulators or game engines deliver physically correct results, which is summarized as the last objective of this thesis.

Objective 7: *Validation* – To provide a collection of selected benchmark examples for nonsmooth mechanical systems that every physics simulator or game engine should pass.

1.3 Outline and Main Contributions

The outline of this thesis is directly related to the previously stated **Research question**. That is, two families of nonsmooth Runge–Kutta methods are proposed, which extend classically used methods for the solution of possibly stiff differential algebraic equations to nonsmooth mechanical systems with frictional contact and impacts. The novel methods address all the stated research objectives, which are formulated as design rules that every non-smooth time integration method should adhere to.

Thereto, the second chapter introduces all the required mathematical preliminaries. This includes a short primer on convex cones given in Section 2.1, presenting different scaling properties of convex sets and the definition of the normal cone. Subsequently, some less-known scaling properties of normal cone inclusions are given, solely using the definition of the normal cone, making this investigation self-contained. In Section 2.3 the relation between normal cone inclusions, variational inequalities and (linear) complementarity problems is discussed. As it will turn out, this is a valuable tool for investigating the (unique) solvability of the discussed nonsmooth time integration methods. The chapter closes with miscellaneous topics from linear algebra like the Kronecker product and the Schur complement, which naturally appear during the description of the proposed families of nonsmooth Runge–Kutta methods.

To establish the notation, Chapter 3 gives a concise introduction to the theory of mechanical systems with frictional contact and impacts. This starts with a short discussion on the necessity of nontrivial kinematic relations between positions and velocities. Subsequently, Section 3.2 introduces the equality of measures that relates the change of velocities with the applied forces acting on the system. The subsequent two sections introduce bilateral constraints and unilateral contact, respectively. A variety of different set-valued friction laws are discussed in Section 3.5. Although different formulations are introduced, they can all be stated by nonlinear normal cone inclusions. Thereby all formulations respect **Objective 4** and do not introduce polyhedral approximations of the desired friction laws. Besides contact and friction laws, the mechanical systems under consideration have to cope with possibly elastic impact laws. This topic is

addressed in Section 3.6. The chapter closes by a short discussion about consistent initial conditions that are required for a correct initialization of the numerical time integration methods proposed later herein.

The subsequent Chapter 4 introduces the first family of nonsmooth Runge–Kutta method addressing **Objectives 1-6**. Specifically, the chapter develops nonsmooth projected stiffly accurate Runge–Kutta methods. This is based on an introductory section that discusses basic concepts of implicit Runge–Kutta methods for the solution of nonlinear ordinary differential equations. Already therein, the excellent stability properties of the proposed methods are discussed. This is followed by Section 4.2, which discusses the application of those methods to differential algebraic equations. At the end of the section, the well-known pendulum problem is investigated, which motivates the introduction of projected Runge–Kutta methods given subsequently in Section 4.3. Hereafter, particular representatives of this method are presented explicitly, including for instance the renowned family of Radau IIA methods as well as a new composite method with user-defined numerical dissipation, similar to the generalized-alpha method already discussed before.

Contribution 1: This thesis contributes a novel implicit Runge–Kutta method for differential algebraic equations, called TR-$\theta(\rho_\infty)$, with optionally user-defined numerical dissipation, similar to the popular generalized-alpha method from structural mechanics.

The section closes by showcasing the outstanding convergence results of the Radau IIA methods and demonstrates the importance of the introduced projection stage. In Section 4.5, the presented methods are naively extended to nonsmooth mechanical systems. As it turns out, all higher-order methods realize an arbitrary impact law, depending on the chosen initial conditions and the applied step-size, making the proposed methods flawed. As shown in the last Section 4.6, application of a nonsmooth projection stage entirely removes this issue and enables the realization of any desired, possibly elastic, impact law.

Contribution 2: It is the contribution of Chapter 4 that projected implicit Runge–Kutta methods are extended to nonsmooth mechanical systems with frictional contact and impacts. Specific representatives of this method address **Objectives 1-6**, making them an excellent choice, especially in the presence of demanding stability requirements.

Based on these findings, Chapter 5 introduces the family of nonsmooth partitioned Runge–Kutta methods. Those methods are classically applied to Hamiltonian systems and are well-known for there excellent long-term energy behavior. Consequently, those methods are by design unsuited for very stiff problems, disrespecting **Objective 6**. Apart from this minor drawback, these methods address **Objectives 1-5**, making them an excellent choice when only moderate stiffnesses are present. In analogy to the previous chapter, basic properties of partitioned Runge–Kutta methods applied to nonlinear ordinary differential equation are discussed in Section 5.1. This is followed by their extension to differential algebraic equations. Since this naturally involves some sort of projection stage, the subsequent Section 5.3 introduces nonsmooth partitioned Runge–Kutta methods that realize a desired, possibly elastic, impact law. The final section introduces prominent representatives of this family as the Lobatto IIIA-IIIB methods, together with a demonstration of their outstanding convergence properties, when applied to differential algebraic equations. Finally, it is shown that this family embeds a half-explicit method that can be interpreted as a variant of Moreau's celebrated mid-point rule.

Contribution 3: Introducing nonsmooth partitioned Runge–Kutta methods for mechanical systems with frictional contact and impacts is the specific contribution of Chapter 5. Different representatives of this method address **Objectives 1-5**. Furthermore, an embedded half-explicit first-order method can be identified as a variant of Moreau's mid-point rule.

During the development of Chapter 4 and 5 it will turn out that the time integration of nonsmooth mechanical systems using both proposed families of nonsmooth Runge–Kutta methods requires the solution of a nonlinear system of equations that is subjected to a set of nonlinear normal cone inclusions. In order to obtain an implementable numerical time integration method, solution algorithms for this kind of problems have to be developed. This is exactly the aim of Chapter 6. Thereto, Section 6.1 shows how normal cone inclusions can equivalently be stated as solutions of implicit equations. This gives rise to Newton-type methods as discussed subsequently in Section 6.2. Due to the often applied modeling assumption of rigid bodies (or rigid cross-sections in structural mechanics), redundant contacts are frequently encountered in nonsmooth mechanical systems. These lead to rank deficient Jacobian matrices, which excludes the usage of Jacobian-based methods discussed so far. In order to solve such scenarios, Section 6.3 introduces the iterative projection method based on simple fixed-

point iterations. The reformulation of normal cone inclusions as implicit equations from Section 6.1 introduces an arbitrary positive parameter. The specific choice of this parameter plays an important role for the convergence of the proposed solution algorithms. Thereto, Section 6.4 concludes this chapter by a simple but sufficient automatic parameter choice.

Contribution 4: Chapter 6 proposes two different solution algorithms for nonlinear normal cone inclusion problems. The second one overcomes problems of rank deficient Jacobians, making it suitable for problems with redundant contacts, thereby addressing **Objective 6**.

Finally, Chapter 7 applies the proposed nonsmooth Runge–Kutta methods to a variety of selected benchmark examples for nonsmooth mechanical systems with frictional contact and impacts. Each example therein demonstrates a specific physical problem or numerical challenge that the proposed numerical methods should be able to cope with, thereby directly reflecting a specific objective of this thesis. As it will turn out, the introduced methods pass all the proposed benchmark examples with ease and even outplay existing methods. Furthermore, for persistent frictional contact, which can alternatively be described by differential algebraic equations (DAE's), the proposed nonsmooth Runge–Kutta methods collapse to a higher-order DAE method, retaining their outstanding orders of convergence and at the same time addressing **Objective 5**. Consequently, both families of nonsmooth Runge–Kutta methods address all[1] the stated research objectives of this thesis.

Contribution 5: An exclusive collection of benchmark examples for nonsmooth time integration methods is proposed in Chapter 7. This collection forms a baseline for the validation and performance comparison of different numerical methods as intended by **Objective 7**.

Concluding remarks are given in Chapter 8, which evaluate the proposed numerical methods depending on the specific application. A comprehensive flowchart gives a systematic approach to choose the most suitable method for a given application. Finally, the thesis closes by giving an outlook on future research in the field of nonsmooth time integration methods.

[1] The nonsmooth partitioned Runge–Kutta methods are only suitable when moderate stiffnesses are encountered, thereby partly disrespecting **Objective 6**.

Chapter 2

Mathematical Preliminaries

Throughout this monograph, $\mathbb{R}$ denotes the real numbers, and $\mathbb{R}^n$ is the usual vector space of real n-tuples $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$. Since $\mathbb{R}^n \cong \mathbb{R}^{n\times 1}$, it is convenient to embed real n-tuples into matrix calculus by denoting them as *column-tuples* $\boldsymbol{x} \in \mathbb{R}^{n\times 1} \cong \mathbb{R}^n$. Let $(\bullet)^{\mathrm{T}}$ denote the matrix transpose, the tuple $\boldsymbol{x}^{\mathrm{T}} \in \mathbb{R}^{1\times n}$ is called a *row-tuple*. Hence, the inner product between two tuples $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ can be written as $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y} \in \mathbb{R}$. For $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{y} \in \mathbb{R}^m$, the outer product $\boldsymbol{x}\boldsymbol{y}^{\mathrm{T}} \in \mathbb{R}^{n\times m}$ gives an $n \times m$ matrix. Let $\mathbf{1}_{n\times n}$ and $\mathbf{0}_{n\times n}$ respectively denote the $n \times n$ identity matrix and the $n \times n$ zero matrix. Moreover, $\mathbf{1}_n = (1, \ldots, 1) \in \mathbb{R}^{n\times 1}$ denotes the column-tuple containing only ones and $\mathbf{0}_n = (0, \ldots, 0) \in \mathbb{R}^{n\times 1}$ is the zero column-tuple. For any regular matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, the inverse matrix is denoted $\boldsymbol{A}^{-1} \in \mathbb{R}^{n\times n}$. The p-norm is defined by $\|\boldsymbol{x}\|_p \coloneqq (\sum_{i=1}^{n} |x_i|^p)^{1/p}$ and includes the 1-norm $\|\boldsymbol{x}\|_1 = |x_1| + |x_2| + \cdots + |x_n|$, the Euclidean norm $\|\boldsymbol{x}\|_2 = (x_1^2 + x_2^2 + \cdots + x_n^2)^{1/2}$ that is subsequently abbreviated as $\|\boldsymbol{x}\|$ and the uniform norm $\|\boldsymbol{x}\|_\infty = \max\{|x_1|, |x_2|, \ldots, |x_n|\}$.

Besides simple $\mathbb{R}^n$-tuples, the description of particles and their motion requires the introduction of the Euclidean 3-space, denoted by $\mathbb{E}^3$, as an abstract three-dimensional real inner-product space [12]. For two given vectors $\vec{\boldsymbol{a}}, \vec{\boldsymbol{b}} \in \mathbb{E}^3$, the inner product $\vec{\boldsymbol{a}} \cdot \vec{\boldsymbol{b}} \in \mathbb{R}$ gives a real number and the cross product $\vec{\boldsymbol{a}} \times \vec{\boldsymbol{b}} \in \mathbb{E}^3$ computes another element of $\mathbb{E}^3$. A basis for $\mathbb{E}^3$ is a linearly independent set of three vectors $\vec{\boldsymbol{e}}_x, \vec{\boldsymbol{e}}_y, \vec{\boldsymbol{e}}_z \in \mathbb{E}^3$. The basis is said to be right-handed if $\vec{\boldsymbol{e}}_x \cdot (\vec{\boldsymbol{e}}_y \times \vec{\boldsymbol{e}}_z) > 0$, i.e., their triple product is positive. If their base vectors are mutually orthogonal and have unit length, the basis is called *orthonormal*. In this monograph, only right-handed orthonormal bases are considered. For a given basis $K = (\vec{\boldsymbol{e}}_x^K, \vec{\boldsymbol{e}}_y^K, \vec{\boldsymbol{e}}_z^K)$, the respective components a_i^K, $i \in \{x, y, z\}$ of a vector $\vec{\boldsymbol{a}} = a_x^K \vec{\boldsymbol{e}}_x^K + a_y^K \vec{\boldsymbol{e}}_y^K + a_z^K \vec{\boldsymbol{e}}_z^K \in \mathbb{E}^3$

can be collected in the 3-tuple ${}_K\vec{\boldsymbol{a}} = (a_x^K, a_y^K, a_z^K) \in \mathbb{R}^3$. Thus, a coordinate representation ${}_K\vec{\boldsymbol{a}} \in \mathbb{R}^3$ of the vector $\vec{\boldsymbol{a}} \in \mathbb{E}^3$ relates the Euclidean 3-space with the real 3-tuples. Moreover, it is convenient to introduce the fixed inertial basis $I = (\vec{\boldsymbol{e}}_x^I, \vec{\boldsymbol{e}}_y^I, \vec{\boldsymbol{e}}_z^I)$, together with the origin O.

2.1 Convex Cones

A point $\boldsymbol{p} \in \mathbb{R}^n$ is a *limit point* of the set $C \subseteq \mathbb{R}^n$ if there exists a convergent sequence $\{\boldsymbol{x}_k\}$ with $\lim_{k\to\infty} \boldsymbol{x}_k = \boldsymbol{p}$ and $\boldsymbol{x}_k \in C$ for all $k \in \mathbb{N}$. A set $C \subseteq \mathbb{R}^n$ is *closed* when it contains all its limit points $\boldsymbol{p}$. For $\alpha \in \mathbb{R}$, the scalar multiple of a set $C \subseteq \mathbb{R}^n$ is introduced as

$$\alpha C \coloneqq \{\alpha \boldsymbol{x} \,|\, \boldsymbol{x} \in C\}\,. \tag{2.1}$$

A set $C \subseteq \mathbb{R}^n$ is said to be *convex* if

$$(1-\alpha)\boldsymbol{x} + \alpha \boldsymbol{y} \in C \tag{2.2}$$

for all $\boldsymbol{x}, \boldsymbol{y} \in C$ and $0 \leq \alpha \leq 1$. Hence, for two arbitrary points of the set, their interconnecting line is contained in the set. A set $K \subseteq \mathbb{R}^n$ is a *cone*, if for any $\boldsymbol{x} \in K$ also

$$\alpha \boldsymbol{x} \in K \quad \forall \alpha > 0\,. \tag{2.3}$$

If condition (2.3) is also met for $\alpha = 0$, the cone contains the origin. Prominent examples for closed convex cones are the non-negative and non-positive orthants, $\mathbb{R}_0^{n+} = \{\boldsymbol{x} \in \mathbb{R}^n \,|\, x_i \geq 0,\, i = 1, \ldots, n\}$ and $\mathbb{R}_0^{n-} = \{\boldsymbol{x} \in \mathbb{R}^n \,|\, x_i \leq 0,\, i = 1, \ldots, n\}$, respectively. The unit ball $B \coloneqq \{\boldsymbol{x} \in \mathbb{R}^n \,|\, \|\boldsymbol{x}\| \leq 1\}$ is a convex set but not a cone.

Given any matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, the linear mapping $A\colon \mathbb{R}^n \to \mathbb{R}^m$, $\boldsymbol{x} \mapsto \boldsymbol{A}\boldsymbol{x}$, defines the image of $C \subseteq \mathbb{R}^n$ under A as

$$AC \coloneqq \{\boldsymbol{A}\boldsymbol{x} \in \mathbb{R}^m \,|\, \boldsymbol{x} \in C \subseteq \mathbb{R}^n\} \tag{2.4}$$

and contains all tuples $\boldsymbol{A}\boldsymbol{x}$ with $\boldsymbol{x} \in C$. Similarly, the inverse image[1] of D under A is defined as

$$A^{-1}D \coloneqq \{\boldsymbol{x} \in \mathbb{R}^n \,|\, \boldsymbol{A}\boldsymbol{x} \in D \subseteq \mathbb{R}^m\} \tag{2.5}$$

and contains all the tuples $\boldsymbol{x}$ that are mapped by A to the set D. Moreover, it is convenient to define

$$A^{\mathrm{T}}D \coloneqq \{\boldsymbol{A}^{\mathrm{T}}\boldsymbol{y} \in \mathbb{R}^n \,|\, \boldsymbol{y} \in D \subseteq \mathbb{R}^m\} \tag{2.6}$$

[1]The notation $A^{-1}D$ here is not meant to imply that the inverse linear transformation, induced by $\boldsymbol{A}^{-1}$, exists.

and

$$A^{-\mathrm{T}}C \coloneqq \{\boldsymbol{y} \in \mathbb{R}^m \mid \boldsymbol{A}^{\mathrm{T}}\boldsymbol{y} \in C \subseteq \mathbb{R}^n\}\,. \tag{2.7}$$

Proposition 2.1 (Rockafellar [149], Theorem 3.4)**.** *Let $A\colon \mathbb{R}^n \to \mathbb{R}^m$ be a linear transformation. Then AC is a convex set in $\mathbb{R}^m$ for every convex set $C \subseteq \mathbb{R}^n$, and $A^{-1}D$ is a convex set in $\mathbb{R}^n$ for every convex set $D \subseteq \mathbb{R}^m$.*

Proof. Let $\boldsymbol{x}, \boldsymbol{y} \in C$ and $0 \leq \alpha \leq 1$. By convexity of C, it holds that $(1-\alpha)\boldsymbol{x} + \alpha\boldsymbol{y} \in C$. Using (2.4), it follows $\boldsymbol{A}[(1-\alpha)\boldsymbol{x} + \alpha\boldsymbol{y}] \in AC$. Therefore,

$$\boldsymbol{A}[(1-\alpha)\boldsymbol{x} + \alpha\boldsymbol{y}] = (1-\alpha)\boldsymbol{A}\boldsymbol{x} + \alpha\boldsymbol{A}\boldsymbol{y} \in AC\,, \tag{2.8}$$

with $\boldsymbol{A}\boldsymbol{x}, \boldsymbol{A}\boldsymbol{y} \in AC$. Consequently, AC is a convex set.

Let $\boldsymbol{A}\boldsymbol{x}, \boldsymbol{A}\boldsymbol{y} \in D$. By convexity of D it follows

$$(1-\alpha)\boldsymbol{A}\boldsymbol{x} + \alpha\boldsymbol{A}\boldsymbol{y} = \boldsymbol{A}[(1-\alpha)\boldsymbol{x} + \alpha\boldsymbol{y}] \in D\,. \tag{2.9}$$

By Definition 2.5, $\boldsymbol{x}, \boldsymbol{y} \in A^{-1}D$ and $(1-\alpha)\boldsymbol{x} + \alpha\boldsymbol{y} \in A^{-1}D$. Consequently, $A^{-1}D$ is convex. □

Definition 2.2. Let $C \subseteq \mathbb{R}^n$ be a closed, non-empty and convex set. The normal cone of C at $\boldsymbol{x} \in C$ is defined as

$$\mathcal{N}_C(\boldsymbol{x}) \coloneqq \{\boldsymbol{y} \in \mathbb{R}^n \mid \boldsymbol{y}^{\mathrm{T}}(\boldsymbol{x}^* - \boldsymbol{x}) \leq 0\,, \quad \forall \boldsymbol{x}^* \in C\}\,, \tag{2.10}$$

whereas $\mathcal{N}_C(\boldsymbol{x}) = \emptyset$ if $\boldsymbol{x} \notin C$ and $\emptyset$ denotes the empty set.

As the name indicates, the normal cone $\mathcal{N}_C(\boldsymbol{x})$ is a cone, i.e., for $\boldsymbol{y} \in \mathcal{N}_C(\boldsymbol{x})$ also $\alpha\boldsymbol{y} \in \mathcal{N}_C(\boldsymbol{x})$ for all $\alpha > 0$. Moreover, it can be shown that the normal cone $\mathcal{N}_C(\boldsymbol{x})$ is closed and convex, see Proposition 6.5 in [150]. Furthermore, it is easily confirmed that $\mathcal{N}_C(\boldsymbol{x}) = \{\boldsymbol{0}_n\}$ if $\boldsymbol{x}$ is an interior point of the set C. This is because, for such interior points, there exists a ball centered at $\boldsymbol{x}$ contained in $C \subseteq \mathbb{R}^n$. Consequently, selecting $\boldsymbol{x}^*$ from this ball enables the production of any direction $\boldsymbol{x}^* - \boldsymbol{x}$ in equation (2.10), leaving only $\boldsymbol{y} = \boldsymbol{0}_n$ to satisfy the inequality. A similar reasoning is found in Section 1.1 of [62].

2.2 Properties of Normal Cone Inclusions

The normal cone possesses several lesser-known scaling properties, which are presented here. These properties can be effectively demonstrated using concepts from convex analysis, such as the subdifferential and the support

function. However, to avoid introducing additional theoretical concepts, these properties are established solely using the definition of the normal cone provided in equation (2.10).

Proposition 2.3. *Consider a closed, non-empty and convex set $C \subseteq \mathbb{R}^n$, a fixed $\boldsymbol{y} \in \mathbb{R}^n$ that does not depend on t, define $C(\alpha) \coloneqq \alpha C$ and let $\mathcal{I} \subseteq \mathbb{R}$. If*

$$\boldsymbol{y} \in \mathcal{N}_{C(\alpha(t))}\big(\boldsymbol{x}(t)\big) \quad \forall t \in \mathcal{I} \subseteq \mathbb{R}\,, \tag{2.11}$$

where $\alpha(t) \geq 0\ \forall t \in \mathcal{I}$, then it holds that

$$\boldsymbol{y} \in \mathcal{N}_{C\left(\int_{\mathcal{I}} \alpha(t)\,\mathrm{d}t\right)}\left(\int_{\mathcal{I}} \boldsymbol{x}(t)\,\mathrm{d}t\right)\,. \tag{2.12}$$

Proof.[2] By definition of the normal cone, given in (2.10), it holds that $\forall t \in \mathcal{I}$ equation (2.11) can be written as

$$\begin{aligned} \boldsymbol{y}^{\mathrm{T}}\big(\boldsymbol{x}^* - \boldsymbol{x}(t)\big) &\leq 0 \quad \forall \boldsymbol{x}^* \in C(\alpha(t)) \\ &\Updownarrow \\ \boldsymbol{y}^{\mathrm{T}}\big(\alpha(t)\bar{\boldsymbol{x}}^* - \boldsymbol{x}(t)\big) &\leq 0 \quad \forall \bar{\boldsymbol{x}}^* \in C\,. \end{aligned} \tag{2.13}$$

Integration of the latter inequality gives

$$\int_{\mathcal{I}} \boldsymbol{y}^{\mathrm{T}}\big(\alpha(t)\bar{\boldsymbol{x}}^* - \boldsymbol{x}(t)\big)\mathrm{d}t = \boldsymbol{y}^{\mathrm{T}}\left[\left(\int_{\mathcal{I}} \alpha(t)\,\mathrm{d}t\right)\bar{\boldsymbol{x}}^* - \int_{\mathcal{I}} \boldsymbol{x}(t)\,\mathrm{d}t\right] \leq 0 \tag{2.14}$$

for all $\bar{\boldsymbol{x}}^* \in C$. Alternatively, this can be stated as

$$\boldsymbol{y}^{\mathrm{T}}\left(\boldsymbol{x}^* - \int_{\mathcal{I}} \boldsymbol{x}(t)\,\mathrm{d}t\right) \leq 0 \quad \forall \boldsymbol{x}^* \in C\left(\int_{\mathcal{I}} \alpha(t)\,\mathrm{d}t\right)\,. \tag{2.15}$$

Using the definition of the normal cone once more, this leads to equation (2.12). □

Proposition 2.4. *For a closed, non-empty and convex set $C \subseteq \mathbb{R}^n$, a regular matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ and the mapping $A\colon \mathbb{R}^n \to \mathbb{R}^n$, $\boldsymbol{x} \mapsto \boldsymbol{A}\boldsymbol{x}$ it holds that*

$$\boldsymbol{y} \in \mathcal{N}_C(\boldsymbol{x}) \quad \Longleftrightarrow \quad \boldsymbol{A}^{\mathrm{T}}\boldsymbol{y} \in \mathcal{N}_{A^{-1}C}(\boldsymbol{A}^{-1}\boldsymbol{x})\,. \tag{2.16}$$

[2] An alternative proof is given in Proposition 2 of Capobianco et al. [41].

Proof. Let $\boldsymbol{z} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{y}$. By injecting $\boldsymbol{A}\boldsymbol{A}^{-1} = \mathbf{1}_{n\times n}$ into the definition of the normal cone (2.10) it can be seen that

$$\begin{aligned}\mathcal{N}_C(\boldsymbol{x}) &= \{\boldsymbol{y} \in \mathbb{R}^n \,|\, \boldsymbol{y}^{\mathrm{T}}(\boldsymbol{x}^* - \boldsymbol{x}) \leq 0\,, \quad \forall \boldsymbol{x}^* \in C\} \\ &= \{\boldsymbol{y} \in \mathbb{R}^n \,|\, \boldsymbol{y}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{A}^{-1}(\boldsymbol{x}^* - \boldsymbol{x}) \leq 0\,, \quad \forall \boldsymbol{x}^* \in C\} \\ &= \{\boldsymbol{A}^{-\mathrm{T}}\boldsymbol{z} \in \mathbb{R}^n \,|\, \boldsymbol{z}^{\mathrm{T}}(\boldsymbol{A}^{-1}\boldsymbol{x}^* - \boldsymbol{A}^{-1}\boldsymbol{x}) \leq 0\,, \quad \forall \boldsymbol{x}^* \in C\}\,.\end{aligned} \tag{2.17}$$

With $\boldsymbol{x}^* = \boldsymbol{A}\boldsymbol{w}^*$ and by definition (2.5), i.e., $\boldsymbol{A}\boldsymbol{w}^* \in C \Leftrightarrow \boldsymbol{w}^* \in A^{-1}C$ it follows

$$\begin{aligned}\mathcal{N}_C(\boldsymbol{x}) &= \{\boldsymbol{A}^{-\mathrm{T}}\boldsymbol{z} \in \mathbb{R}^n \,|\, \boldsymbol{z}^{\mathrm{T}}(\boldsymbol{w}^* - \boldsymbol{A}^{-1}\boldsymbol{x}) \leq 0\,, \quad \forall \boldsymbol{w}^* \in A^{-1}C\} \\ &= \{\boldsymbol{A}^{-\mathrm{T}}\boldsymbol{z} \in \mathbb{R}^n \,|\, \boldsymbol{z} \in \mathcal{N}_{A^{-1}C}(\boldsymbol{A}^{-1}\boldsymbol{x})\}\,.\end{aligned} \tag{2.18}$$

Since $\boldsymbol{z} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{y}$, this implies

$$\boldsymbol{y} \in \mathcal{N}_C(\boldsymbol{x}) \quad \Longleftrightarrow \quad \boldsymbol{A}^{\mathrm{T}}\boldsymbol{y} \in \mathcal{N}_{A^{-1}C}(\boldsymbol{A}^{-1}\boldsymbol{x})\,. \tag{2.19}$$

□

2.3 Normal Cone Inclusions, Variational Inequalities and Complementarity Problems

In this section, the relation between normal cone inclusions, variational inequalities and (linear) complementarity problems is discussed. As it turns out, their relationship is essential for discussing solvability and analyzing specific problems. Depending on the chosen formulation, tailored solution algorithms exist. Subsequently, only the solution of linear complementarity problems is investigated, while the solution of normal cone inclusion problems is addressed in Chapter 6.

Definition 2.5. For a given closed, non-empty and convex set $C \subseteq \mathbb{R}^n$ and a mapping $\boldsymbol{f}\colon C \to \mathbb{R}^n$, the *normal cone inclusion problem*, denoted NIP $(C, \boldsymbol{f})$, is to find a tuple $\boldsymbol{x} \in C$ that satisfies

$$-\boldsymbol{f}(\boldsymbol{x}) \in \mathcal{N}_C(\boldsymbol{x})\,. \tag{2.20}$$

Definition 2.6. For a given closed, non-empty and convex set $C \subseteq \mathbb{R}^n$ and a mapping $\boldsymbol{f}\colon C \to \mathbb{R}^n$, the *variational inequality*, denoted VI $(C, \boldsymbol{f})$, is to find a tuple $\boldsymbol{x} \in C$ such that

$$(\boldsymbol{x}^* - \boldsymbol{x})^{\mathrm{T}}\boldsymbol{f}(\boldsymbol{x}) \geq 0 \quad \forall \boldsymbol{x}^* \in C\,. \tag{2.21}$$

Clearly, the inequality (2.21) together with the definition of the normal cone (2.10) reveals that solutions $\boldsymbol{x} \in C$ of the VI $(C, \boldsymbol{f})$ solve the NIP $(C, \boldsymbol{f})$ and vise versa. This is summarized in the following corollary.

Corollary 2.7. *For a given closed, non-empty and convex set $C \subseteq \mathbb{R}^n$ and a mapping $\boldsymbol{f}\colon C \to \mathbb{R}^n$, a tuple $\boldsymbol{x} \in C$ solves the VI $(C, \boldsymbol{f})$ if and only if it solves the NIP $(C, \boldsymbol{f})$, since*

$$\begin{aligned} -\boldsymbol{f}(\boldsymbol{x}) \in \mathcal{N}_C(\boldsymbol{x}) &= \{\boldsymbol{y} \in \mathbb{R}^n \mid \boldsymbol{y}^{\mathrm{T}}(\boldsymbol{x}^* - \boldsymbol{x}) \leq 0,\, \forall \boldsymbol{x}^* \in C\} \\ &\Updownarrow \\ \boldsymbol{f}(\boldsymbol{x})^{\mathrm{T}}(\boldsymbol{x}^* - \boldsymbol{x}) = (\boldsymbol{x}^* - \boldsymbol{x})^{\mathrm{T}}\boldsymbol{f}(\boldsymbol{x}) &\geq 0 \quad \forall \boldsymbol{x}^* \in C\,. \end{aligned} \tag{2.22}$$

Proposition 2.8. *Let $g\colon \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable and $C \subseteq \mathbb{R}^n$ be closed, non-empty and convex. Assume $\boldsymbol{x}$ is a solution of the optimization problem*

$$\min_{\boldsymbol{x}^* \in C} g(\boldsymbol{x}^*)\,. \tag{2.23}$$

Then, $\boldsymbol{x}$ is a solution of VI $(C, \nabla g)$*, where ∇g denotes the gradient of g.*

Proof. Let $\phi(\alpha) = g\big(\boldsymbol{x} + \alpha(\boldsymbol{x}^* - \boldsymbol{x})\big)$, for $\alpha \in [0,1]$ and $\boldsymbol{x}^* \in C$. Since $\boldsymbol{x} \in C$ solves the minimization problem (2.23), $\phi(\alpha)$ achieves its minimum at $\alpha = 0$ and it follows $0 \leq \phi'(0) = \nabla g(\boldsymbol{x})^{\mathrm{T}}(\boldsymbol{x}^* - \boldsymbol{x})$ for all $\boldsymbol{x}^* \in C$, that is, $\boldsymbol{x}$ is a solution of VI $(C, \nabla g)$. □

Definition 2.9. A function $g\colon \mathbb{R}^n \to \mathbb{R}$ is said to be convex if

$$g\big((1-\alpha)\boldsymbol{x} + \alpha\boldsymbol{y}\big) \leq (1-\alpha)g(\boldsymbol{x}) + \alpha g(\boldsymbol{y}) \tag{2.24}$$

for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ and $\alpha \in [0,1]$.

Proposition 2.10. *If $g\colon \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and convex, then it holds that*

$$g(\boldsymbol{x}^*) \geq g(\boldsymbol{x}) + \nabla g(\boldsymbol{x})^{\mathrm{T}}(\boldsymbol{x}^* - \boldsymbol{x}) \quad \forall \boldsymbol{x}^*, \boldsymbol{x} \in \mathbb{R}^n\,. \tag{2.25}$$

Proof. Let $\boldsymbol{x}, \boldsymbol{x}^* \in \mathbb{R}^n$ and define

$$h(\alpha) = g\big((1-\alpha)\boldsymbol{x}^* + \alpha\boldsymbol{x}\big) - \big((1-\alpha)g(\boldsymbol{x}^*) + \alpha g(\boldsymbol{x})\big)\,, \tag{2.26}$$

i.e., the error of the inequality (2.24). Consequently, $h(\alpha) \leq 0$ for all $\alpha \in [0,1]$ and $h(0) = h(1) = 0$. Clearly, $h'(0) \leq 0$ and $h'(1) \geq 0$. Straightforward evaluation of the second inequality shows

$$h'(1) = \nabla g(\boldsymbol{x})^{\mathrm{T}}(\boldsymbol{x} - \boldsymbol{x}^*) + g(\boldsymbol{x}^*) - g(\boldsymbol{x}) \geq 0\,, \tag{2.27}$$

which concludes the proof. □

Proposition 2.11. *If g is a continuously differentiable convex function and $\boldsymbol{x}$ is a solution to VI $(C, \nabla g)$, then $\boldsymbol{x}$ is a solution to the optimization problem* (2.23).

Proof. Since $\boldsymbol{x}$ is a solution of VI $(C, \nabla g)$, it follows from Definition 2.6 that $\nabla g(\boldsymbol{x})^{\mathrm{T}}(\boldsymbol{x}^* - \boldsymbol{x}) \geq 0$ for all $\boldsymbol{x}^* \in C$. And since g is assumed to be convex, Proposition 2.10 implies

$$g(\boldsymbol{x}^*) - g(\boldsymbol{x}) \geq \nabla g(\boldsymbol{x})^{\mathrm{T}}(\boldsymbol{x}^* - \boldsymbol{x}) \geq 0 \quad \forall \boldsymbol{x}^* \in C\,. \tag{2.28}$$

Consequently, $\boldsymbol{x}$ minimizes (2.23) since $g(\boldsymbol{x}^*) \geq g(\boldsymbol{x})$ for all $\boldsymbol{x}^* \in C$. □

Definition 2.12. Let $C \subseteq \mathbb{R}^n$ be closed, non-empty and convex. There exists a unique tuple $\boldsymbol{x} \in C$ that is closest to $\boldsymbol{z} \in \mathbb{R}^n$ in the Euclidean norm, see Theorem 1.5.5 of [62]. This closest tuple $\boldsymbol{x}$ is called the (Euclidean) projection of $\boldsymbol{z}$ onto C and denoted $\operatorname{prox}_C(\boldsymbol{z})$. The mapping $\operatorname{prox}_C \colon \mathbb{R}^n \to C, \boldsymbol{z} \mapsto \operatorname{prox}_C(\boldsymbol{z})$ is called the *Euclidean projector* onto C. By definition, $\boldsymbol{x} = \operatorname{prox}_C(\boldsymbol{z})$ is the unique solution of the convex minimization problem

$$\boldsymbol{x} = \operatorname{prox}_C(\boldsymbol{z}) = \operatorname*{arg\,min}_{\boldsymbol{x}^* \in C} g(\boldsymbol{x}^*) \quad \text{with} \quad g(\boldsymbol{x}^*) = \frac{1}{2}(\boldsymbol{x}^* - \boldsymbol{z})^{\mathrm{T}}(\boldsymbol{x}^* - \boldsymbol{z})\,. \tag{2.29}$$

Definition 2.13. Let $\boldsymbol{f} \colon C \to \mathbb{R}^n$ and $C \subseteq \mathbb{R}^n$ be closed, non-empty and convex. The mapping $\boldsymbol{F}_C^{\mathrm{nat}} \colon \mathbb{R}^n \to \mathbb{R}^n$ is called the *natural map*[3] and is defined as

$$\boldsymbol{F}_C^{\mathrm{nat}}(\boldsymbol{x}) = \boldsymbol{x} - \operatorname{prox}_C\big(\boldsymbol{x} - \boldsymbol{f}(\boldsymbol{x})\big)\,. \tag{2.30}$$

Proposition 2.14 (Facchinei [62], Proposition 1.5.8)**.** *Let $\boldsymbol{f} \colon C \to \mathbb{R}^n$ and $C \subseteq \mathbb{R}^n$ be closed, non-empty and convex. A tuple $\boldsymbol{x}$ solves the normal cone inclusion problem* (2.20) *if and only if the natural map* (2.30) *vanishes, i.e.,*

$$-\boldsymbol{f}(\boldsymbol{x}) \in \mathcal{N}_C(\boldsymbol{x}) \quad \Longleftrightarrow \quad \boldsymbol{F}_C^{\mathrm{nat}}(\boldsymbol{x}) = \boldsymbol{x} - \operatorname{prox}_C\big(\boldsymbol{x} - \boldsymbol{f}(\boldsymbol{x})\big) = \boldsymbol{0}_n\,. \tag{2.31}$$

Proof. Let $\boldsymbol{z} = \boldsymbol{x} - \boldsymbol{f}(\boldsymbol{x})$ and assume $\boldsymbol{F}_C^{\mathrm{nat}}(\boldsymbol{x}) = \boldsymbol{0}_n$. By Definition 2.12 this can equivalently be written as

$$\boldsymbol{x} = \operatorname{prox}_C(\boldsymbol{z}) = \operatorname*{arg\,min}_{\boldsymbol{x}^* \in C} g(\boldsymbol{x}^*)\,. \tag{2.32}$$

[3] Surprisingly, it is rather unclear who coined the name natural map. According to Facchinei [62], the first occurrence of this name is due to Mangasarian [122], which called (2.30) the *natural residual* or *gradient projection map* for the case of $C = \mathbb{R}^{n+}_0$.

Since g is convex with $\nabla g(\boldsymbol{x}) = \boldsymbol{x} - \boldsymbol{z}$, Proposition 2.8 shows that (2.32) implies VI $(C, \nabla g)$, i.e.,

$$(\boldsymbol{x}^* - \boldsymbol{x})^{\mathrm{T}} \nabla g(\boldsymbol{x}) = (\boldsymbol{x}^* - \boldsymbol{x})^{\mathrm{T}} (\boldsymbol{x} - \boldsymbol{z}) \geq 0 \quad \forall \boldsymbol{x}^* \in C\,. \tag{2.33}$$

Finally, substitution of $\boldsymbol{z} = \boldsymbol{x} - \boldsymbol{f}(\boldsymbol{x})$ shows that (2.33) is equivalent to

$$(\boldsymbol{x}^* - \boldsymbol{x})^{\mathrm{T}} \big(\boldsymbol{x} - (\boldsymbol{x} - \boldsymbol{f}(\boldsymbol{x}))\big) = (\boldsymbol{x}^* - \boldsymbol{x})^{\mathrm{T}} \boldsymbol{f}(\boldsymbol{x}) \geq 0 \quad \forall \boldsymbol{x}^* \in C\,, \tag{2.34}$$

which is exactly the definition of VI $(C, \boldsymbol{f})$ and due to Corollary 2.7 is equivalent to NIP $(C, \boldsymbol{f})$. Using Proposition 2.11 instead of 2.8 shows the reverse direction and concludes the proof. □

Since $\mathcal{N}_C(\boldsymbol{x})$ is a cone, this further implies $-\rho \boldsymbol{f}(\boldsymbol{x}) \in \mathcal{N}_C(\boldsymbol{x}) \Leftrightarrow \boldsymbol{F}_C^{\mathrm{nat}}(\boldsymbol{x}) = \boldsymbol{x} - \mathrm{prox}_C\big(\boldsymbol{x} - \rho \boldsymbol{f}(\boldsymbol{x})\big) = \boldsymbol{0}_n$ for arbitrary $\rho > 0$. Together with the cone property (2.3) this yields $-\boldsymbol{f}(\boldsymbol{x}) \in \mathcal{N}_C(\boldsymbol{x}) \Leftrightarrow \boldsymbol{F}_C^{\mathrm{nat}} = \boldsymbol{x} - \mathrm{prox}_C\big(\boldsymbol{x} - \rho \boldsymbol{f}(\boldsymbol{x})\big) = \boldsymbol{0}_n$. This reformulation is of crucial importance, since the auxiliary parameter $\rho > 0$ can be utilized to manipulate the convergence of numerical solution algorithms proposed in Chapter 6.

Definition 2.15. For a given closed and convex cone $K \subseteq \mathbb{R}^n$ and a mapping $\boldsymbol{f}\colon K \to \mathbb{R}^n$, the *complementarity problem*, denoted CP $(K, \boldsymbol{f})$, is to find a tuple $\boldsymbol{x} \in \mathbb{R}^n$ that satisfies

$$\boldsymbol{x} \in K\,, \quad \boldsymbol{f}(\boldsymbol{x}) \in K^* \quad \text{and} \quad \boldsymbol{x}^{\mathrm{T}} \boldsymbol{f}(\boldsymbol{x}) = 0\,. \tag{2.35}$$

Therein, $K^* = \{\boldsymbol{d} \in \mathbb{R}^n \,|\, \boldsymbol{v}^{\mathrm{T}} \boldsymbol{d} \geq 0\,,\ \forall \boldsymbol{v} \in K\}$ denotes the dual cone of K.

Proposition 2.16 (Facchinei [62], Proposition 1.1.3)**.** *Let K be a closed cone in $\mathbb{R}^n$. A tuple $\boldsymbol{x}$ solves the variational inequality VI $(K, \boldsymbol{f})$ if and only if $\boldsymbol{x}$ solves the complementarity problem CP $(K, \boldsymbol{f})$.*

Proof. Assume $\boldsymbol{x}$ solves the VI $(K, \boldsymbol{f})$. Consequently, $\boldsymbol{x} \in K$. Since the cone is assumed to be closed, it contains the origin $\boldsymbol{0}_n$. Choosing $\boldsymbol{x}^* = \boldsymbol{0}_n$ in (2.21) yields

$$\boldsymbol{x}^{\mathrm{T}} \boldsymbol{f}(\boldsymbol{x}) \leq 0\,. \tag{2.36}$$

Moreover, since K is a cone, also $2\boldsymbol{x} \in K$. The choice $\boldsymbol{x}^* = 2\boldsymbol{x}$ in (2.21) implies

$$\boldsymbol{x}^{\mathrm{T}} \boldsymbol{f}(\boldsymbol{x}) \geq 0\,. \tag{2.37}$$

Combining both inequalities implies $\boldsymbol{x}^{\mathrm{T}} \boldsymbol{f}(\boldsymbol{x}) = 0$. Consequently, equation (2.21) implies $\boldsymbol{f}(\boldsymbol{x})^{\mathrm{T}} \boldsymbol{x}^* \geq 0$ for all $\boldsymbol{x}^* \in K$, which is exactly $\boldsymbol{f}(\boldsymbol{x}) \in K^*$. Therefore, $\boldsymbol{x}$ solves the CP $(K, \boldsymbol{f})$. Conversely, if $\boldsymbol{x}$ solves the CP $(K, \boldsymbol{f})$, it follows immediately that $\boldsymbol{x}$ solves the VI $(K, \boldsymbol{f})$. Specifically, $\boldsymbol{f}(\boldsymbol{x}) \in K^* \Leftrightarrow \boldsymbol{f}(\boldsymbol{x})^{\mathrm{T}} \boldsymbol{x}^* \geq 0$ for all $\boldsymbol{x}^* \in K$. Since $\boldsymbol{x}^{\mathrm{T}} \boldsymbol{f}(\boldsymbol{x}) = 0$ this implies $\boldsymbol{f}(\boldsymbol{x}) \in K^* \Leftrightarrow (\boldsymbol{x}^* - \boldsymbol{x})^{\mathrm{T}} \boldsymbol{f}(\boldsymbol{x}) \geq 0$ for all $\boldsymbol{x}^* \in K$. □

There exist many special cases of complementarity problems that are frequently encountered. The most prominent example is the case where $K = \mathbb{R}^{n+}_0$, i.e., the non-negative orthant. Its dual cone $K^* = \mathbb{R}^{n+}_0$ is again the non-negative orthant itself. In this case, the CP $(\mathbb{R}^{n+}_0, \boldsymbol{f})$ is called *nonlinear complementarity problem*, denoted NCP $(\boldsymbol{f})$. If, in addition, the mapping $\boldsymbol{f}$ is affine, i.e., $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{M}\boldsymbol{x} + \boldsymbol{q}$, with a given tuple $\boldsymbol{q} \in \mathbb{R}^n$ and matrix $\boldsymbol{M} \in \mathbb{R}^{n\times n}$, the NCP $(\boldsymbol{f})$ is called a *linear complementarity problem*.

Definition 2.17. For a given tuple $\boldsymbol{q} \in \mathbb{R}^n$ and matrix $\boldsymbol{M} \in \mathbb{R}^{n\times n}$, the *linear complementarity problem*, denoted as LCP $(\boldsymbol{q}, \boldsymbol{M})$, is to find a tuple $\boldsymbol{x} \in \mathbb{R}^n$ that satisfies

$$\boldsymbol{0}_n \leq \boldsymbol{x}\,, \quad \boldsymbol{0}_n \leq \boldsymbol{y} \quad \text{and} \quad \boldsymbol{x}^{\mathrm{T}}\boldsymbol{y} = 0\,, \tag{2.38}$$

where $\boldsymbol{y} = \boldsymbol{M}\boldsymbol{x} + \boldsymbol{q}$ and both inequalities are understood element-wise, i.e., $0 \leq x_i, y_i$ for all $i = 1, \ldots, n$. The orthogonality notation $\boldsymbol{x} \perp \boldsymbol{y} \Leftrightarrow \boldsymbol{x}^{\mathrm{T}}\boldsymbol{y} = 0$ is commonly used to write (2.38) more concisely as

$$\boldsymbol{0}_n \leq \boldsymbol{x} \perp \boldsymbol{y} \geq \boldsymbol{0}_n\,. \tag{2.39}$$

A tuple $\boldsymbol{x}$ that solves the LCP $(\boldsymbol{q}, \boldsymbol{M})$ also solves VI $(\mathbb{R}^{n+}_0, \boldsymbol{M}\boldsymbol{x} + \boldsymbol{q})$. According to Corollary 2.7 this is equivalent to $-\boldsymbol{y} \in \mathcal{N}_{\mathbb{R}^{n+}_0}(\boldsymbol{x})$, where $\boldsymbol{y} = \boldsymbol{M}\boldsymbol{x} + \boldsymbol{q}$. Since $\boldsymbol{x}^* \in \mathbb{R}^{n+}_0 \Leftrightarrow -\boldsymbol{x}^* \in \mathbb{R}^{n-}_0$, a tuple $\boldsymbol{x} \in \mathbb{R}^{n+}_0$ solves VI $(\mathbb{R}^{n+}_0, \boldsymbol{M}\boldsymbol{x} + \boldsymbol{q})$ if and only if $-\boldsymbol{x} \in \mathbb{R}^{n-}_0$ solves VI $(\mathbb{R}^{n-}_0, -\boldsymbol{M}\boldsymbol{x} - \boldsymbol{q})$. Therefore, Corollary 2.7 finally reveals the equivalences

$$\boldsymbol{0}_n \leq \boldsymbol{x} \perp \boldsymbol{y} \geq \boldsymbol{0}_n \quad \Longleftrightarrow \quad -\boldsymbol{y} \in \mathcal{N}_{\mathbb{R}^{n+}_0}(\boldsymbol{x}) \quad \Longleftrightarrow \quad \boldsymbol{y} \in \mathcal{N}_{\mathbb{R}^{n-}_0}(-\boldsymbol{x})\,. \tag{2.40}$$

This identity will be useful to describe the non-penetration condition, also known as Signorini's law, by a normal cone inclusion or linear complementarity problem. Consequently, the reformulation of normal cone inclusions as linear complementarity problems can be an interesting tool for the solution of contact problems. Further, it can be used to investigate its solvability. To establish the class of matrices $\boldsymbol{M}$ such that the LCP $(\boldsymbol{q}, \boldsymbol{M})$ has a unique solution for all tuples $\boldsymbol{q}$, the notion of P-matrices has to be introduced. This is done in accordance with the monograph [49].

Definition 2.18. Let $\boldsymbol{A} \in \mathbb{R}^{m\times n}$. For index sets $\alpha \subseteq \{1, 2, \ldots, m\}$ and $\beta \subseteq \{1, 2, \ldots, n\}$, the submatrix $\boldsymbol{A}_{\alpha\beta}$ of $\boldsymbol{A}$ is the matrix whose entries lie in the rows of $\boldsymbol{A}$ indexed by α and the columns indexed by β. If $\alpha = \{1, \ldots, m\}$, the submatrix $\boldsymbol{A}_{\alpha\beta}$ is denoted $\boldsymbol{A}_{\bullet\beta}$; similarly, if $\beta = \{1, \ldots, n\}$,

the submatrix $\boldsymbol{A}_{\alpha\beta}$ is denoted $\boldsymbol{A}_{\alpha\bullet}$. If $m = n$ and $\alpha = \beta$, the submatrix $\boldsymbol{A}_{\alpha\alpha}$ is called a *principal submatrix* of $\boldsymbol{A}$; the determinant of $\boldsymbol{A}_{\alpha\alpha}$ is called a *principal minor* of $\boldsymbol{A}$.

Definition 2.19. A matrix $\boldsymbol{M} \in \mathbb{R}^{n\times n}$ is said to be a P-matrix if all its principal minors are positive. If all the principal minors of $\boldsymbol{M}$ are nonnegative, $\boldsymbol{M}$ is said to be a P_0-matrix.

Example 2.20. Let

$$\boldsymbol{M} = \begin{pmatrix} \frac{5}{12} & -\frac{1}{12} \\ \frac{3}{4} & \frac{1}{4} \end{pmatrix} \quad \text{and} \quad \boldsymbol{M}^2 = \boldsymbol{M}\boldsymbol{M} = \begin{pmatrix} \frac{1}{9} & -\frac{1}{18} \\ \frac{1}{2} & 0 \end{pmatrix}.$$

The principal submatrices of $\boldsymbol{M}$ and $\boldsymbol{M}^2$ are

$$\left\{ \begin{pmatrix} \frac{5}{12} \end{pmatrix}, \begin{pmatrix} \frac{1}{4} \end{pmatrix}, \begin{pmatrix} \frac{5}{12} & -\frac{1}{12} \\ \frac{3}{4} & \frac{1}{4} \end{pmatrix} \right\} \quad \text{and} \quad \left\{ \begin{pmatrix} \frac{1}{9} \end{pmatrix}, \begin{pmatrix} 0 \end{pmatrix}, \begin{pmatrix} \frac{1}{9} & -\frac{1}{18} \\ \frac{1}{2} & 0 \end{pmatrix} \right\},$$

respectively. Moreover, their corresponding principal minors are

$$\left\{ \frac{5}{12}, \frac{1}{4}, \det(\boldsymbol{M}) = \frac{1}{6} \right\} \quad \text{and} \quad \left\{ \frac{1}{9}, 0, \det(\boldsymbol{M}^2) = \frac{1}{36} \right\}.$$

Clearly, $\boldsymbol{M}$ belongs to the class of P-matrices, while $\boldsymbol{M}^2$ is only P_0 since one principal minor is exactly zero.

Proposition 2.21 (Cottle et al. [49], Theorem 3.3.7)**.** *A matrix $\boldsymbol{M} \in \mathbb{R}^{n\times n}$ is a P-matrix if and only if the LCP $(\boldsymbol{q}, \boldsymbol{M})$ has a unique solution for all tuples $\boldsymbol{q} \in \mathbb{R}^n$.*

2.4 Miscellaneous Topics from Linear Algebra

Let $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{B} \in \mathbb{R}^{p\times q}$, then the *Kronecker product* $\boldsymbol{A} \otimes \boldsymbol{B}$ is the $mp \times nq$ block matrix

$$\boldsymbol{C} = \boldsymbol{A} \otimes \boldsymbol{B} = \begin{pmatrix} a_{11}\boldsymbol{B} & a_{12}\boldsymbol{B} & \dots & a_{1n}\boldsymbol{B} \\ a_{21}\boldsymbol{B} & a_{22}\boldsymbol{B} & \dots & a_{2n}\boldsymbol{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}\boldsymbol{B} & a_{m2}\boldsymbol{B} & \dots & a_{mn}\boldsymbol{B} \end{pmatrix} \in \mathbb{R}^{mp\times nq}. \tag{2.41}$$

Using index notation, this reads $c_{p(r-1)+v,q(s-1)+w} = a_{rs}b_{vw}$. The definition of Kronecker products can be extended to $\mathbb{R}^m$ tuples. Let $\boldsymbol{a} \in \mathbb{R}^m$ and

$\boldsymbol{B} \in \mathbb{R}^{p\times q}$. Then, the Kronecker product $\boldsymbol{a} \otimes \boldsymbol{B}$ is defined by interpreting the tuple $\boldsymbol{a}$ as column-tuple $\boldsymbol{A} \in \mathbb{R}^{m\times 1}$, i.e., $\boldsymbol{a} \otimes \boldsymbol{B} = \boldsymbol{A} \otimes \boldsymbol{B}$. Similarly, the Kronecker product $\boldsymbol{a}^{\mathrm{T}} \otimes \boldsymbol{B} = \boldsymbol{A}^{\mathrm{T}} \otimes \boldsymbol{B}$. For $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{b} \in \mathbb{R}^{p}$ the identical definition is used where the tuple $\boldsymbol{b}$ is interpreted as column-tuple $\boldsymbol{B} \in \mathbb{R}^{p\times 1}$. Hence, $\boldsymbol{A} \otimes \boldsymbol{b} = \boldsymbol{A} \otimes \boldsymbol{B}$ and $\boldsymbol{A} \otimes \boldsymbol{b}^{\mathrm{T}} = \boldsymbol{A} \otimes \boldsymbol{B}^{\mathrm{T}}$.

A frequently encountered problem is to solve the linear system of equations $\boldsymbol{M}\boldsymbol{z} = \boldsymbol{w}$ with block structure defined in terms of

$$\boldsymbol{M} = \begin{pmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{pmatrix}, \quad \boldsymbol{z} = \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{pmatrix} \quad \text{and} \quad \boldsymbol{w} = \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{v} \end{pmatrix} \tag{2.42}$$

of appropriate dimensions. Assuming $\boldsymbol{A}$ to be regular, the first row can be rearranged as

$$\boldsymbol{x} = \boldsymbol{A}^{-1}(\boldsymbol{u} - \boldsymbol{B}\boldsymbol{y}), \tag{2.43}$$

which eliminates $\boldsymbol{x}$ from the set of unknowns. Substituting this expression into the second set of equations yields

$$(\boldsymbol{D} - \boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B})\boldsymbol{y} = \boldsymbol{v} - \boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{u}. \tag{2.44}$$

Therein, the matrix $\boldsymbol{S}_{\boldsymbol{A}} = \boldsymbol{D} - \boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B}$ is called the *Schur complement* of the upper left block $\boldsymbol{A}$ in $\boldsymbol{M}$. Consequently, the second equation can be solved as

$$\boldsymbol{y} = \boldsymbol{S}_{\boldsymbol{A}}^{-1}(\boldsymbol{v} - \boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{u}). \tag{2.45}$$

Substituting this again into (2.43) yields

$$\boldsymbol{x} = (\boldsymbol{A}^{-1} + \boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{S}_{\boldsymbol{A}}^{-1}\boldsymbol{C}\boldsymbol{A}^{-1})\boldsymbol{u} - \boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{S}_{\boldsymbol{A}}^{-1}\boldsymbol{v}. \tag{2.46}$$

Consequently, the solution of the linear system $\boldsymbol{M}\boldsymbol{z} = \boldsymbol{w}$ can be written as

$$\boldsymbol{z} = \boldsymbol{M}^{-1}\boldsymbol{w} \tag{2.47}$$

with the inverse of the block matrix $\boldsymbol{M}$ given as

$$\boldsymbol{M}^{-1} = \begin{pmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{pmatrix}^{-1} = \begin{pmatrix} \boldsymbol{A}^{-1} + \boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{S}_{\boldsymbol{A}}^{-1}\boldsymbol{C}\boldsymbol{A}^{-1} & -\boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{S}_{\boldsymbol{A}}^{-1} \\ -\boldsymbol{S}_{\boldsymbol{A}}^{-1}\boldsymbol{C}\boldsymbol{A}^{-1} & \boldsymbol{S}_{\boldsymbol{A}}^{-1} \end{pmatrix}. \tag{2.48}$$

Similar steps as outlined above can be performed under the assumption that $\boldsymbol{B}$, $\boldsymbol{C}$ or $\boldsymbol{D}$ are regular. Specifically, for the case of a regular $\boldsymbol{B}$, this leads to

$$\boldsymbol{M}^{-1} = \begin{pmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{pmatrix}^{-1} = \begin{pmatrix} -\boldsymbol{S}_{\boldsymbol{B}}^{-1}\boldsymbol{D}\boldsymbol{B}^{-1} & \boldsymbol{S}_{\boldsymbol{B}}^{-1} \\ \boldsymbol{B}^{-1} + \boldsymbol{B}^{-1}\boldsymbol{A}\boldsymbol{S}_{\boldsymbol{B}}^{-1}\boldsymbol{D}\boldsymbol{B}^{-1} & -\boldsymbol{B}^{-1}\boldsymbol{A}\boldsymbol{S}_{\boldsymbol{B}}^{-1} \end{pmatrix}, \tag{2.49}$$

with the Schur complement $\boldsymbol{S}_{\boldsymbol{B}}$ of the upper right block $\boldsymbol{B}$ in $\boldsymbol{M}$ given as

$$\boldsymbol{S}_{\boldsymbol{B}} = \boldsymbol{C} - \boldsymbol{D}\boldsymbol{B}^{-1}\boldsymbol{A}. \tag{2.50}$$

Chapter 3

Mechanical Systems with Frictional Contact and Impacts

> It is possible that things of so little moment and so easy to set right can occupy and perplex a ripe wit like yours, fit to break through and crush far greater obstacles?
>
> —*Miguel de Cervantes*

This chapter introduces the reader to the theory of nonsmooth mechanical systems with frictional contact and impacts. Thereto, it starts with a short note about kinematics. Subsequently, Section 3.2 introduces a generalization of the classically used equations of motions, formulated as an *equality of measures*. It links the change of velocity of the system with the applied forces. Since the velocities are assumed to be special functions of locally bounded variation [72, 129], they can be decomposed in an absolute continuous part and a collection of step functions. This decomposition reveals the decomposition of the motion in smooth and impulsive parts. Subsequent to a unified description of bilateral constraints on position and velocity level, Section 3.4 introduces normal contacts by means of a set-valued relation, called *Signorini's law*. These are complemented by the statement of set-valued friction laws. Since the precise description of physical processes requires the treatment of a variety of different friction phenomena, Section 3.5 gives a comprehensive overview on different types of friction models. Using appropriate scaling techniques, all of them can be described by a similar set-valued relation. The description of frictional contact is completed by the discussion of impacts. This requires that the impact equation is satisfied together with appropriate impact laws. Finally,

the chapter closes by discussing consistent initial conditions required for a later numerical simulation.

3.1 Kinematics

Consider a finite-dimensional mechanical system whose state is described by the generalized positions $\boldsymbol{q}(t) \in \mathbb{R}^{n_q}$ and by the generalized velocities $\boldsymbol{u}(t) \in \mathbb{R}^{n_u}$, which are both functions of time $t \in \mathbb{R}$. The relation between positions and velocities is a consequence of the kinematics of the system and is in general of the form

$$\mathrm{d}\boldsymbol{q} = \dot{\boldsymbol{q}}(t, \boldsymbol{q}, \boldsymbol{u})\,\mathrm{d}t\,, \quad \text{where} \quad \dot{\boldsymbol{q}}(t, \boldsymbol{q}, \boldsymbol{u}) = \boldsymbol{B}(t, \boldsymbol{q})\boldsymbol{u} + \boldsymbol{\beta}(t, \boldsymbol{q}) \tag{3.1}$$

denotes the kinematic differential equation with $\boldsymbol{B}(t, \boldsymbol{q}) \in \mathbb{R}^{n_q \times n_u}$ and $\boldsymbol{\beta}(t, \boldsymbol{q}) \in \mathbb{R}^{n_q}$. Therein, $n_u \leq n_q$, although often $n_u < n_q$, e.g., when parametrizing finite rotations with unit quaternions and choosing angular velocities as minimal velocities, see [56].

3.2 Equations of Motion

In addition to the applied forces, the mechanical system is subject to bilateral and frictional unilateral constraints. Its dynamics is then described by the equality of measures [5, 30, 113, 128]

$$\begin{aligned} \boldsymbol{M}(t, \boldsymbol{q})\,\mathrm{d}\boldsymbol{u} &= \boldsymbol{h}(t, \boldsymbol{q}, \boldsymbol{u})\,\mathrm{d}t + \boldsymbol{W}_g(t, \boldsymbol{q})\,\mathrm{d}\boldsymbol{\pi}_g + \boldsymbol{W}_\gamma(t, \boldsymbol{q})\,\mathrm{d}\boldsymbol{\pi}_\gamma \\ &\quad + \boldsymbol{W}_N(t, \boldsymbol{q})\,\mathrm{d}\boldsymbol{\pi}_N + \boldsymbol{W}_F(t, \boldsymbol{q})\,\mathrm{d}\boldsymbol{\pi}_F \\ &= \boldsymbol{h}(t, \boldsymbol{q}, \boldsymbol{u})\,\mathrm{d}t + \boldsymbol{W}(t, \boldsymbol{q})\,\mathrm{d}\boldsymbol{\pi}\,. \end{aligned} \tag{3.2}$$

It links the change in velocity $\mathrm{d}\boldsymbol{u}$ to the forces acting on the system. Therein, $\boldsymbol{M}(t, \boldsymbol{q}) \in \mathbb{R}^{n_u \times n_u}$ denotes the symmetric mass matrix, possibly depending explicitly on time t and on the generalized positions $\boldsymbol{q}$. Let $n_\pi = n_g + n_\gamma + n_N + n_F$ denote the total number of constraints. For $\square \in \{g, \gamma, N, F\}$, their corresponding generalized force directions $\boldsymbol{W}_\square \in \mathbb{R}^{n_u \times n_\square}$ and percussion measures $\mathrm{d}\boldsymbol{\pi}_\square \in \mathbb{R}^{n_\square}$ are gathered in $\boldsymbol{W} = (\boldsymbol{W}_g\ \boldsymbol{W}_\gamma\ \boldsymbol{W}_N\ \boldsymbol{W}_F) \in \mathbb{R}^{n_u \times n_\pi}$ and[1] $\mathrm{d}\boldsymbol{\pi} = (\mathrm{d}\boldsymbol{\pi}_g, \mathrm{d}\boldsymbol{\pi}_\gamma, \mathrm{d}\boldsymbol{\pi}_N, \mathrm{d}\boldsymbol{\pi}_F) \in \mathbb{R}^{n_\pi}$, respectively. All forces which are neither constraint nor contact forces, such as spring forces, dashpot forces and gyroscopic terms, are assumed to be nonimpulsive and are contained in $\boldsymbol{h}(t, \boldsymbol{q}, \boldsymbol{u}) \in \mathbb{R}^{n_u}$. Let $\boldsymbol{\lambda}^+(t_i) = \lim_{\tau \downarrow t_i} \boldsymbol{\lambda}(\tau)$ and $\boldsymbol{\lambda}^-(t_i) = \lim_{\tau \uparrow t_i} \boldsymbol{\lambda}(\tau)$

[1] Herein, the following notation is used: For $\boldsymbol{x} \in \mathbb{R}^{n \times 1} \cong \mathbb{R}^n$ and $\boldsymbol{y} \in \mathbb{R}^{m \times 1} \cong \mathbb{R}^m$, $(\boldsymbol{x}, \boldsymbol{y}) := (\boldsymbol{x}^{\mathrm{T}}\ \boldsymbol{y}^{\mathrm{T}})^{\mathrm{T}} \in \mathbb{R}^{n+m \times 1} \cong \mathbb{R}^{n+m}$.

respectively denote the right and left limits of $\boldsymbol{\lambda}$ at t_i. Following [128], the percussion measures $\mathrm{d}\boldsymbol{\pi}_\square$ combine the effects of nonimpulsive forces $\boldsymbol{\lambda}_\square$ and impulsive forces $\boldsymbol{\Lambda}_\square$ in the sense that

$$\mathrm{d}\boldsymbol{\pi}_\square = \boldsymbol{\lambda}_\square \,\mathrm{d}t + \sum_i \left(\boldsymbol{\pi}_\square^+(t_i) - \boldsymbol{\pi}_\square^-(t_i)\right) \mathrm{d}\delta_{t_i} = \boldsymbol{\lambda}_\square \,\mathrm{d}t + \boldsymbol{\Lambda}_\square \,\mathrm{d}\eta\,, \tag{3.3}$$

where $\mathrm{d}\eta$ denotes the atomic measure that can be interpreted as the sum of Dirac point measures $\mathrm{d}\delta_{t_i}$ in accordance with

$$\mathrm{d}\eta = \sum_i \mathrm{d}\delta_{t_i}\,, \quad \int_{[t_a,t_b]} \mathrm{d}\delta_{t_i} = \begin{cases} 1 & t_i \in [t_a,t_b]\,, \\ 0 & t_i \notin [t_a,t_b]\,. \end{cases} \tag{3.4}$$

Similarly, $\boldsymbol{u}^+(t_i) = \lim_{\tau\downarrow t_i} \boldsymbol{u}(\tau)$ and $\boldsymbol{u}^-(t_i) = \lim_{\tau\uparrow t_i} \boldsymbol{u}(\tau)$ respectively denote the right and left limits of $\boldsymbol{u}$ at t_i. The velocity measure $\mathrm{d}\boldsymbol{u}$ can be decomposed in a absolutely continuous part $\dot{\boldsymbol{u}}$ that exists almost everywhere and $\boldsymbol{U}$ that takes the discontinuities of $\boldsymbol{u}$ into account. Specifically,

$$\mathrm{d}\boldsymbol{u} = \dot{\boldsymbol{u}}\,\mathrm{d}t + \sum_i \left(\boldsymbol{u}^+(t_i) - \boldsymbol{u}^-(t_i)\right) \mathrm{d}\delta_{t_i} = \dot{\boldsymbol{u}}\,\mathrm{d}t + \boldsymbol{U}\,\mathrm{d}\eta\,, \tag{3.5}$$

where in the last equality (3.4) was applied.

3.3 Bilateral Constraints

The generalized percussion measures $\boldsymbol{W}_g \mathrm{d}\boldsymbol{\pi}_g$ and $\boldsymbol{W}_\gamma \mathrm{d}\boldsymbol{\pi}_\gamma$ in (3.2) contain the perfect constraint forces of the bilateral position and velocity level constraints

$$\boldsymbol{g}(t,\boldsymbol{q}) = \boldsymbol{0}_{n_g} \quad \text{and} \quad \boldsymbol{\gamma}(t,\boldsymbol{q},\boldsymbol{u}) = \boldsymbol{W}_\gamma^{\mathrm{T}}(t,\boldsymbol{q})\,\boldsymbol{u} + \boldsymbol{\chi}_\gamma(t,\boldsymbol{q}) = \boldsymbol{0}_{n_\gamma}\,, \tag{3.6}$$

respectively. Using the kinematic differential equation (3.1), the time derivative of the bilateral constraints on position level is written as

$$\begin{aligned} \dot{\boldsymbol{g}}(t,\boldsymbol{q},\boldsymbol{u}) &= \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{q}}(t,\boldsymbol{q})\,\dot{\boldsymbol{q}} + \frac{\partial \boldsymbol{g}}{\partial t}(t,\boldsymbol{q}) \\ &\overset{(3.1)}{=} \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{q}}(t,\boldsymbol{q})\Big(\boldsymbol{B}(t,\boldsymbol{q})\boldsymbol{u} + \boldsymbol{\beta}(t,\boldsymbol{q})\Big) + \frac{\partial \boldsymbol{g}}{\partial t}(t,\boldsymbol{q}) \\ &= \boldsymbol{W}_g^{\mathrm{T}}(t,\boldsymbol{q})\,\boldsymbol{u} + \boldsymbol{\chi}_g(t,\boldsymbol{q})\,. \end{aligned} \tag{3.7}$$

Hence, the generalized force directions for bilateral constraints on position and velocity level are given by

$$\boldsymbol{W}_g^{\mathrm{T}} = \frac{\partial \dot{\boldsymbol{g}}}{\partial \boldsymbol{u}} = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{q}}\boldsymbol{B} \quad \text{and} \quad \boldsymbol{W}_\gamma^{\mathrm{T}} = \frac{\partial \boldsymbol{\gamma}}{\partial \boldsymbol{u}}\,. \tag{3.8}$$

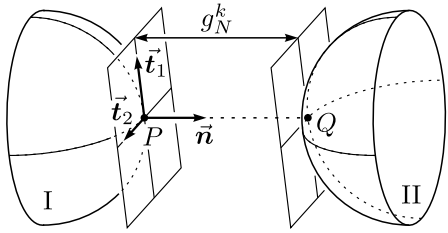

Figure 3.1: Kinematics of the contact k consisting of the material contact points P and Q on the bodies I and II, respectively.

3.4 Unilateral Contact

The not yet discussed generalized force measures of (3.2), i.e., $\boldsymbol{W}_N \mathrm{d}\boldsymbol{\pi}_N + \boldsymbol{W}_F \mathrm{d}\boldsymbol{\pi}_F$, contain the percussion measures describing the frictional contact. Their corresponding force laws can conveniently be described by normal cone inclusions [5, 30, 74, 114]. Consider a pair of contact points on either contacting body, which is simply referred to as contact k. Let

$$g_N^k(t,\boldsymbol{q}) = \vec{\boldsymbol{n}}(t,\boldsymbol{q}) \cdot \big(\vec{\boldsymbol{r}}_{OQ}(t,\boldsymbol{q}) - \vec{\boldsymbol{r}}_{OP}(t,\boldsymbol{q})\big) \tag{3.9}$$

be the real-valued gap function[2] describing the signed distance between the tangent planes of the pairs of contact points. An illustration thereof is given in Figure 3.1. This contact distance is composed of the contact normal $\vec{\boldsymbol{n}}(t,\boldsymbol{q}) \in \mathbb{E}^3$ and the vectors $\vec{\boldsymbol{r}}_{OP}(t,\boldsymbol{q}), \vec{\boldsymbol{r}}_{OQ}(t,\boldsymbol{q}) \in \mathbb{E}^3$, addressing the material points P and Q, respectively. The impenetrability of contact k is enforced by the perfect unilateral constraint $g_N^k \geq 0$. The contact is open if $g_N^k > 0$, i.e., the contacting bodies are separated. Excluding distance effects (e.g. magnetic or gravitational forces), the contact force $\lambda_N^k = 0$ is zero for an open contact. The contact is said to be closed if $g_N^k = 0$, i.e., the bodies are touching. For $g_N^k < 0$, the bodies are penetrating each other. Assuming non-adhesive contacts implies a non-negative contact force $\lambda_N^k \geq 0$. A graphical illustration of this contact law, known as *Signorini's law*, is given in Figure 3.2. According to (2.40), it can equivalently be stated as normal cone inclusion to the set $\mathbb{R}_0^- = \{x \in \mathbb{R} \,|\, x \leq 0\}$ of non-positive

[2]The evaluation of (3.9) is performed using as specific coordinate representation of the involved vectors, e.g., $g_N^k(t,\boldsymbol{q}) = {}_I\vec{\boldsymbol{n}}^{\mathrm{T}}(t,\boldsymbol{q})({}_I\vec{\boldsymbol{r}}_{OQ}(t,\boldsymbol{q}) - {}_I\vec{\boldsymbol{r}}_{OP}(t,\boldsymbol{q}))$, for the inertial I-basis. However, the specific choice of the used basis dramatically influences the complexity of the resulting equations and consequently the performance of numerical implementations.

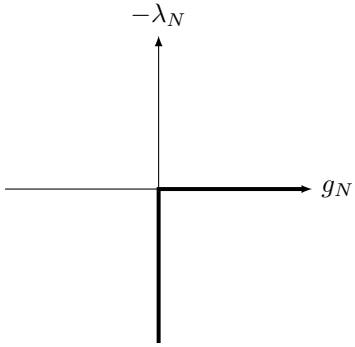

Figure 3.2: Graphical illustration of Signorini's law $g_N \in \mathcal{N}_{\mathbb{R}_0^-}(-\lambda_N) \Leftrightarrow g_N^k(t,\boldsymbol{q}) \geq 0\,,\ \lambda_N^k \geq 0\,,\ g_N^k(t,\boldsymbol{q})\,\lambda_N^k = 0.$

real numbers or as inequality complementarity condition

$$\begin{gathered} g_N^k(t,\boldsymbol{q}) \in \mathcal{N}_{\mathbb{R}_0^-}(-\lambda_N^k) \\ \Updownarrow \\ g_N^k(t,\boldsymbol{q}) \geq 0\,, \quad \lambda_N^k \geq 0\,, \quad g_N^k(t,\boldsymbol{q})\,\lambda_N^k = 0\,. \end{gathered} \tag{3.10}$$

Therein, the normal cone $\mathcal{N}_C$ to the set $C = \mathbb{R}_0^-$ as defined in (2.10) is used. It is convenient to define the set of active (closed or penetrated) contacts as[3]

$$A(t,\boldsymbol{q}) = \left\{k = 1,\dots,n_N \;\middle|\; g_N^k(t,\boldsymbol{q}) \leq 0\right\}, \tag{3.11}$$

together with its complement $A^c = \{1,\dots,n_N\} \setminus A$, which is called the set of inactive (open) contacts. Let the gap functions of all contacts $k = 1,\dots,n_N$ be gathered into a tuple $\boldsymbol{g}_N(t,\boldsymbol{q}) \in \mathbb{R}^{n_N}$. In accordance with (3.7) and by using the kinematic differential equation (3.1), the gap velocity is defined as

$$\begin{aligned} \dot{\boldsymbol{g}}_N(t,\boldsymbol{q},\boldsymbol{u}) &= \frac{\partial \boldsymbol{g}_N}{\partial \boldsymbol{q}}(t,\boldsymbol{q})\,\dot{\boldsymbol{q}} + \frac{\partial \boldsymbol{g}_N}{\partial t}(t,\boldsymbol{q}) \\ &\overset{(3.1)}{=} \frac{\partial \boldsymbol{g}_N}{\partial \boldsymbol{q}}(t,\boldsymbol{q})\Big(\boldsymbol{B}(t,\boldsymbol{q})\boldsymbol{u} + \boldsymbol{\beta}(t,\boldsymbol{q})\Big) + \frac{\partial \boldsymbol{g}_N}{\partial t}(t,\boldsymbol{q}) \\ &= \boldsymbol{W}_N^{\mathrm{T}}(t,\boldsymbol{q})\,\boldsymbol{u} + \boldsymbol{\chi}_N(t,\boldsymbol{q})\,. \end{aligned} \tag{3.12}$$

[3]For the later implementation of a numerical method, it is crucial to consider in (3.11) not only closed contacts but also the potential occurrence of penetrated contacts arising from drift problems or solutions that have not yet reached convergence.

Again, the generalized force directions arising in (3.2) are defined as

$$\boldsymbol{W}_N^{\mathrm{T}} = \frac{\partial \dot{\boldsymbol{g}}_N}{\partial \boldsymbol{u}} = \frac{\partial \boldsymbol{g}_N}{\partial \boldsymbol{q}} \boldsymbol{B} \,. \tag{3.13}$$

In the following, all quantities are considered to be right-continuous, e.g., the velocity $\boldsymbol{u}(t)$ is equal to its right-limit

$$\boldsymbol{u}(t) \coloneqq \boldsymbol{u}^+(t) = \lim_{\tau \downarrow t} \boldsymbol{u}(\tau) \,. \tag{3.14}$$

For active contacts $k \in A(t, \boldsymbol{q})$, this allows to state Signorini's law (3.10) on velocity level, see Section 7.2 and Section 10.4 of Glocker [72]. Specifically,

$$\begin{cases} \dot{g}_N^k(t, \boldsymbol{q}, \boldsymbol{u}) \in \mathcal{N}_{\mathbb{R}_0^-}(-\lambda_N^k) & \text{if } k \in A(t, \boldsymbol{q}) \,, \\ \lambda_N^k = 0 & \text{if } k \notin A(t, \boldsymbol{q}) \,. \end{cases} \tag{3.15}$$

According to Proposition 2.4 in [128], known as *Moreau's viability lemma*, equation (3.15) together with $g_N^k(t_0, \boldsymbol{q}(t_0)) \geq 0$ implies the unilateral constraint on position level

$$g_N^k(t, \boldsymbol{q}(t)) \geq 0 \,, \quad \forall t \geq t_0 \,. \tag{3.16}$$

Hence, it guarantees that the two contacting bodies do not penetrate each other for all future time.

3.5 Set-valued Friction

Let $\vec{\boldsymbol{v}}_P \in \mathbb{E}^3$ and $\vec{\boldsymbol{v}}_Q \in \mathbb{E}^3$ describe the velocities of the material points P and Q, respectively. Moreover, the angular velocities of the bodies I and II are respectively denoted by $\vec{\boldsymbol{\Omega}}_{\mathrm{I}} \in \mathbb{E}^3$ and $\vec{\boldsymbol{\Omega}}_{\mathrm{II}} \in \mathbb{E}^3$. The relative tangent velocity of the contact k is defined by

$$\boldsymbol{\gamma}_T^k(t, \boldsymbol{q}, \boldsymbol{u}) = \begin{pmatrix} \vec{\boldsymbol{t}}_1(t, \boldsymbol{q}) \cdot \big(\vec{\boldsymbol{v}}_Q(t, \boldsymbol{q}, \boldsymbol{u}) - \vec{\boldsymbol{v}}_P(t, \boldsymbol{q}, \boldsymbol{u})\big) \\ \vec{\boldsymbol{t}}_2(t, \boldsymbol{q}) \cdot \big(\vec{\boldsymbol{v}}_Q(t, \boldsymbol{q}, \boldsymbol{u}) - \vec{\boldsymbol{v}}_P(t, \boldsymbol{q}, \boldsymbol{u})\big) \end{pmatrix} \tag{3.17}$$

and is given by the projection of the relative velocity $\vec{\boldsymbol{v}}_Q - \vec{\boldsymbol{v}}_P \in \mathbb{E}^3$ onto both tangents $\vec{\boldsymbol{t}}_1 \in \mathbb{E}^3$ and $\vec{\boldsymbol{t}}_2 \in \mathbb{E}^3$, respectively, see Figure 3.1. Accordingly, the relative angular velocity $\vec{\boldsymbol{\Omega}}_{\mathrm{II}} - \vec{\boldsymbol{\Omega}}_{\mathrm{I}} \in \mathbb{E}^3$ can be projected onto the normal $\vec{\boldsymbol{n}} \in \mathbb{E}^3$ and on both tangents $\vec{\boldsymbol{t}}_1, \vec{\boldsymbol{t}}_2 \in \mathbb{E}^3$. This gives rise to the description of two more friction phenomena. The first one is described by the so-called relative pivoting velocity

$$\gamma_P^k(t, \boldsymbol{q}, \boldsymbol{u}) = \vec{\boldsymbol{n}}(t, \boldsymbol{q}) \cdot \big(\vec{\boldsymbol{\Omega}}_{\mathrm{II}}(t, \boldsymbol{q}, \boldsymbol{u}) - \vec{\boldsymbol{\Omega}}_{\mathrm{I}}(t, \boldsymbol{q}, \boldsymbol{u})\big) \tag{3.18}$$

and accounts for dissipation phenomenon in pure pivoting motions [155]. The kinematic quantity γ_P^k is required for the formulation of Coulomb–Contensou friction, a combined spatial Coulomb and pivoting friction, see [112]. The second additional friction phenomenon depends on the relative rolling velocity

$$\boldsymbol{\gamma}_R^k(t,\boldsymbol{q},\boldsymbol{u}) = \begin{pmatrix} \vec{\boldsymbol{t}}_1(t,\boldsymbol{q}) \cdot \left(\vec{\boldsymbol{\Omega}}_{\mathrm{II}}(t,\boldsymbol{q},\boldsymbol{u}) - \vec{\boldsymbol{\Omega}}_{\mathrm{I}}(t,\boldsymbol{q},\boldsymbol{u})\right) \\ \vec{\boldsymbol{t}}_2(t,\boldsymbol{q}) \cdot \left(\vec{\boldsymbol{\Omega}}_{\mathrm{II}}(t,\boldsymbol{q},\boldsymbol{u}) - \vec{\boldsymbol{\Omega}}_{\mathrm{I}}(t,\boldsymbol{q},\boldsymbol{u})\right) \end{pmatrix} \tag{3.19}$$

and can be used for a macroscopic model of dissipation in pure rolling motions[4], see [4, 111]. For the specific evaluation of $\boldsymbol{\gamma}_T$, γ_P and $\boldsymbol{\gamma}_R$, the same arguments concerning the choice of basis apply as in Section 3.4 for the evaluation of the gap function.

For a contact k, the set-valued friction law can in many cases be described by a normal cone inclusion

$$\boldsymbol{\gamma}_F^k(t,\boldsymbol{q},\boldsymbol{u}) \in \mathcal{N}_{C_F^k(\lambda_N^k)}(-\boldsymbol{\lambda}_F^k)\,, \tag{3.20}$$

where $\boldsymbol{\gamma}_F^k$ gathers all relative friction velocities and C_F^k denotes the set of admissible (negative) friction forces, which possibly depends in the corresponding normal force λ_N^k. Based on the specific choice of $\boldsymbol{\gamma}_F^k$ and C_F^k, different friction models emerge. Typically used models are subsequently discussed in detail.

Isotropic Coulomb Friction with Circular Force Reservoir

The most basic friction law that is typically used for the description of dry friction between isotropic surfaces is called isotropic Coulomb friction [14]. Let $\boldsymbol{\gamma}_F^k = \boldsymbol{\gamma}_T^k$, i.e., the friction velocity coincides with the relative tangent velocity (3.17). Furthermore, $\mu_T^k > 0$ denotes the positive friction coefficient of contact k. By that, spatial isotropic Coulomb friction is described by the set of admissible friction forces

$$C_F^k(\lambda_N^k) = B_2(\mu_T^k \lambda_N^k)\,, \tag{3.21}$$

where the $\mathbb{R}^f$-ball of radius r is defined as

$$B_f(r) = \{\boldsymbol{x} \in \mathbb{R}^f \,|\, r \geq \|\boldsymbol{x}\|\}\,. \tag{3.22}$$

[4] In some cases rolling friction coincides with contour friction, a friction law that describes the resisting moment against the movement of the immaterial contact point over the rim of a disk, see [108, 111, 114].

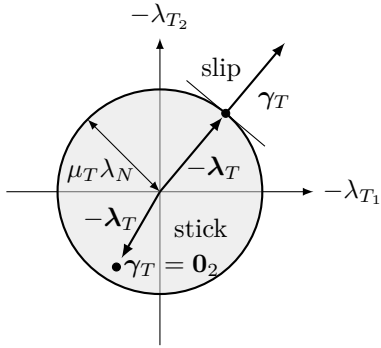

Figure 3.3: Spatial isotropic Coulomb friction $\boldsymbol{\gamma}_T \in \mathcal{N}_{B_2(\mu_T\lambda_N)}(-\boldsymbol{\lambda}_T)$.

Consequently, the friction law reads

$$\boldsymbol{\gamma}_T^k(t, \boldsymbol{q}, \boldsymbol{u}) \in \mathcal{N}_{B_2(\mu_T^k\lambda_N^k)}(-\boldsymbol{\lambda}_T^k)\,. \tag{3.23}$$

Using the definition of the normal cone (2.10), it can be verified that this describes isotropic Coulomb friction. Therefore two different cases have to be considered (for simplicity, the index k and the arguments of $\boldsymbol{\gamma}_T$ are suppressed):

(i) The friction force $\boldsymbol{\lambda}_T$ is in the interior of the set of admissible forces $B_2(\mu_T\lambda_N)$, i.e., $\|\boldsymbol{\lambda}_T\| < \mu_T\lambda_N$. Since for this case the normal cone reads $\mathcal{N}_{B_2(\mu_T\lambda_N)}(-\boldsymbol{\lambda}_T) = \{\mathbf{0}_2\}$, this implies *stick*, i.e.,

$$\boldsymbol{\gamma}_T \in \mathcal{N}_{B_2(\mu_T\lambda_N)}(-\boldsymbol{\lambda}_T) = \{\mathbf{0}_2\} \Leftrightarrow \boldsymbol{\gamma}_T = \mathbf{0}_2\,. \tag{3.24}$$

(ii) The friction force $\boldsymbol{\lambda}_T$ is on the boundary of the set of admissible forces $B_2(\mu_T\lambda_N)$, i.e., $\|\boldsymbol{\lambda}_T\| = \mu_T\lambda_N$. In this case, the normal cone is given by the outward normal ray to the disk, which starts at the origin and points in the direction $-\boldsymbol{\lambda}_T$, i.e.,

$$\begin{aligned}\mathcal{N}_{B_2(\mu_T\lambda_N)}(-\boldsymbol{\lambda}_T) &= \{-\alpha\boldsymbol{\lambda}_T/\|\boldsymbol{\lambda}_T\| \,|\, \alpha \geq 0\} \\ &= \{-\alpha\boldsymbol{\lambda}_T/(\mu_T\lambda_N) \,|\, \alpha \geq 0\} \\ &= \frac{\alpha}{\mu_T\lambda_N}\{-\boldsymbol{\lambda}_T\} \quad \text{for} \quad \alpha \geq 0\,.\end{aligned} \tag{3.25}$$

For $\alpha = 0$, this results again in stick with $\boldsymbol{\gamma}_T = \mathbf{0}_2$. For $\alpha > 0$, the inclusion (3.23) can be recast to $\boldsymbol{\gamma}_T/\alpha = -\boldsymbol{\lambda}_T/(\mu_T\lambda_N) = -\boldsymbol{\lambda}_T/\|\boldsymbol{\lambda}_T\|$.

Since $\boldsymbol{\gamma}_T$ and $\boldsymbol{\lambda}_T$ are colinear, but pointing in opposite directions, this implies $\alpha = \|\boldsymbol{\gamma}_T\|$, leading to the force law for *slip*, i.e.,

$$\boldsymbol{\lambda}_T = -\mu_T \lambda_N \frac{\boldsymbol{\gamma}_T}{\|\boldsymbol{\gamma}_T\|} \,. \tag{3.26}$$

A graphical illustration of spatial isotropic Coulomb friction is given in Figure 3.3.

Anisotropic Coulomb Friction with Elliptical Force Reservoir

Besides the description of idealized processes, typically used for academic examples, isotopic Coulomb friction is not sufficient to correctly describe the experimentally observed friction phenomena [14]. Manufacturing processes or material properties introduce anisotropic friction conditions. In order to describe such phenomena correctly, anisotropic Coulomb friction has to be considered. Similar to the isotropic case, we set $\boldsymbol{\gamma}_F^k = \boldsymbol{\gamma}_T^k$ but the set of admissible friction forces is chosen differently. For $\boldsymbol{a}^k = (a_1^k, a_2^k) \in \mathbb{R}^2$ with $a_i^k \in (0, 1]$, the set of admissible friction forces is given by

$$C_F^k(\lambda_N^k) = E_2(\mu_T^k \lambda_N^k; \boldsymbol{a}^k) \,, \tag{3.27}$$

where the $\mathbb{R}^f$-ellipsoid is defined as

$$E_f(r; \boldsymbol{a}) = \left\{ \boldsymbol{x} \in \mathbb{R}^f \;\middle|\; r^2 \geq \left(\frac{x_1}{a_1}\right)^2 + \left(\frac{x_2}{a_2}\right)^2 + \cdots + \left(\frac{x_f}{a_f}\right)^2 \right\} . \tag{3.28}$$

A graphical illustration of spatial anisotropic Coulomb friction is given in Figure 3.4 (a). As it will turn out in Chapter 6, such an ellipsoidal set of admissible friction forces is unsuitable for an efficient numerical implementation, since the projection onto the ellipsoidal set has no closed form solution. Moreover, efficient calculation of this projection is crucial for the numerical implementation of solution algorithms [62]. Computing the projection on a three dimensional ellipse E_3 requires the computationally expensive solution of a 4^{th}-order polynomial, see e.g. [59, 171]. As proposed in [126], a reformulation of the friction law can be used that results in a ball-like set of admissible friction forces, similar to the isotropic case. See also the investigations in [177]. By introducing the linear mappings

$$A^k \colon \mathbb{R}^2 \to \mathbb{R}^2 \,, \quad \boldsymbol{x} \mapsto \boldsymbol{A}^k \boldsymbol{x} \,, \quad \text{with} \quad \boldsymbol{A}^k = \operatorname{diag}(\boldsymbol{a}^k) \,, \tag{3.29}$$

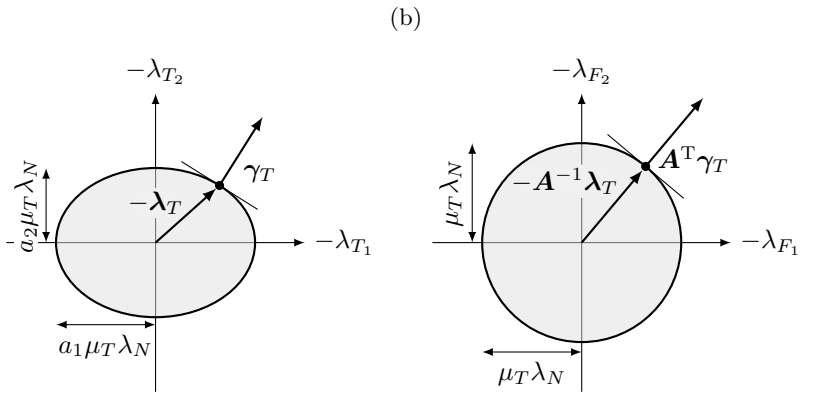

Figure 3.4: (a) Ellipsoidal set of admissible friction forces for spatial anisotropic Coulomb friction $\boldsymbol{\gamma}_T \in \mathcal{N}_{E_2(\mu_T^k\lambda_N^k;\boldsymbol{a}^k)}(-\boldsymbol{\lambda}_T)$. (b) Ball-like set of transformed admissible friction forces with friction law $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{\gamma}_T \in \mathcal{N}_{B_2(\mu\lambda_N)}(-\boldsymbol{A}^{-1}\boldsymbol{\lambda}_T)$.

the set (3.28) can be transformed to $B_2(\mu_T^k\lambda_N^k)$ via inverse scaling with $(A^k)^{-1}$, i.e., $B_2(\mu_T^k\lambda_N^k) = (A^k)^{-1}E_2(\mu_T^k\lambda_N^k;\boldsymbol{a}^k)$, see Section 2.1. Consequently, Proposition 2.4 reveals the equivalence

$$\begin{aligned} \boldsymbol{\gamma}_T^k(t,\boldsymbol{q},\boldsymbol{u}) &\in \mathcal{N}_{E_2(\mu_T^k\lambda_N^k;\boldsymbol{a}^k)}(-\boldsymbol{\lambda}_T^k) \\ &\Updownarrow \\ (\boldsymbol{A}^k)^{\mathrm{T}}\boldsymbol{\gamma}_T^k(t,\boldsymbol{q},\boldsymbol{u}) &\in \mathcal{N}_{(A^k)^{-1}E_2(\mu_T^k\lambda_N^k;\boldsymbol{a}^k)}\big(-(\boldsymbol{A}^k)^{-1}\boldsymbol{\lambda}_T\big)\,. \end{aligned} \tag{3.30}$$

By choosing friction velocities and forces in accordance with

$$\boldsymbol{\gamma}_F^k(t,\boldsymbol{q},\boldsymbol{u}) = (\boldsymbol{A}^k)^{\mathrm{T}}\boldsymbol{\gamma}_T^k(t,\boldsymbol{q},\boldsymbol{u}) \quad \text{and} \quad \boldsymbol{\lambda}_F^k = (\boldsymbol{A}^k)^{-1}\boldsymbol{\lambda}_T^k\,, \tag{3.31}$$

together with the set of admissible friction forces

$$C_F^k(\lambda_N^k) = (A^k)^{-1}E_2(\mu_T^k\lambda_N^k;\boldsymbol{a}^k) = B_2(\mu_T^k\lambda_N^k)\,, \tag{3.32}$$

spatial anisotropic Coulomb friction with elliptical force reservoir can again be stated as normal cone inclusion to the ball-like set (3.22). It should be noted that the friction forces $\boldsymbol{\lambda}_F^k$ are related to the true tangent friction forces $\boldsymbol{\lambda}_T^k$ via $\boldsymbol{\lambda}_T^k = \boldsymbol{A}^k\boldsymbol{\lambda}_F^k$ and can be determined a posteriori to a numerical simulation.

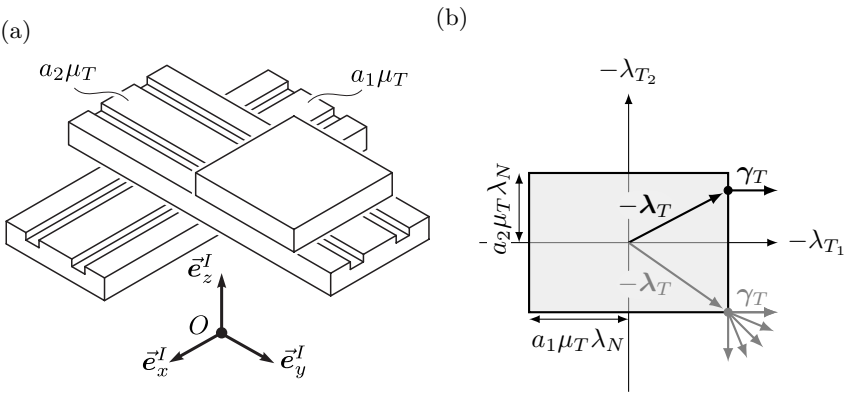

Figure 3.5: (a) Graphical illustration of an x-y table. (b) Anisotropic Coulomb friction with rectangular force reservoir.

Anisotropic Coulomb Friction with Rectangular Force Reservoir

A much simpler description of anisotropic Coulomb friction is obtained by assuming a rectangular force reservoir. Again, the relative friction velocity is chosen as $\boldsymbol{\gamma}_F^k = \boldsymbol{\gamma}_T^k$. For $\boldsymbol{a}^k = (a_1^k, a_2^k) \in \mathbb{R}^2$ with $a_i^k \in (0, 1]$ and $\mu_T^k > 0$, the set of admissible friction forces is given by the rectangular force reservoir

$$\begin{aligned} C_F^k(\lambda_N^k) &= \{\boldsymbol{\lambda}_F^k \in \mathbb{R}^2 \,|\, |\lambda_{F_1}^k| \leq a_1^k \mu_T^k \lambda_N^k \,,\, |\lambda_{F_2}^k| \leq a_2^k \mu_T^k \lambda_N^k\} \\ &= B_1(a_1^k \mu_T^k \lambda_N^k) \times B_1(a_2^k \mu_T^k \lambda_N^k)\,, \end{aligned} \tag{3.33}$$

with $\boldsymbol{\lambda}_F^k = \boldsymbol{\lambda}_T^k$. Such a force reservoir is visualized in Figure 3.5 (b) and results in a friction law where stick and slip along the two tangent directions are independent of each other. A physical model exhibiting such a behavior is the frictional x-y table shown in Figure 3.5 (a). Each of the two slide bearings, with friction coefficient $a_1\mu_T$ and $a_2\mu_T$, respectively, can move independently of the other, see [177]. Clearly, such a behavior is only a crude approximation of the elliptical anisotropic Coulomb friction model presented before. A higher resolution can be obtained by introducing a polyhedral approximation of the (anisotropic) set of admissible friction forces (3.27). Several authors have applied this approach, see e.g. [11, 107, 167]. Depending on the chosen time-discretization method, the resulting equations end up with an (non-)linear complementarity problem. This has different appealing properties, but independent of the number of chosen

facets, Coulomb friction with elliptical/circular force reservoir is only modeled approximately and isotropy is lost. Hence, this approach will not be investigated further here.

Pivoting and Rolling Friction

Many physical processes require the description of even more elaborated friction phenomena [14, 108, 111, 155]. A commonly used extension incorporates the effects of pivoting and rolling friction, see [4, 59, 111, 112, 114, 155, 177]. Thereto, force laws related to the relative pivoting velocity γ_P^k and two relative rolling velocities $\boldsymbol{\gamma}_R^k$, already introduced in (3.18) and (3.19), have to be formulated. Since these force laws are assumed to be decoupled, three independent normal cone inclusions can be stated. They are given as

$$\gamma_P^k(t, \boldsymbol{q}, \boldsymbol{u}) \in \mathcal{N}_{B_1(\mu_P^k \lambda_N^k)}(-\lambda_P^k) \tag{3.34}$$

and

$$\gamma_{R_i}^k(t, \boldsymbol{q}, \boldsymbol{u}) \in \mathcal{N}_{B_1(\mu_{R_i}^k \lambda_N^k)}(-\lambda_{R_i}^k) \tag{3.35}$$

with $i = 1, 2$. Therein, $\mu_P^k, \mu_{R_1}^k, \mu_{R_2}^k > 0$ denote the coefficients of friction for pivoting and both rolling frictions, respectively. Typically chosen values for μ_P and μ_{R_1}, μ_{R_2} are discussed in [51, 88, 161] by means of analytical models or performed experiments. Clearly, the introduced force laws share their dependence on the normal force λ_N^k that uniformly scales the respective force reservoirs.

Coulomb–Contensou Friction

Sometimes the isolated description of pivoting friction is not sufficient to adequately describe the experimentally observed friction phenomena. Especially, this is the case for friction induced inversion phenomena [14, 108, 154, 155]. Their description requires a coupling of sliding and pivoting friction, the so-called Coulomb–Contensou friction [112]. As it is pointed out in [126], a fairly good approximation to this is given in terms of an ellipsoidal force reservoir. Thereto, for $\boldsymbol{a}^k \in \mathbb{R}^3$, with $a_i^k \in (0, 1]$, the set of admissible friction forces is given by

$$C_F^k(\lambda_N^k) = E_3(\mu_T^k \lambda_N^k; \boldsymbol{a}^k)\,. \tag{3.36}$$

Often, the value of a_3^k is related to a radius of assumed circular contact area R^k via $a_3^k = \frac{3}{16}\pi R^k$, see [29, 41, 126, 155]. For isotropic sliding friction $a_1^k = a_2^k = 1$. The corresponding force law augments the relative contact

tangent velocity $\boldsymbol{\gamma}_T^k$ from (3.17) by a relative pivoting velocity γ_P^k, see (3.18). Moreover, it appends the frictional torque λ_P^k to $\boldsymbol{\lambda}_T^k$. Once again, the set of admissible friction force (3.36) is an ellipsoid. As in the case of anisotropic Coulomb friction with elliptical force reservoir, introducing the mapping

$$A^k \colon \mathbb{R}^3 \to \mathbb{R}^3\,, \quad \boldsymbol{x} \mapsto \boldsymbol{A}^k \boldsymbol{x}\,, \quad \text{with} \quad \boldsymbol{A}^k = \operatorname{diag}(\boldsymbol{a}^k)\,, \tag{3.37}$$

allows to reformulate the force law as normal cone inclusion to the ball-like set of admissible friction forces (3.22), with

$$\boldsymbol{\gamma}_F^k(t,\boldsymbol{q},\boldsymbol{u}) = (\boldsymbol{A}^k)^{\mathrm{T}} \begin{pmatrix} \boldsymbol{\gamma}_T^k(t,\boldsymbol{q},\boldsymbol{u}) \\ \gamma_P^k(t,\boldsymbol{q},\boldsymbol{u}) \end{pmatrix}\,, \quad \boldsymbol{\lambda}_F^k = (\boldsymbol{A}^k)^{-1} \begin{pmatrix} \boldsymbol{\lambda}_T^k \\ \lambda_P^k \end{pmatrix} \tag{3.38}$$

and

$$C_F^k(\lambda_N^k) = (A^k)^{-1} E_3(\mu_T^k \lambda_N^k; \boldsymbol{a}^k) = B_3(\mu_T^k \lambda_N^k)\,. \tag{3.39}$$

As already discussed, the true frictional forces can be computed a posteriori via $(\boldsymbol{\lambda}_T^k,\ \lambda_P^k) = \boldsymbol{A}^k \boldsymbol{\lambda}_F^k$.

Unified Description of Frictional Contact

In conclusion, the various friction formulations introduced above can be formulated (approximated) using ellipsoidal force reservoirs and subsequently be reformulated as normal cone inclusions to ball-like sets of admissible friction forces. Consequently, a subsequent numerical treatment only investigates such force laws. It is therefore assumed for each contact k that the force laws describing the frictional contact have the form

$$\boldsymbol{\psi}^k(t,\boldsymbol{q},\boldsymbol{u}) \in \mathcal{K}^k(\boldsymbol{\lambda})\,, \quad k = 1,\dots,n_N\,, \tag{3.40}$$

with

$$\boldsymbol{\psi}^k(t,\boldsymbol{q},\boldsymbol{u}) = \begin{pmatrix} g_N^k(t,\boldsymbol{q}) \\ \boldsymbol{\gamma}_{F_1}^k(t,\boldsymbol{q},\boldsymbol{u}) \\ \vdots \\ \boldsymbol{\gamma}_{F_{f_k}}^k(t,\boldsymbol{q},\boldsymbol{u}) \end{pmatrix} \tag{3.41}$$

and

$$\mathcal{K}^k(\boldsymbol{\lambda}) = \mathcal{N}_{\mathbb{R}_0^-}(-\lambda_N^k) \times \mathcal{N}_{C_{F_1}^k(\lambda_N^k)}(-\boldsymbol{\lambda}_{F_1}^k) \times \cdots \times \mathcal{N}_{C_{F_{f_k}}^k(\lambda_N^k)}(-\boldsymbol{\lambda}_{F_{f_k}}^k)\,, \tag{3.42}$$

where $C_{F_{f_k}}^k(\lambda_N^k) = B_{n_{f_k}}(\mu_T^{f_k} \lambda_N^k)$. Therein, for every contact k, Signorini's law (3.10) is combined with f^k set-valued friction laws as outlined above. To keep the subsequent investigations simple, it is assumed that $f^k = 1$ for $k = 1,\dots,n_N$, i.e., every contact only includes a single friction law. Thereby, cumbersome multi indices are omitted and the notation is kept clean. Besides the cumbersome bookkeeping such an extension is trivial.

3.6 Impact Equation and Impact Laws

The equality of measures (3.2) can be integrated over the singleton $\{t\}$, leading to the impact equation [72, 73, 114, 142]

$$\boldsymbol{M}(t,\boldsymbol{q})\left(\boldsymbol{u}^{+}-\boldsymbol{u}^{-}\right)=\boldsymbol{W}(t,\boldsymbol{q})\,\boldsymbol{\Lambda}\,, \tag{3.43}$$

where the abbreviation $\boldsymbol{\Lambda}=\boldsymbol{\pi}^{+}-\boldsymbol{\pi}^{-}$ is introduced in accordance with (3.3). Clearly, the already stated contact and friction laws defining $\boldsymbol{\lambda}$ have to be complemented by corresponding impact laws, relating the impulsive change of velocity $\boldsymbol{u}^{+}-\boldsymbol{u}^{-}$ with the impulsive constraint forces $\boldsymbol{\Lambda}$. Therefore, each contact k is equipped with a restitution coefficient $0\le e_N^k\le 1$. For active contacts, i.e., $k\in A(t,\boldsymbol{q})$, the classical Newtonian impact law [142] assumes the post impact velocity to be given by $\dot{g}_N^k(t,\boldsymbol{q},\boldsymbol{u}^{+})=-e_N^k\,\dot{g}_N^k(t,\boldsymbol{q},\boldsymbol{u}^{-})$. By introducing the kinematic quantity [73, 114, 142]

$$\xi_N^k(t,\boldsymbol{q},\boldsymbol{u}^{-},\boldsymbol{u}^{+})=\dot{g}_N^k(t,\boldsymbol{q},\boldsymbol{u}^{+})+e_N^k\,\dot{g}_N^k(t,\boldsymbol{q},\boldsymbol{u}^{-})\,, \tag{3.44}$$

a generalized Newton-type impact law [73, 114, 142] can equivalently be stated as normal cone inclusion to the set $\mathbb{R}_0^-$ or as inequality complementarity condition, i.e.,

$$\begin{aligned}
&\xi_N^k(t,\boldsymbol{q},\boldsymbol{u}^{-},\boldsymbol{u}^{+})\in\mathcal{N}_{\mathbb{R}_0^-}(-\Lambda_N^k)\\
&\qquad\qquad\Updownarrow\\
&\xi_N^k(t,\boldsymbol{q},\boldsymbol{u}^{-},\boldsymbol{u}^{+})\ge 0\,,\quad \Lambda_N^k\ge 0\,,\quad \xi_N^k(t,\boldsymbol{q},\boldsymbol{u}^{-},\boldsymbol{u}^{+})\,\Lambda_N^k=0\,.
\end{aligned} \tag{3.45}$$

For a strictly positive impulsive force $\Lambda_N^k>0$, this imposes the classical Newtonian impact law $\xi_N^k(t,\boldsymbol{q},\boldsymbol{u}^{-},\boldsymbol{u}^{+})=0$. Moreover, it encompasses superfluous contact for which $\Lambda_N^k=0$ and $\xi_N^k\ge 0$. The second case occurs if a closed contact does not at all participate in the impact ($\Lambda_N^k=0$), and could therefore be removed from the system, without changing the post-impact results, see [74].

Similar to the normal direction (3.44), the kinematic quantity

$$\boldsymbol{\xi}_F^k(t,\boldsymbol{q},\boldsymbol{u}^{-},\boldsymbol{u}^{+})=\boldsymbol{\gamma}_F^k(t,\boldsymbol{q},\boldsymbol{u}^{+})+e_F^k\,\boldsymbol{\gamma}_F^k(t,\boldsymbol{q},\boldsymbol{u}^{-}) \tag{3.46}$$

can be introduced, together with a tangential restitution coefficient $0\le e_F^k\le 1$, see [73]. This allows to combine the friction law (3.20) with a Newton-type impact law in tangential direction by the normal cone inclusion

$$\boldsymbol{\xi}_F^k(t,\boldsymbol{q},\boldsymbol{u}^{-},\boldsymbol{u}^{+})\in\mathcal{N}_{C_F^k(\Lambda_N^k)}(-\boldsymbol{\Lambda}_F^k)\,. \tag{3.47}$$

More details about the introduced contact model and impact laws can be found in [72, 73, 74, 114, 142, 178].

3.7 Consistent Initial Conditions

In the following the determination of so-called *consistent initial conditions* is discussed. Assume a given initial time t_0, initial position $\boldsymbol{q}_0 = \boldsymbol{q}(t_0)$ and initial velocity $\boldsymbol{u}_0 = \boldsymbol{u}^+(t_0)$. The following presentation is restricted to the case where $\boldsymbol{q}_0$ and $\boldsymbol{u}_0$ are already feasible, i.e., they satisfy the bilateral constraint conditions $\boldsymbol{g}(t_0, \boldsymbol{q}_0) = \boldsymbol{0}_{n_g}$ and $\boldsymbol{\gamma}(t_0, \boldsymbol{q}_0, \boldsymbol{u}_0) = \boldsymbol{0}_{n_\gamma}$. Moreover, the positions have to satisfy the unilateral constraint conditions $g_N^k(t_0, \boldsymbol{q}_0) \geq 0$ for all $k = 1, 2, \ldots, n$. Let $A_0 = A(t_0, \boldsymbol{q}_0) = \left\{ k = 1, \ldots, n_N \,\middle|\, g_N^k(t_0, \boldsymbol{q}_0) \leq 0 \right\}$ denote the set of initially active contacts on position level. Feasible velocities $\boldsymbol{u}_0$ have to satisfy the unilateral constraint conditions on velocity level, i.e., $\dot{g}_N^k(t_0, \boldsymbol{q}_0, \boldsymbol{u}_0) \geq 0$ for all $k \in A_0$.

The initial time derivative of the generalized positions are obtained by simple evaluation of

$$\dot{\boldsymbol{q}}_0 = \dot{\boldsymbol{q}}(t_0, \boldsymbol{q}_0, \boldsymbol{u}_0)\,. \tag{3.48}$$

Further, the goal of this section is to compute the initial accelerations $\dot{\boldsymbol{u}}_0 \coloneqq \dot{\boldsymbol{u}}(t_0)$ as well as bilateral and unilateral constraint forces (Lagrange multipliers) $\boldsymbol{\lambda}_0 \coloneqq \boldsymbol{\lambda}(t_0)$ that are consistent, i.e., satisfy the equations of motion on acceleration level. Hence, the bilateral constraint conditions $\boldsymbol{0}_{n_g} = \ddot{\boldsymbol{g}}(t, \boldsymbol{q}, \boldsymbol{u}, \dot{\boldsymbol{u}}) = \boldsymbol{W}_g(t, \boldsymbol{q})^{\mathrm{T}} \dot{\boldsymbol{u}} + \boldsymbol{\zeta}_g(t, \boldsymbol{q}, \boldsymbol{u})$ and $\boldsymbol{0}_{n_\gamma} = \dot{\boldsymbol{\gamma}}(t, \boldsymbol{q}, \boldsymbol{u}, \dot{\boldsymbol{u}}) = \boldsymbol{W}_\gamma(t, \boldsymbol{q})^{\mathrm{T}} \dot{\boldsymbol{u}} + \boldsymbol{\zeta}_\gamma(t, \boldsymbol{q}, \boldsymbol{u})$ have to be enforced on acceleration level. The unilateral constraint conditions require more involved considerations, which are outlined subsequently. More information can be found in Section 10.4 and 10.5 of Glocker [72].

Let $B(t, \boldsymbol{q}, \boldsymbol{u}) = \left\{ k \in A(t, \boldsymbol{q}) \,\middle|\, \dot{g}_N^k(t, \boldsymbol{q}, \boldsymbol{u}) \leq 0 \right\}$ denote the set of active contacts on velocity level. Hence, contacts that are initially active on velocity level[5] are collected in $B_0 = B(t_0, \boldsymbol{q}_0, \boldsymbol{u}_0)$. For those contacts $k \in B_0$, Signorini's law has to be enforced on acceleration level [72, 141], i.e.,

$$\ddot{g}_N^k(t_0, \boldsymbol{q}_0, \boldsymbol{u}_0, \dot{\boldsymbol{u}}_0) \in \mathcal{N}_{\mathbb{R}_0^-}(-\lambda_{N,0}^k)\,, \tag{3.49}$$

which relates the initial accelerations $\dot{\boldsymbol{u}}_0$ with the initial contact forces $\boldsymbol{\lambda}_{N,0}$. For all other contacts $k \notin B_0$, the initial contact forces are zero.

Following Section 5.1 of [141], the set of frictional contacts that are possibly sticking in the initial configuration[6] is $C_0 = C(t_0, \boldsymbol{q}_0, \boldsymbol{u}_0)$ with $C(t, \boldsymbol{q}, \boldsymbol{u}) = \left\{ k \in A(t, \boldsymbol{q}, \boldsymbol{u}) \,\middle|\, \|\boldsymbol{\gamma}_F^k(t, \boldsymbol{q}, \boldsymbol{u})\| = 0 \right\}$. For those contacts

[5] Due to possible round-off errors, it may be beneficial to also consider contacts as active on velocity level, i.e, $k \in B_0$, if $k \in A_0$ and $|\dot{g}_N^k(t, \boldsymbol{q}, \boldsymbol{u})| < \varepsilon \ll 1$.

[6] Again, it may be computationally beneficial to also consider contacts as possibly sicking, i.e., $k \in C_0$, using the modified set $C(t, \boldsymbol{q}, \boldsymbol{u}) = \{k = 1, \ldots, n_N \mid \|\boldsymbol{\gamma}_F^k(t, \boldsymbol{q}, \boldsymbol{u})\| < \varepsilon \ll 1\}$.

$k \in C_0$, the set-valued friction law has to be enforced on acceleration level. Specifically,

$$\dot{\boldsymbol{\gamma}}_F^k(t_0, \boldsymbol{q}_0, \boldsymbol{u}_0, \dot{\boldsymbol{u}}_0) \in \mathcal{N}_{C_F^k(\lambda_{N,0}^k)}(-\boldsymbol{\lambda}_{F,0}^k) \tag{3.50}$$

relates the initial accelerations $\dot{\boldsymbol{u}}_0$ with the initial friction forces $\boldsymbol{\lambda}_{F,0}$. For those contacts that are active on position and velocity level, i.e., $k \in B_0$ but have a non-vanishing friction velocity with $\|\boldsymbol{\gamma}_F^k(t_0, \boldsymbol{q}_0, \boldsymbol{u}_0)\| > 0$, i.e., $k \notin C_0$, the set-valued friction force has to be enforced on velocity level. Since this monograph is restricted on set-valued friction forces that can be expressed by ball-like force reservoirs (3.22), the initial friction forces $\boldsymbol{\lambda}_{F,0}$ can be computed as

$$\boldsymbol{\lambda}_{F,0}^k = -\mu^k \lambda_{N,0}^k \frac{\boldsymbol{\gamma}_{F,0}^k}{\|\boldsymbol{\gamma}_{F,0}^k\|}\,, \tag{3.51}$$

where $\boldsymbol{\gamma}_{F,0}^k = \boldsymbol{\gamma}_F^k(t_0, \boldsymbol{q}_0, \boldsymbol{u}_0)$ was used. Similar to the normal direction, for all contacts $k \notin B_0$, the initial friction forces are zero.

Let $\boldsymbol{x}_0 = (\dot{\boldsymbol{u}}_0, \boldsymbol{\lambda}_{g,0}, \boldsymbol{\lambda}_{\gamma,0}) \in \mathbb{R}^{n_u+n_g+n_\gamma}$ and $\boldsymbol{y}_0 = (\boldsymbol{\lambda}_{N,0}, \boldsymbol{\lambda}_{F,0}) \in \mathbb{R}^{n_N+n_F}$ denote the tuples of unknowns. For the sake of compactness it is subsequently written $\boldsymbol{M}_0 = \boldsymbol{M}(t_0, \boldsymbol{q}_0)$ and $\boldsymbol{h}_0 = \boldsymbol{h}(t_0, \boldsymbol{q}_0, \boldsymbol{u}_0)$, a notational abbreviation that is analogously applied for all other appearing functions. The consistent initial conditions are given by solving the zeros of the linear system

$$\begin{aligned}\boldsymbol{\varphi}_0(\boldsymbol{x}_0, \boldsymbol{y}_0) = &\begin{pmatrix} \boldsymbol{M}_0 & -\boldsymbol{W}_{g,0} & -\boldsymbol{W}_{\gamma,0} \\ -\boldsymbol{W}_{g,0}^{\mathrm{T}} & \boldsymbol{0}_{n_g \times n_g} & \boldsymbol{0}_{n_g \times n_\gamma} \\ -\boldsymbol{W}_{\gamma,0}^{\mathrm{T}} & \boldsymbol{0}_{n_\gamma \times n_g} & \boldsymbol{0}_{n_\gamma \times n_\gamma} \end{pmatrix} \begin{pmatrix} \dot{\boldsymbol{u}}_0 \\ \boldsymbol{\lambda}_{g,0} \\ \boldsymbol{\lambda}_{\gamma,0} \end{pmatrix} \\ &- \begin{pmatrix} \boldsymbol{h}_0 + \boldsymbol{W}_{N,0}\boldsymbol{\lambda}_{N,0} + \boldsymbol{W}_{F,0}\boldsymbol{\lambda}_{F,0} \\ \boldsymbol{\zeta}_{g,0} \\ \boldsymbol{\zeta}_{\gamma,0} \end{pmatrix}\,,\end{aligned} \tag{3.52}$$

such that for $k = 1, \ldots, n_N$ the following conditions are satisfied:

$$\begin{cases} \ddot{g}_N^k(t_0, \boldsymbol{q}_0, \boldsymbol{u}_0, \dot{\boldsymbol{u}}_0) \in \mathcal{N}_{\mathbb{R}_0^-}(-\lambda_{N,0}^k) & \text{if } k \in B_0\,, \\ \lambda_{N,0}^k = 0 & \text{otherwise}\,, \end{cases} \tag{3.53a}$$

$$\begin{cases} \dot{\boldsymbol{\gamma}}_F^k(t_0, \boldsymbol{q}_0, \boldsymbol{u}_0, \dot{\boldsymbol{u}}_0) \in \mathcal{N}_{C_F^k(\lambda_{N,0}^k)}(-\boldsymbol{\lambda}_{F,0}^k) & k \in B_0 \wedge \|\boldsymbol{\gamma}_{F,0}^k\| = 0\,, \\ \boldsymbol{\lambda}_{F,0}^k = -\mu^k \lambda_{N,0}^k \frac{\boldsymbol{\gamma}_{F,0}^k}{\|\boldsymbol{\gamma}_{F,0}^k\|} & k \in B_0 \wedge \|\boldsymbol{\gamma}_{F,0}^k\| > 0\,, \\ \boldsymbol{\lambda}_{F,0}^k = \boldsymbol{0}_{n_{F_k}} & \text{otherwise}\,. \end{cases} \tag{3.53b}$$

Hence, consistent initial conditions are obtained by solving the linear system $\boldsymbol{\varphi}_0(\boldsymbol{x}_0, \boldsymbol{y}_0) = \boldsymbol{0}_{n_u+n_g+n_\gamma}$ subject to the normal cone inclusions

$$\boldsymbol{\psi}^k(\boldsymbol{x}_0) \in \mathcal{K}^k(\boldsymbol{y}_0)\,, \quad k = 1, \ldots, n_N\,, \tag{3.54}$$

where the left hand side of the inclusion is given by

$$\boldsymbol{\psi}^k(\boldsymbol{x}_0) = \begin{pmatrix} \ddot{g}_N^k(t_0, \boldsymbol{q}_0, \boldsymbol{u}_0, \dot{\boldsymbol{u}}_0) \\ \dot{\boldsymbol{\gamma}}_F^k(t_0, \boldsymbol{q}_0, \boldsymbol{u}_0, \dot{\boldsymbol{u}}_0) \end{pmatrix} . \tag{3.55}$$

Moreover, the set can be decomposed as

$$\mathcal{K}^k(\boldsymbol{y}_0) = \mathcal{K}_N^k(\boldsymbol{y}_0) \times \mathcal{K}_F^k(\boldsymbol{y}_0) , \tag{3.56}$$

with

$$\mathcal{K}_N^k(\boldsymbol{y}_0) = \begin{cases} \mathcal{N}_{\mathbb{R}_0^-}(-\lambda_{N,0}^k) & \text{if } k \in B_0 , \\ \mathcal{N}_{\{0\}}(-\lambda_{N,0}^k) & \text{otherwise} , \end{cases} \tag{3.57}$$

and

$$\mathcal{K}_F^k(\boldsymbol{y}_0) = \begin{cases} \mathcal{N}_{C_F^k(\lambda_{N,0}^k)}(-\boldsymbol{\lambda}_{F,0}^k) & \text{if } k \in B_0 \wedge \|\boldsymbol{\gamma}_{F,0}^k\| = 0 , \\ \mathcal{N}_{\{\mathbf{0}_{n_{F_k}}\}}\left(\boldsymbol{\lambda}_{F,0}^k - \mu^k \lambda_{N,0}^k \frac{\boldsymbol{\gamma}_{F,0}^k}{\|\boldsymbol{\gamma}_{F,0}^k\|}\right) & \text{if } k \in B_0 \wedge \|\boldsymbol{\gamma}_{F,0}^k\| > 0 , \\ \mathcal{N}_{\{\mathbf{0}_{n_{F_k}}\}}(-\boldsymbol{\lambda}_{F,0}^k) & \text{otherwise} . \end{cases} \tag{3.58}$$

Therein, it is used that for $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ it holds that $\boldsymbol{y} \in \mathcal{N}_{\{\mathbf{0}_n\}}(\boldsymbol{x}) = \mathbb{R}^n \Leftrightarrow \boldsymbol{x} = \mathbf{0}_n$ and $\boldsymbol{y} \in \mathbb{R}^n$. Consequently, the normal cone inclusion encoding $\boldsymbol{x} = \mathbf{0}_n$ is only used for a unified description of the inclusions (3.53a) and (3.53b). A detailed discussion of how such class of problems can be solved is given in Chapter 6.

Chapter 4

Nonsmooth Projected Stiffly Accurate Runge–Kutta Methods

> Und bist du nicht willig, so brauch ich Gewalt.
>
> —*Johann Wolfgang von Goethe, Der Erlkönig*

In this chapter the concept of nonsmooth projected stiffly accurate Runge–Kutta methods is introduced. To keep the presentation self-contained and didactic, the first section introduces basic concepts and notation of Runge–Kutta methods applied to nonlinear ordinary differential equations. Based on that, it is outlined how these methods can be applied to differential algebraic equations with possible differentiation index 2. It will turn out that an order reduction takes place that can be alleviated by a subsequent velocity projection. The resulting methods are called projected Runge–Kutta methods. Subsequently, their extension to nonsmooth mechanical systems is investigated. The naive extension to measure differential inclusions with mechanical structure is shown to be flawed. Moreover, it turns out that it is of crucial importance to add again a nonsmooth equivalent to the velocity projection applied for differential algebraic equations. Such a projection can be interpreted as the discrete counterpart of the impact equation together with corresponding impact laws. In the nonsmooth setting, the velocity projection is mandatory to force the velocities to satisfy the possible set-valued constraint equations on velocity level[1].

[1]Since the unprojected method does not respect this requirement, the quote below the title can be interpreted by the need of a certain degree of insistence to make the method obey the underlying physics.

4.1 Stiffly Accurate Runge–Kutta Methods

To introduce the basic properties of implicit Runge–Kutta methods when applied to ordinary differential equations, this section focuses on mechanical systems of the form $\boldsymbol{M}(t,\boldsymbol{q})\dot{\boldsymbol{u}} = \boldsymbol{h}(t,\boldsymbol{q},\boldsymbol{u})$, i.e., systems without bilateral or unilateral constraints, as presented in Chapter 3. Consequently, the kinematic relation (3.1) and the equality of measures (3.2) can be recast as

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t,\boldsymbol{x}(t)) \quad \text{with} \quad \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \in \mathbb{R}^m\,. \tag{4.1}$$

Therein, $\boldsymbol{x} = (\boldsymbol{q},\boldsymbol{u}) \in \mathbb{R}^m$ denotes the tuple of unknowns with $m = n_q + n_u$ and the function $\boldsymbol{f}$ is given as

$$\boldsymbol{f}(t,\boldsymbol{x}) = \begin{pmatrix} \boldsymbol{B}(t,\boldsymbol{q})\boldsymbol{u} + \boldsymbol{\beta}(t,\boldsymbol{q}) \\ \boldsymbol{M}(t,\boldsymbol{q})^{-1}\boldsymbol{h}(t,\boldsymbol{q},\boldsymbol{u}) \end{pmatrix}\,. \tag{4.2}$$

Implicit Runge–Kutta methods are well established methods for the solution of possibly stiff differential equations of the form (4.1), see [82].

Let $h \in \mathbb{R}$ denote the step-size of the method. Assume a step from $t_n \mapsto t_{n+1} = t_n + h$ that computes an approximation $\boldsymbol{x}_{n+1} \approx \boldsymbol{x}(t_{n+1}) = \boldsymbol{x}(t_n) + \int_{t_n}^{t_{n+1}} \dot{\boldsymbol{x}}(t)\,\mathrm{d}t$. According to [82, 84], application of an s-stage Runge–Kutta method to (4.1) reads

$$\boldsymbol{X}_i = \boldsymbol{x}_n + h\sum\nolimits_{j=1}^{s} a_{ij}\boldsymbol{f}(t_n + c_i h, \boldsymbol{X}_i)\,, \quad i = 1,\dots,s \tag{4.3a}$$

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + h\sum\nolimits_{i=1}^{s} b_i\boldsymbol{f}(t_n + c_i h, \boldsymbol{X}_i)\,. \tag{4.3b}$$

The coefficients a_{ij}, b_i and c_i entirely characterize the specific method and are discussed later in this section. Let the coefficients a_{ij} be collected in the matrix $\boldsymbol{A} \in \mathbb{R}^{s\times s}$. By denoting $\tau_i = t_n + c_i h$ and introducing the solution tuple $\boldsymbol{X} = (\boldsymbol{X}_1,\dots,\boldsymbol{X}_s) \in \mathbb{R}^{sm}$, each step of the proposed Runge–Kutta method requires the solution of the $s\cdot m$ nonlinear equations[2] $\boldsymbol{\varphi}(\boldsymbol{X}) = \boldsymbol{0}_{s\cdot m}$, where

$$\boldsymbol{\varphi}(\boldsymbol{X}) = \boldsymbol{X} - (\boldsymbol{1}_s \otimes \boldsymbol{x}_n) - h(\boldsymbol{A}\otimes\boldsymbol{1}_{m\times m})\begin{pmatrix} \boldsymbol{f}\big(\tau_1, (\boldsymbol{e}_1^{\mathrm{T}}\otimes\boldsymbol{1}_{m\times m})\boldsymbol{X}\big) \\ \vdots \\ \boldsymbol{f}\big(\tau_s, (\boldsymbol{e}_s^{\mathrm{T}}\otimes\boldsymbol{1}_{m\times m})\boldsymbol{X}\big) \end{pmatrix}\,. \tag{4.4}$$

[2] It is noteworthy to mention that in the case of a non-constant mass matrix $\boldsymbol{M}(t,\boldsymbol{q})$ that depends on the generalized coordinates $\boldsymbol{q}$, the solution of the time discretized system (4.4) can be significantly simplified (differentiation of $\boldsymbol{M}(t,\boldsymbol{q})^{-1}$ with respect to $\boldsymbol{q}$ is not required), by application of Runge–Kutta methods to *implicit differential equations* as originally proposed by Ascher and Petzold [17] and outlined in great detail in Section 5.3.5 of Fabien [61]. The same argument applies for all subsequent extensions of this method.

Therein, the Kronecker product $\otimes$ was used as defined in (2.41), $\mathbf{1}_s = (1, \dots, 1) \in \mathbb{R}^s$ denotes the tuple containing only ones and $\boldsymbol{e}_i = (0, \dots, 0, 1, 0, \dots, 0) \in \mathbb{R}^s$ is the tuple with all components being zero, except the ith, which is one. When the nonlinear system (4.4) is solved successfully, the future state $\boldsymbol{x}_{n+1}$ is computed by (4.3b) using the already computed evaluations of the function $\boldsymbol{f}$. With the notation from above and $\boldsymbol{F}(\boldsymbol{X}) = \big(\boldsymbol{f}(\tau_1, \boldsymbol{X}_1), \dots, \boldsymbol{f}(\tau_s, \boldsymbol{X}_s)\big) \in \mathbb{R}^{sm}$, this reads

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + h(\boldsymbol{b}^{\mathrm{T}} \otimes \mathbf{1}_{m\times m})\boldsymbol{F}(\boldsymbol{X})\,. \tag{4.5}$$

Since this method solves for the state variables $\boldsymbol{X}$, it is subsequently called the *state formulation*.

An alternative formulation of implicit Runge–Kutta methods was already proposed in the early works of John Butcher [34] and later in [17, 27, 84]. It is given by

$$\dot{\boldsymbol{X}}_i = \boldsymbol{f}\left(t_n + c_i h, \boldsymbol{x}_n + h\sum\nolimits_{j=1}^{s} a_{ij}\dot{\boldsymbol{X}}_j\right)\,, \quad i = 1, \dots, s \tag{4.6a}$$

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + h\sum\nolimits_{i=1}^{s} b_i\dot{\boldsymbol{X}}_i\,, \tag{4.6b}$$

where the stage values can be identified as $\boldsymbol{X}_i = \boldsymbol{x}_n + h\sum_{j=1}^{s} a_{ij}\dot{\boldsymbol{X}}_j$. Introducing another solution tuple $\dot{\boldsymbol{X}} = (\dot{\boldsymbol{X}}_1, \dots, \dot{\boldsymbol{X}}_s) \in \mathbb{R}^{sm}$, the states can be computed via $\boldsymbol{X} = (\mathbf{1}_s \otimes \boldsymbol{x}_n) + h(\boldsymbol{A} \otimes \mathbf{1}_{m\times m})\dot{\boldsymbol{X}}$. Consequently, each Runge–Kutta step requires the solution of $s \cdot m$ nonlinear equations

$$\mathbf{0}_{s\cdot m} = \boldsymbol{\varphi}(\dot{\boldsymbol{X}}) = \dot{\boldsymbol{X}} - \begin{pmatrix} \boldsymbol{f}\big(\tau_1, \boldsymbol{x}_n + h(\boldsymbol{a}_1^{\mathrm{T}} \otimes \mathbf{1}_{m\times m})\dot{\boldsymbol{X}}\big) \\ \vdots \\ \boldsymbol{f}\big(\tau_s, \boldsymbol{x}_n + h(\boldsymbol{a}_s^{\mathrm{T}} \otimes \mathbf{1}_{m\times m})\dot{\boldsymbol{X}}\big) \end{pmatrix}\,, \tag{4.7}$$

where the tuple $\boldsymbol{a}_i \in \mathbb{R}^s$ denotes the ith row of $\boldsymbol{A}$. Since this second method solves for the state derivatives $\dot{\boldsymbol{X}}$ it is subsequently called the *derivative formulation*. Similar to (4.5), the future state is computed as

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + h(\boldsymbol{b}^{\mathrm{T}} \otimes \mathbf{1}_{m\times m})\dot{\boldsymbol{X}}\,. \tag{4.8}$$

Both approaches only differ in the choice of the primary unknowns. The former method solves for the stage variables $\boldsymbol{X}_i$, while the latter chooses the stage derivatives $\dot{\boldsymbol{X}}_i$ as the unknown quantities. As it will turn out later, the second approach fits better to the discretization of nonsmooth mechanical systems and is subsequently applied. Typically, the arising system of $s \cdot m$ nonlinear algebraic equations is solved by simplified Newton

iterations, where a single evaluation of the Jacobian matrix $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}$ is used as approximation for all stages $i = 1, \ldots, s$. Moreover, the special structure of the nonlinear systems (4.4) and (4.7) can be exploited to considerably reduce the required numerical work at the price of dealing with complex linear systems, see e.g. [24, 37, 61, 81].

Definition 4.1. A Runge–Kutta method (4.3) or (4.6) has *order* p if for sufficiently smooth problems (4.1),

$$\|\boldsymbol{x}(t_n + h) - \boldsymbol{x}_{n+1}\| \leq K h^{p+1}\,, \quad K \in \mathbb{R}\,, \tag{4.9}$$

i.e., if the Taylor series of the exact solution $\boldsymbol{x}(t_n + h)$ coincides up to (and including) the term h^p with $\boldsymbol{x}_{n+1}$. The constant K depends on bounds of the vector field $\boldsymbol{f}$, on its Lipschitz constant and on the total integration domain, but is independent on the step size h. Using the big $\mathcal{O}$ notation a pth-order accurate method is notated as

$$\|\boldsymbol{x}(t_n + h) - \boldsymbol{x}_{n+1}\| = \mathcal{O}(h^{p+1})\,. \tag{4.10}$$

An s-stage implicit Runge–Kutta method is commonly defined in terms of a so-called *Butcher tableau*

$$\begin{array}{c|c} \boldsymbol{c} & \boldsymbol{A} \\ \hline & \boldsymbol{b}^{\mathrm{T}} \end{array} = \begin{array}{c|cccccc} c_1 & a_{11} & a_{12} & \ldots & a_{1,s-1} & a_{1s} \\ c_2 & a_{21} & a_{22} & \ldots & a_{2,s-1} & a_{2s} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ c_{s-1} & a_{s-1,1} & a_{s-1,2} & \ldots & a_{s-1,s-1} & a_{s-1,s} \\ c_s & a_{s,1} & a_{s,2} & \ldots & a_{s,s-1} & a_{s,s} \\ \hline & b_1 & b_2 & \ldots & b_{s-1} & b_s \end{array}\,, \tag{4.11}$$

originating from John Butcher [35]. In the tableau (4.11), the tuple $\boldsymbol{c} \in \mathbb{R}^s$ contains the nodes of the underlying quadrature formula. For the implicit methods discussed in this monograph they satisfy

$$c_i = \sum\nolimits_{j=1}^{s} a_{ij} \quad \text{for} \quad i = 1, \ldots, s\,, \tag{4.12}$$

which is imposed by the condition $C(1)$ introduced below. The corresponding quadrature weights are determined by the quadrature conditions

$$\sum_{i=1}^{s} b_i c_i^{q-1} = \frac{1}{q} \quad \text{for} \quad q = 1, \ldots, p\,, \tag{$B(p)$}$$

which ensure that the quadrature formula with weights b_i and nodes c_i is of order p on the interval $[0, 1]$, see [82]. The coefficients a_{ij} are determined by the requirement that the obtained numerical solution coincides with the true solution up to and including an order p of powers of the step-size h. It turns out, that these requirements are closely related to the so-called *simplifying assumptions* $B(p)$,

$$\sum_{j=1}^{s} a_{ij} c_j^{q-1} = \frac{c_i^q}{q} \quad \text{for} \quad i = 1, \dots, s \quad \text{and} \quad q = 1, \dots, \eta \qquad C(\eta)$$

as well as

$$\sum_{i=1}^{s} b_i c_i^{q-1} a_{ij} = \frac{b_j}{q}(1 - c_j^q) \quad \text{for} \quad j = 1, \dots, s \quad \text{and} \quad q = 1, \dots, \zeta\,, \quad D(\zeta)$$

see [34, 38, 82, 84]. Therein, $C(\eta)$ ensures that the internal stages $\boldsymbol{X}_i - \boldsymbol{x}(\tau_i) = \mathcal{O}(h^{\eta+1})$, i.e., the quadrature formula with weights $a_{i1}, \dots, a_{is}$ and nodes c_i is of order η on the interval $[0, c_i]$, see Lemma 7.5 in [84]. A Runge–Kutta method satisfying $B(p)$ and $C(\eta)$ is said to be of *stage order* $\min(p, \eta)$. The stage order is an important property of Runge–Kutta methods for the solution of stiff differential equations and differential algebraic equations, since their classical convergence order possibly drops to the stage order for such class of systems [82].

The presented simplifying assumptions are important to construct higher-order implicit Runge–Kutta methods and can be summarized by the following theorem.

Theorem 4.2 (Butcher [34], Theorem 7)**.**[3] *If $B(p)$, $C(\eta)$ and $D(\zeta)$ are satisfied with $p \leq 2\eta + 2$ and $p \leq \zeta + \eta + 1$, then the Runge–Kutta method is of order p.*

Definition 4.3. Application of a Runge–Kutta method (4.3) or (4.6) to the Dahlquist test equation $\dot{x} = \lambda x$ results in

$$x_{n+1} = R(z) x_n \quad \text{with} \quad z = h\lambda\,. \tag{4.13}$$

Therein, $R(z)$ denotes the *stability function* of the method and the set

$$S = \{z \in \mathbb{C} \,|\, |R(z)| \leq 1\} \tag{4.14}$$

is called its *stability domain*.

[3] See also Hairer et al. [84], Theorem 7.4.

Proposition 4.4. *The stability function $R(z)$ of the s-stage implicit Runge–Kutta methods* (4.3) *or* (4.6) *is given by*

$$R(z) = 1 + z\boldsymbol{b}^{\mathrm{T}}(\mathbf{1}_{s\times s} - z\boldsymbol{A})^{-1}\mathbf{1}_s\,. \tag{4.15}$$

Proof. For $f(t,x) = \lambda x$, the equations (4.4) and (4.7) are linear in $\boldsymbol{X}$ and $\dot{\boldsymbol{X}}$, respectively. Solving for these quantities and inserting $\boldsymbol{X}$ into (4.5) and $\dot{\boldsymbol{X}}$ into (4.8), respectively leads to (4.15). □

As shown in [82], the stability function (4.15) may alternatively be written as

$$R(z) = \frac{P(z)}{Q(z)} = \frac{\det(\mathbf{1}_{s\times s} - z\boldsymbol{A} + z\mathbf{1}_s\boldsymbol{b}^{\mathrm{T}})}{\det(\mathbf{1}_{s\times s} - z\boldsymbol{A})}\,. \tag{4.16}$$

Consequently, for implicit Runge–Kutta methods, the stability function becomes a rational function with numerator $P(z)$ and denominator $Q(z)$ of degree $\leq s$. The exact solution $x(t) = x(t_0)e^{\lambda(t-t_0)}$ of the Dahlquist test equation is stable for $\lambda \in \mathbb{C}^- = \{z \in \mathbb{C} \,|\, \mathrm{Re}\, z \leq 0\}$. For a numerical method it is a desirable property to preserve this stability. Hence, the method is called *A-stable* if its stability domain satisfies $\mathbb{C}^- \subseteq S$. It can be shown that a Runge–Kutta method is A-stable if and only if $|R(iy)| \leq 1$ for all real y and if aditionally $R(z)$ is analytic for $\mathrm{Re}\, z < 0$, see [82]. The second condition is satisfied if $R(z)$ has only poles with $\mathrm{Re}\, z \geq 0$, i.e., the zeros of $Q(z)$ have nonnegative real parts.

If in addition, $\lim_{z\to\infty} R(z) = 0$, the method is called *L-stable* [57, 82]. This property is of crucial importance for the solution of stiff differential equations, singularly perturbed problems and differential algebraic equations, since L-stable methods quickly damp out the transient response of stiff components of the differential equation, see [61, 85].

Definition 4.5 (Prothero and Robinson [145])**.** A Runge–Kutta method satisfying

$$a_{sj} = b_j \quad \text{for} \quad j = 1,\ldots,s \quad \Longleftrightarrow \quad \boldsymbol{A}^{\mathrm{T}}\boldsymbol{e}_s = \boldsymbol{b} \tag{4.17}$$

is called *stiffly accurate*.

It can be readily verified that for stiffly accurate Runge–Kutta methods it holds that $\boldsymbol{x}_{n+1} = \boldsymbol{X}_s$, i.e., the future state $\boldsymbol{x}_{n+1}$ coincides with the last stage $\boldsymbol{X}_s$. As noted by Ehle [57], this is very beneficial for solving stiff differential equations, since for methods with invertible coefficient matrix $\boldsymbol{A}$, the internal stages are very close to the exact solution, i.e., $\boldsymbol{Y}_i \approx \boldsymbol{y}(t_n + c_i h)$, but the numerical approximation $\boldsymbol{y}_{n+1} = \boldsymbol{y}_n + \sum_{i=1}^{s} b_i \dot{\boldsymbol{Y}}_i$ may be far away [81].

Proposition 4.6. *A stiffly accurate s-stage implicit Runge–Kutta method with regular coefficient matrix $\boldsymbol{A}$ satisfies $\lim_{z\to\infty} R(z) = 0$. This makes an A-stable method L-stable.*

Proof. By (4.15) and (4.17),

$$\begin{aligned}\lim_{z\to\infty} R(z) &= \lim_{z\to\infty} \left(1 + \boldsymbol{b}^{\mathrm{T}} \left(\tfrac{1}{z}\mathbf{1}_{s\times s} - \boldsymbol{A}\right)^{-1} \mathbf{1}_s\right) \\ &= 1 - \boldsymbol{b}^{\mathrm{T}}\boldsymbol{A}^{-1}\mathbf{1}_s = 1 - \boldsymbol{e}_s^{\mathrm{T}}\mathbf{1}_s = 1 - 1 = 0\,.\end{aligned} \tag{4.18}$$

□

Definition 4.7 (Butcher [36])**.**[4] Let the nonlinear system (4.1) satisfy the contractivity condition

$$(\boldsymbol{f}(t,\boldsymbol{x}) - \boldsymbol{f}(t,\boldsymbol{y}))^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{y}) \le 0 \quad \forall \boldsymbol{x},\boldsymbol{y}\,. \tag{4.19}$$

Then, a Runge–Kutta method applied to this system is called *B-stable*, if for two numerical solutions $\boldsymbol{x}_{n+1}, \boldsymbol{y}_{n+1}$ and an arbitrary step-size $h > 0$ this condition implies

$$\|\boldsymbol{x}_{n+1} - \boldsymbol{y}_{n+1}\| \le \|\boldsymbol{x}_n - \boldsymbol{y}_n\| \quad \forall \boldsymbol{x}_n, \boldsymbol{y}_n\,. \tag{4.20}$$

That is, the distance between two numerical solutions is nonincreasing for increasing n.

Since B-stability is a generalization of A-stability, the former one implies the latter. To see this, the definition from above is applied to $\dot{x} = \lambda x$, with $\lambda = a + ib \in \mathbb{C}$ and $x = x_1 + ix_2 \in \mathbb{C}$. This can equivalently be written as

$$\begin{pmatrix}\dot{x}_1\\ \dot{x}_2\end{pmatrix} = \begin{pmatrix}a & -b\\ b & a\end{pmatrix}\begin{pmatrix}x_1\\ x_2\end{pmatrix}\,. \tag{4.21}$$

Now for $a \le 0$ this implies $(\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{y}))^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{y}) = a\|\boldsymbol{x}-\boldsymbol{y}\|^2 \le 0$. Consequently, the contractivity assumption (4.19) is met. With $y_n = 0$ it follows that $y_{n+1} = R(\lambda h)y_n = 0$. Due to the assumption of B-stability also (4.20) is met. Consequently, $|x_{n+1}| = |R(\lambda h)|\,|x_n| \le |x_n|$ holds, which implies that $|R(\lambda h)| \le 1$.

Proposition 4.8 (Hairer and Wanner [82], Theorem 12.4)**.** *If the coefficients of a Runge–Kutta method* (4.3) *or* (4.6) *satisfy*

(i) non-negative weights, i.e., $b_i \ge 0$ for $i = 1,\dots,s$

(ii) the matrix $\boldsymbol{Q}$ is non-negative definite, with $q_{ij} = b_i a_{ij} + b_j a_{ji} - b_i b_j$,

then the method is B-stable.

[4] See also Hairer and Wanner [82], Definition 12.2.

4.2 Bilaterally Constrained Systems

Returning back to constrained mechanical systems, it is now of interest to solve the system of differential algebraic equations given by

$$\dot{\boldsymbol{q}} = \boldsymbol{f}(t, \boldsymbol{q}, \boldsymbol{u}) \tag{4.22a}$$
$$\dot{\boldsymbol{u}} = \boldsymbol{k}(t, \boldsymbol{q}, \boldsymbol{u}) + \boldsymbol{l}(t, \boldsymbol{q})\dot{\boldsymbol{\pi}} \tag{4.22b}$$
$$\mathbf{0}_{n_g} = \boldsymbol{g}(t, \boldsymbol{q}) \tag{4.22c}$$
$$\mathbf{0}_{n_\gamma} = \boldsymbol{\gamma}(t, \boldsymbol{q}, \boldsymbol{u})\,, \tag{4.22d}$$

with the functions

$$\boldsymbol{f}(t, \boldsymbol{q}, \boldsymbol{u}) = \boldsymbol{B}(t, \boldsymbol{q})\boldsymbol{u} + \boldsymbol{\beta}(t, \boldsymbol{q}) \tag{4.23a}$$
$$\boldsymbol{k}(t, \boldsymbol{q}, \boldsymbol{u}) = \boldsymbol{M}(t, \boldsymbol{q})^{-1}\boldsymbol{h}(t, \boldsymbol{q}, \boldsymbol{u}) \tag{4.23b}$$
$$\boldsymbol{l}(t, \boldsymbol{q}) = \boldsymbol{M}(t, \boldsymbol{q})^{-1}\boldsymbol{W}(t, \boldsymbol{q}) \tag{4.23c}$$

and $\boldsymbol{W}(t, \boldsymbol{q}) = \begin{pmatrix} \boldsymbol{W}_g(t, \boldsymbol{q}) & \boldsymbol{W}_\gamma(t, \boldsymbol{q}) \end{pmatrix} \in \mathbb{R}^{n_u \times n_g + n_\gamma}$. Assuming $\boldsymbol{f}$, $\boldsymbol{g}$ and $\boldsymbol{\gamma}$ to be sufficiently differentiable and that

$$\begin{pmatrix} \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{q}} \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}} \\ \frac{\partial \boldsymbol{\gamma}}{\partial \boldsymbol{u}} \end{pmatrix} \boldsymbol{l} \in \mathbb{R}^{n_g + n_\gamma \times n_g + n_\gamma} \tag{4.24}$$

is invertible in a vicinity of the exact solution, equation (4.22) defines a system of semi-explicit differential algebraic equations in Hessenberg form of index 1 or 2, see [28, 67]. Additionally, the first equation is linear in the velocities $\boldsymbol{u}$ and the second one linear in the Lagrange multipliers $\dot{\boldsymbol{\pi}}$. Such systems described by (4.22) can efficiently be treated by stiffly accurate implicit Runge–Kutta methods, see [80, 83, 98]. Its application reads

$$\boldsymbol{q}_{n+1} = \boldsymbol{q}_n + h \sum\nolimits_{i=1}^{s} b_i \dot{\boldsymbol{Q}}_i\,, \qquad \boldsymbol{u}_{n+1} = \boldsymbol{u}_n + h \sum\nolimits_{i=1}^{s} b_i \dot{\boldsymbol{U}}_i \tag{4.25a}$$
$$\boldsymbol{Q}_i = \boldsymbol{q}_n + h \sum\nolimits_{j=1}^{s} a_{ij} \dot{\boldsymbol{Q}}_j\,, \qquad \boldsymbol{U}_i = \boldsymbol{u}_n + h \sum\nolimits_{j=1}^{s} a_{ij} \dot{\boldsymbol{U}}_j \tag{4.25b}$$
$$\boldsymbol{\pi}_{n+1} = \boldsymbol{\pi}_n + h \sum\nolimits_{i=1}^{s} b_i \dot{\boldsymbol{\Pi}}_i\,, \qquad \boldsymbol{\Pi}_i = \boldsymbol{\pi}_n + h \sum\nolimits_{j=1}^{s} a_{ij} \dot{\boldsymbol{\Pi}}_j\,, \tag{4.25c}$$

such that each state $i = 1, \ldots, s$ satisfies

$$\dot{\boldsymbol{Q}}_i = \boldsymbol{f}(\tau_i, \boldsymbol{Q}_i, \boldsymbol{U}_i) \tag{4.26a}$$
$$\dot{\boldsymbol{U}}_i = \boldsymbol{k}(\tau_i, \boldsymbol{Q}_i, \boldsymbol{U}_i) + \boldsymbol{l}(\tau_i, \boldsymbol{Q}_i)\dot{\boldsymbol{\Pi}}_i \tag{4.26b}$$
$$\mathbf{0}_{n_g} = \boldsymbol{g}(\tau_i, \boldsymbol{Q}_i) \tag{4.26c}$$
$$\mathbf{0}_{n_\gamma} = \boldsymbol{\gamma}(\tau_i, \boldsymbol{Q}_i, \boldsymbol{U}_i)\,. \tag{4.26d}$$

Let $\dot{\boldsymbol{Q}} = (\dot{\boldsymbol{Q}}_1, \dots, \dot{\boldsymbol{Q}}_s) \in \mathbb{R}^{sn_q}$, $\dot{\boldsymbol{U}} = (\dot{\boldsymbol{U}}_1, \dots, \dot{\boldsymbol{U}}_s) \in \mathbb{R}^{sn_u}$ and $\dot{\boldsymbol{\Pi}}_\square = (\dot{\boldsymbol{\Pi}}_{\square,1}, \dots, \dot{\boldsymbol{\Pi}}_{\square,s}) \in \mathbb{R}^{sn_\square}$, where $\square \in \{g, \gamma\}$. The internal stages can be computed via $\boldsymbol{Q} = (\mathbf{1}_s \otimes \boldsymbol{q}_n) + h(\boldsymbol{A} \otimes \mathbf{1}_{n_q \times n_q})\dot{\boldsymbol{Q}}$ and $\boldsymbol{U} = (\mathbf{1}_s \otimes \boldsymbol{u}_n) + h(\boldsymbol{A} \otimes \mathbf{1}_{n_u \times n_u})\dot{\boldsymbol{U}}$. With $\dot{\boldsymbol{X}} = (\dot{\boldsymbol{Q}}, \dot{\boldsymbol{U}}, \dot{\boldsymbol{\Pi}}_g, \dot{\boldsymbol{\Pi}}_\gamma) \in \mathbb{R}^{s(n_q+n_u+n_g+n_\gamma)}$, each Runge–Kutta step requires the solution of the system of nonlinear equations $\boldsymbol{\varphi}(\dot{\boldsymbol{X}}) = \mathbf{0}_{s(n_u+n_q+n_g+n_\gamma)}$, where

$$\boldsymbol{\varphi}(\dot{\boldsymbol{X}}) = \begin{pmatrix} \dot{\boldsymbol{Q}} - \boldsymbol{F}(\boldsymbol{Q}, \boldsymbol{U}) \\ \dot{\boldsymbol{U}} - \boldsymbol{K}(\boldsymbol{Q}, \boldsymbol{U}) - \boldsymbol{L}(\boldsymbol{Q})\dot{\boldsymbol{\Pi}} \\ \boldsymbol{G}(\boldsymbol{Q}) \\ \boldsymbol{\Gamma}(\boldsymbol{Q}, \boldsymbol{U}) \end{pmatrix}. \tag{4.27}$$

Therein, the abbreviations

$$\begin{aligned} \boldsymbol{F}(\boldsymbol{Q}, \boldsymbol{U}) &= \begin{pmatrix} \boldsymbol{f}(\tau_1, \boldsymbol{Q}_1, \boldsymbol{U}_1) \\ \vdots \\ \boldsymbol{f}(\tau_s, \boldsymbol{Q}_s, \boldsymbol{U}_s) \end{pmatrix}, \qquad \boldsymbol{K}(\boldsymbol{Q}, \boldsymbol{U}) = \begin{pmatrix} \boldsymbol{k}(\tau_1, \boldsymbol{Q}_1, \boldsymbol{U}_1) \\ \vdots \\ \boldsymbol{k}(\tau_s, \boldsymbol{Q}_s, \boldsymbol{U}_s) \end{pmatrix} \\ \boldsymbol{L}(\boldsymbol{Q}) &= \begin{pmatrix} \boldsymbol{l}(\tau_1, \boldsymbol{Q}_1) \\ \vdots \\ \boldsymbol{l}(\tau_s, \boldsymbol{Q}_s) \end{pmatrix} \end{aligned} \tag{4.28}$$

and

$$\boldsymbol{G}(\boldsymbol{Q}) = \begin{pmatrix} \boldsymbol{g}(\tau_1, \boldsymbol{Q}_1) \\ \vdots \\ \boldsymbol{g}(\tau_s, \boldsymbol{Q}_s) \end{pmatrix}, \quad \boldsymbol{\Gamma}(\boldsymbol{Q}, \boldsymbol{U}) = \begin{pmatrix} \boldsymbol{\gamma}(\tau_1, \boldsymbol{Q}_1, \boldsymbol{U}_1) \\ \vdots \\ \boldsymbol{\gamma}(\tau_s, \boldsymbol{Q}_s, \boldsymbol{U}_s) \end{pmatrix} \tag{4.29}$$

were used. When the system is solved successfully, the future positions, velocities and integrated Lagrange multipliers are computed as

$$\begin{aligned} \boldsymbol{q}_{n+1} &= \boldsymbol{q}_n + h(\boldsymbol{b}^{\mathrm{T}} \otimes \mathbf{1}_{n_q \times n_q})\dot{\boldsymbol{Q}}, \quad \boldsymbol{u}_{n+1} = \boldsymbol{u}_n + h(\boldsymbol{b}^{\mathrm{T}} \otimes \mathbf{1}_{n_u \times n_u})\dot{\boldsymbol{U}}, \\ \boldsymbol{\pi}_{\square,n+1} &= \boldsymbol{\pi}_{\square,n} + h(\boldsymbol{b}^{\mathrm{T}} \otimes \mathbf{1}_{n_\square \times n_\square})\dot{\boldsymbol{\Pi}}_\square. \end{aligned} \tag{4.30}$$

For stiffly accurate methods their values are already given by $\boldsymbol{q}_{n+1} = \boldsymbol{Q}_s$, $\boldsymbol{u}_{n+1} = \boldsymbol{U}_s$ and $\boldsymbol{\pi}_{\square,n+1} = \boldsymbol{\Pi}_{\square,s}$.

Example 4.9 (Pendulum)**.** Consider a Cartesian pendulum of mass $m = 1.25$ and length $L = 1$ as depicted in Figure 4.1. The pendulum is subjected to gravitational forces with acceleration $\mathrm{g} = 10$. A spring with stiffness

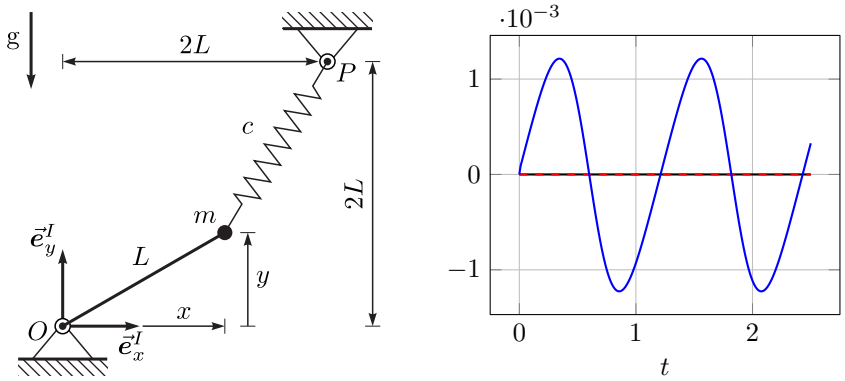

Figure 4.1: Sketch of the pendulum system (left) and simulated time evolution of the constraints g (——), $\dot{g}$ (——) for the two-stage Radau IIA method and $\dot{g}$ (- - -) for the projected two-stage Radau IIA method introduced in Section 4.3 (right).

$c = 1$ is attached to the mass that is relaxed for $(x, y) = (2L, 2L)$. The system is described by the equations

$$\begin{aligned} \dot{x} &= u_x \\ \dot{y} &= u_y \\ m\dot{u}_x &= c(2L - x) + 2x\dot{\pi}_g \\ m\dot{u}_y &= c(2L - y) + 2y\dot{\pi}_g - mg \\ 0 &= x^2 + y^2 - L^2\,, \end{aligned} \tag{4.31}$$

where the last line denotes the constraint equation $g = x^2 + y^2 - L^2$. Its time derivative reads $\dot{g} = 2xu_x + 2yu_y$. Starting at rest with $x = L$ and $y = 0$, for $t \in [0, 2.5]$, the system is integrated by the application of a two-stage Radau IIA method that will be introduced in Section 4.4. The evolution of the constraint equation g and its time derivative $\dot{g}$ are depicted in Figure 4.1. Clearly, the constraint equation g is satisfied up to machine precision, but the constraint derivative $\dot{g}$ shows oscillations with an amplitude around 10^{-3}.

4.3 Projected Runge–Kutta Methods

To eliminate the undesired oscillations of the constraint derivatives observed in Example 4.9, the numerical solution can additionally be projected on the

velocity constraints $\dot{\boldsymbol{g}}$, see [82]. The obtained methods, originally proposed in 1991 by Ascher and Petzold [17], are called projected Runge–Kutta methods. In the same year, the short communication of Lubich [119] showed that for methods with invertible coefficient matrix $\boldsymbol{A}$ such a projection can be interpreted as another Runge–Kutta method, where the original Butcher tableau

$$\begin{array}{c|c} \boldsymbol{c} & \boldsymbol{A} \\ \hline & \boldsymbol{b}^{\mathrm{T}} \end{array} \tag{4.32}$$

is appended by an additional stage. This leads to the extended Butcher tableau[5]

$$\begin{array}{c|c} \hat{\boldsymbol{c}} & \hat{\boldsymbol{A}} \\ \hline & \hat{\boldsymbol{b}}^{\mathrm{T}} \end{array} = \begin{array}{c|cc} \boldsymbol{c} & \boldsymbol{A} & \boldsymbol{0}_s \\ 1+\varepsilon & \boldsymbol{b}^{\mathrm{T}} & \varepsilon \\ \hline & \boldsymbol{b}^{\mathrm{T}} & \varepsilon \end{array} \quad \text{with} \quad \varepsilon \to 0\,. \tag{4.33}$$

For the considered constrained mechanical systems, the last stage of this extended method reads

$$\begin{aligned} \boldsymbol{Q}_{s+1} &= \boldsymbol{q}_{n+1} + h\varepsilon \boldsymbol{f}(t_{n+1} + \varepsilon h, \boldsymbol{Q}_{s+1}, \boldsymbol{U}_{s+1}) && \text{(4.34a)} \\ \boldsymbol{U}_{s+1} &= \boldsymbol{u}_{n+1} + h\varepsilon \boldsymbol{k}(t_{n+1} + \varepsilon h, \boldsymbol{Q}_{s+1}, \boldsymbol{U}_{s+1}) && \text{(4.34b)} \\ &\qquad + h\varepsilon \boldsymbol{l}(t_{n+1} + \varepsilon h, \boldsymbol{Q}_{s+1}) \dot{\boldsymbol{\Pi}}_{s+1} \\ \boldsymbol{0}_{n_g} &= \dot{\boldsymbol{g}}(t_{n+1} + \varepsilon h, \boldsymbol{Q}_{s+1}, \boldsymbol{U}_{s+1}) && \text{(4.34c)} \\ \boldsymbol{0}_{n_\gamma} &= \boldsymbol{\gamma}(t_{n+1} + \varepsilon h, \boldsymbol{Q}_{s+1}, \boldsymbol{U}_{s+1})\,. && \text{(4.34d)} \end{aligned}$$

Note, the velocity constraints $\boldsymbol{\gamma}$ are included to the projection stage again, in order to guarantee that they are also satisfied by the projected velocities $\boldsymbol{U}_{s+1}$. Inspecting (4.34) reveals that this is exactly the result of an implicit Euler step with step-size $h\varepsilon$ starting form $(t_{n+1}, \boldsymbol{q}_{n+1}, \boldsymbol{u}_{n+1})$. Let $\delta\boldsymbol{\Pi}_{n+1} = \boldsymbol{\Pi}_{s+1} - \boldsymbol{\pi}_{n+1} = h\varepsilon\dot{\boldsymbol{\Pi}}_{s+1}$ and $\delta\boldsymbol{u}_{n+1} = \boldsymbol{U}_{s+1} - \boldsymbol{u}_{n+1}$. For $\varepsilon \to 0$ this last stage results in $\boldsymbol{Q}_{s+1} = \boldsymbol{q}_{n+1}$, i.e., the positions are unaffected by the projection. The velocities $\boldsymbol{u}_{n+1}^{+} \coloneqq \boldsymbol{U}_{s+1} = \boldsymbol{u}_{n+1} + \delta\boldsymbol{u}_{n+1}$ are determined

[5] It is important to remark that for methods with singular coefficient matrix $\boldsymbol{A}$, the velocity projection has to be performed using (4.35) instead of considering the extended Butcher tableau from (4.33). This can be explained by the fact that all these methods share a first explicit stage. Consequently, the value of the first stage corresponds to the value of the last stage of the previous step. Due to the projection, this value would get distorted, destroying the order of convergence of the used method.

by solving the velocity projection

$$\begin{aligned}
\delta \boldsymbol{u}_{n+1} &= \boldsymbol{l}(t_{n+1}, \boldsymbol{q}_{n+1}) \delta \boldsymbol{\pi}_{n+1} && \text{(4.35a)} \\
\boldsymbol{0}_{n_g} &= \dot{\boldsymbol{g}}(t_{n+1}, \boldsymbol{q}_{n+1}, \boldsymbol{u}_{n+1} + \delta \boldsymbol{u}_{n+1}) && \text{(4.35b)} \\
&= \boldsymbol{W}_g^{\mathrm{T}}(t_{n+1}, \boldsymbol{q}_{n+1})(\boldsymbol{u}_{n+1} + \delta \boldsymbol{u}_{n+1}) + \boldsymbol{\chi}_g(t_{n+1}, \boldsymbol{q}_{n+1}) \\
\boldsymbol{0}_{n_\gamma} &= \boldsymbol{\gamma}(t_{n+1}, \boldsymbol{q}_{n+1}, \boldsymbol{u}_{n+1} + \delta \boldsymbol{u}_{n+1}) && \text{(4.35c)} \\
&= \boldsymbol{W}_\gamma^{\mathrm{T}}(t_{n+1}, \boldsymbol{q}_{n+1})(\boldsymbol{u}_{n+1} + \delta \boldsymbol{u}_{n+1}) + \boldsymbol{\chi}_\gamma(t_{n+1}, \boldsymbol{q}_{n+1}) \,.
\end{aligned}$$

Consequently, the velocities $\boldsymbol{u}_{n+1}^{+} = \boldsymbol{u}_{n+1} + \delta \boldsymbol{u}_{n+1}$ satisfy all bilateral constraint conditions on velocity level. The next step of the method is started by setting $\boldsymbol{u}_n = \boldsymbol{u}_{n+1}^{+}$. The resulting decoupled velocity projection stage requires very low additional numerical work since by assumption $\dot{\boldsymbol{g}}$ and $\boldsymbol{\gamma}$ are both linear in the velocities $\boldsymbol{u}$. Hence, the projection stage is performed by solving a single additional linear system of equations, with already computed evaluations $\boldsymbol{l}(t_{n+1}, \boldsymbol{q}_{n+1})$, $\boldsymbol{W}_g(t_{n+1}, \boldsymbol{q}_{n+1})$ and $\boldsymbol{W}_\gamma(t_{n+1}, \boldsymbol{q}_{n+1})$ from the last stage of the classical stiffly accurate Runge–Kutta method.

Example 4.10 (Pendulum continued)**.** As depicted in Figure 4.1, adding the aforementioned projection stage to the two-stage Radau IIA method, the constraint velocities $\dot{g}$ are as well satisfied up to machine precision. For the pendulum example, the specific projection stage is given by

$$\begin{aligned}
\delta u_{x,n+1} &= 2 x_{n+1} \delta \pi_{g,n+1} \\
\delta u_{y,n+1} &= 2 y_{n+1} \delta \pi_{g,n+1} \\
0 &= 2 x_{n+1}(u_{x,n+1} + \delta u_{x,n+1}) + 2 y_{n+1}(u_{y,n+1} + \delta u_{y,n+1}) \,,
\end{aligned} \tag{4.36}$$

which can be recast to the form

$$\begin{aligned}
&\begin{pmatrix} 1 & 0 & -2x_{n+1} \\ 0 & 1 & -2y_{n+1} \\ -2x_{n+1} & -2y_{n+1} & 0 \end{pmatrix} \begin{pmatrix} \delta u_{x,n+1} \\ \delta u_{y,n+1} \\ \delta \pi_{g,n+1} \end{pmatrix} \\
&\qquad = \begin{pmatrix} 0 \\ 0 \\ 2x_{n+1} u_{x,n+1} + 2y_{n+1} u_{y,n+1} \end{pmatrix} .
\end{aligned} \tag{4.37}$$

4.4 Butcher Tableaus of Selected Methods

In order to complete the description of the above introduced Runge–Kutta methods, the yet unknown Butcher tableaus have to be specified. Therefore, this section introduces a selection of well-established Butcher tableaus, together with their most important properties. Moreover, a novel composite method with optionally user-defined numerical dissipation is presented.

$$\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array} \qquad \begin{array}{c|cc} \frac{1}{3} & \frac{5}{12} & -\frac{1}{12} \\ 1 & \frac{3}{4} & \frac{1}{4} \\ \hline & \frac{3}{4} & \frac{1}{4} \end{array} \qquad \begin{array}{c|ccc} \frac{4-\sqrt{6}}{10} & \frac{88-7\sqrt{6}}{360} & \frac{296-169\sqrt{6}}{1800} & \frac{-2+3\sqrt{6}}{225} \\ \frac{4+\sqrt{6}}{10} & \frac{296+169\sqrt{6}}{1800} & \frac{88+7\sqrt{6}}{360} & \frac{-2-3\sqrt{6}}{225} \\ 1 & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \\ \hline & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \end{array}$$

Table 4.1: Butcher tableaus for single-stage (backward Euler), two-stage and three-stage Radau IIA methods.

Radau IIA

As shown in [84, 97, 98], the family of Radau IIA methods is well suited for the solution of stiff index 2 and index 3 differential algebraic equations. Hence, it can be expected that these methods are an appropriate choice for the solution of problems of structural mechanics involving frictional contact and impacts. According to [83], Radau IIA methods are uniquely defined by the simplifying assumptions $B(2s-1)$, $C(s)$, $D(s-1)$, together with $c_s = 1$. The methods are A-stable and stiffly accurate with invertible coefficient matrix $\boldsymbol{A}$, hence they are L-stable. Moreover, it can be shown that they are also B-stable, see [36] and Definition 4.7. According to [81, 82], for an s-stage method, the corresponding nodes c_j are given by the roots of the polynomial

$$f(c) = \frac{\mathrm{d}^{s-1}}{\mathrm{d}c^{s-1}} \left(c^{s-1}(c-1)^s \right) . \tag{4.38}$$

The coefficients a_{ij} are obtained by solving for each stage $i = 1, \ldots, s$ the linear system obtained by the simplifying assumptions $C(s)$. Specifically,

$$\sum_{j=1}^{s} a_{ij} c_j^{k-1} = \frac{c_i^k}{k} , \quad k = 1, \ldots, s . \tag{4.39}$$

Since the Raudau IIA methods are stiffly accurate Runge–Kutta methods, their weights are given by

$$b_i = a_{si} , \quad i = 1, \ldots, s . \tag{4.40}$$

For the orders 1 (backward Euler), 3 and 5, their corresponding Butcher-tableaus are shown in Table 4.1. Since the numerical effort seems to be unreasonable and exceeds the benefits of higher-order integration methods, this investigation is restricted on methods with $s \leq 3$.

$$\begin{array}{c|cc} 0 & \frac{1}{2} & -\frac{1}{2} \\ 1 & \frac{1}{2} & \frac{1}{2} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \qquad \begin{array}{c|ccc} 0 & \frac{1}{6} & -\frac{1}{3} & \frac{1}{6} \\ \frac{1}{2} & \frac{1}{6} & \frac{5}{12} & -\frac{1}{12} \\ 1 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array}$$

Table 4.2: Butcher tableaus for two- and three-stage Lobatto IIIC methods.

Lobatto IIIC

Another family of stiffly accurate Runge–Kutta methods are called Lobatto IIIC methods. Although their order is lower compared to the family of Radau IIA methods, they are successfully applied to differential algebraic equations with higher-index, see [84, 98]. Lobatto IIIC methods are uniquely defined by the simplifying assumptions $B(2s-2)$, $C(s-1)$, $D(s-1)$ as well as $c_1 = 0$ and $c_s = 1$. The methods are A-stable and stiffly accurate with invertible coefficient matrix $\boldsymbol{A}$, hence they are L-stable. Moreover, it is shown in [100] that they are also B-stable. Following [82, 100], for an s-stage method, the corresponding nodes c_j are given by the roots of the polynomial

$$f(c) = \frac{\mathrm{d}^{s-2}}{\mathrm{d}c^{s-2}} \left(c^{s-1}(c-1)^{s-1} \right) . \tag{4.41}$$

Following [148], the weights b_i are determined by $B(s)$, i.e., by solving the linear system

$$\sum_{i=1}^{s} b_i c_i^{q-1} = \frac{1}{q} \quad \text{for} \quad q = 1, \ldots, s \, . \tag{4.42}$$

An explicit formula for the nodes is given in [100]. The coefficients a_{ij} are obtained by setting $a_{i1} = b_1$ for $i = 1, \ldots, s$ and $a_{sj} = b_j$ for $j = 1, \ldots, s$. The remaining $(s-1)^2$ coefficients a_{ij} for $i = 1, \ldots, s-1$ and $j = 2, \ldots, s$ are defined by $C(s-1)$. For each stage $i = 1, \ldots, s-1$ the linear system, obtained by the simplifying assumptions

$$\sum_{j=1}^{s} a_{ij} c_j^{k-1} = \frac{c_i^k}{k} \, , \quad k = 1, \ldots, s-1 \, , \tag{4.43}$$

is solved for a_{ij} with $j = 2, \ldots, s$. For the orders 2 and 4, their corresponding Butcher-tableaus are shown in Table 4.2. The Butcher tableau's for order 6 and 8 are reported in [100].

$$
\begin{array}{c|cc}
\alpha & \alpha & 0 \\
1 & 1-\alpha & \alpha \\
\hline
 & 1-\alpha & \alpha
\end{array}
\qquad
\begin{array}{c|ccc}
\beta & \beta & 0 & 0 \\
\frac{1+\beta}{2} & \frac{1-\beta}{2} & \beta & 0 \\
1 & -\frac{6\beta^2-16\beta+1}{4} & \frac{6\beta^2-20\beta+5}{4} & \beta \\
\hline
 & -\frac{6\beta^2-16\beta+1}{4} & \frac{6\beta^2-20\beta+5}{4} & \beta
\end{array}
$$

Table 4.3: Butcher tableaus for the second- and third-order SDIRK method of Alexander [8]. For the two-stage method, $\alpha = 1 - \sqrt{2}/2 \approx 0.29289322$ is the smaller root of the polynomial $\alpha^2 - 2\alpha + \frac{1}{2}$, since the larger one would result in $c_1 > 1$. For the three-stage method $\beta \approx 0.43586652$ is chosen as the root of the polynomial $\beta^3 - 3\beta^2 + \frac{3}{2}\beta - \frac{1}{6}$ that lies in $(\frac{1}{6}, \frac{1}{2})$.

Single Diagonally Implicit

Diagonally implicit Runge–Kutta (DIRK) methods are characterized by a lower triangular coefficient matrix $\boldsymbol{A}$. Consequently, all s stages of these methods are decoupled and can be solved independently. If in addition the diagonal elements of $\boldsymbol{A}$ are identical, these methods are called single diagonally implicit Runge–Kutta (SDIRK) methods and share the same Jacobian matrix required for the applied (simplified) Newton iterations. Consequently, computational performance is expected to increase since a sequence of s nonlinear systems can be solved instead of a huge nonlinear system containing all the implicit equations of the s coupled systems. L-stable and stiffly accurate SDIRK methods are investigated by Alexander [8]. The corresponding Butcher tableaus for the second- and third-order method are shown in Table 4.3.

Both methods are A-stable and stiffly accurate with invertible coefficient matrix $\boldsymbol{A}$. Hence, they are also L-stable. Alternative SDIRK methods with embedded lower-order methods for error estimation are discussed in [43]. As pointed out by Skvortsov [163], the solution of index 2 and index 3 differential algebraic equations requires the chosen method to satisfy additional conditions. Hence, the methods proposed by Alexander [8] suffer from order reduction problems when applied to differential-algebraic equations. For index 3 problems, a second-order four-stage method is presented in [163]. Although the method is diagonally implicit with a single diagonal element, its computational complexity predominates due to the high number of stages required for relatively poor second-order convergence.

A natural idea is to extend SDIRK methods to almost SDIRK methods in the sense that an additional first explicit stage is allowed. For stiffly

$$
\begin{array}{c|c}
\boldsymbol{c} & \boldsymbol{A} \\
\hline
 & \boldsymbol{b}^{\mathrm{T}} \\
\hline
 & \hat{\boldsymbol{b}}^{\mathrm{T}}
\end{array}
=
\begin{array}{c|ccc}
0 & 0 & 0 & 0 \\
\gamma & d & d & 0 \\
1 & w & w & d \\
\hline
 & w & w & d \\
\hline
 & \frac{1-w}{3} & \frac{3w+1}{3} & \frac{d}{3}
\end{array}
$$

Table 4.4: Butcher tableau with embedded third-order formula of the TR-BDF2 method [93]. Therein, $\gamma = 2 - \sqrt{2}$, $d = \gamma/2$ and $w = \sqrt{2}/4$.

accurate methods, this is obtained without an extra function evaluation. Such methods are called stiffly accurate explicit single diagonally implicit Runge–Kutta (ESDIRK) methods. A well-established ESDIRK method is called TR-BDF2 and was originally proposed by Bank et al. [18]. It combines a step of an implicit trapezoidal rule with a second-order backward difference step (BDF2), both with possibly different step-sizes. Hence, in contrast to the classical BDF2 method, no history of previously computed states is required. Hosea and Shampine [93] have shown that for specific choices of the step-sizes, an ESDIRK method is obtained. The corresponding Butcher tableau is depicted in Table 4.4. The method can be shown to be A-, L- and strongly S-stable, see [18, 93], making it well suited for the integration of stiff differential and differential algebraic equations. Moreover, there is an embedded third-order method, defined in terms of $\hat{\boldsymbol{b}}$, which can be used for efficient error estimation, required for adaptive step-size implementations.

Besides the TR-BDF2, Hosea and Shampine [93] presented a full family of ESDIRK methods starting from a generic three-stage stiffly accurate ESDIRK Butcher tableau

$$
\begin{array}{c|c}
\boldsymbol{c} & \boldsymbol{A} \\
\hline
 & \boldsymbol{b}^{\mathrm{T}} \\
\hline
 & \hat{\boldsymbol{b}}^{\mathrm{T}}
\end{array}
=
\begin{array}{c|ccc}
0 & 0 & 0 & 0 \\
c_2 & a_{21} & b_3 & 0 \\
c_3 & b_1 & b_2 & b_3 \\
\hline
 & b_1 & b_2 & b_3 \\
\hline
 & \hat{b}_1 & \hat{b}_2 & \hat{b}_3
\end{array}
. \tag{4.44}
$$

Now the coefficients are restricted by the second-order conditions $B(2)$ together with consistency $C(1)$. Specifically, it is required that

$$b_1 + b_2 + b_3 = 1 \tag{4.45a}$$

$$b_2 c_2 + b_3 c_3 = \frac{1}{2} \tag{4.45b}$$

$$a_{21} + b_3 = c_2 \tag{4.45c}$$

$$b_1 + b_2 + b_3 = c_3 \,. \tag{4.45d}$$

Four additional restrictions are obtained by the requirement that $\hat{\boldsymbol{b}} = (\hat{b}_1, \hat{b}_2, \hat{b}_3)$ yields an embedded third-order method. Since there are eight equations for nine unknown, introducing $b_3 = \frac{\theta}{2}$ leads to a one parameter family, defined in terms of the Butcher tableau

$$\begin{array}{c|c} \boldsymbol{c} & \boldsymbol{A} \\ \hline & \boldsymbol{b}^{\mathrm{T}} \\ \hline & \hat{\boldsymbol{b}}^{\mathrm{T}} \end{array} = \begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \theta & \frac{\theta}{2} & \frac{\theta}{2} & 0 \\ 1 & \frac{3\theta-\theta^2-1}{2\theta} & \frac{1-\theta}{2\theta} & \frac{\theta}{2} \\ \hline & \frac{3\theta-\theta^2-1}{2\theta} & \frac{1-\theta}{2\theta} & \frac{\theta}{2} \\ \hline & \frac{3\theta-1}{6\theta} & \frac{1}{6\theta(1-\theta)} & \frac{2-3\theta}{6(1-\theta)} \end{array} . \tag{4.46}$$

According to (4.15) this implies a Runge–Kutta method with stability function

$$\begin{aligned} R(z) &= 1 + z\boldsymbol{b}^{\mathrm{T}}(\mathbf{1}_{s\times s} - z\boldsymbol{A})^{-1}\mathbf{1}_s \\ &= \frac{(\theta^2 - 4\theta + 2)z^2 + 4(1-\theta)z + 4}{(2-\theta z)^2} \end{aligned} \tag{4.47}$$

and $\lim_{z\to\infty} R(z) = \frac{\theta^2-4\theta+2}{\theta^2}$. The method is L-stable for $\theta^2 - 4\theta + 2 = 0$ and hence, $\theta = 2 \pm \sqrt{2}$. Since $2 + \sqrt{2} > 1$, this choice would lead to $c_2 = \theta > 1$. Consequently, the authors have chosen $\theta = 2 - \sqrt{2} < 1$, which finally leads to the TR-BDF2 method. The choice $\theta = \frac{1}{2}$ results in a method that corresponds to a double step of the implicit trapezoidal rule, they called TRX2. It is defined by the Butcher tableau

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4} & 0 \\ 1 & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \hline & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{array} . \tag{4.48}$$

Since this choice leads to $\lim_{z\to\infty} R(z) = \frac{\theta^2-4\theta+2}{\theta^2} = 1$, the TRX2 method is not L-stable.

Composite Trapezoidal-Generalized-alpha

In the year 2000, Jansen et al. [96] proposed a generalized-alpha method for the integration of the filtered Navier–Stokes equation. This method is capable to deal with arbitrary first-order differential equations. Since the mechanical systems under consideration typically exhibit a non-trivial kinematic equation (3.1), making them first-order differential equations, the extension of Jansen's method to nonsmooth mechanical systems is desirable. Applied to the first-order differential equation (4.1), the mentioned generalized-alpha method reads

$$\dot{\boldsymbol{x}}_{n+1} = \boldsymbol{f}(t_{n+1}, \boldsymbol{x}_{n+1}) \tag{4.49a}$$

$$\alpha_m \dot{\boldsymbol{x}}_n + (1-\alpha_m)\dot{\boldsymbol{x}}_{n+1} = \alpha_m \boldsymbol{y}_n + (1-\alpha_m)\boldsymbol{y}_{n+1} \tag{4.49b}$$

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + h\big((1-\gamma)\boldsymbol{y}_n + \gamma \boldsymbol{y}_{n+1}\big)\,, \tag{4.49c}$$

see [33]. Therein, $\boldsymbol{y} \neq \dot{\boldsymbol{x}}$ denotes an auxiliary derivative. The simple initialization $\boldsymbol{y}_0 = \dot{\boldsymbol{x}}_0$ results in spurious oscillations, especially when the method is applied to differential algebraic equations. Advanced initialization methods are required to reduce such undesired effects, see [16]. Even worse, when an adaptive step-size should be used, after each change of the step-size, the auxiliary variables have to be modified in order to ensure second-order convergence, see [33, 101].

Composite Method with User-defined Damping

To address these challenges, a new and yet unpublished composite method, similar to TR-BDF2 method, is developed subsequently. This novel approach, like the previously discussed generalized-alpha method, allows for user-defined damping. Assume two steps of the generalized-alpha method (4.49), the first one with $t_n \mapsto t_{n+\beta} = t_n + \beta h$ and another one with $t_{n+\beta} \mapsto t_{n+1} = t_n + h$, i.e., with step-size $(1-\beta)h$. The auxiliary derivatives $\boldsymbol{y}$ can be eliminated by combining both steps, resulting in the multi-step formulation

$$\sum_{k\in\{0,\beta,1\}} d_k\, \boldsymbol{x}_{n+k} = h \sum_{k\in\{0,\beta,1\}} e_k\, \boldsymbol{f}(t_{n+k}, \boldsymbol{x}_{n+k})\,, \tag{4.50}$$

with the coefficients

$$\begin{aligned} d_0 &= -(1-\beta)\alpha_m\,, & e_0 &= \beta(1-\beta)(1-\gamma)\alpha_f \\ d_\beta &= \alpha_m - \beta\,, & e_\beta &= \beta(1-\beta)[(1-\gamma)(1-\alpha_f)+\gamma\alpha_f] \\ d_1 &= \beta(1-\alpha_m)\,, & e_1 &= \beta(1-\beta)\gamma(1-\alpha_f)\,. \end{aligned} \tag{4.51}$$

With $\beta = \frac{1}{2}$ and using a step size $h = 2\tilde{h}$, this is exactly the formulation derived in [15, 33]. The free parameters γ, α_f and α_m are chosen such that second-order accuracy and user-defined numerical dissipation is obtained. As outlined in [33], this choice results in

$$\gamma = \frac{1}{2} + \alpha_f - \alpha_m\,, \quad \alpha_f = \frac{\rho_\infty}{\rho_\infty + 1}\,, \quad \alpha_m = \frac{3\rho_\infty - 1}{2(\rho_\infty + 1)}\,. \tag{4.52}$$

Therein, user-defined numerical dissipation is controlled by $\rho_\infty \in [0, 1)$. An undamped method is characterized by $\rho_\infty = 1$, whereas $\rho_\infty = 0$ means asymptotic annihilation of the high-frequency response [15, 45]. Combining the implicit trapezoidal step

$$\boldsymbol{x}_{n+\beta} - \boldsymbol{x}_n = \beta\frac{h}{2}\big(\boldsymbol{f}(t_n, \boldsymbol{x}_n) + \boldsymbol{f}(t_{n+\beta}, \boldsymbol{x}_{n+\beta})\big) \tag{4.53}$$

with (4.51) and by verifying $d_0 + d_\beta = d_1$, simple algebraic manipulations reveal that the composite method can be written as Runge–Kutta method, defined in terms of the Butcher tableau

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ c_2 & \frac{\beta}{2} & \frac{\beta}{2} & 0 \\ c_3 & \frac{2e_0 - d_\beta\beta}{2d_1} & \frac{2e_\beta - d_\beta\beta}{2a_1} & \frac{e_1}{d_1} \\ \hline & \frac{2e_0 - d_\beta\beta}{2d_1} & \frac{2e_\beta - d_\beta\beta}{2d_1} & \frac{e_1}{d_1} \end{array}\,. \tag{4.54}$$

The values for c_2 and c_3 are determined by requiring consistency $C(1)$, i.e., $c_i = \sum_{j=1}^{3} a_{ij}$ for $i = 2, 3$. In order to obtain an ESDIRK method the coefficients have to satisfy $\beta d_1 = 2e_1$. Additionally, the second-order conditions (4.45a) and (4.45b) have to be satisfied. And finally, the damping at infinity property requires $\lim_{z\to\infty} R(z) = \rho_\infty$.

Clearly, it requires tedious computations to satisfy all these stated conditions. Even more, it is completely unnecessary since a Butcher tableau with similar structure, given in (4.46), was already introduced by Hosea and Shampine [93]. As already discussed before, it encodes a Runge–Kutta method with $\lim_{z\to\infty} R(z) = \frac{\theta^2 - 4\theta + 2}{\theta^2}$. Hence, in the spirit

of the generalized-alpha method, a second-order accurate Runge–Kutta method with user-defined damping at infinity is obtained by the constraint

$$\frac{\theta^2 - 4\theta + 2}{\theta^2} = \rho_\infty \quad \Longleftrightarrow \quad \theta = \frac{2 \pm \sqrt{2(1+\rho_\infty)}}{1-\rho_\infty} \quad \text{for} \quad \rho_\infty \in [0,1)\,. \tag{4.55}$$

Using the same arguments as for the TR-BDF2 method, the smaller value of θ in (4.55) is preferred since it results in $c_2 < 1$. For $\rho_\infty = 0$, this leads to the TR-BDF2 method with $\theta = 2 - \sqrt{2}$. Moreover, the TRX2 method is recovered for $\theta = \frac{1}{4}$, which leads to $\rho_\infty = 1$. For all other values of $\rho_\infty \in (0,1)$, user-defined numerical dissipation is obtained which leads to a generalized-alpha type Runge–Kutta method. The resulting method is subsequently called TR-$\theta(\rho_\infty)$ method, with $\theta(\rho_\infty)$ defined by (4.55) for $\rho_\infty \in [0,1)$ and $\theta = \frac{1}{2}$ for $\rho_\infty = 1$.

Another family of similar composite methods is presented in [180]. Therein, the idea of composite trapezoidal methods is continued for composite implicit mid-point rules. The combination of an implicit mid-point rule with a second-order BDF method leads to a composite method they called IM-BDF2. In contrast to the TR-BDF2 method [18, 93], the IM-BDF2 has no first explicit stage. Although this reduces the memory requirements of the resulting method this comes at the price that no embedded third-order formula can be constructed, see [93]. Apart from that, Ying et al. [180] developed higher-order composite BDF methods. Since this section already introduced a huge number of Runge–Kutta methods, the composite methods of [180] are not further investigated but might be interesting candidates for future research.

Stability Functions and Convergence

To close this section, the stability functions of the presented methods should be investigated. Therefore, their numerical values are visualized in Figure 4.2. As already proven in Proposition 4.6, all presented methods with invertible coefficient matrix $\boldsymbol{A}$ are L-stable, i.e., $\lim_{z\to\infty} R(z) = 0$. This is also the case for the TR-BDF2 method, which coincides to the TR-$\theta(\rho_\infty)$ method with $\rho_\infty = 0$. For different values of ρ_∞ the desired property $\lim_{z\to\infty} R(z) = \rho_\infty$ can be observed that describes user-defined numerical dissipation of high frequencies.

Example 4.11 (Pendulum continued)**.** In what follows the convergence of the proposed implicit Runge–Kutta methods is investigated when applied to the pendulum example from before. Thereto, the system is integrated

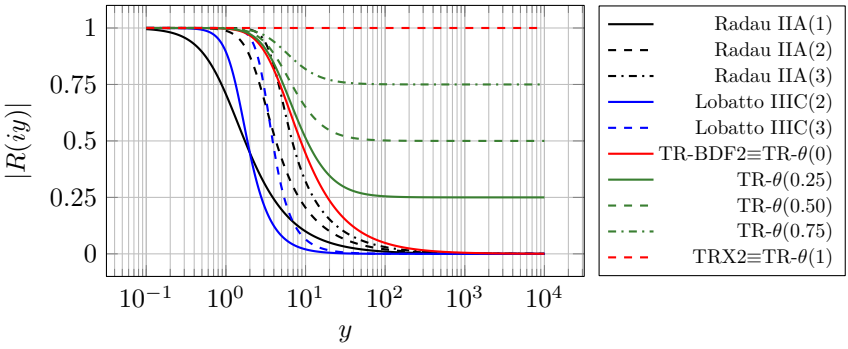

Figure 4.2: Visualized stability functions of the presented stiffly accurate Runge–Kutta methods.

with different step-sizes $h \in \{5.12 \cdot 10^{-2},\ 2.56 \cdot 10^{-2},\ 1.28 \cdot 10^{-2},\ 6.4 \cdot 10^{-3},\ 3.2 \cdot 10^{-3},\ 1.6 \cdot 10^{-3},\ 8 \cdot 10^{-4}\}$ and the convergence behavior of the solutions with respect to an accurate numerical solution computed with $h^{\text{ref}} = 2 \cdot 10^{-4}$ is studied. For $n = 1, 2, \ldots, N$ let $\boldsymbol{x}_n, \boldsymbol{x}_n^{\text{ref}} \in \mathbb{R}^{n_x}$ be the time evolution of a quantity (e.g. $\boldsymbol{x} = \boldsymbol{q}$ or $\boldsymbol{x} = \boldsymbol{u}$) computed with the step-sizes h and h^{ref}, respectively. Moreover, assume that $\boldsymbol{x}_n$ and $\boldsymbol{x}_n^{\text{ref}}$ approximate the values of $\boldsymbol{x}$ at the times $t_n = nh$. The time evolution of the error between the simulated and the reference solution is therefore $\boldsymbol{e}_n^{\boldsymbol{x}} = \boldsymbol{x}_n - \boldsymbol{x}_n^{\text{ref}}$ and the overall simulation error is computed as

$$\|\boldsymbol{e}^{\boldsymbol{x}}\|_1 = h \sum_{n=1}^{N} \|\boldsymbol{e}_n^{\boldsymbol{x}}\|_1 = h \sum_{n=1}^{N} \sum_{k=1}^{n_x} |e_n^{\boldsymbol{x},k}|\,, \tag{4.56}$$

where $e_n^{\boldsymbol{x},k}$ is the kth component of the tuple $\boldsymbol{e}_n^{\boldsymbol{x}}$.

For the evaluation of each time-step, a nonlinear system of equations has to be solved. Therefore, a Newton–Raphson method was applied with relative and absolute tolerances of 10^{-14}, see (6.17) for details. Moreover, the time-span of the integration was restricted to $t \in [0, 0.2048]$. The convergence behavior of the discussed Runge–Kutta methods is visualized in Figure 4.3. The slope of the straight lines indicate the global order of convergence. The outstanding convergence rates of the Radau IIA methods are in line with the theoretically obtained convergence results given by [98]. For methods with $s \geq 2$ these are given by

$$\boldsymbol{q}_n - \boldsymbol{q}(t_n) = \mathcal{O}(h^{2s-1}) \quad \text{and} \quad \boldsymbol{u}_n - \boldsymbol{u}(t_n) = \mathcal{O}(h^s)\,, \tag{4.57}$$

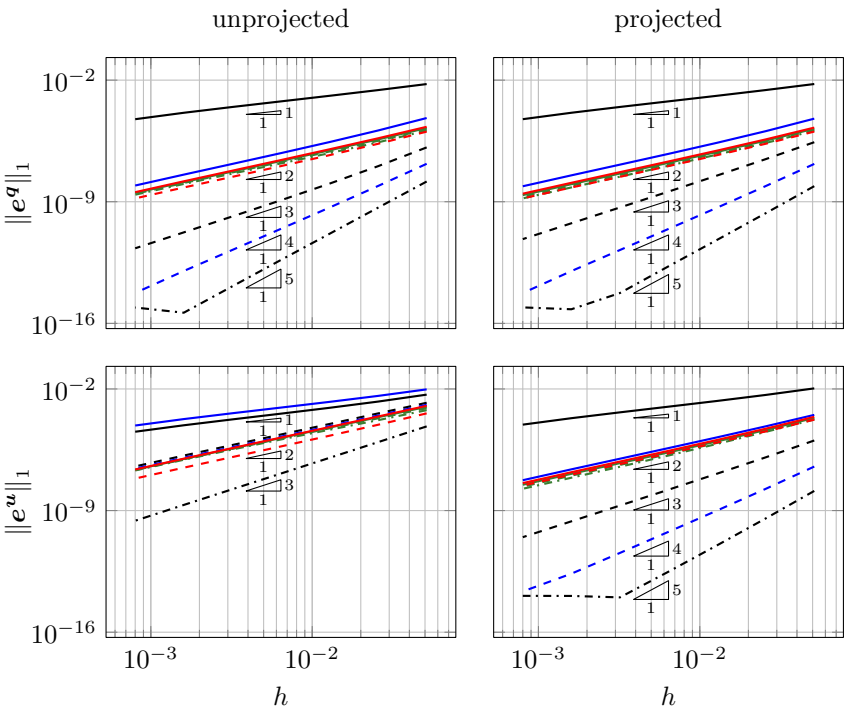

Figure 4.3: Convergence results of the unprojected (left) and projected (right) implicit Runge–Kutta methods applied to the pendulum problem. The legend coincides with the one used in Figure 4.2.

i.e., the convergence order of the velocity components drops to the stage order s (order reduction). By application of the velocity projection introduced in Section 4.3, the same order of convergence is obtained for the positions $\boldsymbol{q}$ and velocities $\boldsymbol{u}$, i.e.,

$$\boldsymbol{u}_n^+ - \boldsymbol{u}(t_n) = \mathcal{O}(h^{2s-1})\,. \tag{4.58}$$

Again, this was already theoretically predicted by Jay [98] and demonstrates the importance of the velocity projection for differential algebraic equations. Besides that, the numerical solution satisfies the constraint equations as well as their time derivative up to machine precision.

4.5 Naive Extension to Nonsmooth Mechanical Systems

The extension of the presented stiffly accurate Runge–Kutta methods to nonsmooth mechanical systems requires the quadrature of the measures $\mathrm{d}\boldsymbol{q}$, $\mathrm{d}\boldsymbol{u}$ and $\mathrm{d}\boldsymbol{\pi}$ that are encountered in the measure differential equations (3.1) and (3.2). Hence, it is now of interest to solve the system of measure differential inclusions given by

$$\mathrm{d}\boldsymbol{q} = \boldsymbol{f}(t,\boldsymbol{q},\boldsymbol{u})\mathrm{d}t \tag{4.59a}$$
$$\mathrm{d}\boldsymbol{u} = \boldsymbol{k}(t,\boldsymbol{q},\boldsymbol{u})\mathrm{d}t + \boldsymbol{l}(t,\boldsymbol{q})\mathrm{d}\boldsymbol{\pi}\,, \tag{4.59b}$$

with

$$\boldsymbol{f}(t,\boldsymbol{q},\boldsymbol{u}) = \boldsymbol{B}(t,\boldsymbol{q})\boldsymbol{u} + \boldsymbol{\beta}(t,\boldsymbol{q}) \tag{4.60a}$$
$$\boldsymbol{k}(t,\boldsymbol{q},\boldsymbol{u}) = \boldsymbol{M}(t,\boldsymbol{q})^{-1}\boldsymbol{h}(t,\boldsymbol{q},\boldsymbol{u}) \tag{4.60b}$$
$$\boldsymbol{l}(t,\boldsymbol{q}) = \boldsymbol{M}(t,\boldsymbol{q})^{-1}\boldsymbol{W}(t,\boldsymbol{q}) \tag{4.60c}$$

and $\boldsymbol{W}(t,\boldsymbol{q}) = \begin{pmatrix}\boldsymbol{W}_g(t,\boldsymbol{q}) & \boldsymbol{W}_\gamma(t,\boldsymbol{q}) & \boldsymbol{W}_N(t,\boldsymbol{q}) & \boldsymbol{W}_F(t,\boldsymbol{q})\end{pmatrix}$. Furthermore, the system (4.59) is subjected to bilateral and unilateral constraint conditions that are incorporated below. Application of an s-stage implicit Runge–Kutta method on (4.59) reads

$$\mathrm{d}\boldsymbol{Q}_i = h\boldsymbol{f}(\tau_i,\boldsymbol{Q}_i,\boldsymbol{U}_i)\,, \qquad \mathrm{d}\boldsymbol{U}_i = h\boldsymbol{k}(\tau_i,\boldsymbol{Q}_i,\boldsymbol{U}_i) + \boldsymbol{l}(\tau_i,\boldsymbol{Q}_i)\mathrm{d}\boldsymbol{\Pi}_i\,, \tag{4.61a}$$
$$\boldsymbol{Q}_i = \boldsymbol{q}_n + \sum\nolimits_{j=1}^{s} a_{ij}\mathrm{d}\boldsymbol{Q}_j\,, \qquad \boldsymbol{U}_i = \boldsymbol{u}_n + \sum\nolimits_{j=1}^{s} a_{ij}\mathrm{d}\boldsymbol{U}_j\,, \tag{4.61b}$$
$$\boldsymbol{\Pi}_i = \boldsymbol{\pi}_n + \sum\nolimits_{j=1}^{s} a_{ij}\mathrm{d}\boldsymbol{\Pi}_j \tag{4.61c}$$
$$\boldsymbol{q}_{n+1} = \boldsymbol{q}_n + \sum\nolimits_{i=1}^{s} b_i\mathrm{d}\boldsymbol{Q}_i\,, \qquad \boldsymbol{u}_{n+1} = \boldsymbol{u}_n + \sum\nolimits_{i=1}^{s} b_i\mathrm{d}\boldsymbol{U}_i\,, \tag{4.61d}$$
$$\boldsymbol{\pi}_{n+1} = \boldsymbol{\pi}_n + \sum\nolimits_{i=1}^{s} b_i\mathrm{d}\boldsymbol{\Pi}_i\,. \tag{4.61e}$$

Instead of applying a classical quadrature, e.g.,

$$\boldsymbol{u}_{n+1} = \boldsymbol{u}_n + h\sum\nolimits_{j=i}^{s} b_i\widetilde{\mathrm{d}\boldsymbol{U}}_i \quad \text{and} \quad \boldsymbol{\pi}_{n+1} = \boldsymbol{\pi}_n + h\sum\nolimits_{j=i}^{s} b_i\widetilde{\mathrm{d}\boldsymbol{\Pi}}_i\,, \tag{4.62}$$

a version of the quadrature with rescaled measures in the discrete setting is chosen, as obviously $\mathrm{d}\boldsymbol{U}_i = h\,\widetilde{\mathrm{d}\boldsymbol{U}}_i$ and analogously for the discrete measures $\mathrm{d}\boldsymbol{Q}$ and $\mathrm{d}\boldsymbol{\Pi}$. This scaling with the time-step is important in order to avoid

unbounded growth of discrete constraint and contact forces for $h \to 0$. Let

$$\Delta\boldsymbol{\pi}(t;h) = \boldsymbol{\pi}(t+h) - \boldsymbol{\pi}(t) = \int_t^{t+h} \mathrm{d}\boldsymbol{\pi} \approx \sum_{i=1}^{s} b_i \mathrm{d}\boldsymbol{\Pi}_i = h \sum_{i=1}^{s} b_i \widetilde{\mathrm{d}\boldsymbol{\Pi}}_i \,. \tag{4.63}$$

In fact, if t is a discontinuity point, then $\Delta\boldsymbol{\pi}(t;0) \neq 0$ is finite. It is clear that in the limit $h \to 0$, the right-hand side of (4.63) can only be finite if $\widetilde{\mathrm{d}\boldsymbol{\Pi}}_i \to \infty$ for some i, which is numerically unfavorable. The unknowns $\mathrm{d}\boldsymbol{\Pi}_i$ can exactly represent this finite value and show no such bad behavior, see [42] for a similar discussion. Hence, the approximation (4.61) is preferred, as it does not have this issue.

As in the bilaterally constrained case, the internal stages are required to satisfy the bilateral position and velocity constraints

$$\boldsymbol{g}(\tau_i, \boldsymbol{Q}_i) = \boldsymbol{0}_{n_g} \quad \text{and} \quad \boldsymbol{\gamma}(\tau_i, \boldsymbol{Q}_i, \boldsymbol{U}_i) = \boldsymbol{0}_{n_\gamma} \tag{4.64}$$

for $i = 1, \ldots, s$. The last open question is, how to enforce the discrete equivalents to the unilateral constraint conditions for contact and friction. A possible solution is that each stage is forced to satisfy an integrated version of Signorini's law, cf. [41, 42]. With $J_i = (t_n, \tau_i] = (t_n, t_n + c_i h]$, for every contact $k = 1, \ldots, n_N$ and for every stage $i = 1, \ldots, s$ it is enforced that[6]

$$g_N^k(\tau_i, \boldsymbol{Q}_i) \in \mathcal{N}_{\mathbb{R}_0^-}\big(-\textstyle\int_{J_i} \mathrm{d}\pi_N^k\big) \approx \mathcal{N}_{\mathbb{R}_0^-}\big(-\Delta\Pi_{N,i}^k\big)\,, \tag{4.66}$$

where Proposition 2.3 was used (see Proposition 2 of Capobianco et al. [41]), together with the abbreviation $\Delta\Pi_{N,i}^k = \Pi_{N,i}^k - \pi_{N,n}^k = \sum_{j=1}^{s} a_{ij} \mathrm{d}\Pi_{N,j}^k \approx \int_{J_i} \mathrm{d}\pi_N^k$. For the discretization of the set-valued friction laws a similar path is taken. For every contact $k = 1, \ldots, n_N$ and for every stage $i = 1, \ldots, s$ the integrated friction law

$$\begin{aligned} \boldsymbol{\gamma}_F^k(\tau_i, \boldsymbol{Q}_i, \boldsymbol{U}_i) &\in \mathcal{N}_{C_F^k(\int_{J_i} \mathrm{d}\pi_N^k)}\big(-\textstyle\int_{J_i} \mathrm{d}\boldsymbol{\pi}_F^k\big) \\ &\approx \mathcal{N}_{C_F^k(\Delta\Pi_{N,i}^k)}\big(-\Delta\boldsymbol{\Pi}_{F,i}^k\big) \end{aligned} \tag{4.67}$$

is enforced. Therein, the abbreviation $\Delta\boldsymbol{\Pi}_{F,i}^k = \boldsymbol{\Pi}_{F,i}^k - \boldsymbol{\pi}_{F,n}^k$ was introduced with $\Delta\boldsymbol{\Pi}_{F,i}^k = \sum_{j=1}^{s} a_{ij} \mathrm{d}\boldsymbol{\Pi}_{F,j}^k \approx \int_{J_i} \mathrm{d}\boldsymbol{\pi}_F^k$.

[6] One might be tempt to enforce the discrete version of Signorini's law using the discrete percussion measures $\mathrm{d}\Pi_{N,i}^k$, i.e.,

$$g_N^k(\tau_i, \boldsymbol{Q}_i) \in \mathcal{N}_{\mathbb{R}_0^-}\big(-\mathrm{d}\Pi_{N,i}^k\big)\,. \tag{4.65}$$

As it will turn out in Example 4.12, this choice leads to non-uniqueness of the contact forces, even for the simple example of a one-dimensional bouncing ball and in the absence of friction.

Let $\mathrm{d}\boldsymbol{Q} = (\mathrm{d}\boldsymbol{Q}_1, \dots, \mathrm{d}\boldsymbol{Q}_s) \in \mathbb{R}^{sn_q}$, $\mathrm{d}\boldsymbol{U} = (\mathrm{d}\boldsymbol{U}_1, \dots, \mathrm{d}\boldsymbol{U}_s) \in \mathbb{R}^{sn_u}$ and $\mathrm{d}\boldsymbol{\Pi}_\square = (\mathrm{d}\boldsymbol{\Pi}_{\square,1}, \dots, \mathrm{d}\boldsymbol{\Pi}_{\square,s}) \in \mathbb{R}^{sn_\square}$, where $\square \in \{g, \gamma, N, F\}$. The internal stages can be computed via $\boldsymbol{Q} = (\mathbf{1}_s \otimes \boldsymbol{q}_n) + (\boldsymbol{A} \otimes \mathbf{1}_{n_q \times n_q})\mathrm{d}\boldsymbol{Q}$, $\boldsymbol{U} = (\mathbf{1}_s \otimes \boldsymbol{u}_n) + (\boldsymbol{A} \otimes \mathbf{1}_{n_u \times n_u})\mathrm{d}\boldsymbol{U}$ and $\boldsymbol{\Pi}_\square = (\mathbf{1}_s \otimes \boldsymbol{\pi}_{\square,n}) + (\boldsymbol{A} \otimes \mathbf{1}_{n_\square \times n_\square})\mathrm{d}\boldsymbol{\Pi}_\square$. With $\mathrm{d}\boldsymbol{x} = (\mathrm{d}\boldsymbol{Q}, \mathrm{d}\boldsymbol{U}, \mathrm{d}\boldsymbol{\Pi}_g, \mathrm{d}\boldsymbol{\Pi}_\gamma) \in \mathbb{R}^{s(n_q+n_u+n_g+n_\gamma)}$, $\mathrm{d}\boldsymbol{y} = (\mathrm{d}\boldsymbol{\Pi}_N, \mathrm{d}\boldsymbol{\Pi}_F) \in \mathbb{R}^{s(n_N+n_F)}$ and $\mathrm{d}\boldsymbol{\Pi} = (\mathrm{d}\boldsymbol{\Pi}_g, \mathrm{d}\boldsymbol{\Pi}_\gamma, \mathrm{d}\boldsymbol{\Pi}_N, \mathrm{d}\boldsymbol{\Pi}_F) \in \mathbb{R}^{s(n_g+n_\gamma+n_N+n_F)}$, each Runge–Kutta step requires the solution of the system of nonlinear equations $\boldsymbol{\varphi}(\mathrm{d}\boldsymbol{x}, \mathrm{d}\boldsymbol{y}) = \mathbf{0}_{s(n_u+n_q+n_g+n_\gamma)}$, where

$$\boldsymbol{\varphi}(\mathrm{d}\boldsymbol{x}, \mathrm{d}\boldsymbol{y}) = \begin{pmatrix} \mathrm{d}\boldsymbol{Q} - h\boldsymbol{F}(\boldsymbol{Q}, \boldsymbol{U}) \\ \mathrm{d}\boldsymbol{U} - h\boldsymbol{K}(\boldsymbol{Q}, \boldsymbol{U}) - \boldsymbol{L}(\boldsymbol{Q})\mathrm{d}\boldsymbol{\Pi} \\ \boldsymbol{G}(\boldsymbol{Q}) \\ \boldsymbol{\Gamma}(\boldsymbol{Q}, \boldsymbol{U}) \end{pmatrix} . \tag{4.68}$$

The definitions for $\boldsymbol{F}$, $\boldsymbol{K}$, $\boldsymbol{L}$, $\boldsymbol{G}$ and $\boldsymbol{\Gamma}$ used therein are already given in (4.28) and (4.29). The nonlinear equation (4.68) is subjected to the normal cone inclusions

$$\boldsymbol{\psi}_i^k(\mathrm{d}\boldsymbol{x}) \in \mathcal{K}_i^k(\boldsymbol{\mathcal{A}}\,\mathrm{d}\boldsymbol{y}) , \quad k = 1, \dots, n_N , \quad i = 1, \dots, s , \tag{4.69}$$

where

$$\boldsymbol{\psi}_i^k(\mathrm{d}\boldsymbol{x}) = \begin{pmatrix} g_N^k(\tau_i, \boldsymbol{Q}_i) \\ \boldsymbol{\gamma}_F^k(\tau_i, \boldsymbol{Q}_i, \boldsymbol{U}_i) \end{pmatrix} \tag{4.70}$$

and

$$\mathcal{K}_i^k(\mathrm{d}\boldsymbol{\mu}) = \mathcal{N}_{\mathbb{R}_0^-}(-\mathrm{d}\mu_{N,i}^k) \times \mathcal{N}_{C_F^k(\mathrm{d}\mu_{N,i}^k)}(-\mathrm{d}\boldsymbol{\mu}_{F,i}^k) . \tag{4.71}$$

From the quadrature of the percussion measures $\mathrm{d}\boldsymbol{\pi}_N$ and $\mathrm{d}\boldsymbol{\pi}_F$, it can be deduced that

$$\boldsymbol{\mathcal{A}} = \boldsymbol{A} \otimes \begin{pmatrix} \mathbf{1}_{n_N \times n_N} & \mathbf{0}_{n_N \times n_F} \\ \mathbf{0}_{n_F \times n_N} & \mathbf{1}_{n_F \times n_F} \end{pmatrix} , \tag{4.72}$$

which leads to $\mathrm{d}\boldsymbol{\mu} = \boldsymbol{\mathcal{A}}\,\mathrm{d}\boldsymbol{y}$. A detailed discussion of how such a class of problems can be solved is given in Chapter 6.

Finally, the future positions and velocities are computed as

$$\boldsymbol{q}_{n+1} = \boldsymbol{q}_n + (\boldsymbol{b}^{\mathrm{T}} \otimes \mathbf{1}_{n_q \times n_q})\mathrm{d}\boldsymbol{Q} , \quad \boldsymbol{u}_{n+1} = \boldsymbol{u}_n + (\boldsymbol{b}^{\mathrm{T}} \otimes \mathbf{1}_{n_u \times n_u})\mathrm{d}\boldsymbol{U} , \tag{4.73}$$

while the percussion increments are obtained by

$$\Delta\boldsymbol{\pi}_{\square,n+1} = \boldsymbol{\pi}_{\square,n+1} - \boldsymbol{\pi}_{\square,n} = (\boldsymbol{b}^{\mathrm{T}} \otimes \mathbf{1}_{n_\square \times n_\square})\mathrm{d}\boldsymbol{\Pi}_\square . \tag{4.74}$$

Since this investigation is restricted on stiffly accurate Runge–Kutta methods, $\boldsymbol{q}_{n+1} = \boldsymbol{Q}_s$, $\boldsymbol{u}_{n+1} = \boldsymbol{U}_s$ and $\Delta\boldsymbol{\pi}_{\square,s} = \Delta\boldsymbol{\Pi}_{\square,n+1} = \boldsymbol{\Pi}_{\square,s} - \boldsymbol{\pi}_{\square,n}$ are already available from the last internal stage.

Example 4.12 (One-dimensional Bouncing Ball)**.** Consider the simple example of a one-dimensional bouncing ball of mass m subjected to gravity with acceleration g. Its height is described by the coordinate y and its velocity is denoted by u. The system can be described by

$$\begin{aligned} \mathrm{d}y &= u\,\mathrm{d}t \\ m\,\mathrm{d}u &= -mg\,\mathrm{d}t + \mathrm{d}\pi_N \end{aligned} \tag{4.75}$$

and is subjected to frictionless contact with $g_N = y$. Application of an s-stage stiffly accurate Runge–Kutta method as described in (4.61) results in the discrete system

$$\mathrm{d}\boldsymbol{Y} = h\left(u_n \mathbf{1}_s + \boldsymbol{A}\mathrm{d}\boldsymbol{U}\right) \tag{4.76a}$$

$$\mathrm{d}\boldsymbol{U} = -hg\mathbf{1}_s + \tfrac{1}{m}\mathrm{d}\boldsymbol{\Pi}_N \tag{4.76b}$$

$$\begin{aligned} &\boldsymbol{Y} \in \mathcal{N}_{\mathbb{R}_0^-}(-\Delta\Pi_{N,1}) \times \cdots \times \mathcal{N}_{\mathbb{R}_0^-}(-\Delta\Pi_{N,s}) \\ &\qquad \Updownarrow \\ &0 \le \boldsymbol{Y} \perp \Delta\boldsymbol{\Pi}_N \ge 0\,, \end{aligned} \tag{4.76c}$$

with

$$\boldsymbol{Y} = y_n \mathbf{1}_s + \boldsymbol{A}\mathrm{d}\boldsymbol{Y}\,, \quad \boldsymbol{U} = u_n \mathbf{1}_s + \boldsymbol{A}\mathrm{d}\boldsymbol{U} \quad \text{and} \quad \Delta\boldsymbol{\Pi}_N = \boldsymbol{A}\mathrm{d}\boldsymbol{\Pi}_N\,. \tag{4.77}$$

Substituting (4.76b) into (4.76a), using $\mathrm{d}\boldsymbol{\Pi}_N = \boldsymbol{A}^{-1}\Delta\boldsymbol{\Pi}_N$ from (4.77) and introducing $\boldsymbol{w} \coloneqq \boldsymbol{Y}$ and $\boldsymbol{z} \coloneqq \Delta\boldsymbol{\Pi}_N$ reveals that the system can be recast as linear complementarity problem (LCP) of the form

$$\boldsymbol{w} = \boldsymbol{M}\boldsymbol{z} + \boldsymbol{q}\,, \quad 0 \le \boldsymbol{w} \perp \boldsymbol{z} \ge 0\,, \tag{4.78}$$

with

$$\boldsymbol{M} = \frac{h}{m}\boldsymbol{A} \quad \text{and} \quad \boldsymbol{q} = y_n \mathbf{1}_s + h\boldsymbol{A}\left[u_n \mathbf{1}_s - hg\boldsymbol{A}\mathbf{1}_s\right]\,. \tag{4.79}$$

According to Proposition 2.21, the LCP $(\boldsymbol{q}, \boldsymbol{M})$ defined by (4.78) and (4.79) possesses a unique solution if $\boldsymbol{M}$ is a P-matrix. Since $\frac{h}{m} > 0$, this requires that the coefficient matrix $\boldsymbol{A}$ of the underlying Runge–Kutta method is a P-matrix. It can be shown, that for all presented Runge–Kutta methods with invertible coefficient matrix $\boldsymbol{A}$ this condition is satisfied. For the one- and two-stage Radau IIA methods from Section 4.4 this leads to the matrices

$$\boldsymbol{A}_1 = \begin{pmatrix} 1 \end{pmatrix} \quad \text{and} \quad \boldsymbol{A}_2 = \begin{pmatrix} \frac{5}{12} & -\frac{1}{12} \\ \frac{3}{4} & \frac{1}{4} \end{pmatrix}\,. \tag{4.80}$$

Obviously, $\boldsymbol{A}_1$ is a P-matrix since it solely contains a single positive real number. As outlined in Example 2.20, $\boldsymbol{A}_2$ is again a P-matrix and hence, both methods possess a unique solution of the LCP $(\boldsymbol{q}, \boldsymbol{M})$. Besides the methods with invertible coefficient matrices, Section 4.4 also introduced methods with a first explicit stage. For those methods the values w_1 and z_1 are already known from the previous time-step and satisfy $0 \leq w_1 \perp z_1 \geq 0$. Hence, it is sufficient to solve the reduced LCP $(\overline{\boldsymbol{q}}, \overline{\boldsymbol{M}})$

$$\overline{\boldsymbol{w}} = \overline{\boldsymbol{M}}\overline{\boldsymbol{z}} + \overline{\boldsymbol{q}}\,, \quad 0 \leq \overline{\boldsymbol{w}} \perp \overline{\boldsymbol{z}} \geq 0\,, \tag{4.81}$$

defined in terms of

$$\overline{\boldsymbol{w}} = \begin{pmatrix} w_2 \\ w_3 \\ \vdots \\ w_s \end{pmatrix}, \quad \overline{\boldsymbol{z}} = \begin{pmatrix} z_2 \\ z_3 \\ \vdots \\ z_s \end{pmatrix}, \quad \overline{\boldsymbol{q}} = \begin{pmatrix} q_2 + a_{21} z_1 \\ q_3 + a_{31} z_1 \\ \vdots \\ q_s + a_{s1} z_1 \end{pmatrix} \tag{4.82}$$

and

$$\overline{\boldsymbol{M}} = \frac{h}{m} \boldsymbol{A}_{\alpha\alpha} = \frac{h}{m} \begin{pmatrix} a_{22} & a_{23} & \dots & a_{2s} \\ a_{32} & a_{33} & & a_{3s} \\ \vdots & & \ddots & \vdots \\ a_{s2} & a_{s3} & \dots & a_{ss} \end{pmatrix}. \tag{4.83}$$

Therein, the submatrix $\boldsymbol{A}_{\alpha\alpha}$ is defined by the set $\alpha = \{2, 3, \dots, s\}$. For the presented implicit Runge–Kutta methods with explicit first stage, the submatrix $\boldsymbol{A}_{\alpha\alpha}$ is again a P-matrix. And since $\frac{h}{m} > 0$, the reduced LCP $(\overline{\boldsymbol{q}}, \overline{\boldsymbol{M}})$ possesses again a unique solution.

As previously noted, enforcing $g_N^k(\tau_i, \boldsymbol{Q}_i) \in \mathcal{N}_{\mathbb{R}_0^-}(-\mathrm{d}\Pi_{N,i}^k)$ is not advisable. Performing the very same steps as outlined before would lead to $\boldsymbol{z} \coloneqq \mathrm{d}\boldsymbol{\Pi}$ and $\boldsymbol{M} = \frac{h}{m}\boldsymbol{A}^2$. As shown in Example 2.20, choosing $\boldsymbol{A}$ form the two-stage Radau IIA method would result in $\boldsymbol{A}^2$ not being a P-matrix. Consequently, $\boldsymbol{M}$ is not a P-matrix and the contact problem described by the LCP $(\boldsymbol{q}, \boldsymbol{M})$ from (4.78) does not possess a unique solution for this method, even for such a simple example and in the absence of friction.

Without loss of generality let $m = 1$, $g = 1$ and $u_n = -1$. Subsequently, two consecutive time-steps from $t_n \mapsto t_{n+1}$ and $t_{n+1} \mapsto t_{n+2}$ are investigated, depending on the initial position y_n and the step-size h for the single- and two-stage Radau IIA method.

Single-stage Radau IIA (backward Euler) Since $s = 1$, for each time-step $2^s = 2$ possible solutions of the underlying LCP have to be investigated. They will subsequently be labeled by

- **case 1:** $z_1 \geq 0$ and $w_1 = 0$
- **case 2:** $z_1 = 0$ and $w_1 \geq 0$

Since for this investigation two consecutive time-steps are of interest, the labels are combined, e.g., case 1.2 denotes that in the first time-step $t_n \mapsto t_{n+1}$ case 1 is chosen, while in the second time-step $t_{n+1} \mapsto t_{n+2}$ case 2 is applied. This results in a totality of $(2^s)^2 = 4$ possible solutions, depending on the chosen values of y_n and h. Evaluating (4.79) for the given initial conditions and the single-stage Radau IIA method (backward Euler) results in

$$M = h \quad \text{and} \quad q = y_n - h - h^2 \,. \tag{4.84}$$

All possible solutions can be computed by setting either $z_1 = 0$ or $w_1 = 0$ in the LCP (4.78) and solving for the remaining unknown. Subsequently, the feasibility of the solution can be investigated. For the first time-step this gives

- **case 1:** Feasible for $0 \leq y_n \leq h + h^2$ with

$$z_{n+1}^1 = 1 + h - \frac{y_n}{h} \,, \quad w_{n+1}^1 = 0 \,, \quad u_{n+1} = -\frac{y_n}{h} \,. \tag{4.85}$$

- **case 2:** Feasible for $y_n \geq h + h^2$ with

$$z_{n+1}^1 = 0 \,, \quad w_{n+1}^1 = y_n - h - h^2 \,, \quad u_{n+1} = -h - 1 \,, \tag{4.86}$$

  but no contact occurs since $w_{n+1}^1 \geq 0$.

Since the second case does not result in contact, the subsequent investigation is restricted on the first case only. Application of a similar procedure subsequent to case 1 with

$$q = h \left(\frac{y_{n+1}}{h} + u_{n+1} - h \right) \tag{4.87}$$

results in two feasible combined cases:

- **case 1.1:** Feasible for $h, y_n \geq 0$ with

$$z_{n+2} = h + \frac{y_n}{h} \,, \quad w_{n+2} = 0 \,, \quad u_{n+1} = 0 \,. \tag{4.88}$$

- **case 1.2:** Infeasible since $0 \leq y_n \leq -h^2$ with

$$z_{n+2} = 0 \,, \quad w_{n+2} = -h^2 - y_n \,, \quad u_{n+1} = -\frac{y_n}{h} \,. \tag{4.89}$$

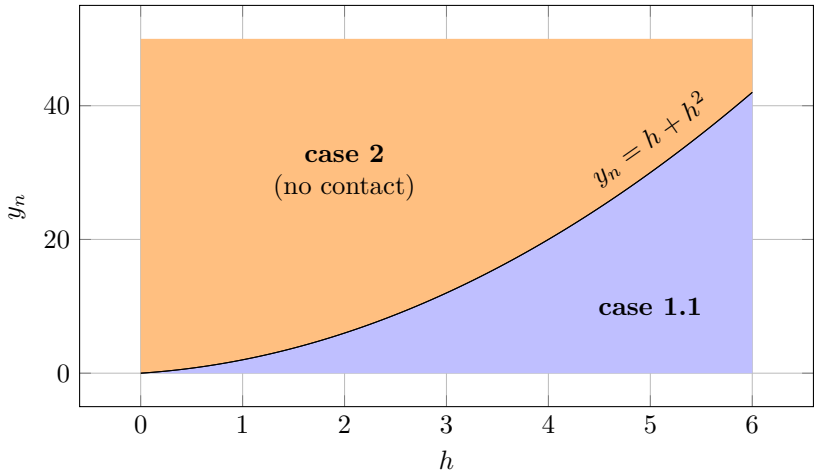

Figure 4.4: Feasible initial conditions for the one-dimensional bouncing ball example solved by two consecutive single-stage Radau IIA steps.

Its clear from this investigation that either no contact occurs (case 2) or the only feasible combined case 1.1 is obtained, see Figure 4.4. Further it is remarkable that according to (4.88), the single-stage Radau IIA (backward Euler) method performs an inelastic impact law over two consecutive time-steps. This is in line with the investigations of Paoli and Schatzman [138, 139], which investigated the enforcement of impact laws on position level over two consecutive time-steps.

Two-stage Radau IIA Since $s = 2$, for each time-step $2^2 = 4$ possible solutions of the underlying LCP have to be investigated. They will subsequently be labeled by

- **case 1:** $z_1 \geq 0$, $z_2 \geq 0$, $w_1 = 0$ and $w_2 = 0$
- **case 2:** $z_1 \geq 0$, $z_2 = 0$, $w_1 = 0$ and $w_2 \geq 0$
- **case 3:** $z_1 = 0$, $z_2 \geq 0$, $w_1 \geq 0$ and $w_2 = 0$
- **case 4:** $z_1 = 0$, $z_2 = 0$, $w_1 \geq 0$ and $w_2 \geq 0$

Similar to the investigation above, two consecutive time-steps are performed. Therefore, the labels are combined, e.g., case 1.4 denotes that in the first time-step $t_n \mapsto t_{n+1}$ case 1 is chosen, while in the second time-step $t_{n+1} \mapsto t_{n+2}$ case 4 is applied. This results in a totality of $(2^s)^2 = 16$ possible solutions, depending on the chosen values of y_n and h. Evaluating (4.79)

for the given initial conditions and the two-stage Radau IIA method results in

$$\boldsymbol{M} = h \begin{pmatrix} \frac{5}{12} & -\frac{1}{12} \\ \frac{3}{4} & \frac{1}{4} \end{pmatrix} \quad \text{and} \quad \boldsymbol{q} = h \begin{pmatrix} \frac{y_n}{h} + \frac{u_n}{3} - \frac{h}{18} \\ \frac{y_n}{h} + u_n - \frac{h}{2} \end{pmatrix} . \tag{4.90}$$

All possible solutions can be computed by eliminating the known values from the LCP (4.78) and solving for the remaining unknowns. Subsequently, the feasibility of the solution can be investigated. For the first time-step this gives

- **case 1:** Feasible for $0 \leq y_n \leq \frac{h^2}{6} + \frac{h}{2}$ with

$$\boldsymbol{z}^1_{n+1} = \begin{pmatrix} -\frac{2y_n}{h} + \frac{h}{3} + 1 \\ \frac{2y_n}{h} + h + 1 \end{pmatrix} , \quad \boldsymbol{w}^1_{n+1} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} . \tag{4.91}$$

- **case 2:** Infeasible since $0 > -\frac{h^2}{2} - \frac{h}{2} \geq y_n$ with

$$\boldsymbol{z}^2_{n+1} = \begin{pmatrix} \frac{-36y_n+2h^2+12h}{15h} \\ 0 \end{pmatrix} , \quad \boldsymbol{w}^2_{n+1} = \begin{pmatrix} 0 \\ -\frac{4y_n+2h^2+2h}{5} \end{pmatrix} . \tag{4.92}$$

- **case 3:** Feasible for $\frac{h^2}{6} + \frac{h}{2} \leq y_n \leq \frac{h^2}{2} + h$ with

$$\boldsymbol{z}^3_{n+1} = \begin{pmatrix} 0 \\ \frac{4h+2h^2-4y_n}{h} \end{pmatrix} , \quad \boldsymbol{w}^3_{n+1} = \begin{pmatrix} \frac{12y_n-6h-2h^2}{9} \\ 0 \end{pmatrix} . \tag{4.93}$$

- **case 4:** Feasible for $y_n \geq \frac{h^2}{2} + h$ with

$$\boldsymbol{z}^4_{n+1} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} , \quad \boldsymbol{w}^4_{n+1} = \begin{pmatrix} \frac{18y_n-h^2-6h}{18} \\ \frac{2y_n-h^2-2h}{2} \end{pmatrix} , \tag{4.94}$$

  but no contact occurs, since for feasible initial heights $y_n \geq \frac{h^2}{2} + h$ the second component of $\boldsymbol{w}^4_{n+1}$ is nonnegative.

Since case 2 is infeasible and case 4 does not result in a contact, the subsequent investigation is restricted on the cases 1 and 3. Moreover, according to the feasibility of case 4, an upper bound for the initial conditions that still results in contact is characterized by

$$y^*_n \leq \frac{h^2}{2} + h \, . \tag{4.95}$$

Application of a similar procedure subsequent to cases 1 and 3 with

$$\boldsymbol{q} = h \begin{pmatrix} \frac{y_{n+1}}{h} + \frac{u_{n+1}}{3} - \frac{h}{18} \\ \frac{y_{n+1}}{h} + u_{n+1} - \frac{h}{2} \end{pmatrix} \tag{4.96}$$

results in six feasible combined cases:

- **case 1.1:** Feasible for $0 \leq y_n \leq \frac{h^2}{6}$ with

$$\boldsymbol{z}_{n+2} = \begin{pmatrix} \frac{h^2-6y_n}{3h} \\ \frac{h^2-2y_n}{h} \end{pmatrix}, \quad \boldsymbol{w}_{n+2} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{4.97}$$

- **case 1.3:** Feasible for $\frac{h^2}{6} \leq y_n \leq \min(\frac{h^2}{4}, \frac{h^2}{6} + \frac{h}{2})$ with

$$\boldsymbol{z}_{n+2} = \begin{pmatrix} 0 \\ \frac{2h^2-8y_n}{h} \end{pmatrix}, \quad \boldsymbol{w}_{n+2} = \begin{pmatrix} \frac{12y_n-2h^2}{9} \\ 0 \end{pmatrix}. \tag{4.98}$$

- **case 1.4:** Feasible for $\max(\frac{h^2}{4}, \frac{h^2}{6} + \frac{h}{2}) \leq y_n \leq \frac{h^2}{6} + \frac{h}{2}$ with

$$\boldsymbol{z}_{n+2} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \boldsymbol{w}_{n+2} = \begin{pmatrix} \frac{12y_n-h^2}{18} \\ \frac{4y_n-h^2}{2} \end{pmatrix}. \tag{4.99}$$

- **case 3.1:** Feasible for $\frac{h^2}{6} + \frac{3h}{4} \leq y_n \leq \frac{h^2}{2} + h$ with

$$\boldsymbol{z}_{n+2} = \begin{pmatrix} \frac{12y_n-9h-2h^2}{3h} \\ \frac{4y_n-3h}{h} \end{pmatrix}, \quad \boldsymbol{w}_{n+2} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{4.100}$$

- **case 3.3:** Feasible for $\max(\frac{h^2}{8} + \frac{3h}{4}, \frac{h^2}{6} + \frac{h}{2}) \leq y_n \leq \frac{h^2}{6} + \frac{3h}{4}$ with

$$\boldsymbol{z}_{n+2} = \begin{pmatrix} 0 \\ \frac{16y_n-12h-2h^2}{h} \end{pmatrix}, \quad \boldsymbol{w}_{n+2} = \begin{pmatrix} \frac{4h^2+18h-24y_n}{9} \\ 0 \end{pmatrix}. \tag{4.101}$$

- **case 3.4:** Feasible for $\frac{h^2}{6} + \frac{h}{2} \leq y_n \leq \frac{h^2}{8} + \frac{3h}{4}$ with

$$\boldsymbol{z}_{n+2} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \boldsymbol{w}_{n+2} = \begin{pmatrix} \frac{5h^2+18h-24y_n}{18} \\ \frac{h^2+6h-8y_n}{2} \end{pmatrix}. \tag{4.102}$$

All these cases are summarized in Figure 4.5, where a normalized initial height y_n/y_n^* is depicted in order to visually remove the domain where no contact happens in the first time-step.

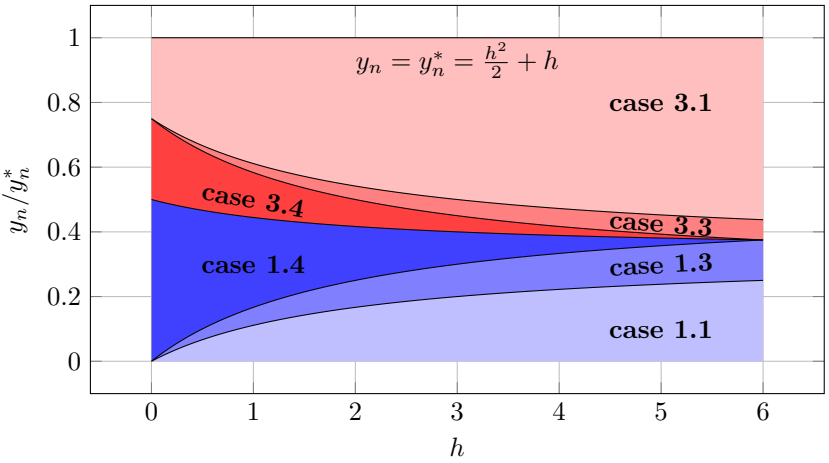

Figure 4.5: Feasible initial conditions for the one-dimensional bouncing ball example that result in contact, solved by two consecutive two-stage Radau IIA steps.

For the specific choice $h = 3$, Figure 4.6 shows the positions y_{n+2} and velocities u_{n+2} after two consecutive time-steps of the proposed method for different initial conditions that result in contact within the first time-step. It can be observed that only for the cases 1.1 and 3.1 ($y_n \in [0, \frac{3}{2}] \cup [\frac{90}{24}, \frac{15}{2}]$), the method performs an inelastic impact ($y_{n+2} = u_{n+2} = 0$) over two consecutive time-steps. For the cases 1.3 and 3.3 ($y_n \in [\frac{3}{2}, \frac{9}{4}] \cup [\frac{27}{8}, \frac{90}{24}]$) the ball stays on the ground with $y_{n+2} = 0$ but its velocity u_{n+2} is negative, clearly violating physics. Moreover, this violates Proposition 2.2 of [128], i.e., for closed or penetrated contacts the post impact velocity has to be non-negative. More dramatically, for the cases 1.4 and 3.4 ($y_n \in [\frac{9}{4}, \frac{27}{8}]$), the ball departs from the ground with different velocities. Consequently, an arbitrary elastic impact law is realized, depending on the chosen initial height y_n. For specific initial heights y_n chosen from the respective cases, the numerically obtained values are depicted in Figure 4.7, together with the corresponding collocation polynomial of the two-stage Radau IIA method, i.e., $y(t_n + \tau h) = l_0(\tau)y_n + l_1(\tau)Y_1 + l_2(\tau)Y_2$ with l_i being the Lagrange basis functions for nodes $c_0 = 0$ and c_i from (4.1) for $i > 0$. Once more, the observation can be made that the proposed method realizes an arbitrary impact law depending on the chosen initial height.

Clearly, the naive extension of higher-order implicit Runge–Kutta methods to nonsmooth mechanical systems is flawed, justifying the name of this

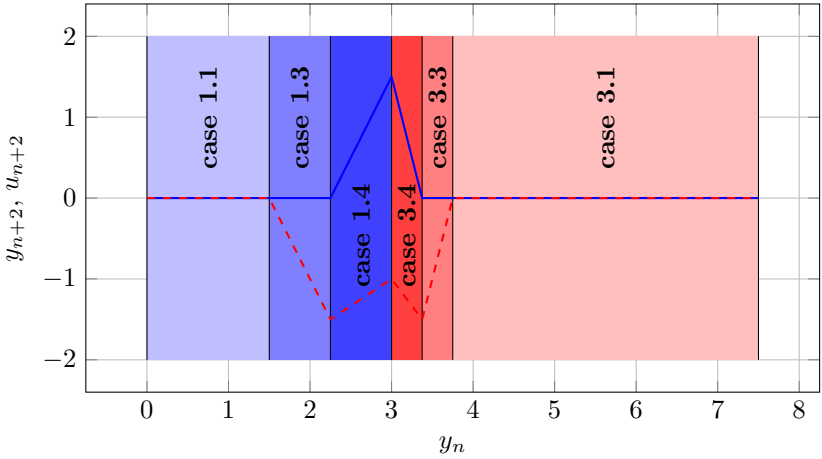

Figure 4.6: Evolution of positions and velocities after two consecutive time-steps of the two-stage Radau IIA method with $h = 3$, different initial heights y_n and contact in the first time-step. Positions y_{n+2} are drawn by (——) and velocities u_{n+2} by (- - -).

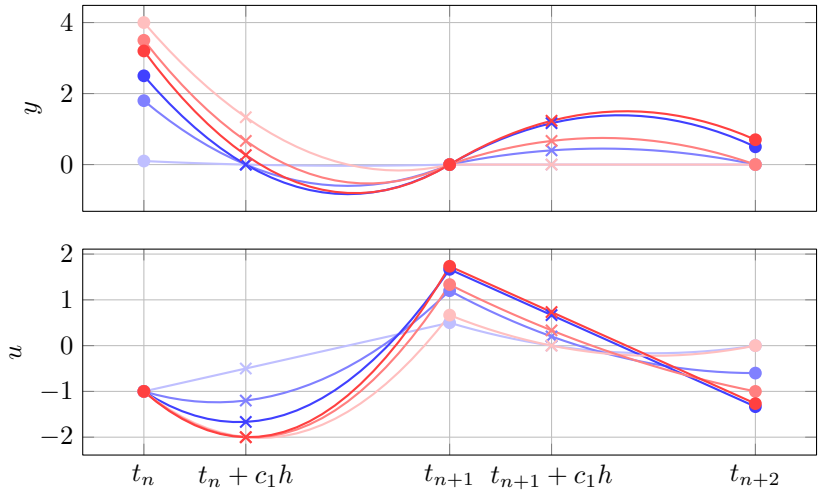

Figure 4.7: Time evolution of position and velocity of two consecutive time-steps of the two-stage Radau IIA method with different initial conditions and $h = 3$. Grid points and internal stages are depicted by bullets and crosses, respectively. Solid lines show the collocation polynomial. The colored lines coincide with the regions of Figure 4.5 and 4.6.

section. The investigated two-stage method does not perform an inelastic impact law as it is the case for the simple single-stage method. Moreover, it is not possible to realize an intended elastic impact law. Both results are not very surprising, since an explicit impact law was never added to the discrete system. To close this ambiguity, the next section introduces projected Runge–Kutta methods for nonsmooth mechanical systems. Those methods are capable to enforce a desired impact law and entirely remove the discussed ambiguities.

4.6 Nonsmooth Projected Runge–Kutta Methods

The aforementioned ambiguities can be eliminated by application of a nonsmooth projected stiffly accurate Runge–Kutta method with extended Butcher tableau already introduced in (4.33). Performing the very same steps as in the bilaterally constrained case (4.35), for $\varepsilon \to 0$, the additionally introduced last stage results in the velocity projection

$$\begin{aligned} \delta \boldsymbol{u}_{n+1} &= \boldsymbol{l}(t_{n+1}, \boldsymbol{q}_{n+1}) \delta \boldsymbol{\pi}_{n+1} && \text{(4.103a)} \\ \mathbf{0}_{n_g} &= \dot{\boldsymbol{g}}(t_{n+1}, \boldsymbol{q}_{n+1}, \boldsymbol{u}_{n+1} + \delta \boldsymbol{u}_{n+1}) && \text{(4.103b)} \\ \mathbf{0}_{n_\gamma} &= \boldsymbol{\gamma}(t_{n+1}, \boldsymbol{q}_{n+1}, \boldsymbol{u}_{n+1} + \delta \boldsymbol{u}_{n+1}) \,. && \text{(4.103c)} \end{aligned}$$

As in the bilaterally constrained case, the positions $\boldsymbol{q}_{n+1}^+ = \boldsymbol{q}_{n+1}$ are not affected by this projection. The final velocities and percussions are given by $\boldsymbol{u}_{n+1}^+ = \boldsymbol{u}_{n+1} + \delta \boldsymbol{u}_{n+1}$ and $\boldsymbol{\pi}_{n+1}^+ = \boldsymbol{\pi}_{n+1} + \delta \boldsymbol{\pi}_{n+1}$, respectively. Investigating the resulting quadrature formula reveals a decomposition of the velocity measures quadrature as

$$\begin{aligned} \boldsymbol{u}_{n+1}^+ &= \boldsymbol{u}_n + \lim_{\varepsilon \to 0} \int\limits_{(t_n, t_{n+1} + \varepsilon h]} \mathrm{d}\boldsymbol{u} = \boldsymbol{u}_n + \int\limits_{(t_n, t_{n+1})} \mathrm{d}\boldsymbol{u} + \int\limits_{\{t_{n+1}\}} \mathrm{d}\boldsymbol{u} \\ &\approx \boldsymbol{u}_n + \sum_{i=1}^{s} b_i \mathrm{d}\boldsymbol{U}_i + \delta \boldsymbol{u}_{n+1} = \boldsymbol{u}_{n+1} + \delta \boldsymbol{u}_{n+1} \,. \end{aligned} \tag{4.104}$$

Since the percussions are integrated using the same quadrature formula, a similar decomposition as already seen for the velocities is obtained. Specifically,

$$\begin{aligned} \boldsymbol{\pi}_{n+1}^+ &= \boldsymbol{\pi}_n + \lim_{\varepsilon \to 0} \int\limits_{(t_n, t_{n+1} + \varepsilon h]} \mathrm{d}\boldsymbol{\pi} = \boldsymbol{\pi}_n + \int\limits_{(t_n, t_{n+1})} \mathrm{d}\boldsymbol{\pi} + \int\limits_{\{t_{n+1}\}} \mathrm{d}\boldsymbol{\pi} \\ &\approx \boldsymbol{\pi}_n + \sum_{i=1}^{s} b_i \mathrm{d}\boldsymbol{\Pi}_i + \delta \boldsymbol{\pi}_{n+1} = \boldsymbol{\pi}_{n+1} + \delta \boldsymbol{\pi}_{n+1} \,. \end{aligned} \tag{4.105}$$

The obtained decompositions beautifully encode that the additional stage of the nonsmooth projected stiffly accurate Runge–Kutta method can be interpreted as the missing term of the measure decompositions (3.3) and (3.5) that accounts for impulsive changes of the velocities $\boldsymbol{u}$ and percussions $\boldsymbol{\pi}$. Moreover, these impulsive changes are condensed onto the singleton $\{t_{n+1}\}$. The key idea is now to augment this velocity projection by unilateral constraint conditions on velocity level and incorporate impact laws for normal and tangent directions over one time-step $I_n = (t_n, t_{n+1}]$.

In what follows, it is assumed that if the contact k is active at the end of the time-step, i.e., $k \in A_{n+1} = A(t_{n+1}, \boldsymbol{q}_{n+1})$, then it has been active during the whole time-step $I_n = (t_n, t_{n+1}] = \lim_{\varepsilon \to 0} J_{s+1}$, with $J_{s+1} = (t_n, \tau_s + \varepsilon] = (t_n, t_n + (c_s + \varepsilon)h]$. Since all the presented methods share $c_s = 1$ this coincides with $J_{s+1} = (t_n, t_{n+1} + \varepsilon h]$. Furthermore, it is assumed that ξ_N^k is constant on I_n and corresponds to

$$\xi_{N,n+1}^k = \dot{g}_N^k(t_{n+1}, \boldsymbol{q}_{n+1}, \boldsymbol{u}_{n+1}^+) + e_N^k \, \dot{g}_N^k(t_n, \boldsymbol{q}_n, \boldsymbol{u}_n^+)\,, \tag{4.106}$$

which is regarded as a discrete approximation of (3.44) over one time-step, see [29, 42, 74]. The position and velocity of the system is continuous between velocity jumps, implying $\dot{g}_N^{k+} = \dot{g}_N^{k-} = \dot{g}_N^k$ for almost all t. Hence, in such cases it holds that

$$(1 + e_N^k)\dot{g}_N^k = \dot{g}_N^{k+} + e_N^k \dot{g}_N^{k-} = \xi_N^k \tag{4.107}$$

and Signorini's law can be written on velocity level (3.15) using ξ_N^k instead of $\dot{g}_N^k$. Using all these preliminary assumptions, together with Proposition 2.3, allows to combine Signorini's law on velocity level (3.15) and the Newton-type impact law in normal direction (3.45) as (see [41])

$$\begin{cases} \xi_{N,n+1}^k \in \mathcal{N}_{\mathbb{R}_0^-}(-\int_{I_n} \mathrm{d}\pi_N^k) & \text{if } k \in A_{n+1}\,, \\ 0 = \int_{I_n} \mathrm{d}\pi_N^k & \text{if } k \notin A_{n+1}\,. \end{cases} \tag{4.108}$$

Substitution of $\int_{I_n} \mathrm{d}\pi_N^k \approx \sum_{i=1}^{s+1} \hat{b}_i \mathrm{d}\Pi_{N,i}^k = \sum_{i=1}^{s} b_i \mathrm{d}\Pi_{N,i}^k + \delta\pi_{N,n+1}^k = \Delta\pi_{N,n+1}^{k+} = \pi_{N,n+1}^{k+} - \pi_{N,n}^k = \Delta\pi_{N,n+1}^k + \delta\pi_{N,n+1}^k$ results in the discrete impact law. A discussion of how this is done in detail as well as an interpretation of the integrated contact law can be found in Section 6 of [41].

For the discretization of the set-valued friction laws a similar path is taken. Using the same assumptions as above, the friction law (3.20) can be combined with the Newton-type impact law in tangential direction (3.47) as

$$\boldsymbol{\xi}_{F,n+1}^k \in \mathcal{N}_{C_F^k(\int_{I_n} \mathrm{d}\pi_N^k)}(-\textstyle\int_{I_n} \mathrm{d}\boldsymbol{\pi}_F^k)\,, \tag{4.109}$$

where (3.46) is discretized as

$$\boldsymbol{\xi}^k_{F,n+1} = \boldsymbol{\gamma}_F(t_{n+1}, \boldsymbol{q}_{n+1}, \boldsymbol{u}^+_{n+1}) + e^k_F \, \boldsymbol{\gamma}^k_F(t_n, \boldsymbol{q}_n, \boldsymbol{u}^+_n) \tag{4.110}$$

and the approximation $\int_{I_n} \mathrm{d}\boldsymbol{\pi}^k_F \approx \sum_{i=1}^{s+1} \hat{b}_i \mathrm{d}\boldsymbol{\Pi}^k_{F,i} = \sum_{i=1}^{s} b_i \mathrm{d}\boldsymbol{\Pi}^k_{F,i} + \delta\boldsymbol{\pi}^k_{F,n+1}$ $= \Delta\boldsymbol{\pi}^{k+}_{F,n+1} = \boldsymbol{\pi}^{k+}_{F,n+1} - \boldsymbol{\pi}^k_{F,n} = \Delta\boldsymbol{\pi}^k_{F,n+1} + \delta\boldsymbol{\pi}^k_{F,n+1}$ is used.

Let $\boldsymbol{x} = (\delta\boldsymbol{u}_{n+1}, \delta\boldsymbol{\pi}_{g,n+1}, \delta\boldsymbol{\pi}_{\gamma,n+1}) \in \mathbb{R}^{n_u+n_g+n_\gamma}$ and $\boldsymbol{y} = (\delta\boldsymbol{\pi}_{N,n+1}, \delta\boldsymbol{\pi}_{F,n+1}) \in \mathbb{R}^{n_N+n_F}$. The proposed nonsmooth projection stage is given by solving the linear system $\boldsymbol{\varphi}(\boldsymbol{x}, \boldsymbol{y}) = \mathbf{0}_{n_u+n_g+n_\gamma}$, where

$$\begin{aligned} \boldsymbol{\varphi}(\boldsymbol{x}, \boldsymbol{y}) = &\begin{pmatrix} \boldsymbol{M}_{n+1} & -\boldsymbol{W}_{g,n+1} & -\boldsymbol{W}_{\gamma,n+1} \\ -\boldsymbol{W}^{\mathrm{T}}_{g,n+1} & \mathbf{0}_{n_g \times n_g} & \mathbf{0}_{n_g \times n_\gamma} \\ -\boldsymbol{W}^{\mathrm{T}}_{\gamma,n+1} & \mathbf{0}_{n_\gamma \times n_g} & \mathbf{0}_{n_\gamma \times n_\gamma} \end{pmatrix} \begin{pmatrix} \delta\boldsymbol{u}_{n+1} \\ \delta\boldsymbol{\pi}_{g,n+1} \\ \delta\boldsymbol{\pi}_{\gamma,n+1} \end{pmatrix} \\ &- \begin{pmatrix} \boldsymbol{W}_{N,n+1}\delta\boldsymbol{\pi}_{N,n+1} + \boldsymbol{W}_{F,n+1}\delta\boldsymbol{\pi}_{F,n+1} \\ \dot{\boldsymbol{g}}(t_{n+1}, \boldsymbol{q}_{n+1}, \boldsymbol{u}_{n+1}) \\ \boldsymbol{\gamma}(t_{n+1}, \boldsymbol{q}_{n+1}, \boldsymbol{u}_{n+1}) \end{pmatrix}, \end{aligned} \tag{4.111}$$

subject to the normal cone inclusions

$$\boldsymbol{\psi}^k(\boldsymbol{x}) \in \mathcal{K}^k(\boldsymbol{y}) \quad \text{for} \quad k \in A_{n+1}\,. \tag{4.112}$$

Therein, the left-hand side of the inclusion reads

$$\boldsymbol{\psi}^k(\boldsymbol{x}) = \begin{pmatrix} \xi^k_{N,n+1} \\ \boldsymbol{\xi}^k_{F,n+1} \end{pmatrix} \tag{4.113}$$

and the set is given by

$$\mathcal{K}^k(\boldsymbol{y}) = \mathcal{N}_{\mathbb{R}^-_0}(-\Delta\pi^{k+}_{N,n+1}) \times \mathcal{N}_{C^k_F(\Delta\pi^{k+}_{N,n+1})}(-\Delta\boldsymbol{\pi}^{k+}_{F,n+1})\,. \tag{4.114}$$

Moreover, the percussion differences $\Delta\pi^{k+}_{N,n+1} = \Delta\pi^k_{N,n+1} + \delta\pi^k_{N,n+1}$ and $\Delta\boldsymbol{\pi}^{k+}_{F,n+1} = \Delta\boldsymbol{\pi}^k_{F,n+1} + \delta\boldsymbol{\pi}^k_{F,n+1}$ are used.

Putting it all together, the naive approach introduced in Section 4.5 can be combined with the above discussed velocity projection resulting in the overall nonsmooth projected stiffly accurate Runge–Kutta method that is summarized in Figure 4.8.

Example 4.13 (One-dimensional Bouncing Ball continued)**.** Using the nonsmooth projected stiffly accurate Runge–Kutta methods introduced above, the example of the one-dimensional bouncing can be solved without any issues, see Figure 7.1. Furthermore, in Section 7.1, the rotating bouncing ball is investigated in more detail, continuing this simple investigation.

Nonsmooth projected stiffly accurate Runge–Kutta

Stage 1: Solve the nonlinear system

$$\begin{aligned} \mathrm{d}\boldsymbol{Q}_i &= h\boldsymbol{f}(\tau_i, \boldsymbol{Q}_i, \boldsymbol{U}_i)\,, & \mathrm{d}\boldsymbol{U}_i &= h\boldsymbol{k}(\tau_i, \boldsymbol{Q}_i, \boldsymbol{U}_i) + \boldsymbol{l}(\tau_i, \boldsymbol{Q}_i)\mathrm{d}\boldsymbol{\Pi}_i\,, \\ \boldsymbol{0}_{n_g} &= \boldsymbol{g}(\tau_i, \boldsymbol{Q}_i)\,, & \boldsymbol{0}_{n_\gamma} &= \boldsymbol{\gamma}(\tau_i, \boldsymbol{Q}_i, \boldsymbol{U}_i)\,, \end{aligned}$$

for $i = 1, \dots, s$, where $\tau_i = t_n + c_i h$, $\boldsymbol{Q}_i = \boldsymbol{q}_n + \sum_{j=1}^{s} a_{ij}\mathrm{d}\boldsymbol{Q}_j$ and $\boldsymbol{U}_i = \boldsymbol{u}_n + \sum_{j=1}^{s} a_{ij}\mathrm{d}\boldsymbol{U}_j$, subject to the normal cone inclusions

$$\begin{pmatrix} g_N^k(\tau_i, \boldsymbol{Q}_i) \\ \boldsymbol{\gamma}_F^k(\tau_i, \boldsymbol{Q}_i, \boldsymbol{U}_i) \end{pmatrix} \in \mathcal{N}_{\mathbb{R}_0^-}(-\mathrm{d}\mu_{N,i}^k) \times \mathcal{N}_{C_F^k(\mathrm{d}\mu_{N,i}^k)}(-\mathrm{d}\boldsymbol{\mu}_{F,i}^k)$$

for $k = 1, 2, \dots, n_N$ and $i = 1, \dots, s$, together with $\mathrm{d}\boldsymbol{\mu} = \boldsymbol{\mathcal{A}}\mathrm{d}\boldsymbol{y}$, $\mathrm{d}\boldsymbol{y} = (\mathrm{d}\boldsymbol{\Pi}_N, \mathrm{d}\boldsymbol{\Pi}_F)$ and

$$\boldsymbol{\mathcal{A}} = \boldsymbol{A} \otimes \begin{pmatrix} \boldsymbol{1}_{n_N \times n_N} & \boldsymbol{0}_{n_N \times n_F} \\ \boldsymbol{0}_{n_F \times n_N} & \boldsymbol{1}_{n_F \times n_F} \end{pmatrix}.$$

Set $\boldsymbol{q}_{n+1} = \boldsymbol{Q}_s$ and $\boldsymbol{u}_{n+1} = \boldsymbol{U}_s$.

Stage 2: Solve the linear system

$$\begin{pmatrix} \boldsymbol{M}_{n+1} & -\boldsymbol{W}_{g,n+1} & -\boldsymbol{W}_{\gamma,n+1} \\ -\boldsymbol{W}_{g,n+1}^{\mathrm{T}} & \boldsymbol{0}_{n_g \times n_g} & \boldsymbol{0}_{n_g \times n_\gamma} \\ -\boldsymbol{W}_{\gamma,n+1}^{\mathrm{T}} & \boldsymbol{0}_{n_\gamma \times n_g} & \boldsymbol{0}_{n_\gamma \times n_\gamma} \end{pmatrix} \begin{pmatrix} \delta\boldsymbol{u}_{n+1} \\ \delta\boldsymbol{\pi}_{g,n+1} \\ \delta\boldsymbol{\pi}_{\gamma,n+1} \end{pmatrix} = \begin{pmatrix} \boldsymbol{W}_{N,n+1}\delta\boldsymbol{\pi}_{N,n+1} + \boldsymbol{W}_{F,n+1}\delta\boldsymbol{\pi}_{F,n+1} \\ \dot{\boldsymbol{g}}(t_{n+1}, \boldsymbol{q}_{n+1}, \boldsymbol{u}_{n+1}) \\ \boldsymbol{\gamma}(t_{n+1}, \boldsymbol{q}_{n+1}, \boldsymbol{u}_{n+1}) \end{pmatrix},$$

subject to the normal cone inclusions

$$\begin{pmatrix} \xi_{N,n+1}^k \\ \boldsymbol{\xi}_{F,n+1}^k \end{pmatrix} \in \mathcal{N}_{\mathbb{R}_0^-}(-\Delta\pi_{N,n+1}^{k+}) \times \mathcal{N}_{C_F^k(\Delta\pi_{N,n+1}^{k+})}(-\Delta\boldsymbol{\pi}_{F,n+1}^{k+})$$

for $k \in A_{n+1}$. With $\boldsymbol{u}_{n+1}^+ = \boldsymbol{u}_{n+1} + \delta\boldsymbol{u}_{n+1}$ and $\Delta\boldsymbol{\pi}_{n+1}^+ = \sum_{i=1}^{s} b_i \mathrm{d}\boldsymbol{\Pi}_i + \delta\boldsymbol{\pi}_{n+1}$, the involved kinematic quantities are given by $\xi_{N,n+1}^k = \dot{g}_N^k(t_{n+1}, \boldsymbol{q}_{n+1}, \boldsymbol{u}_{n+1}^+) + e_N^k\, \dot{g}_N^k(t_n, \boldsymbol{q}_n, \boldsymbol{u}_n^+)$ and $\boldsymbol{\xi}_{F,n+1}^k = \boldsymbol{\gamma}_F(t_{n+1}, \boldsymbol{q}_{n+1}, \boldsymbol{u}_{n+1}^+) + e_F^k\, \boldsymbol{\gamma}_F^k(t_n, \boldsymbol{q}_n, \boldsymbol{u}_n^+)$.

Figure 4.8: Summary of nonsmooth projected stiffly accurate Runge–Kutta methods.

Chapter 5

Nonsmooth Partitioned Runge–Kutta Methods

> The Eminent Professor asked us what method of integration we had tried. "Runge–Kutta," we replied. "Runge–Kutta! Runge–Kutta!" he exclaimed, banging his head. "That's all you physicists know!"
>
> —*William Henry Press, Saul Arno Teukolsky*

As outlined in the previous chapter, the extension of higher-order Runge–Kutta methods to nonsmooth mechanical systems requires the introduction of an additional velocity projection stage in order to ensure that the desired impact laws are satisfied within each time-step. Such a projection stage was motivated by projected Runge–Kutta methods for differential algebraic equations. Another approach that ensures that constraint conditions are satisfied on position and velocity level are so-called partitioned Runge–Kutta methods. Classically, these methods are used for the simulation of Hamiltonian systems, due to their excellent long-term energy behavior. The extension of these methods to nonsmooth mechanical systems will be investigated in this chapter.[1] For mechanical systems with trivial kinematic differential equation $\dot{\boldsymbol{q}} = \boldsymbol{u}$, a similar method is already proposed in [42], based on an early draft of the nonsmooth RATTLE algorithm [29]. As it will turn out, the latter is a special case of the two-stage Lobatto IIIA-IIIB method, which is proposed herein along with other prominent

[1] To keep this chapter self-consistent and only based on a minimal number of required concepts, some argument from the proceeding chapter are recapitulated here.

examples belonging to the class of partitioned Runge–Kutta methods. Their numerically observed order of convergence applied to differential algebraic equations will be investigated within the already introduced pendulum example at the end of the chapter.

5.1 Partitioned Differential Equations

To introduce the basic properties of partitioned Runge–Kutta methods when applied to ordinary differential equations, this section focuses on mechanical systems of the form $\boldsymbol{M}(t,\boldsymbol{q})\dot{\boldsymbol{u}} = \boldsymbol{h}(t,\boldsymbol{q},\boldsymbol{u})$, i.e., systems without bilateral or unilateral constraints, as presented in Chapter 3. Consequently, the kinematic relation (3.1) and the equality of measures (3.2) can be recast as

$$\dot{\boldsymbol{q}} = \boldsymbol{f}(t,\boldsymbol{q},\boldsymbol{u})\,, \quad \boldsymbol{f}(t,\boldsymbol{q},\boldsymbol{u}) = \boldsymbol{B}(t,\boldsymbol{q})\boldsymbol{u} + \boldsymbol{\beta}(t,\boldsymbol{q}) \tag{5.1a}$$

$$\dot{\boldsymbol{u}} = \boldsymbol{k}(t,\boldsymbol{q},\boldsymbol{u})\,, \quad \boldsymbol{k}(t,\boldsymbol{q},\boldsymbol{u}) = \boldsymbol{M}(t,\boldsymbol{q})^{-1}\boldsymbol{h}(t,\boldsymbol{q},\boldsymbol{u})\,. \tag{5.1b}$$

The idea of partitioned Runge–Kutta methods is to integrate both equations using two different Runge–Kutta methods, i.e., treat the $\boldsymbol{q}$-variables with the first method (a_{ij}, b_i, c_i) and the $\boldsymbol{u}$-variables by the second method $(\hat{a}_{ij}, \hat{b}_i, \hat{c}_i)$, see Section II.16 in [84]. Such methods were originally proposed for the solution of differential equations that can be split into stiff and nonstiff parts [77, 91]. The authors applied an explicit method to the nonstiff part and an implicit method to the stiff part of a system of differential equations. Moreover, it is commonly assumed that $c_i = \hat{c}_i$, which can be shown to be a necessary condition for second-order convergence, see [84].

Let $\tau_i = t_n + hc_i$, application of a partitioned Runge–Kutta method to (5.1) reads

$$\dot{\boldsymbol{Q}}_i = \boldsymbol{f}(\tau_i,\boldsymbol{Q}_i,\boldsymbol{U}_i)\,, \qquad \dot{\boldsymbol{U}}_i = \boldsymbol{k}(\tau_i,\boldsymbol{Q}_i,\boldsymbol{U}_i)\,, \tag{5.2a}$$

$$\boldsymbol{Q}_i = \boldsymbol{q}_n + h\sum\nolimits_{j=1}^{s} a_{ij}\dot{\boldsymbol{Q}}_j\,, \qquad \boldsymbol{U}_i = \boldsymbol{u}_n + h\sum\nolimits_{j=1}^{s} \hat{a}_{ij}\dot{\boldsymbol{U}}_j\,, \tag{5.2b}$$

$$\boldsymbol{q}_{n+1} = \boldsymbol{q}_n + h\sum\nolimits_{i=1}^{s} b_i\dot{\boldsymbol{Q}}_i\,, \qquad \boldsymbol{u}_{n+1} = \boldsymbol{u}_n + h\sum\nolimits_{i=1}^{s} \hat{b}_i\dot{\boldsymbol{U}}_i\,. \tag{5.2c}$$

Introducing the solution tuples $\dot{\boldsymbol{Q}} = (\dot{\boldsymbol{Q}}_1,\ldots,\dot{\boldsymbol{Q}}_s) \in \mathbb{R}^{sn_q}$ and $\dot{\boldsymbol{U}} = (\dot{\boldsymbol{U}}_1,\ldots,\dot{\boldsymbol{U}}_s) \in \mathbb{R}^{sn_u}$, the internal stages can be computed via $\boldsymbol{Q} = (\boldsymbol{1}_s \otimes \boldsymbol{q}_n) + h(\boldsymbol{A} \otimes \boldsymbol{1}_{n_q\times n_q})\dot{\boldsymbol{Q}}$ and $\boldsymbol{U} = (\boldsymbol{1}_s \otimes \boldsymbol{u}_n) + h(\hat{\boldsymbol{A}} \otimes \boldsymbol{1}_{n_u\times n_u})\dot{\boldsymbol{U}}$. Consequently, each Runge–Kutta step requires the solution of $s(n_q + n_u)$

nonlinear equations[2]

$$\mathbf{0}_{n_u+n_q} = \boldsymbol{\varphi}(\dot{\boldsymbol{Q}}, \dot{\boldsymbol{U}}) = \begin{pmatrix} \dot{\boldsymbol{Q}} - \boldsymbol{F}(\boldsymbol{Q}, \boldsymbol{U}) \\ \dot{\boldsymbol{U}} - \boldsymbol{K}(\boldsymbol{Q}, \boldsymbol{U}) \end{pmatrix}, \tag{5.3}$$

where

$$\boldsymbol{F}(\boldsymbol{Q}, \boldsymbol{U}) = \begin{pmatrix} \boldsymbol{f}(\tau_1, \boldsymbol{Q}_1, \boldsymbol{U}_1) \\ \vdots \\ \boldsymbol{f}(\tau_s, \boldsymbol{Q}_s, \boldsymbol{U}_s) \end{pmatrix} \quad \text{and} \quad \boldsymbol{K}(\boldsymbol{Q}, \boldsymbol{U}) = \begin{pmatrix} \boldsymbol{k}(\tau_1, \boldsymbol{Q}_1, \boldsymbol{U}_1) \\ \vdots \\ \boldsymbol{k}(\tau_s, \boldsymbol{Q}_s, \boldsymbol{U}_s) \end{pmatrix}. \tag{5.4}$$

Finally, when the above system is solved successfully, the future positions and velocities can be computed a posteriori as

$$\boldsymbol{q}_{n+1} = \boldsymbol{q}_n + h(\boldsymbol{b}^{\mathrm{T}} \otimes \mathbf{1}_{n_q \times n_q}) \dot{\boldsymbol{Q}} \tag{5.5a}$$

$$\boldsymbol{u}_{n+1} = \boldsymbol{u}_n + h(\hat{\boldsymbol{b}}^{\mathrm{T}} \otimes \mathbf{1}_{n_u \times n_u}) \dot{\boldsymbol{U}}\,. \tag{5.5b}$$

5.2 Bilaterally Constrained Systems

As different works suggest, see e.g., [85] and [99], partitioned Runge–Kutta methods show very beneficial properties for the solution of constrained (Hamiltonian) systems. Hence, the ODE system defined in (5.1) is extended as

$$\dot{\boldsymbol{q}} = \boldsymbol{f}(t, \boldsymbol{q}, \boldsymbol{u}) \tag{5.6a}$$

$$\dot{\boldsymbol{u}} = \boldsymbol{k}(t, \boldsymbol{q}, \boldsymbol{u}) + \boldsymbol{l}(t, \boldsymbol{q}) \dot{\boldsymbol{\pi}}\,, \tag{5.6b}$$

where

$$\boldsymbol{l}(t, \boldsymbol{q}) = \boldsymbol{M}(t, \boldsymbol{q})^{-1} \boldsymbol{W}(t, \boldsymbol{q}) \quad \text{and} \quad \boldsymbol{W}(t, \boldsymbol{q}) = \begin{pmatrix} \boldsymbol{W}_g(t, \boldsymbol{q}) & \boldsymbol{W}_\gamma(t, \boldsymbol{q}) \end{pmatrix}. \tag{5.7}$$

Further, the system (5.6) is now subjected to the bilateral constraint conditions (3.6), i.e.,

$$\boldsymbol{g}(t, \boldsymbol{q}) = \mathbf{0}_{n_g} \tag{5.8a}$$

$$\boldsymbol{\gamma}(t, \boldsymbol{q}, \boldsymbol{u}) = \mathbf{0}_{n_\gamma}\,. \tag{5.8b}$$

[2] It is noteworthy to mention that in the case of a non-constant mass matrix $\boldsymbol{M}(t, \boldsymbol{q})$ that depends on the generalized coordinates $\boldsymbol{q}$, the solution of the time discretized system (5.2) can be significantly simplified (differentiation of $\boldsymbol{M}(t, \boldsymbol{q})^{-1}$ with respect to $\boldsymbol{q}$ is not required), by application of Runge–Kutta methods to *implicit differential equations* as originally proposed by Ascher and Petzold [17] and outlined in great detail in Section 5.3.5 of Fabien [61]. The same argument applies for all subsequent extensions of this method.

The corresponding partitioned Runge–Kutta method reads

$$\dot{\boldsymbol{Q}}_i = \boldsymbol{f}(\tau_i, \boldsymbol{Q}_i, \boldsymbol{U}_i)\,, \qquad \dot{\boldsymbol{U}}_i = \boldsymbol{k}(\tau_i, \boldsymbol{Q}_i, \boldsymbol{U}_i) + \boldsymbol{l}(\tau_i, \boldsymbol{Q}_i)\dot{\boldsymbol{\Pi}}_i\,, \tag{5.9a}$$

$$\boldsymbol{Q}_i = \boldsymbol{q}_n + h\sum\nolimits_{j=1}^{s} a_{ij}\dot{\boldsymbol{Q}}_j\,, \qquad \boldsymbol{U}_i = \boldsymbol{u}_n + h\sum\nolimits_{j=1}^{s} \hat{a}_{ij}\dot{\boldsymbol{U}}_j\,, \tag{5.9b}$$

$$\boldsymbol{q}_{n+1} = \boldsymbol{q}_n + h\sum\nolimits_{i=1}^{s} b_i\dot{\boldsymbol{Q}}_i\,, \qquad \boldsymbol{u}_{n+1} = \boldsymbol{u}_n + h\sum\nolimits_{i=1}^{s} \hat{b}_i\dot{\boldsymbol{U}}_i\,, \tag{5.9c}$$

$$\boldsymbol{\pi}_{n+1} = \boldsymbol{\pi}_n + h\sum\nolimits_{i=1}^{s} \hat{b}_i\dot{\boldsymbol{\Pi}}_i\,, \tag{5.9d}$$

where the Lagrange multipliers of the individual stages $\dot{\boldsymbol{\Pi}}_i = (\dot{\boldsymbol{\Pi}}_{g,i}, \dot{\boldsymbol{\Pi}}_{\gamma,i})$ are defined by the constraint equations

$$\boldsymbol{g}(\tau_i, \boldsymbol{Q}_i) = \boldsymbol{0}_{n_g} \quad \text{for} \quad i = 1, \dots, s \tag{5.10a}$$

$$\boldsymbol{\gamma}(\tau_i, \boldsymbol{Q}_i, \boldsymbol{U}_i) = \boldsymbol{0}_{n_\gamma} \quad \text{for} \quad i = 1, \dots, s\,. \tag{5.10b}$$

Jay [99] and Hairer et al. [85] were interested in symplectic methods for constrained Hamiltonian systems. Although the mechanical systems from Chapter 3 are not assumed to be Hamiltonian, the presented analysis is beneficial for general constrained mechanical systems. Hence, the investigations on symplectic partitioned Runge–Kutta methods are subsequently extended to general constrained mechanical systems. For Hamiltonian systems a symplectic partitioned Runge–Kutta method is obtained by restricting the given Butcher tableaus by the subsequent theorem.

Theorem 5.1 (Suris [172] and Sanz-Serna [157])**.**[3] *If the Butcher tableaus* (a_{ij}, b_i, c_i) *and* $(\hat{a}_{ij}, \hat{b}_i, \hat{c}_i)$ *satisfy*

$$b_i = \hat{b}_i \quad \textit{for} \quad i = 1, \dots, s \tag{5.11a}$$

$$b_i\hat{a}_{ij} + \hat{b}_j a_{ji} - b_i\hat{b}_j = 0 \quad \textit{for} \quad i, j = 1, \dots, s\,, \tag{5.11b}$$

then the method (5.3) *is symplectic.*

Therein, equation (5.11a) indicates that both quadrature formulas share the same weights. Since second- and higher-order convergence requires $c_i = \hat{c}_i$, both quadrature formulas coincide. Moreover, the positions $\boldsymbol{q}$ are assumed to be integrated by a stiffly accurate Runge–Kutta method, i.e.,

$$a_{sj} = b_j \quad \text{for} \quad j = 1, \dots, s\,, \tag{5.12}$$

[3] See also Hairer et al. [84], Theorem 16.10.

which implies $\boldsymbol{q}_{n+1} = \boldsymbol{Q}_s$, see Definition 4.5. By condition $B(1)$ this further implies $\sum_{i=1}^{s} b_i = 1$. Due to the consistency condition $C(1)$, i.e., $c_i = \sum_{j=1}^{s} a_{ij}$, also $\tau_s = t_n + h = t_{n+1}$. Consequently, the bilateral constraint conditions $\boldsymbol{g}(t_{n+1}, \boldsymbol{q}_{n+1})$ are automatically satisfied by the method. Assuming nonzero b_i, condition (5.11a) and (5.11b), together with (5.12) further imply

$$\hat{a}_{is} = 0 \quad \text{for} \quad i = 1, \ldots, s\,. \tag{5.13}$$

Thus, the velocities $\boldsymbol{U}_s$ no longer depend on the Lagrange multipliers $\dot{\boldsymbol{\Pi}}_s$. Since these multipliers are involved in the computation of $\boldsymbol{u}_{n+1}$ the idea is to determine $\dot{\boldsymbol{\Pi}}_s$ by additionally enforcing the constraint conditions

$$\dot{\boldsymbol{g}}(t_{n+1}, \boldsymbol{q}_{n+1}, \boldsymbol{u}_{n+1}) = \boldsymbol{0}_{n_g} \quad \text{and} \quad \boldsymbol{\gamma}(t_{n+1}, \boldsymbol{q}_{n+1}, \boldsymbol{u}_{n+1}) = \boldsymbol{0}_{n_\gamma}\,, \tag{5.14}$$

see [10, 99, 110]. In order to obtain a solvable system, the authors in [85, 99] additionally impose the condition

$$a_{1j} = 0 \quad \text{for} \quad j = 1, \ldots, s\,, \tag{5.15}$$

so that $\boldsymbol{Q}_1 = \boldsymbol{q}_n$ and with $c_1 = \sum_{j=1}^{s} a_{1j} = 0$ also $\tau_1 = t_n$. Assuming consistent initial conditions this implies (5.10a) to be automatically satisfied for the first stage, i.e., $\boldsymbol{g}(\tau_1, \boldsymbol{Q}_1) = \boldsymbol{g}(t_n, \boldsymbol{q}_n) = \boldsymbol{0}_{n_g}$. Hence, only the remaining constraints

$$\boldsymbol{g}(\tau_i, \boldsymbol{Q}_i) = \boldsymbol{0}_{n_g} \quad \text{for} \quad i = 2, \ldots, s \tag{5.16}$$

are enforced. Although there might be different possibilities how to deal with the bilateral constraints on velocity level (5.10b), the analogous procedure is chosen. Specifically,

$$\boldsymbol{\gamma}(\tau_i, \boldsymbol{Q}_i, \boldsymbol{U}_i) = \boldsymbol{0}_{n_\gamma} \quad \text{for} \quad i = 2, \ldots, s \tag{5.17}$$

is enforced.[4] By (5.11b) and assuming nonzero b_i, condition (5.15) further implies

$$\hat{a}_{i1} = b_1 \quad \text{for} \quad i = 1, \ldots, s\,. \tag{5.18}$$

[4]This step requires some special attention. For $i = 2, \ldots, s$, the constraint equations $\boldsymbol{g}(\tau_i, \boldsymbol{Q}_i)$ and $\boldsymbol{\gamma}(\tau_i, \boldsymbol{Q}_i, \boldsymbol{U}_i)$ are enforced by the Lagrange multipliers $\dot{\boldsymbol{\Pi}}_{i-1}$. Consequently, an index shift is present between the constraint equations and the corresponding Lagrange multipliers. Keeping this in mind is of crucial importance for the subsequent extension of the proposed method to nonsmooth mechanical system and will be investigated in Example 5.2.

In conclusion the above discussed requirements result in Butcher tableaus of the form

$$\begin{array}{c|c} \boldsymbol{c} & \boldsymbol{A} \\ \hline & \boldsymbol{b}^{\mathrm{T}} \end{array} = \begin{array}{c|cccccc} c_1 = 0 & 0 & 0 & \dots & 0 & 0 \\ c_2 & a_{21} & a_{22} & \dots & a_{2,s-1} & a_{2s} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ c_{s-1} & a_{s-1,1} & a_{s-1,2} & \dots & a_{s-1,s-1} & a_{s-1,s} \\ c_s = 1 & b_1 & b_2 & \dots & b_{s-1} & b_s \\ \hline & b_1 & b_2 & \dots & b_{s-1} & b_s \end{array} \tag{5.19}$$

and

$$\begin{array}{c|c} \boldsymbol{c} & \hat{\boldsymbol{A}} \\ \hline & \boldsymbol{b}^{\mathrm{T}} \end{array} = \begin{array}{c|cccccc} c_1 = 0 & b_1 & \hat{a}_{12} & \dots & \hat{a}_{1,s-1} & 0 \\ c_2 & b_1 & \hat{a}_{22} & \dots & \hat{a}_{2,s-1} & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ c_{s-1} & b_1 & \hat{a}_{s-1,2} & \dots & \hat{a}_{s-1,s-1} & 0 \\ c_s = 1 & b_1 & \hat{a}_{s2} & \dots & \hat{a}_{s,s-1} & 0 \\ \hline & b_1 & b_2 & \dots & b_{s-1} & b_s \end{array} . \tag{5.20}$$

Introducing two novel solution tuples $\dot{\boldsymbol{\Pi}}_g = (\dot{\boldsymbol{\Pi}}_{g,1}, \dots, \dot{\boldsymbol{\Pi}}_{g,s}) \in \mathbb{R}^{sn_g}$ and $\dot{\boldsymbol{\Pi}}_\gamma = (\dot{\boldsymbol{\Pi}}_{\gamma,1}, \dots, \dot{\boldsymbol{\Pi}}_{\gamma,s}) \in \mathbb{R}^{sn_\gamma}$ that are gathered in $\dot{\boldsymbol{\Pi}} = (\dot{\boldsymbol{\Pi}}_g, \dot{\boldsymbol{\Pi}}_\gamma) \in \mathbb{R}^{s(n_g+n_\gamma)}$ and with $\dot{\boldsymbol{X}} = (\dot{\boldsymbol{Q}}, \dot{\boldsymbol{U}}, \dot{\boldsymbol{\Pi}}) \in \mathbb{R}^{s(n_q+n_u+n_g+n_\gamma)}$, each time-step of the presented partitioned Runge–Kutta methods requires the solution of the system of nonlinear equations $\boldsymbol{\varphi}(\dot{\boldsymbol{X}}) = \boldsymbol{0}_{s(n_u+n_q+n_g+n_\gamma)}$, where

$$\boldsymbol{\varphi}(\dot{\boldsymbol{X}}) = \begin{pmatrix} \dot{\boldsymbol{Q}} - \boldsymbol{F}(\boldsymbol{Q}, \boldsymbol{U}) \\ \dot{\boldsymbol{U}} - \boldsymbol{K}(\boldsymbol{Q}, \boldsymbol{U}) - \boldsymbol{L}(\boldsymbol{Q})\dot{\boldsymbol{\Pi}} \\ \boldsymbol{G}(\boldsymbol{Q}) \\ \boldsymbol{\Gamma}(\boldsymbol{Q}, \boldsymbol{U}) \end{pmatrix} . \tag{5.21}$$

Therein, the abbreviations

$$\boldsymbol{F}(\boldsymbol{Q}, \boldsymbol{U}) = \begin{pmatrix} \boldsymbol{f}(\tau_1, \boldsymbol{Q}_1, \boldsymbol{U}_1) \\ \vdots \\ \boldsymbol{f}(\tau_s, \boldsymbol{Q}_s, \boldsymbol{U}_s) \end{pmatrix} , \qquad \boldsymbol{K}(\boldsymbol{Q}, \boldsymbol{U}) = \begin{pmatrix} \boldsymbol{k}(\tau_1, \boldsymbol{Q}_1, \boldsymbol{U}_1) \\ \vdots \\ \boldsymbol{k}(\tau_s, \boldsymbol{Q}_s, \boldsymbol{U}_s) \end{pmatrix}$$
$$\boldsymbol{L}(\boldsymbol{Q}) = \begin{pmatrix} \boldsymbol{l}(\tau_1, \boldsymbol{Q}_1) \\ \vdots \\ \boldsymbol{l}(\tau_s, \boldsymbol{Q}_s) \end{pmatrix} \tag{5.22}$$

and

$$\boldsymbol{G}(\boldsymbol{Q}) = \begin{pmatrix} \boldsymbol{g}(\tau_2, \boldsymbol{Q}_2) \\ \vdots \\ \boldsymbol{g}(\tau_s, \boldsymbol{Q}_s) \\ \dot{\boldsymbol{g}}(t_{n+1}, \boldsymbol{q}_{n+1}, \boldsymbol{u}_{n+1}) \end{pmatrix}, \quad \boldsymbol{\Gamma}(\boldsymbol{Q}, \boldsymbol{U}) = \begin{pmatrix} \boldsymbol{\gamma}(\tau_2, \boldsymbol{Q}_2, \boldsymbol{U}_2) \\ \vdots \\ \boldsymbol{\gamma}(\tau_s, \boldsymbol{Q}_s, \boldsymbol{U}_s) \\ \boldsymbol{\gamma}(t_{n+1}, \boldsymbol{q}_{n+1}, \boldsymbol{u}_{n+1}) \end{pmatrix} \tag{5.23}$$

were used.

5.3 Extension to Nonsmooth Mechanical Systems

The extension of the presented method to nonsmooth mechanical systems requires the quadrature of the measures $\mathrm{d}\boldsymbol{q}$, $\mathrm{d}\boldsymbol{u}$ and $\mathrm{d}\boldsymbol{\pi}$ that are encountered in the measure differential equations (3.1) and (3.2). Hence, it is now of interest to solve the system of measure differential inclusions given by

$$\mathrm{d}\boldsymbol{q} = \boldsymbol{f}(t, \boldsymbol{q}, \boldsymbol{u})\mathrm{d}t \tag{5.24a}$$

$$\mathrm{d}\boldsymbol{u} = \boldsymbol{k}(t, \boldsymbol{q}, \boldsymbol{u})\mathrm{d}t + \boldsymbol{l}(t, \boldsymbol{q})\mathrm{d}\boldsymbol{\pi}\,, \tag{5.24b}$$

with

$$\boldsymbol{f}(t, \boldsymbol{q}, \boldsymbol{u}) = \boldsymbol{B}(t, \boldsymbol{q})\boldsymbol{u} + \boldsymbol{\beta}(t, \boldsymbol{q}) \tag{5.25a}$$

$$\boldsymbol{k}(t, \boldsymbol{q}, \boldsymbol{u}) = \boldsymbol{M}(t, \boldsymbol{q})^{-1}\boldsymbol{h}(t, \boldsymbol{q}, \boldsymbol{u}) \tag{5.25b}$$

$$\boldsymbol{l}(t, \boldsymbol{q}) = \boldsymbol{M}(t, \boldsymbol{q})^{-1}\boldsymbol{W}(t, \boldsymbol{q}) \tag{5.25c}$$

and $\boldsymbol{W}(t, \boldsymbol{q}) = \begin{pmatrix} \boldsymbol{W}_g(t, \boldsymbol{q}) & \boldsymbol{W}_\gamma(t, \boldsymbol{q}) & \boldsymbol{W}_N(t, \boldsymbol{q}) & \boldsymbol{W}_F(t, \boldsymbol{q}) \end{pmatrix}$. Moreover, the system defined by (5.24) and (5.25) is subjected to bilateral and unilateral constraint conditions that are incorporated below. Adopting the procedure introduced in Section 5.2, this results in the nonsmooth partitioned Runge–Kutta method

$$\mathrm{d}\boldsymbol{Q}_i = h\boldsymbol{f}(\tau_i, \boldsymbol{Q}_i, \boldsymbol{U}_i)\,, \qquad \mathrm{d}\boldsymbol{U}_i = h\boldsymbol{k}(\tau_i, \boldsymbol{Q}_i, \boldsymbol{U}_i) + \boldsymbol{l}(\tau_i, \boldsymbol{Q}_i)\mathrm{d}\boldsymbol{\Pi}_i\,, \tag{5.26a}$$

$$\boldsymbol{Q}_i = \boldsymbol{q}_n + \sum\nolimits_{j=1}^{s} a_{ij}\mathrm{d}\boldsymbol{Q}_j\,, \qquad \boldsymbol{U}_i = \boldsymbol{u}_n + \sum\nolimits_{j=1}^{s} \hat{a}_{ij}\mathrm{d}\boldsymbol{U}_j\,, \tag{5.26b}$$

$$\boldsymbol{\Pi}_i = \boldsymbol{\pi}_n + \sum\nolimits_{j=1}^{s} \hat{a}_{ij}\mathrm{d}\boldsymbol{\Pi}_j\,, \tag{5.26c}$$

$$\boldsymbol{q}_{n+1} = \boldsymbol{q}_n + \sum\nolimits_{i=1}^{s} b_i\mathrm{d}\boldsymbol{Q}_i\,, \qquad \boldsymbol{u}_{n+1} = \boldsymbol{u}_n + \sum\nolimits_{i=1}^{s} b_i\mathrm{d}\boldsymbol{U}_i\,, \tag{5.26d}$$

$$\boldsymbol{\pi}_{n+1} = \boldsymbol{\pi}_n + \sum\nolimits_{i=1}^{s} b_i\mathrm{d}\boldsymbol{\Pi}_i\,. \tag{5.26e}$$

Similar to the nonsmooth projected stiffly accurate Runge–Kutta methods from the chapter before, the application of a classical quadrature is replaced by a version of the quadrature with rescaled measures in the discrete setting, see (4.62) and (4.63) and the corresponding discussion.

As in the bilaterally case, the internal stages are required to satisfy the bilateral constraints (5.16) and (5.17), while the final stage is forced to satisfy (5.14). Specifically,

$$\boldsymbol{g}(\tau_i, \boldsymbol{Q}_i) = \boldsymbol{0}_{n_g} \quad \text{for} \quad i = 2, \dots, s \tag{5.27a}$$

$$\boldsymbol{\gamma}(\tau_i, \boldsymbol{Q}_i, \boldsymbol{U}_i) = \boldsymbol{0}_{n_\gamma} \quad \text{for} \quad i = 2, \dots, s \tag{5.27b}$$

as well as

$$\dot{\boldsymbol{g}}(t_{n+1}, \boldsymbol{q}_{n+1}, \boldsymbol{u}_{n+1}) = \boldsymbol{0}_{n_g} \quad \text{and} \quad \boldsymbol{\gamma}(t_{n+1}, \boldsymbol{q}_{n+1}, \boldsymbol{u}_{n+1}) = \boldsymbol{0}_{n_\gamma} \,. \tag{5.28}$$

The last open question is, how to enforce the discrete equivalents to the unilateral constraint condition for contact and friction. In analogy to the bilateral constraints and as proposed by Capobianco et al. [42], all but the first internal stage are forced to satisfy a discrete version of Signorini's law. Specifically, for every contact $k = 1, \dots, n_N$ and for every stage $i = 2, \dots, s$ it is required that[5]

$$g_N^k(\tau_i, \boldsymbol{Q}_i) \in \mathcal{N}_{\mathbb{R}_0^-}(-\mathrm{d}\Pi_{N,i-1}^k) \,. \tag{5.30}$$

Therein, an index shift is present between the evaluation of the contact distances g_N^k and the descrete percussion measures $\mathrm{d}\Pi_N^k$, as already observed in the case of bilateral constraints before. A similar index shift is present in the methods proposed in [42] and will be investigated in Example 5.2.

Up to now, the percussions $\boldsymbol{\pi}_{N,n+1}$ are unspecified. Thereto, using the same arguments from Chapter 4 and in analogy with the bilateral constraints, additionally the time derivative of the unilateral constraints will be enforced. Moreover, a possibly elastic impact law will be incorporated. In what follows, it is assumed that if the contact k is active at the end of the time-step, i.e., $k \in A_{n+1} = A(t_{n+1}, \boldsymbol{q}_{n+1})$, then it has been active

[5] In analogy to the previous chapter, one might be tempt to enforce the discrete version of Signorini's law using the percussion differences $\Delta\Pi_{N,i-1}^k$, i.e.,

$$g_N^k(\tau_i, \boldsymbol{Q}_i) \in \mathcal{N}_{\mathbb{R}_0^-}(-\Delta\Pi_{N,i-1}^k) \,. \tag{5.29}$$

As demonstrated in Section 7.7, this choice leads to an order reduction in the subsequently proposed family of Lobatto IIIA-IIIB methods and is therefore not recommended. The same applies to the discrete friction laws proposed below.

during the whole time-step $I_n = (t_n, t_{n+1}]$. Furthermore, it is assumed that ξ_N^k is constant on I_n and corresponds to

$$\xi_{N,n+1}^k = \dot{g}_N^k(t_{n+1}, \boldsymbol{q}_{n+1}, \boldsymbol{u}_{n+1}) + e_N^k \, \dot{g}_N^k(t_n, \boldsymbol{q}_n, \boldsymbol{u}_n)\,, \tag{5.31}$$

which is regarded as a discrete approximation of (3.44) over one time-step, see [29, 41, 42, 74]. The position and velocity of the system is continuous between velocity jumps, implying $\dot{g}_N^{k+} = \dot{g}_N^{k-} = \dot{g}_N^k$ for almost all t. Hence, in such cases it holds that

$$(1 + e_N^k)\dot{g}_N^k = \dot{g}_N^{k+} + e_N^k \dot{g}_N^{k-} = \xi_N^k \tag{5.32}$$

and Signorini's law can be written on velocity level (3.15) using ξ_N^k instead of $\dot{g}_N^k$. Using all these preliminary assumptions together with Proposition 2.3, allows to combine Signorini's law on velocity level (3.15) and the Newton-type impact law in normal direction (3.45) as (see [41])

$$\begin{cases} \xi_{N,n+1}^k \in \mathcal{N}_{\mathbb{R}_0^-}(-\int_{I_n} \mathrm{d}\pi_N^k) & \text{if } k \in A_{n+1}\,, \\ 0 = \int_{I_n} \mathrm{d}\pi_N^k & \text{if } k \notin A_{n+1}\,. \end{cases} \tag{5.33}$$

Substitution of $\int_{I_n} \mathrm{d}\pi_N^k \approx \sum_{i=1}^s b_i \mathrm{d}\Pi_{N,i}^k = \Delta\pi_{N,n+1}^k = \pi_{N,n+1}^k - \pi_{N,n}^k$ results in the discrete force law. A discussion of how this is done in detail as well as an interpretation of the integrated contact law can be found in Section 6 of [41].

For the discretization of the set-valued friction laws a similar path is taken. Using the same assumptions as above, the friction law (3.20) can be combined with the Newton-type impact law in tangential direction (3.47) as

$$\boldsymbol{\xi}_{F,n+1}^k \in \mathcal{N}_{C_F^k(\int_{I_n} \mathrm{d}\pi_N^k)}(-\textstyle\int_{I_n} \mathrm{d}\boldsymbol{\pi}_F^k)\,, \tag{5.34}$$

where (3.46) is discretized as

$$\boldsymbol{\xi}_{F,n+1}^k = \boldsymbol{\gamma}_F(t_{n+1}, \boldsymbol{q}_{n+1}, \boldsymbol{u}_{n+1}) + e_F^k \, \boldsymbol{\gamma}_F^k(t_n, \boldsymbol{q}_n, \boldsymbol{u}_n) \tag{5.35}$$

and the approximation $\int_{I_n} \mathrm{d}\boldsymbol{\pi}_F^k \approx \sum_{i=1}^s b_i \mathrm{d}\boldsymbol{\Pi}_{F,i}^k = \Delta\boldsymbol{\pi}_{F,n+1}^k = \boldsymbol{\pi}_{F,n+1}^k - \boldsymbol{\pi}_{F,n}^k$ is used. Moreover, for every contact $k = 1, \ldots, n_N$ and for every stage $i = 2, \ldots, s$ the discrete friction law

$$\boldsymbol{\gamma}_F^k(\tau_i, \boldsymbol{Q}_i, \boldsymbol{U}_i) \in \mathcal{N}_{C_F^k(\mathrm{d}\Pi_{N,i-1}^k)}\big(-\mathrm{d}\boldsymbol{\Pi}_{F,i-1}^k\big) \tag{5.36}$$

is enforced, which is in line with the bilateral constraints (5.17) and the method proposed in [42].

Let $\mathrm{d}\boldsymbol{Q} = (\mathrm{d}\boldsymbol{Q}_1, \dots, \mathrm{d}\boldsymbol{Q}_s) \in \mathbb{R}^{sn_q}$, $\mathrm{d}\boldsymbol{U} = (\mathrm{d}\boldsymbol{U}_1, \dots, \mathrm{d}\boldsymbol{U}_s) \in \mathbb{R}^{sn_u}$ and $\mathrm{d}\boldsymbol{\Pi}_\square = (\mathrm{d}\boldsymbol{\Pi}_{\square,1}, \dots, \mathrm{d}\boldsymbol{\Pi}_{\square,s}) \in \mathbb{R}^{sn_\square}$ denote the tuples of unknowns, which are gathered in $\mathrm{d}\boldsymbol{x} = (\mathrm{d}\boldsymbol{Q}, \mathrm{d}\boldsymbol{U}, \mathrm{d}\boldsymbol{\Pi}_g, \mathrm{d}\boldsymbol{\Pi}_\gamma) \in \mathbb{R}^{s(n_q+n_u+n_g+n_\gamma)}$ and $\mathrm{d}\boldsymbol{y} = (\mathrm{d}\boldsymbol{\Pi}_N, \mathrm{d}\boldsymbol{\Pi}_F) \in \mathbb{R}^{s(n_N+n_F)}$. A single time-step $t_n \mapsto t_{n+1}$ of the presented method is given by solving the system of nonlinear equations $\boldsymbol{\varphi}(\mathrm{d}\boldsymbol{x}, \mathrm{d}\boldsymbol{y}) = \boldsymbol{0}_{s(n_u+n_q+n_g+n_\gamma)}$, where

$$\boldsymbol{\varphi}(\mathrm{d}\boldsymbol{x}, \mathrm{d}\boldsymbol{y}) = \begin{pmatrix} \mathrm{d}\boldsymbol{Q} - h\boldsymbol{F}(\boldsymbol{Q}, \boldsymbol{U}) \\ \mathrm{d}\boldsymbol{U} - h\boldsymbol{K}(\boldsymbol{Q}, \boldsymbol{U}) - \boldsymbol{L}(\boldsymbol{Q})\mathrm{d}\boldsymbol{\Pi} \\ \boldsymbol{G}(\boldsymbol{Q}) \\ \boldsymbol{\Gamma}(\boldsymbol{Q}, \boldsymbol{U}) \end{pmatrix}, \tag{5.37}$$

where the definitions for $\boldsymbol{F}$, $\boldsymbol{K}$, $\boldsymbol{L}$, $\boldsymbol{G}$ and $\boldsymbol{\Gamma}$ are already given in (5.22) and (5.23). The nonlinear equation (5.37) is subjected to the normal cone inclusions

$$\boldsymbol{\psi}_i^k(\mathrm{d}\boldsymbol{x}) \in \mathcal{K}_i^k(\boldsymbol{\mathcal{A}}\,\mathrm{d}\boldsymbol{y}), \quad k = 1, \dots, n_n, \quad i = 1, \dots, s, \tag{5.38}$$

where

$$\boldsymbol{\psi}_s^k(\mathrm{d}\boldsymbol{x}) = \begin{pmatrix} \xi_{N,n+1}^k \\ \boldsymbol{\xi}_{F,n+1}^k \end{pmatrix} \quad \text{and} \quad \boldsymbol{\psi}_i^k(\mathrm{d}\boldsymbol{x}) = \begin{pmatrix} g_N^k(\tau_i, \boldsymbol{Q}_i) \\ \boldsymbol{\gamma}_F^k(\tau_i, \boldsymbol{Q}_i, \boldsymbol{U}_i) \end{pmatrix} \tag{5.39}$$

for $i = 1, \dots, s-1$. Moreover,

$$\mathcal{K}_i^k(\mathrm{d}\boldsymbol{\mu}) = \mathcal{N}_{\mathbb{R}_0^-}(-\mathrm{d}\mu_{N,i}^k) \times \mathcal{N}_{C_F^k(\mathrm{d}\mu_{N,i}^k)}(-\mathrm{d}\boldsymbol{\mu}_{F,i}^k) \quad \text{and} \quad \mathrm{d}\boldsymbol{\mu} = \boldsymbol{\mathcal{A}}\,\mathrm{d}\boldsymbol{y}\,. \tag{5.40}$$

From the quadrature of the percussion measures $\mathrm{d}\boldsymbol{\pi}_N$, $\mathrm{d}\boldsymbol{\pi}_F$ defined in (5.26), due to the special structure of the coefficient matrix $\hat{\boldsymbol{A}}$, given in (5.20), and using the inclusions (5.30), (5.33), (5.36) and (5.34), it can be deduced that

$$\boldsymbol{\mathcal{A}} = \begin{pmatrix} \boldsymbol{1}_{s-1\times s-1} & \boldsymbol{0}_{s-1} \\ \boldsymbol{b}_\beta^{\mathrm{T}} & b_s \end{pmatrix} \otimes \begin{pmatrix} \boldsymbol{1}_{n_N\times n_N} & \boldsymbol{0}_{n_N\times n_F} \\ \boldsymbol{0}_{n_F\times n_N} & \boldsymbol{1}_{n_F\times n_F} \end{pmatrix}, \tag{5.41}$$

where $\beta = \{1, 2, \dots, s-1\}$. When alternatively, the inclusion (5.29) is enforced, together with a similarly integrated friction law, this leads to

$$\boldsymbol{\mathcal{A}} = \begin{pmatrix} \hat{\boldsymbol{A}}_{\beta\beta} & \boldsymbol{0}_{s-1} \\ \boldsymbol{b}_\beta^{\mathrm{T}} & b_s \end{pmatrix} \otimes \begin{pmatrix} \boldsymbol{1}_{n_N\times n_N} & \boldsymbol{0}_{n_N\times n_F} \\ \boldsymbol{0}_{n_F\times n_N} & \boldsymbol{1}_{n_F\times n_F} \end{pmatrix}. \tag{5.42}$$

A detailed discussion of how the nonlinear equation (5.37) subject to the normal cone inclusions (5.38) can be solved is given in Chapter 6.

Example 5.2 (One-dimensional Bouncing Ball revisited)**.** Subsequently, the one-dimensional bouncing ball Example 4.12, already discussed in the previous chapter, is revisited. Application of an s-stage nonsmooth partitioned Runge–Kutta method as described in (5.26) and enforcing the unilateral constraint conditions (5.30) and (5.33) results in the discrete system

$$\mathrm{d}\boldsymbol{Y} = h\left(u_n\mathbf{1}_s + \hat{\boldsymbol{A}}\mathrm{d}\boldsymbol{U}\right) \tag{5.43a}$$

$$\mathrm{d}\boldsymbol{U} = -hg\mathbf{1}_s + \tfrac{1}{m}\mathrm{d}\boldsymbol{\Pi}_N \tag{5.43b}$$

$$Y_2 \in \mathcal{N}_{\mathbb{R}_0^-}(-\mathrm{d}\Pi_{N,1}) \tag{5.43c}$$

$$\vdots$$

$$Y_s \in \mathcal{N}_{\mathbb{R}_0^-}(-\mathrm{d}\Pi_{N,s-1}) \tag{5.43d}$$

$$\begin{cases} \xi_{N,n+1} \in \mathcal{N}_{\mathbb{R}_0^-}(-\Delta\pi_{N,n+1}) & \text{if } Y_s \le 0 \\ 0 = \Delta\pi_{N,n+1} & \text{otherwise}\,. \end{cases} \tag{5.43e}$$

Therein, $\xi_{N,n+1} = u_{n+1} + e_N u_n$ and the quadratures are given by

$$\boldsymbol{Y} = y_n\mathbf{1}_s + \boldsymbol{A}\mathrm{d}\boldsymbol{Y} \quad \text{and} \quad \boldsymbol{U} = u_n\mathbf{1}_s + \hat{\boldsymbol{A}}\mathrm{d}\boldsymbol{U} \tag{5.44}$$

as well as

$$u_{n+1} = \boldsymbol{b}^{\mathrm{T}}\mathrm{d}\boldsymbol{U} \quad \text{and} \quad \Delta\pi_{N,n+1} = \boldsymbol{b}^{\mathrm{T}}\mathrm{d}\boldsymbol{\Pi}_N\,. \tag{5.45}$$

Introducing the two tuples

$$\boldsymbol{w} = \begin{pmatrix} Y_2 \\ \vdots \\ Y_s \\ \xi_{N,n+1} \end{pmatrix}, \quad \boldsymbol{z} = \begin{pmatrix} \mathrm{d}\Pi_{N,1} \\ \vdots \\ \mathrm{d}\Pi_{N,s-1} \\ \Delta\pi_{N,n+1} \end{pmatrix} = \underbrace{\begin{pmatrix} \mathbf{1}_{s-1\times s-1} & \mathbf{0}_{s-1} \\ \boldsymbol{b}_\beta^{\mathrm{T}} & b_s \end{pmatrix}}_{\boldsymbol{\mathcal{A}}} \mathrm{d}\boldsymbol{\Pi} \tag{5.46}$$

and by assuming $Y_s = 0$, the normal cone inclusions (5.43c)–(5.43e) can be written as the inequality complementarity

$$0 \le \boldsymbol{w} \perp \boldsymbol{z} \ge 0\,. \tag{5.47}$$

Let $\alpha = \{2, 3, \ldots, s\}$ and use $\mathrm{d}\boldsymbol{\Pi} = \boldsymbol{\mathcal{A}}^{-1}\boldsymbol{z}$. Substituting (5.43b) into (5.43a) and the result into (5.46) reveals

$$\boldsymbol{w} = \begin{pmatrix} y_n \\ \vdots \\ y_n \\ e_N u_n \end{pmatrix} + h\begin{pmatrix} \boldsymbol{A}_{\alpha\bullet} \\ \mathbf{0}_s^{\mathrm{T}} \end{pmatrix}\mathrm{d}\boldsymbol{Y} + \begin{pmatrix} \mathbf{0}_{s-1\times s} \\ \boldsymbol{b}^{\mathrm{T}} \end{pmatrix}\mathrm{d}\boldsymbol{U} = \boldsymbol{q} + \boldsymbol{M}\boldsymbol{z}\,, \tag{5.48}$$

with

$$\boldsymbol{q} = \begin{pmatrix} y_n \\ \vdots \\ y_n \\ e_N u_n \end{pmatrix} + h u_n \begin{pmatrix} \boldsymbol{A}_{\alpha\bullet} \\ \boldsymbol{0}_s^{\mathrm{T}} \end{pmatrix} \boldsymbol{1}_s - hg\boldsymbol{B}\boldsymbol{1}_s \quad \text{and} \quad \boldsymbol{M} = \frac{1}{m}\boldsymbol{B}\boldsymbol{\mathcal{A}}^{-1}. \tag{5.49}$$

Due to the special structure of the Butcher tableaus (5.19) and (5.20), the matrix $\boldsymbol{B}$ can be written as

$$\boldsymbol{B} = h \begin{pmatrix} \boldsymbol{A}_{\alpha\bullet} \\ \boldsymbol{0}_s^{\mathrm{T}} \end{pmatrix} \hat{\boldsymbol{A}} + \begin{pmatrix} \boldsymbol{0}_{s-1\times s} \\ \boldsymbol{b}^{\mathrm{T}} \end{pmatrix} = \begin{pmatrix} h\boldsymbol{A}_{\alpha\bullet}\hat{\boldsymbol{A}}_{\bullet\beta} & \boldsymbol{0}_{s-1} \\ \boldsymbol{b}_\beta^{\mathrm{T}} & b_s \end{pmatrix}. \tag{5.50}$$

Consequently, with $Y_s = 0$, equation (5.43) can equivalently be stated as linear complementarity problem

$$\mathrm{LCP}\,(\boldsymbol{q}, \boldsymbol{M}) \quad \Longleftrightarrow \quad \boldsymbol{w} = \boldsymbol{M}\boldsymbol{z} + \boldsymbol{q}\,, \quad 0 \leq \boldsymbol{w} \perp \boldsymbol{z} \geq 0\,. \tag{5.51}$$

As shown in (2.48), the inverse of $\boldsymbol{\mathcal{A}}$ can be computed using block Gaussian elimination yielding

$$\boldsymbol{\mathcal{A}}^{-1} = \begin{pmatrix} \boldsymbol{1}_{s-1\times s-1} & \boldsymbol{0}_{s-1} \\ \boldsymbol{b}_\beta^{\mathrm{T}} & b_s \end{pmatrix}^{-1} = \begin{pmatrix} \boldsymbol{1}_{s-1\times s-1} & \boldsymbol{0}_{s-1} \\ -\frac{1}{b_s}\boldsymbol{b}_\beta^{\mathrm{T}} & \frac{1}{b_s} \end{pmatrix}. \tag{5.52}$$

Finally, the matrix $\boldsymbol{M}$, required for the LCP $(\boldsymbol{q}, \boldsymbol{M})$, can be computed as

$$\boldsymbol{M} = \frac{1}{m}\boldsymbol{B}\boldsymbol{\mathcal{A}}^{-1} = \frac{1}{m}\begin{pmatrix} h\boldsymbol{A}_{\alpha\bullet}\hat{\boldsymbol{A}}_{\bullet\beta} & \boldsymbol{0}_{s-1} \\ \boldsymbol{0}_{s-1}^{\mathrm{T}} & 1 \end{pmatrix}. \tag{5.53}$$

Investigation of the structure of $\boldsymbol{M}$ gives more inside into the problem. On the one hand, this reveals that the last stage can be computed independently from the previous stages $i = 1, \ldots, s-1$. This is similar to the velocity projection introduced for the nonsmooth projected stiffly accurate Runge–Kutta methods from the previous chapter. On the other hand, it enables an investigation of the (unique) solvability of the linear complementarity problem. According to Proposition 2.21, the LCP $(\boldsymbol{q}, \boldsymbol{M})$ possesses a unique solution if $\boldsymbol{M}$ is a P-matrix. From Definition (2.19) it is clear that it is sufficient to investigate if the upper-left block of $\boldsymbol{M}$ given by

$$\widetilde{\boldsymbol{M}} = \frac{h}{m}\boldsymbol{A}_{\alpha\bullet}\hat{\boldsymbol{A}}_{\bullet\beta} \tag{5.54}$$

is itself a P-matrix. It can be demonstrated by numerical evaluations that this is the case for all subsequently proposed combinations of Butcher tableaus and is independent of the nonnegative factor $\frac{h}{m} > 0$.

The proposed nonsmooth partitioned Runge–Kutta methods enforce the unilateral constraint conditions (5.30) and (5.36) with a yet unmotivated index shift between the kinematic quantities and the scaled discrete percussion measures. Repeating the derivations above without this index shift, i.e., enforcing $Y_i \in \mathcal{N}_{\mathbb{R}_0^-}(-\mathrm{d}\Pi_{N,i})$ for $i = 2, \ldots, s$, would result in the matrices

$$\boldsymbol{\mathcal{A}} = \begin{pmatrix} \mathbf{0}_{s-1} & \mathbf{1}_{s-1\times s-1} \\ b_1 & \boldsymbol{b}_\alpha^{\mathrm{T}} \end{pmatrix} \quad \text{and} \quad \boldsymbol{\mathcal{A}}^{-1} = \begin{pmatrix} -\frac{1}{b_1}\boldsymbol{b}_\alpha^{\mathrm{T}} & \frac{1}{b_1} \\ \mathbf{1}_{s-1\times s-1} & \mathbf{0}_{s-1} \end{pmatrix}, \tag{5.55}$$

where (2.49) was applied for computing the inverse. For this case, numerical evaluations reveal that $\widetilde{\boldsymbol{M}}$ is no longer a P-matrix for the specific methods that will be introduced below. Consequently, the numerical solution of the LCP $(\boldsymbol{q}, \boldsymbol{M})$ does not possess a unique solution for this simple problem, leading to an ill-posed integration method.

Using the alternative discrete contact law proposed in (5.29), involving integrated percussion differences, results in the matrices

$$\boldsymbol{\mathcal{A}} = \begin{pmatrix} \hat{\boldsymbol{A}}_{\beta\beta} & \mathbf{0}_{s-1} \\ \boldsymbol{b}_\beta^{\mathrm{T}} & b_s \end{pmatrix} \quad \text{and} \quad \boldsymbol{\mathcal{A}} = \begin{pmatrix} \hat{\boldsymbol{A}}_{\alpha\beta} & \mathbf{0}_{s-1} \\ \boldsymbol{b}_\beta^{\mathrm{T}} & b_s \end{pmatrix}, \tag{5.56}$$

for the shifted and non-shifted case, respectively. Performing similar steps as outlined above leads to another LCP $(\boldsymbol{q}, \boldsymbol{M})$. Numerical evaluations show that for $s \geq 3$, the non-shifted method does not possess a unique solution since the corresponding matrix $\boldsymbol{M}$ is not a P-matrix. For the two-stage method, both variants lead to a matrix $\boldsymbol{M}$ that belongs to the class of P-matrices and hence results in a uniquely solvable LCP $(\boldsymbol{q}, \boldsymbol{M})$.

To conclude this section, the overall nonsmooth partitioned Runge–Kutta method, used in the subsequent numerical experiments, is summarized in Figure 5.1.

5.4 Butcher Tableaus of Selected Methods

In order to complete the description of the above introduced nonsmooth partitioned Runge–Kutta methods, the yet unknown Butcher tableaus have to be specified. Therefore, this section introduces a selection of well-established Butcher tableaus, together with their most important properties.

Nonsmooth partitioned Runge–Kutta

Solve the nonlinear system

$$\begin{aligned} \mathrm{d}\boldsymbol{Q}_i &= h\boldsymbol{f}(\tau_i, \boldsymbol{Q}_i, \boldsymbol{U}_i) \\ \mathrm{d}\boldsymbol{U}_i &= h\boldsymbol{k}(\tau_i, \boldsymbol{Q}_i, \boldsymbol{U}_i) + \boldsymbol{l}(\tau_i, \boldsymbol{Q}_i)\mathrm{d}\boldsymbol{\Pi}_i \end{aligned}$$

for $i = 1, 2, \ldots, s$, where $\tau_i = t_n + c_i h$, $\boldsymbol{Q}_i = \boldsymbol{q}_n + \sum_{j=1}^{s} a_{ij}\mathrm{d}\boldsymbol{Q}_j$ and $\boldsymbol{U}_i = \boldsymbol{u}_n + \sum_{j=1}^{s} \hat{a}_{ij}\mathrm{d}\boldsymbol{U}_j$. For $i = 2, \ldots, s$, the system is subject to the bilateral and unilateral constraints

$$\begin{gathered} \boldsymbol{g}(\tau_i, \boldsymbol{Q}_i) = \boldsymbol{0}_{n_g}\,, \quad \boldsymbol{\gamma}(\tau_i, \boldsymbol{Q}_i, \boldsymbol{U}_i) = \boldsymbol{0}_{n_\gamma} \quad \text{for} \quad i = 2, \ldots, s \\ \begin{pmatrix} g_N^k(\tau_i, \boldsymbol{Q}_i) \\ \boldsymbol{\gamma}_F^k(\tau_i, \boldsymbol{Q}_i, \boldsymbol{U}_i) \end{pmatrix} \in \mathcal{N}_{\mathbb{R}_0^-}(-\mathrm{d}\Pi_{N,i-1}^k) \times \mathcal{N}_{C_F^k(\mathrm{d}\Pi_{N,i-1}^k)}(-\mathrm{d}\boldsymbol{\Pi}_{F,i-1}^k)\,, \end{gathered}$$

where $k = 1, \ldots, n_N$. Further, with $\boldsymbol{q}_{n+1} = \boldsymbol{Q}_s$, $\boldsymbol{u}_{n+1} = \boldsymbol{u}_n + \sum_{i=1}^{s} b_i\mathrm{d}\boldsymbol{U}_i$ and $\Delta\boldsymbol{\pi}_{n+1} = \sum_{i=1}^{s} b_i\mathrm{d}\boldsymbol{\Pi}_i$, it has to be enforced that

$$\begin{gathered} \dot{\boldsymbol{g}}(t_{n+1}, \boldsymbol{q}_{n+1}, \boldsymbol{u}_{n+1}) = \boldsymbol{0}_{n_g}\,, \quad \boldsymbol{\gamma}(t_{n+1}, \boldsymbol{q}_{n+1}, \boldsymbol{u}_{n+1}) = \boldsymbol{0}_{n_\gamma} \\ \begin{pmatrix} \xi_{N,n+1}^k \\ \boldsymbol{\xi}_{F,n+1}^k \end{pmatrix} \in \mathcal{N}_{\mathbb{R}_0^-}(-\Delta\pi_{N,n+1}^k) \times \mathcal{N}_{C_F^k(\Delta\pi_{N,n+1}^k)}(-\Delta\boldsymbol{\pi}_{F,n+1}^k)\,, \end{gathered}$$

for $k \in A_{n+1}$.

Figure 5.1: Summary of nonsmooth partitioned Runge–Kutta methods.

Lobatto IIIA-IIIB

A prominent example, satisfying all the aforementioned requirements, is the pair of Lobatto IIIA and IIIB methods, respectively used for the tableaus (a_{ij}, b_i, c_i) and $(\hat{a}_{ij}, \hat{b}_i, \hat{c}_i)$. Both methods share their quadrature points and weights, i.e., $c_i = \hat{c}_i$ and $b_i = \hat{b}_i$. According to [84], for an s-stage method, the nodes are given by the roots of the polynomial

$$f(c) = \frac{\mathrm{d}^{s-2}}{\mathrm{d}c^{s-2}}\left(c^{s-1}(c-1)^{s-1}\right)\,. \tag{5.57}$$

The corresponding weights b_i are determined by the simplifying assumptions $B(2s-2)$. The remaining coefficients for Lobatto IIIA can be computed from solving s linear systems defined by $C(s)$. Since the method is stiffly accurate it follows $b_j = a_{sj}$. The coefficients for Lobatto IIIB

(a)

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & \frac{1}{2} & \frac{1}{2} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \qquad \begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{5}{24} & \frac{1}{3} & -\frac{1}{24} \\ 1 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array}$$

(b)

$$\begin{array}{c|cc} 0 & \frac{1}{2} & 0 \\ 1 & \frac{1}{2} & 0 \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \qquad \begin{array}{c|ccc} 0 & \frac{1}{6} & -\frac{1}{6} & 0 \\ \frac{1}{2} & \frac{1}{6} & \frac{1}{3} & 0 \\ 1 & \frac{1}{6} & \frac{5}{6} & 0 \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array}$$

Table 5.1: Butcher tableaus for two- and three-stage Lobatto IIIA (a) and Lobatto IIIB (b) methods.

can be computed from solving s linear systems defined by $D(s)$. Specific Butcher tableaus for both methods and $s = 2, 3$ are given in Table 5.1. The tableaus for $s = 4, 5$ can be found in [100].

Example 5.3 (Pendulum continued)**.** The investigations from Example 4.9 are continued with the proposed partitioned Runge–Kutta methods. In contrast to the unprojected Runge–Kutta methods, proposed in the previous chapter, these methods enforce the constraint equations on position and velocity level up to machine precision. Hence, no additional velocity projection is required since this is already incorporated into the method.

Since the proposed Lobatto IIIA-IIIB methods can be applied with an arbitrary number of stages s, it is interesting to continue the convergence analysis from Example 4.11 with those methods. Their convergence behavior is visualized in Figure 5.2. The slope of the straight lines indicate the order of convergence. The outstanding convergence rates of the Lobatto IIIA-IIIB methods are in line with the theoretically obtained convergence results given in [99]. For methods with $s \geq 2$ the global errors satisfy

$$\boldsymbol{q}_n - \boldsymbol{q}(t_n) = \mathcal{O}(h^{2s-2}) \quad \text{and} \quad \boldsymbol{u}_n - \boldsymbol{u}(t_n) = \mathcal{O}(h^{2s-2})\,. \tag{5.58}$$

Hence, no order reduction takes place for constrained mechanical systems, making these methods well-suited for such class of problems under the assumption that the system is not too stiff, see [99]. As investigated later in Section 7.5, violating this assumption will results in serious step-size restrictions. Specifically, in order to get a stable discrete system, the step-size has to be chosen relatively small, compared to the stiffly accurate Runge–Kutta methods presented in Chapter 4.

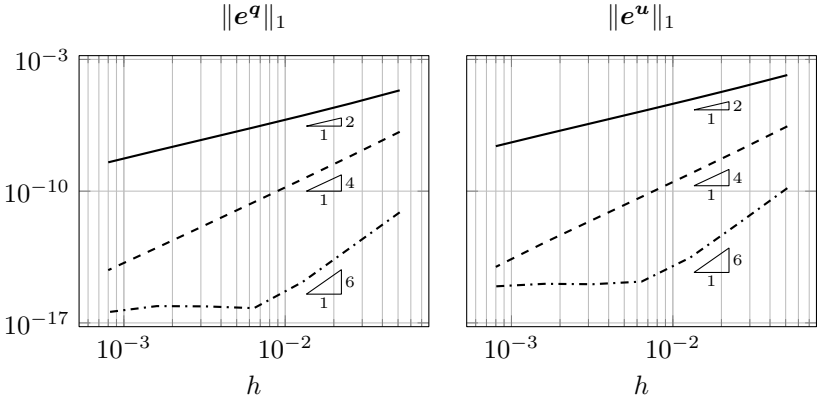

Figure 5.2: Convergence results of the partitioned implicit Runge–Kutta methods with Lobatto IIIA-IIIB tableaus and stages $s = 2, 3, 4$ applied to the pendulum problem.

Nonsmooth RATTLE

The two-stage Lobatto IIIA-IIIB method presented above deserves special attention. Besides a scaling factor of $\frac{1}{2}$, applied to the Lagrange multipliers in the discrete contact and friction laws, this method exactly coincides with the nonsmooth RATTLE algorithm presented in [29]. Moreover, it is the simplest partitioned method that simultaneously enforces bilateral as well as unilateral constraints on position and velocity level and allows to incorporates an elastic impact law. Inspecting the two-stage tableaus in Table 5.1 reveals $\tau_1 = t_n$, $\tau_2 = t_n + h = t_{n+1}$, $\boldsymbol{Q}_1 = \boldsymbol{q}_n$ and $\boldsymbol{U}_1 = \boldsymbol{U}_2 =: \boldsymbol{u}_{n+1/2}$. Substituting the other quantities from the Butcher tableaus results in two discrete systems. The first one is given by

$$\boldsymbol{u}_{n+1/2} = \boldsymbol{u}_n + \tfrac{h}{2}\boldsymbol{k}(t_n, \boldsymbol{q}_n, \boldsymbol{u}_{n+1/2}) + \tfrac{1}{2}\boldsymbol{l}(t_n, \boldsymbol{q}_n)\mathrm{d}\boldsymbol{\Pi}_1 \tag{5.59a}$$

$$\boldsymbol{q}_{n+1} = \boldsymbol{q}_n + \tfrac{h}{2}\left(\boldsymbol{f}(t_n, \boldsymbol{q}_n, \boldsymbol{u}_{n+1/2}) + \boldsymbol{f}(t_{n+1}, \boldsymbol{q}_{n+1}, \boldsymbol{u}_{n+1/2})\right) \tag{5.59b}$$

$$\boldsymbol{0}_{n_g} = \boldsymbol{g}(t_{n+1}, \boldsymbol{q}_{n+1}) \tag{5.59c}$$

$$\boldsymbol{0}_{n_\gamma} = \boldsymbol{\gamma}(t_{n+1}, \boldsymbol{q}_{n+1}, \boldsymbol{u}_{n+1/2}) \tag{5.59d}$$

$$g_N^k(t_{n+1}, \boldsymbol{q}_{n+1}) \in \mathcal{N}_{\mathbb{R}_0^-}(-\mathrm{d}\Pi_{N,1}^k), \quad k = 1, \ldots, n_N \tag{5.59e}$$

$$\boldsymbol{\gamma}_F^k(t_{n+1}, \boldsymbol{q}_{n+1}, \boldsymbol{u}_{n+1/2}) \in \mathcal{N}_{C_F^k(\mathrm{d}\Pi_{N,1}^k)}(-\mathrm{d}\boldsymbol{\Pi}_{F,1}^k), \quad k = 1, \ldots, n_N \tag{5.59f}$$

and enables the evaluation of the set of active contacts $A_{n+1} = A(t_{n+1}, \boldsymbol{q}_{n+1})$. Subsequently, a second (velocity) stage solves the system

$$\boldsymbol{u}_{n+1} = \boldsymbol{u}_{n+1/2} + \tfrac{h}{2}\boldsymbol{k}(t_{n+1}, \boldsymbol{q}_{n+1}, \boldsymbol{u}_{n+1/2}) + \tfrac{1}{2}\boldsymbol{l}(t_{n+1}, \boldsymbol{q}_{n+1})\mathrm{d}\boldsymbol{\Pi}_2 \tag{5.60a}$$

$$\boldsymbol{0}_{n_g} = \dot{\boldsymbol{g}}(t_{n+1}, \boldsymbol{q}_{n+1}, \boldsymbol{u}_{n+1}) \tag{5.60b}$$

$$\boldsymbol{0}_{n_\gamma} = \boldsymbol{\gamma}(t_{n+1}, \boldsymbol{q}_{n+1}, \boldsymbol{u}_{n+1}) \tag{5.60c}$$

$$\begin{cases} \xi^k_{N,n+1} \in \mathcal{N}_{\mathbb{R}_0^-}(-\Delta\pi^k_{N,n+1}) & \text{if } k \in A_{n+1} \\ 0 = \Delta\pi^k_{N,n+1} & \text{if } k \notin A_{n+1} \end{cases} \tag{5.60d}$$

$$\begin{cases} \boldsymbol{\xi}^k_{F,n+1} \in \mathcal{N}_{C^k_F(\Delta\pi^k_{N,n+1})}(-\Delta\boldsymbol{\pi}^k_{F,n+1}) & \text{if } k \in A_{n+1} \\ 0 = \Delta\boldsymbol{\pi}^k_{F,n+1} & \text{if } k \notin A_{n+1}\,, \end{cases} \tag{5.60e}$$

where $\Delta\boldsymbol{\pi}_{n+1} = \frac{1}{2}\mathrm{d}\boldsymbol{\Pi}_1 + \frac{1}{2}\mathrm{d}\boldsymbol{\Pi}_2$. It is noteworthy to mention that the presented nonsmooth RATTLE algorithm is decomposed into two independent stages, which reduced the computational costs of the algorithm significantly. First, solve (5.59) and subsequently (5.60), which consists of a linear system subject to normal cone inclusions, very much related to the velocity projection of Section 4.6.

Shifted Moreau

Besides the presented Lobatto IIIA-IIIB methods, other possible Butcher tableaus can be applied to the presented nonsmooth partitioned Runge–Kutta methods. Thereto, the requirement of a stiffly accurate first method is dropped. This implies that the bilateral and unilateral constraint conditions are not satisfied on position level. Moreover, in the case of an explicit method, the active constraints $k \in A_{n+1}$ are determined by an explicit prediction. The most simple choice is given by using twice the explicit Euler method, defined in terms of the tableau

$$\begin{array}{c|c} \boldsymbol{c} & \boldsymbol{A} \\ \hline & \boldsymbol{b}^{\mathrm{T}} \end{array} = \begin{array}{c|c} \boldsymbol{c} & \hat{\boldsymbol{A}} \\ \hline & \boldsymbol{b}^{\mathrm{T}} \end{array} = \begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}. \tag{5.61}$$

Since this Butcher tableau consists of a single explicit stage, the resulting method performs an explicit prediction of the positions and the set of active contacts, i.e.,

$$\boldsymbol{q}_{n+1} = \boldsymbol{q}_n + h\boldsymbol{f}(t_n, \boldsymbol{q}_n, \boldsymbol{u}_n) \quad \text{and} \quad A_{n+1} = A(t_{n+1}, \boldsymbol{q}_{n+1})\,, \tag{5.62}$$

respectively. The velocities and Lagrange multipliers are subsequently computed in accordance with

$$\boldsymbol{u}_{n+1} = \boldsymbol{u}_n + h\boldsymbol{k}(t_n, \boldsymbol{q}_n, \boldsymbol{u}_n) + \boldsymbol{l}(t_n, \boldsymbol{q}_n)\Delta\boldsymbol{\pi}_{n+1} \tag{5.63a}$$

$$\mathbf{0}_{n_g} = \dot{\boldsymbol{g}}(t_{n+1}, \boldsymbol{q}_{n+1}, \boldsymbol{u}_{n+1}) \tag{5.63b}$$

$$\mathbf{0}_{n_\gamma} = \boldsymbol{\gamma}(t_{n+1}, \boldsymbol{q}_{n+1}, \boldsymbol{u}_{n+1}) \tag{5.63c}$$

$$\begin{cases} \xi^k_{N,n+1} \in \mathcal{N}_{\mathbb{R}_0^-}(-\Delta\pi^k_{N,n+1}) & \text{if } k \in A_{n+1} \\ 0 = \Delta\pi^k_{N,n+1} & \text{if } k \notin A_{n+1} \end{cases} \tag{5.63d}$$

$$\begin{cases} \boldsymbol{\xi}^k_{F,n+1} \in \mathcal{N}_{C^k_F(\Delta\pi^k_{N,n+1})}(-\Delta\boldsymbol{\pi}^k_{F,n+1}) & \text{if } k \in A_{n+1} \\ 0 = \Delta\boldsymbol{\pi}^k_{F,n+1} & \text{if } k \notin A_{n+1}\,. \end{cases} \tag{5.63e}$$

Astonishingly, the resulting method exactly coincides with the shifted Moreau scheme presented in [40]. A closer inspection of the method (5.63) reveals that the classically used Moreau's mid-point rule can easily be recovered by the subsequent arguments. Let $\boldsymbol{q}_{n+1}$ denote the positions of the mid-points of the classical Moreau scheme. Starting the time integration with the positions $\boldsymbol{q}_0 = \boldsymbol{q}(t_0) - \frac{h}{2}\boldsymbol{f}\big(t_0, \boldsymbol{q}(t_0), \boldsymbol{u}(t_0)\big)$ results exactly in Moreaus mid-point rule, where the numerically obtained values of $\boldsymbol{q}_n$ coincide with $\boldsymbol{q}_{n+1/2}$ of Moreau's mid-point rule, cf. [74, 130]. The true positions at the grid points, denoted $\boldsymbol{q}^*_n$, can be computed a posteriori to a numerical simulation via $\boldsymbol{q}^*_n = \boldsymbol{q}_n + \frac{h}{2}\boldsymbol{f}\big(t_n, \boldsymbol{q}_n, \boldsymbol{u}_n\big)$.

Chapter 6

Solving Nonlinear Normal Cone Inclusions

This chapter discusses how to solve a system of nonlinear equations subject to a set of different nonlinear normal cone inclusion with respect to closed nonempty and convex sets. Such problems commonly emerge from the time discretization of mechanical systems subject to bilateral and unilateral constraints as presented in the previous chapters. First of all, the equivalence relation between normal cone inclusions and the natural map, already introduced in Proposition 2.14, is revisited in Section 6.1. Based on this reformulation, a semi-smooth Newton method is presented subsequently. In order to cope with redundant contacts, Section 6.3 introduces an iterative method for solving the nonlinear normal cone inclusion problem. Since the performance and convergence of the presented algorithms crucially depend on the choice of arbitrary positive parameters, the final section briefly discusses a possible estimation of these parameters.

6.1 Proximal Point Equation and Natural Map

Let $\boldsymbol{x} \in \mathbb{R}^m$ and $\boldsymbol{y} \in \mathbb{R}^n$ denote the tuples of unknowns. Further, $\boldsymbol{\varphi}\colon \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^m$ denotes a nonlinear function for which the gradient $\partial\boldsymbol{\varphi}/\partial\boldsymbol{x}(\boldsymbol{x},\boldsymbol{y}) \in \mathbb{R}^{m\times m}$ is assumed to be regular. Moreover, the matrix $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{m\times m}$ is assumed to be regular. This chapter discusses different methods for solving

$$\boldsymbol{\varphi}(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{0}_m \quad \text{subject to} \quad \boldsymbol{\psi}^k(\boldsymbol{x}) \in \mathcal{K}^k(\boldsymbol{\mathcal{A}}\,\boldsymbol{y})\,. \tag{6.1}$$

Therein, the function $\boldsymbol{\psi}^k$ can be decomposed as

$$\boldsymbol{\psi}^k(\boldsymbol{x}) = \begin{pmatrix} \psi_N^k(\boldsymbol{x}) \\ \boldsymbol{\psi}_F^k(\boldsymbol{x}) \end{pmatrix} . \tag{6.2}$$

Let $\boldsymbol{y} = (y_N^1, \dots, y_N^p, \boldsymbol{y}_F^1, \dots, \boldsymbol{y}_F^p) \in \mathbb{R}^n$ and consequently $\boldsymbol{\mu} = \boldsymbol{\mathcal{A}}\boldsymbol{y} = (\mu_N^1, \dots, \mu_N^p, \boldsymbol{\mu}_F^1, \dots, \boldsymbol{\mu}_F^p) \in \mathbb{R}^n$. This investigation is restricted to sets of the form

$$\mathcal{K}^k(\boldsymbol{\mu}) = \mathcal{N}_{\mathbb{R}_0^-}(-\mu_N^k) \times \mathcal{N}_{B_{f^k}(r^k \mu_N^k)}(-\boldsymbol{\mu}_F^k) , \tag{6.3}$$

i.e., Cartesian products of the normal cone to the non-positive real numbers $\mathbb{R}_0^-$ and the normal cone to the $\mathbb{R}^{f^k}$-ball $B_{f^k}(r^k \mu_N^k)$ with radius $r^k \mu_N^k$. As shown in Chapter 3, such a setup can describe a huge class of mechanical systems with frictional contact. The set (6.3) is restricted to a single $\mathbb{R}^{f^k}$-ball normal cone of radius $r^k \mu_N^k$. Its extension to a variety of such sets is straightforward but requires the introduction of multiple different indices. To keep the subsequent investigations concise and in order to improve readability this is not included herein, as already discussed at the end of Section 3.5.

Although normal cone inclusions are valuable tools for describing physical force and impact laws, they are rather unsuitable to use within a numerical method. Consequently, this section deals with the reformulation of normal cone inclusions as implicit equations. This enables different solution strategies known from literature. From Proposition (2.14) it is known that normal cone inclusions can be equivalently stated as implicit functions using the natural map. Consequently, the normal cone inclusions $\boldsymbol{\psi}^k(\boldsymbol{x}) \in \mathcal{K}^k(\boldsymbol{\mu})$ from (6.1) are equivalent to finding the zeros of the implicit equations

$$F_{N,k}^{\mathrm{nat}}(\boldsymbol{x}, \boldsymbol{\mu}) = \mu_N^k + \mathrm{prox}_{\mathbb{R}_0^-}\left(\rho_N^k \psi_N^k(\boldsymbol{x}) - \mu_N^k\right) = 0 \tag{6.4a}$$

$$\boldsymbol{F}_{F,k}^{\mathrm{nat}}(\boldsymbol{x}, \boldsymbol{\mu}) = \boldsymbol{\mu}_F^k + \mathrm{prox}_{B_{f^k}(r^k \mu_N^k)}\left(\rho_F^k \boldsymbol{\psi}_F^k(\boldsymbol{x}) - \boldsymbol{\mu}_F^k\right) = \mathbf{0}_{f^k} , \tag{6.4b}$$

where $\rho_N^k, \rho_F^k > 0$ denote arbitrary positive real parameters that will be specified in Section 6.4. Appending the nonlinear equations (6.4) to $\boldsymbol{\varphi}(\boldsymbol{x}, \boldsymbol{y}) = \mathbf{0}_m$ gives rise to Newton-type solution procedures. Besides that, rearrangement of (6.4) can be utilized to obtain a fixed-point iteration scheme. Both strategies are discussed in the subsequent sections. It should be noted that more elaborated solution strategies exists. For instance, a reformulation of the frictional contact problem as second-order cone complementarity problem gives rise to novel interior point algorithms, see [7]. Moreover, variants of nonlinear Gauss–Seidel methods [135] are

promising alternatives to a simple semi-smooth Newton method. Since these methods are topics of current research they are considered to be out of scope of this thesis and will not be investigated further.

6.2 Semi-smooth Newton Method

As already indicated above, using the natural maps introduced in (6.4) reveals that solving (6.1) is equivalent to finding the zero of the nonlinear system of equations

$$\boldsymbol{F}(\boldsymbol{z}) = \begin{pmatrix} \boldsymbol{\varphi}(\boldsymbol{x},\boldsymbol{y}) \\ F^{\mathrm{nat}}_{N,1}(\boldsymbol{x},\boldsymbol{\mathcal{A}y}) \\ \vdots \\ F^{\mathrm{nat}}_{N,p}(\boldsymbol{x},\boldsymbol{\mathcal{A}y}) \\ \boldsymbol{F}^{\mathrm{nat}}_{F,1}(\boldsymbol{x},\boldsymbol{\mathcal{A}y}) \\ \vdots \\ \boldsymbol{F}^{\mathrm{nat}}_{F,p}(\boldsymbol{x},\boldsymbol{\mathcal{A}y}) \end{pmatrix} =: \begin{pmatrix} \boldsymbol{\varphi}(\boldsymbol{x},\boldsymbol{y}) \\ \boldsymbol{F}^{\mathrm{nat}}(\boldsymbol{x},\boldsymbol{\mathcal{A}y}) \end{pmatrix}, \tag{6.5}$$

where $\boldsymbol{z} = (\boldsymbol{x},\ \boldsymbol{y}) \in \mathbb{R}^{m+n}$. Since the proximal point equation contained in the natural maps (6.4) is semi-smooth, the composed system (6.5) can be solved efficiently by application of a semi-smooth Newton method [62]. Let $\boldsymbol{z}_0 \in \mathbb{R}^{m+n}$ be a given initial guess that satisfies (6.1). The semi-smooth Newton method computes

$$\boldsymbol{z}_{i+1} = \boldsymbol{z}_i + \Delta\boldsymbol{z}_i \quad \text{with} \quad \boldsymbol{H}_i \Delta\boldsymbol{z}_i = -\boldsymbol{F}(\boldsymbol{z}_i)\,, \tag{6.6}$$

where $\boldsymbol{H}_i$ is an element of the generalized Jacobian at $\boldsymbol{z}_i$, see e.g. [6, 146] for more details. The update (6.6) is performed until a given convergence criterion is reached. Typically, $\|\boldsymbol{F}(\boldsymbol{z}_{i+1})\|_2 < \varepsilon$ or $\|\boldsymbol{F}(\boldsymbol{z}_{i+1})\|_\infty < \varepsilon$, where $\varepsilon \ll 1$ denotes some user-defined tolerance. In the latter case, each individual equation contained in $\boldsymbol{F}$ is solved with a precision of at least ε. This approach should be preferred if very accurate results are required.

More sophisticated error measures are obtained by introducing the concept of relative and absolute tolerances, typically denoted *rtol* and *atol*, respectively. Thereto, the Newton iteration is accepted as converged if

$$\left\| \frac{\boldsymbol{F}(\boldsymbol{z}_{i+1})}{(m+n)\big(atol + rtol|\boldsymbol{F}(\boldsymbol{z}_0)|\big)} \right\|_2 < 1 \tag{6.7}$$

is satisfied, cf. [104, 105]. Therein, the division and absolute value function have to be understood element-wise. Such combinations of relative and

absolute errors are commonly used in numerical methods for ordinary differential equations or differential algebraic equations, see for instance [28, 82, 84, 140].

From an algorithmic viewpoint it is essential to replace (6.4) by slight reformulations. Assume $\rho_N^k \psi_N(\boldsymbol{x}) - \mu_N^k \in \mathbb{R}_0^-$, hence, the proximal point is the identity mapping and the zero of $F_{N,k}^{\text{nat}}(\boldsymbol{x}, \boldsymbol{\mathcal{A}y}) = \rho_N^k \psi_N(\boldsymbol{x})$ solves the normal cone inclusion (6.4a). Depending on the chosen value of ρ_N^k this results in a disturbed residual. Let $\varepsilon \ll 1$ denote a required tolerance of the semi-smooth Newton method. For small values of ρ_N^k the method seems to be converged since $\|\rho_N^k \psi_N(\boldsymbol{x})\| < \varepsilon$, although $\|\psi_N(\boldsymbol{x})\| > \varepsilon$ indicates the contrary. Similarly, for large values of ρ_N^k, the criterion $\|\rho_N^k \psi_N(\boldsymbol{x})\| \ll \varepsilon$ might never be reached and Newton's method seems not to converge although $\|\psi_N(\boldsymbol{x})\| \ll \varepsilon$ is already met. The identical arguments apply for (6.4b). Consequently, the usage of the implicit equations

$$F_{N,k}^{\text{nat}}(\boldsymbol{x}, \boldsymbol{\mu}) = \begin{cases} \psi_N(\boldsymbol{x}) & \text{if } a_N^k \in \mathbb{R}_0^- \,, \\ \mu_N^k & \text{if } a_N^k \notin \mathbb{R}_0^- \,, \end{cases} \tag{6.8a}$$

$$\boldsymbol{F}_{F,k}^{\text{nat}}(\boldsymbol{x}, \boldsymbol{\mu}) = \begin{cases} \boldsymbol{\psi}_F(\boldsymbol{x}) & \text{if } \boldsymbol{a}_F^k \in B_{f^k}(r^k \mu_N^k) \,, \\ \boldsymbol{\mu}_F^k + r^k \mu_N^k \frac{\boldsymbol{a}_F^k}{\|\boldsymbol{a}_F^k\|} & \text{if } \boldsymbol{a}_F^k \notin B_{f^k}(r^k \mu_N^k) \,, \end{cases} \tag{6.8b}$$

is beneficial and entirely resolves this issue. Therein, the abbreviations $a_N^K = \rho_N^k \psi_N(\boldsymbol{x}) - \mu_N^k$ and $\boldsymbol{a}_F^k = \rho_F^k \boldsymbol{\psi}_F(\boldsymbol{x}) - \boldsymbol{\mu}_F^k$ were used. The usage of (6.8a) and (6.8b) is known as *primal-dual active set strategy*. It is shown by Hintermueller et al. [90] that this can again be interpreted as a semi-smooth Newton method. The extension to frictional contacts is given in [95]. For the case $\boldsymbol{a}_F^k \notin B_{f^k}(r^k \mu_N^k)$ there exist further useful reformulations, entirely eliminating the parameter ρ_F^k from the modified natural map (6.8b). The interested reader is referred to [6, 39, 41] for details of such reformulations. The basic structure of the proposed semi-smooth Newton method is summarized in Algorithm 1.

Noteworthy performance improvements can be obtained by application of a block LU-decomposition, naturally leading to the Schur complement of the matrix $\boldsymbol{H}_i$, see [21, 76] and the corresponding discussion in Section 2.4. Let $\boldsymbol{x}_0 \in \mathbb{R}^m$, $\boldsymbol{y}_0 \in \mathbb{R}^n$ be a given initial guess that satisfies (6.5). The semi-smooth Newton method (6.6) computes the updates

$$\begin{pmatrix} \boldsymbol{x}_{i+1} \\ \boldsymbol{y}_{i+1} \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_i \\ \boldsymbol{y}_i \end{pmatrix} + \begin{pmatrix} \Delta \boldsymbol{x}_i \\ \Delta \boldsymbol{y}_i \end{pmatrix} \,, \quad \boldsymbol{H}_i \begin{pmatrix} \Delta \boldsymbol{x}_i \\ \Delta \boldsymbol{y}_i \end{pmatrix} = -\boldsymbol{F}(\boldsymbol{x}_i, \boldsymbol{y}_i) \,. \tag{6.9}$$

Algorithm 1: Semi-smooth Newton method

Input: $\boldsymbol{z}_0 \in \mathbb{R}^{m+n}$, $0 < rtol, atol \ll 1$,
NEWTON_MAXITER $\in \mathbb{N}$
$\boldsymbol{s} = (m+n)(atol + rtol|\boldsymbol{F}(\boldsymbol{z}_0)|)$ `# scaling for error measure`
$i = 0$
while $\|\boldsymbol{F}(\boldsymbol{z}_i)/\boldsymbol{s}\|_2 \geq 1$ **do**
  $\boldsymbol{z}_{i+1} = \boldsymbol{z}_i - \boldsymbol{H}(\boldsymbol{z}_i)^{-1}\boldsymbol{F}(\boldsymbol{z}_i)$
  $i = i + 1$
  if $i \geq$ NEWTON_MAXITER **then**
    return $\boldsymbol{z}_i$ `# Newton not convergence`
  end
end
return $\boldsymbol{z}_i$ `# Newton converged`

Subsequently, it is assumed that $\boldsymbol{H}_i$ takes the form

$$\boldsymbol{H}_i = \begin{pmatrix} \boldsymbol{H}_{\boldsymbol{xx}}(\boldsymbol{x}_i, \boldsymbol{y}_i) & \boldsymbol{H}_{\boldsymbol{xy}}(\boldsymbol{x}_i, \boldsymbol{y}_i) \\ \boldsymbol{H}_{\boldsymbol{yx}}(\boldsymbol{x}_i, \boldsymbol{y}_i) & \boldsymbol{H}_{\boldsymbol{yy}}(\boldsymbol{x}_i, \boldsymbol{y}_i) \end{pmatrix}, \tag{6.10}$$

with $\boldsymbol{H}_{\boldsymbol{xx}}(\boldsymbol{x}_i, \boldsymbol{y}_i)$ being regular. Hence, the first line of (6.9) can be solved for

$$\Delta\boldsymbol{x}_i = -\boldsymbol{H}_{\boldsymbol{xx}}(\boldsymbol{x}_i, \boldsymbol{y}_i)^{-1}\left(\boldsymbol{\varphi}(\boldsymbol{x}, \boldsymbol{y}) + \boldsymbol{H}_{\boldsymbol{xy}}(\boldsymbol{x}_i, \boldsymbol{y}_i)\Delta\boldsymbol{y}_i\right). \tag{6.11}$$

This can be substituted into the second line of (6.9) and after minor refactoring results in the linear system

$$\boldsymbol{S}_i \Delta\boldsymbol{y}_i = \boldsymbol{b}_i \tag{6.12}$$

with

$$\begin{aligned} \boldsymbol{S}_i &= \boldsymbol{H}_{\boldsymbol{yy}}(\boldsymbol{x}_i, \boldsymbol{y}_i) - \boldsymbol{H}_{\boldsymbol{yx}}(\boldsymbol{x}_i, \boldsymbol{y}_i)\boldsymbol{H}_{\boldsymbol{xx}}(\boldsymbol{x}_i, \boldsymbol{y}_i)^{-1}\boldsymbol{H}_{\boldsymbol{xy}}(\boldsymbol{x}_i, \boldsymbol{y}_i) \\ \boldsymbol{b}_i &= \boldsymbol{H}_{\boldsymbol{yx}}(\boldsymbol{x}_i, \boldsymbol{y}_i)\boldsymbol{H}_{\boldsymbol{xx}}(\boldsymbol{x}_i, \boldsymbol{y}_i)^{-1}\boldsymbol{\varphi}(\boldsymbol{x}_i, \boldsymbol{y}_i) - \boldsymbol{F}^{\mathrm{nat}}(\boldsymbol{x}_i, \boldsymbol{y}_i), \end{aligned} \tag{6.13}$$

where the matrix $\boldsymbol{S}_i$ is denoted Schur complement of the block $\boldsymbol{H}_{\boldsymbol{xx}}$ of $\boldsymbol{H}_i$. At a first sight (6.13) looks quiet involved, compared to the simple Newton update proposed in (6.6). Therefore, it requires a more in-depth discussion. On the one hand the LU-decomposition of $\boldsymbol{H}_{\boldsymbol{xx}}$ can be reused for multiple iterations (or even multiple time-steps), resulting in an inexact Newton method, which only requires rather cheap forward and backward substitutions of triangular matrices. Moreover, the gradient $\boldsymbol{H}_{\boldsymbol{xy}}(\boldsymbol{x}_i, \boldsymbol{y}_i)$ can be kept constant until a new LU-decomposition of $\boldsymbol{H}_{\boldsymbol{xx}}(\boldsymbol{x}_i, \boldsymbol{y}_i)$ is

requested. Both strategies together significantly speed up the numerical method. Since $\boldsymbol{F}^{\mathrm{nat}}$ is semi-smooth, the corresponding generalized gradients $\boldsymbol{H}_{\boldsymbol{yx}}$ and $\boldsymbol{H}_{\boldsymbol{yy}}$ have to be updated within each iteration.

6.3 The Iterative Projection Method

In order to compute the update $\Delta \boldsymbol{z}_i$ appearing in the semi-smooth Newton update (6.6) the matrix $\boldsymbol{H}_i$ is required to be regular. Similarly, the update of $\Delta \boldsymbol{y}_i$ in (6.12) requires $\boldsymbol{S}_i$ to be regular. For applications involving multiple contacts, typically present in rigid body simulations like granular media and robotics, this assumption cannot be made in general. Often redundant contacts have to be solved due to rigid body contact interactions [71, 179]. A key concept for the solution of such a class of systems is to solve (6.1) by methods that do not require such a regularity assumption. The most simple method of this type is based on fixed-point iterations. The basic structure of such an iterative procedure is given in the following. For given initial guesses $\boldsymbol{x}_0 \in \mathbb{R}^m$ and $\boldsymbol{y}_0 \in \mathbb{R}^n$, fulfilling $\boldsymbol{\varphi}(\boldsymbol{x}_0, \boldsymbol{y}_0) = \boldsymbol{0}_m$, the resulting algorithm incrementally computes

$$\boldsymbol{\mu}_i = \boldsymbol{\mathcal{A}} \boldsymbol{y}_i \tag{6.14a}$$

$$\boldsymbol{\mu}_{i+1} = -\begin{pmatrix} \operatorname{prox}_{\mathbb{R}_0^-}\left(\rho_N^1 \psi_N^1(\boldsymbol{x}_i) - \mu_{N,i}^1\right) \\ \vdots \\ \operatorname{prox}_{\mathbb{R}_0^-}\left(\rho_N^p \psi_N^p(\boldsymbol{x}_i) - \mu_{N,i}^p\right) \\ \operatorname{prox}_{B_{f^1}(r^1 \mu_{N,i}^1)}\left(\rho_F^1 \boldsymbol{\psi}_F^1(\boldsymbol{x}_i) - \boldsymbol{\mu}_{F,i}^1\right) \\ \vdots \\ \operatorname{prox}_{B_{f^p}(r^p \mu_{N,i}^p)}\left(\rho_F^p \boldsymbol{\psi}_F^p(\boldsymbol{x}_i) - \boldsymbol{\mu}_{F,i}^p\right) \end{pmatrix} =: -\operatorname{prox}(\boldsymbol{x}_i, \boldsymbol{\mu}_i) \tag{6.14b}$$

$$\boldsymbol{y}_{i+1} = \boldsymbol{\mathcal{A}}^{-1} \boldsymbol{\mu}_{i+1} \text{ and find } \boldsymbol{x}_{i+1} \text{ such that } \boldsymbol{\varphi}(\boldsymbol{x}_{i+1}, \boldsymbol{y}_{i+1}) = \boldsymbol{0}_m \tag{6.14c}$$

until either $\|\boldsymbol{x}_{i+1} - \boldsymbol{x}_i\| < \varepsilon$ or $\|\boldsymbol{y}_{i+1} - \boldsymbol{y}_i\| < \varepsilon$. Although both convergence criteria are valid choices they require further discussion. Assume a problem with non-unique contact forces. There might be the possibility that the iteration method ends up in a cyclic jump between different Lagrange multipliers contained in $\boldsymbol{y}$, although the values of $\boldsymbol{x}$ are already converged. This motivates the usage of the first convergence criteria. Moreover, the tuple $\boldsymbol{x}$ might also contain Lagrange multipliers related to bilateral constraints. When such a described cycling behavior occurs, the possibility arises that also the bilateral Lagrange multipliers show such a cycling.

Hence, they also have to be excluded from the convergence criteria in order to improve the stability of the resulting method.

Similar to the discussion of the previous section and as detailed in Section II.4 of [84], an even more sophisticated convergence criterion is obtained by introducing absolute and relative tolerances, commonly denoted *atol* and *rtol*, respectively. Consequently, the fixed-point iteration is accepted as converged if

$$\left\| \frac{\boldsymbol{x}_{i+1} - \boldsymbol{x}_i}{m\big(atol + rtol \max(|\boldsymbol{x}_{i+1}|, |\boldsymbol{x}_i|)\big)} \right\|_2 < 1 \,. \tag{6.15}$$

Therein, the division, absolute value and maximum function have to be understood element-wise.

The basic fixed-point iteration method presented in (6.14) will now be outlined with more care. First of all, the nonlinear equation (6.14c) has to be solved. This is typically done using a Newton–Raphson method. The iteration is given by

$$\boldsymbol{x}_{i+1}^{k+1} = \boldsymbol{x}_{i+1}^{k} + \Delta\boldsymbol{x}_{i+1}^{k+1} \,, \qquad \boldsymbol{x}_{i+1}^{0} = \boldsymbol{x}_i \tag{6.16a}$$

$$\boldsymbol{H}_{i+1}^{k} \Delta\boldsymbol{x}_{i+1}^{k+1} = -\boldsymbol{\varphi}(\boldsymbol{x}_{i+1}^{k}, \boldsymbol{y}_{i+1}) \,, \qquad \boldsymbol{H}_{i+1}^{k} = \nabla_{\boldsymbol{x}} \boldsymbol{\varphi}(\boldsymbol{x}_{i+1}^{k}, \boldsymbol{y}_{i+1}) \,, \tag{6.16b}$$

where (6.16b) is solved by LU-decomposition and forward/backward substitution. The iteration is continued until a given prescribed tolerance $\varepsilon \ll 1$, i.e., $\|\boldsymbol{\varphi}(\boldsymbol{x}_{i+1}^{k+1}, \boldsymbol{y}_{i+1})\| < \varepsilon$ is satisfied. Again, this can be improved by using the convergence criteria

$$\left\| \frac{\boldsymbol{\varphi}(\boldsymbol{x}_{i+1}^{k+1}, \boldsymbol{y}_{i+1})}{m\big(atol + rtol|\boldsymbol{\varphi}(\boldsymbol{x}_{i+1}^{0}, \boldsymbol{y}_{i+1})|\big)} \right\|_2 < 1 \,, \tag{6.17}$$

respecting different magnitudes of the quantities collected in $\boldsymbol{\varphi}(\boldsymbol{x}, \boldsymbol{y})$. For typical problems, a huge number of fixed-point iterations (6.14b) is required. Hence, solving the nonlinear system (6.14c) via (6.16) gets the bottleneck of the proposed algorithm. It is natural to consider the application of an inexact Newton method. That is, keeping the Jacobian $\boldsymbol{H}_{i+1}^{k} \coloneqq \nabla_{\boldsymbol{x}} \boldsymbol{\varphi}(\boldsymbol{x}_i, \boldsymbol{y}_i)$ constant during the fixed-point iterations. Moreover, a considerable amount of computation resources can be saved by reusing the Jacobian for various time-steps until a maximal number of Newton steps is exceeded or no convergence is obtained. The basic structure of the proposed iterative projection method is summarized in Algorithm 2.

Algorithm 2: Fixed-point iteration with inexact Newton

Input: $\boldsymbol{x}_0 \in \mathbb{R}^m$, $\boldsymbol{y}_0 \in \mathbb{R}^n$, $0 < rtol_F, atol_F, rtol_N, atol_N \ll 1$, FIXEDPOINT_MAXITER, NEWTON_MAXITER $\in \mathbb{N}$

$\boldsymbol{H} = \nabla_{\boldsymbol{x}} \boldsymbol{\varphi}(\boldsymbol{x}_0, \boldsymbol{y}_0)$ `# compute and store LU-decomposition`

for $i = 0, \ldots,$ FIXEDPOINT_MAXITER $- 1$ **do**

- $\boldsymbol{y}_{i+1} = -\boldsymbol{\mathcal{A}}^{-1}\mathrm{prox}(\boldsymbol{x}_i, \boldsymbol{\mathcal{A}}\boldsymbol{y}_i)$
- $\boldsymbol{s}_N = m(atol_N + rtol_N|\boldsymbol{\varphi}(\boldsymbol{x}_i, \boldsymbol{y}_i)|)$ `# scaling Newton method`
- $k = 0$
- $\boldsymbol{x}_{i+1}^k = \boldsymbol{x}_i$
- **while** $\|\boldsymbol{\varphi}(\boldsymbol{x}_{i+1}^k, \boldsymbol{y}_{i+1})/\boldsymbol{s}_N\|_2 \geq 1$ **do**
  - **if** $k \geq$ NEWTON_MAXITER **then**
    - **return** $\boldsymbol{x}_{i+1}$, $\boldsymbol{y}_{i+1}$ `# Newton not converged`
  - **end**
  - `# solve with forward/backward substitution and`
  - `# stored LU-decomposition of` $\boldsymbol{H}$
  - $\boldsymbol{x}_{i+1}^{k+1} = \boldsymbol{x}_{i+1}^k - \boldsymbol{H}^{-1}\boldsymbol{\varphi}(\boldsymbol{x}_{i+1}^k, \boldsymbol{y}_{i+1})$
  - $k = k + 1$
- **end**
- $\boldsymbol{x}_{i+1} = \boldsymbol{x}_{i+1}^k$ `# Newton converged`
- $\boldsymbol{s}_F = m\big(atol_F + rtol_F \max(|\boldsymbol{x}_{i+1}|, |\boldsymbol{x}_i|)\big)$ `# scaling`
- `# fixed-point`
- **if** $\|(\boldsymbol{x}_{i+1} - \boldsymbol{x}_i)/\boldsymbol{s}_F\|_2 < 1$ **then**
  - **return** $\boldsymbol{x}_{i+1}$, $\boldsymbol{y}_{i+1}$ `# fixed-point iteration converged`
- **end**

end

return $\boldsymbol{x}_{i+1}$, $\boldsymbol{y}_{i+1}$ `# fixed-point iteration not converged`

6.4 Choice of Proximal Point Parameters

The convergence and solvability properties of the aforementioned solution strategies crucially depend on the choice of the proximal point parameters $\rho_N^1, \ldots, \rho_N^p$ and $\rho_F^1, \ldots, \rho_F^p$. Different choices can be found in literature, e.g., [6, 125, 160, 169, 171]. A simple and at the same time very efficient choice is obtained by computing the parameters depending on the current value of the generalized force directions $\boldsymbol{W}_N$, $\boldsymbol{W}_F$ and the mass matrix $\boldsymbol{M}$. In order to save computational resources, a lazy evaluation of these quantities, let's say once every time-step, is essential. Since the involved quantities are already computed for all the presented methods, no additional

function evaluations are required. For the normal contacts the proximal point parameters are estimated by[1]

$$\rho_N^i = \frac{\alpha}{\boldsymbol{G}_{NN,ii}}, \quad \text{with} \quad \boldsymbol{G}_{NN} = \boldsymbol{W}_N^{\mathrm{T}} \boldsymbol{M}^{-1} \boldsymbol{W}_N \tag{6.18}$$

and $\alpha \in (0, 2)$. Typically, a smaller value α slows down the convergence but results in a convergent scheme if otherwise the method tends to diverge. For the friction forces a similar estimation is performed. For $k = 1, 2, \ldots, p$, the parameters are chosen in accordance with

$$\rho_F^k = \frac{\alpha}{\min\left(\boldsymbol{G}_{FF,11}^k, \ldots, \boldsymbol{G}_{FF,f^k f^k}^k\right)}, \quad \text{where} \quad \boldsymbol{G}_{FF}^k = (\boldsymbol{W}_F^k)^{\mathrm{T}} \boldsymbol{M}^{-1} \boldsymbol{W}_F^k . \tag{6.19}$$

To further reduce the computational costs of this estimation, the evaluation of (6.18) and (6.19) can be performed in a vectorized manner using a single LU-decomposition of the mass matrix $\boldsymbol{M}$.

A more sophisticated estimation strategy is mentioned in Section 2.5.5 of Möller [125]. Therein, the sensitivity of $\boldsymbol{\varphi}$ is taken into account, making the estimation aware of the underlying numerical method. Since for the simple estimation strategy outlined above no convergence issues are encountered within the numerical examples of the next chapter, a more involved estimation strategy is out of the scope of this thesis.

[1] Numerical experiments have shown that for unilateral constraints on position level it is beneficial to scale the estimated proximal point parameters ρ_N^i by the inverse step-size $\frac{1}{h}$. This observation is in line with the investigations of Macklin et al. [120] and can alternatively be justified by an analysis of the involved units.

Chapter 7

Application on Selected Benchmark Examples

This chapter introduces a collection of selected benchmark examples for nonsmooth mechanical systems with frictional contact and impacts. Each example therein investigates a specific class of systems that represents a typical real world problem. In order to keep the visualizations feasible, only the solution curves of a few integration methods are shown subsequently. When appropriate, at least one method of both presented families of nonsmooth Runge–Kutta methods is investigated. Specifically, most examples are investigated with the nonsmooth two-stage Radau IIA method as representative of the family of nonsmooth projected stiffly accurate Runge–Kutta methods proposed in Chapter 4 and the two-stage Lobatto IIIA-IIIB method[1] from the family of nonsmooth partitioned Runge–Kutta methods of Chapter 5. Besides these two methods, Moreau's mid-point rule [74, 130] is often chosen as reference. Sometimes additional other methods are investigated depending on the type of system and the demonstrated physical/numerical problem under consideration.

Unless stated otherwise, all subsequent examples are simulated by application of the iterative projection method presented in Section 6.3 and share the same tolerances. Specifically, the fixed-point iterations as well as the involved inexact Newton method use the relative and absolute tolerances $rtol = atol = 10^{-6}$. Moreover, the simple estimation for the proximal point parameters presented in Section 6.4 is used together with a possible scaling defined in terms of $\alpha \in (0, 2)$. Since the used geometry

[1] This method is subsequently abbreviated as RATTLE, although it does not exactly coincide with the nonsmooth RATTLE algorithm, presented in [29].

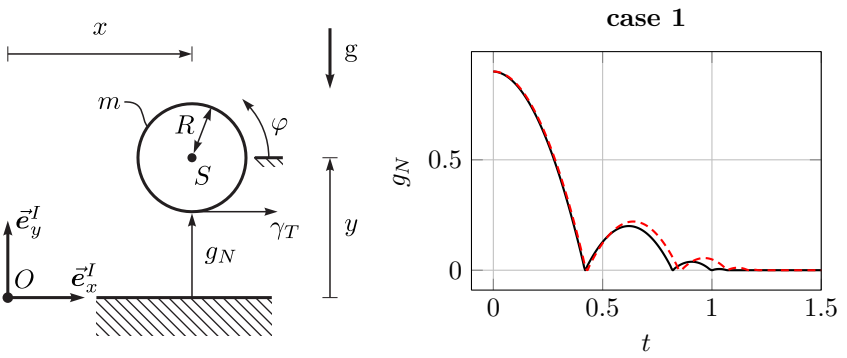

Figure 7.1: Sketch of the bouncing ball system (left) and simulated time evolution of the contact distance g_N for case 1, solved with the one-stage (——) and two-stage (----) nonsmooth Radau IIA method (right).

and material properties involved in the subsequent examples are chosen rather arbitrarily, no units are appended to the individual values. Every valid unit system could be chosen and the numerical results remain valid.

7.1 Rotating Bouncing Ball

This example continues the investigations of the one-dimensional bouncing ball, already introduced in Example 4.12, 4.13 and 5.2. Thereto, it follows [41, 71] and investigates the motion of a homogeneous sphere of radius $R = 0.1$ and mass $m = 1$ subjected to gravitational forces with gravitational acceleration $g = 10$, falling on a horizontal plane, as depicted in Figure 7.1. The sphere is constrained to move in the $\vec{e}_x^I$-$\vec{e}_y^I$-plane. Hence, it can be described by the generalized positions $\boldsymbol{q} = (x,\ y,\ \varphi)$. The tuple ${}_I\vec{\boldsymbol{r}}_{OS} = (x,\ y,\ 0)$ addresses the center of mass S with respect to the inertial basis I. Moreover, the balls orientation is described by the absolute angle φ with respect to the $\vec{e}_x^I$-axis. The generalized velocities $\boldsymbol{u} = (u_x,\ u_y,\ u_\varphi)$ are chosen to correspond to $\dot{\boldsymbol{q}}$ whenever the time derivative of the generalized positions exists. Consequently, the ingredients for the equality of measures (3.2) are given in terms of

$$\boldsymbol{M} = \operatorname{diag}(m,\ m,\ \Theta_S) \quad \text{and} \quad \boldsymbol{h} = (0,\ -mg,\ 0)\,, \tag{7.1}$$

with $\Theta_S = \frac{2}{5}mR^2$. For the description of planar Coulomb friction in line with (3.21), the contact distance and relative tangent velocity are given by

$$g_N = y - R \quad \text{and} \quad \gamma_T = u_x + Ru_\varphi\,, \tag{7.2}$$

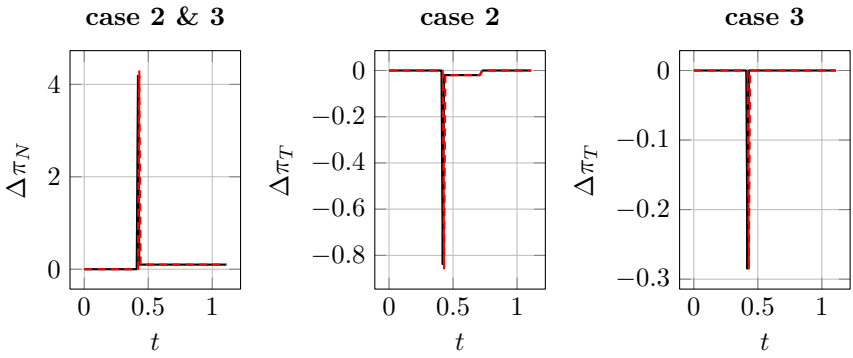

Figure 7.2: Simulated normal and friction contact percussion differences $\Delta\pi_N$ and $\Delta\pi_T$ for case 2 ($\omega = 50$) and case 3 ($\omega = 10$) solved with the one-stage (——) and two-stage (----) nonsmooth Radau IIA method.

together with friction coefficient $\mu_T = 0.2$ and restitution coefficient in tangent direction, $e_F = 0$. A number of different cases will be considered for the restitution coefficient in normal direction, e_N.

Specifically, three different cases are considered to extend the investigations from Example 4.12 and 4.13. Again, the one- and two-stage nonsmooth Radau IIA methods are applied. All cases share the same initial configuration given by $t_0 = 0$, $\boldsymbol{q}(t_0) = (0,\ 1,\ 0)$ and $\boldsymbol{u}(t_0) = (0,\ 0,\ \omega)$, such that the ball has an initial rotational velocity ω. The simulations are performed using a constant step-size $h = 10^{-2}$.

case 1: Starting from rest ($\omega = 0$) and assuming elastic impacts with $e_N = 0.5$, the well-known bouncing ball motion exhibiting the Zeno phenomenon is obtained. The simulation results shown in Figure 7.1 confirm that the nonsmooth projected stiffly accurate Runge–Kutta methods, presented in Section 4.6, can overcome accumulation points and enforce a desired elastic impact law. Even for the applied moderate time-step no penetration is obtained, since this is prohibited by the methods. Clearly, the first-order method ($s = 1$) dissipates energy during the free flight motion compared to the third-order method ($s = 2$). For $s \geq 3$, the solutions cannot be distinguished anymore since during the free flight, the results are already exactly calculated up to machine precision.

The subsequent two cases are used to test the behavior of the proposed methods with respect to friction forces. For both cases, inelastic impacts

are enforced with $e_N = 0$ implying that once the contact closes it remains closed, i.e., the post impact velocity is $u_y^+ = 0$, which allows a validation of friction. Since the ball is constantly accelerated by gravity, the dynamics of the ball induced by (7.1) reveals that the frictional impact occurs at $t_1 = \sqrt{2(y(t_0) - R)/g}$. The corresponding pre-impact velocity is given by $u_y^- = -\sqrt{2g(y(t_0) - R)}$. It follows from the impact equation (3.43) that the impulsive normal contact force is $\Lambda_N = m\sqrt{2g(y(t_0) - R)} \approx 4.2426$. Furthermore, after the impact the non-impulsive normal contact force compensates the gravitational force and therefore takes the value $\lambda_N = mg = 10$. Consequently, the percussion difference takes the value $\Delta\pi_N = h\lambda_N + \Lambda_N \approx 4.3426$. Depending on the value of ω, two additional cases arise.

case 2: For a high initial rotational velocity ($\omega = 50$) sliding occurs after the impact. This implies that the friction forces attain the maximally allowed values $\Lambda_T = \mu\Lambda_N \approx 0.8485$ and $\lambda_T = \mu\lambda_N = 2$, leading to the percussion difference $\Delta\pi_T = h\lambda_T + \Lambda_T \approx 0.8685$. After undergoing a period of sliding contact, the ball slows down enough to enter into a slip-stick transition. At this point, the ball begins to move in a pure rolling motion, which is described by the kinematic condition $\gamma_T = 0$. Since the rolling motion is described by constant velocities, no net friction forces occur, implying $\lambda_T = 0$. Hence, at the slip-stick transition the non-impulsive friction force instantly jumps to zero. Figure 7.2 illustrates that the presented nonsmooth projected stiffly accurate Runge–Kutta methods perfectly replicates the behavior described above.

case 3: In the third scenario, a rotational velocity of $\omega = 10$ is selected, which is small enough to cause the ball to stick upon impact, exhibiting a pure rolling motion. Again, this behavior is described by zero friction forces, i.e., $\lambda_T = 0$ for all times. The pre-impact velocities are given by $u_x^- = 0$ and $u_\varphi^- = \omega$. The contact sticks immediately after the impact, i.e., $\gamma_T^+ = 0 \Leftrightarrow u_x^+ = -Ru_\varphi^+$. Using this relation, the impact equations in horizontal direction and angular rotation can be combined yielding $u_\varphi^+ = \omega\Theta_S/(\Theta_S + mR^2)$. Consequently, the impulsive friction force as well as the percussion differences are given by $\Lambda_T = \Delta\pi_T = -mRu_\varphi^+ = -mR\omega\Theta_S/(\Theta_S + mR^2) = -\frac{2}{7}mR\omega \approx -0.2857$. As illustrated in Figure 7.2, the numerical solutions perfectly align with the results of [41] and reproduce the behavior described above.

In [29] and [42], similar investigations are performed by application of the nonsmooth RATTLE algorithm and the family of Lobatto IIIA-IIIB methods, respectively. For Hamiltonian systems with a trivial kinematic differential equation, the family of nonsmooth partitioned Runge–Kutta methods, presented in Chapter 5, coincide with the method of [42]. Furthermore, the nonsmooth RATTLE algorithm is a special method of this family with slightly different definitions of the involved Lagrange multipliers. Consequently, this demonstrates that these class of methods are well-suited for the solution of nonsmooth mechanical systems with frictional contact. Moreover, these methods can overcome accumulation points and exactly describe the desired frictional force and impact laws.

7.2 Rolling Ball on Plane with Analytical Solution

In what follows, the previously introduced rotating bouncing ball example will be extended by rolling friction. Hence, this section analyzes a combination of set-valued rolling and Coulomb friction. Thereto, it is assumed that the ball lies on the ground, i.e., $y = R$ and $\dot{y} = \ddot{y} = 0$. Consequently, the dynamic equilibrium yields $\lambda_N = mg > 0$. The description of sliding and rolling friction requires the relative tangent and spinning velocities

$$\gamma_T = u_x + Ru_\varphi \quad \text{and} \quad \gamma_R = u_\varphi \,, \tag{7.3}$$

respectively. The motion of the rolling ball can be distinguished by two different phases. The first one is characterized by a combined rolling and sliding motion, while in the second one, the relative tangent velocity vanishes and a pure rolling motion is exhibited.

phase 1: For $\gamma_T = u_x + Ru_\varphi > 0 \Leftrightarrow u_x > -Ru_\varphi$, the ball is sliding. Together with $u_\varphi < 0$, the friction forces are given by

$$\lambda_T = -\mu_T \lambda_N \quad \text{and} \quad \lambda_R = \mu_R \lambda_N \,. \tag{7.4}$$

Consequently, the motion of the ball is determined by the equations of motion

$$\begin{pmatrix} m & 0 \\ 0 & \Theta_S \end{pmatrix} \begin{pmatrix} \dot{u}_x \\ \dot{u}_\varphi \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ -R & 1 \end{pmatrix} \begin{pmatrix} \lambda_T \\ \lambda_R \end{pmatrix} = mg \begin{pmatrix} \mu_T \\ \mu_R - R\mu_T \end{pmatrix} , \tag{7.5}$$

which leads to the constant velocities $u_x = v_0 - \mu_T g t$ and $u_\varphi = \omega_0 + (\mu_R - R\mu_T) mgt/\Theta_S$. The time t_1, when pure rolling is attained can be computed from the condition for vanishing relative tangent velocity $\gamma_T = 0$:

$$u_x(t_1) \stackrel{!}{=} -Ru_\varphi(t_1) \quad \Longleftrightarrow \quad t_1 = \frac{v_0 + R\omega_0}{\mu_T g + R(\mu_R - R\mu_T) mg/\Theta_S} \,. \tag{7.6}$$

phase 2: At $t = t_1$, the relative tangent velocity vanishes, i.e., $\gamma_T = 0$. Consequently, stick can be described by the condition

$$\dot{\gamma}_T = 0 \quad \Longleftrightarrow \quad \dot{u}_x = -R\dot{u}_\varphi \,. \tag{7.7}$$

Together with the first line of (7.5) this leads to the force law

$$\lambda_T = -mR\dot{u}_\varphi \,. \tag{7.8}$$

Using this result in the second row of (7.5) reveals that the motion in the second phase is uniquely defined by the equation

$$\left(\Theta_S + mR^2\right)\dot{u}_\varphi = \lambda_R = \mu_R mg \,. \tag{7.9}$$

The ball comes to rest when the angular velocity vanishes, i.e.,

$$u_\varphi(t_2) = 0 \quad \Longleftrightarrow \quad t_2 = t_1 - \frac{\Theta_S + mR^2}{\mu_R mg} u_\varphi(t_1) \,. \tag{7.10}$$

Since pure rolling is enforced during the second phase, the horizontal velocity is characterized by $u_x = -Ru_\varphi$. The entire course of the ball's piecewise quadratic motion is determined by simple integration and can be summarized as

$$\dot{u}_\varphi(t) = \begin{cases} \frac{(\mu_R + R\mu_T)mg}{\Theta_S} =: a_{\varphi,0} & \text{for } 0 \le t < t_1 \\ \frac{\mu_R mg}{\Theta_S + mR^2} =: a_{\varphi,1} & \text{for } t_1 \le t < t_2 \end{cases} \tag{7.11a}$$

$$\dot{u}_x(t) = \begin{cases} -\mu_T g =: a_{x,0} & \text{for } 0 \le t < t_1 \\ -R\frac{\mu_R mg}{\Theta_S + mR^2} =: a_{x,1} & \text{for } t_1 \le t < t_2 \end{cases} \tag{7.11b}$$

$$u_\varphi(t) = \begin{cases} a_{\varphi,0}t + \omega_0 & \text{for } 0 \le t < t_1 \\ a_{\varphi,1}(t - t_1) + u_\varphi(t_1) & \text{for } t_1 \le t < t_2 \end{cases} \tag{7.11c}$$

$$u_x(t) = \begin{cases} a_{x,0}t + v_0 & \text{for } 0 \le t < t_1 \\ a_{x,1}(t - t_1) + u_x(t_1) & \text{for } t_1 \le t < t_2 \end{cases} \tag{7.11d}$$

$$\varphi(t) = \begin{cases} \frac{1}{2}a_{\varphi,0}t^2 + \omega_0 t + \varphi_0 & \text{for } 0 \le t < t_1 \\ \frac{1}{2}a_{\varphi,1}(t - t_1)^2 + u_\varphi(t_1)(t - t_1) + \varphi(t_1) & \text{for } t_1 \le t < t_2 \end{cases} \tag{7.11e}$$

$$x(t) = \begin{cases} \frac{1}{2}a_{x,0}t^2 + v_0 t + x_0 & \text{for } 0 \le t < t_1 \\ \frac{1}{2}a_{x,1}(t - t_1)^2 + u_x(t_1)(t - t_1) + x(t_1) & \text{for } t_1 \le t < t_2. \end{cases} \tag{7.11f}$$

Most of the parameters coincide with the previous example, i.e., $m = 1$, $R = 0.1$ and $g = 10$. The initial positions and velocities are given by

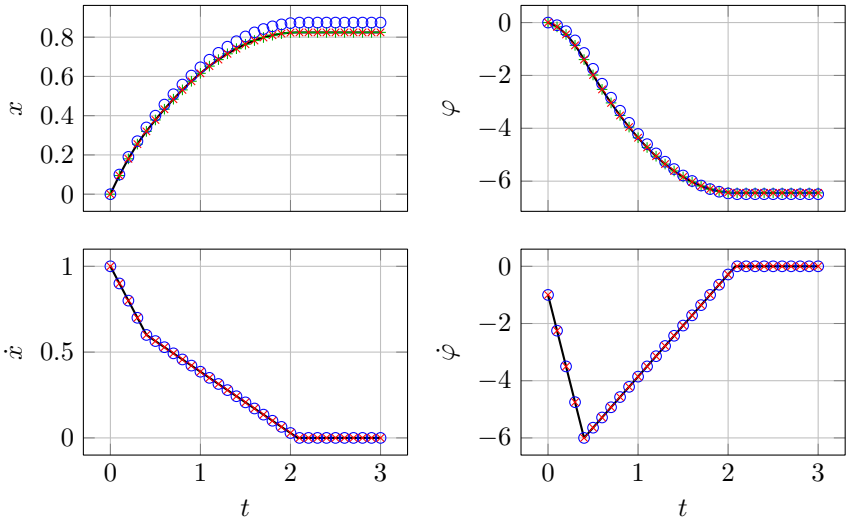

Figure 7.3: Time evolution of positions and velocities of the rolling ball on plane example. The analytical solution (——), given in (7.11), is compared with the shifted Moreau method (○) and Moreau's midpoint rule (×). The recovered positions of the shifted Moreau method are depicted as (+).

$t_0 = 0$, $x(t_0) = 0$, $y(t_0) = R$, $\varphi(t_0) = 0$, $\dot{x}(t_0) = 1$, $\dot{y} = 0$, $\dot{\varphi}(t_0) = -1$. The tangential and rolling friction coefficient are given by $\mu_T = 0.1$ and $\mu_R = 0.005$, respectively. Figure 7.3 compares the analytical solution of the problem, outlined above, with the values obtained with the shifted Moreau method and Moreau's mid-point rule [74]. Both methods succeed to solve the problem with a large step-size $h = 10^{-1}$. As expected, the solution of the velocities coincides for both methods, see (5.63a), but the positions differ. Clearly, the computed positions of Moreau's mid-point rule are superior compared to the ones obtained by the shifted Moreau method. Application of the shifted initial positions together with a post-processing step, outlined below (5.63), recovers the same order of accuracy, as depicted in Figure 7.3.

7.3 Ball in Hollow Sphere and Penetration Problems

As depicted in Figure 7.4, the subsequent example investigates the dynamics of a solid ball of mass $m = 1$ and radius $R = 0.1$ inside a hollow sphere of radius $R_s = 1$. Due to the curved contact geometry classical nonsmooth

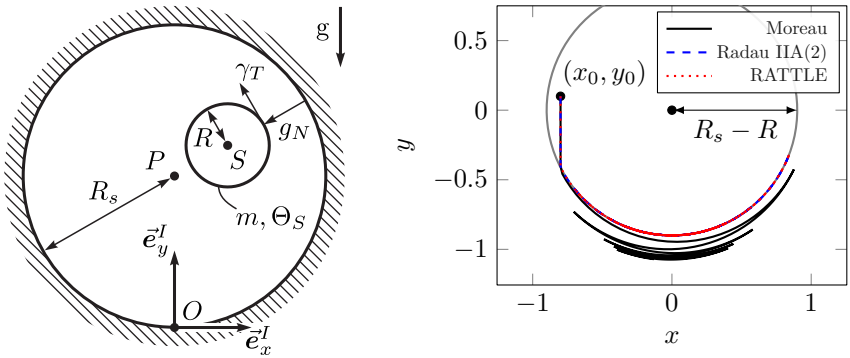

Figure 7.4: Sketch of the ball in hollow sphere (left). Trajectories of the ball's center of mass ${}_I\vec{\boldsymbol{r}}_{OS} = (x,\ y,\ 0)$ (right).

time integration methods, stated on velocity level only, suffer from unilateral drift or contact penetration problems. Therefore, the presented nonsmooth Runge–Kutta methods that enforce Signorini's law on position level are assumed to show better numerical results prohibiting penetration.

For simplicity, the investigation is again restricted to the planar case and the dynamics of the previous examples are still valid. Consequently, the ingredients for the equality of measures (3.2) coincide with those of Section 7.1 and 7.2. The normal contact between the ball and the hollow sphere is described by the gap function

$$g_N = R_\mathrm{s} - R - \|\vec{\boldsymbol{r}}_{SP}\|\,, \tag{7.12}$$

together with $e_N = 0$. Using ${}_I\vec{\boldsymbol{n}} = (n_x,\ n_y,\ 0)$ to denote the inertial components of the inward normal $\vec{\boldsymbol{n}} = \vec{\boldsymbol{r}}_{SP}/||\vec{\boldsymbol{r}}_{SP}||$ of the sphere, the inertial components of the tangent can be defined as ${}_I\vec{\boldsymbol{t}} = (n_y,\ -n_x,\ 0)$. Consequently, the relative tangent velocity is given by

$$\gamma_T = {}_I\vec{\boldsymbol{v}}_S^{\mathrm{T}}\,{}_I\vec{\boldsymbol{t}} + Ru_\varphi\,, \tag{7.13}$$

where the inertial components of the velocity of S are ${}_I\vec{\boldsymbol{v}}_S = (u_x,\ u_y,\ 0)$. The friction coefficient is chosen as $\mu_T = 0.1$ and inelastic tangential impacts are incorporated with restitution coefficient $e_F = 0$.

At $t_0 = 0$, the ball starts at rest from the initial position $\boldsymbol{q}(t_0) = (-0.8,\ 0.1,\ 0)$. Using a step-size $h = 2 \cdot 10^{-2}$, numerical simulations are performed with Moreau's mid-point rule [74, 126], the two-stage Lobatto IIIA-IIIB method (RATTLE) and the two-stage Radau IIA method. In

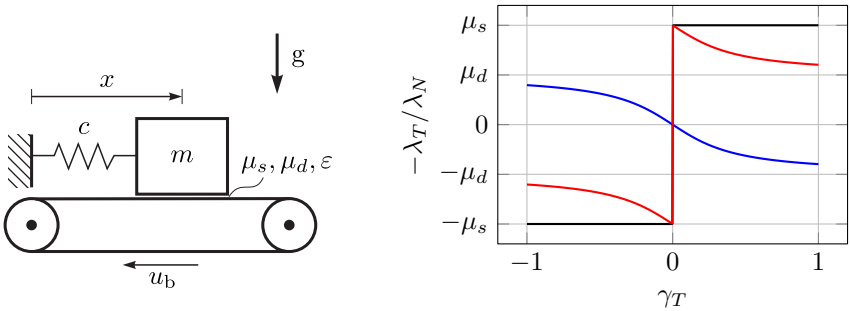

Figure 7.5: Sketch of the block on belt system (left). Decomposition of the friction law with Stribeck effect (right). Therein, $\lambda_T = \lambda_{T,0} + \lambda_{T,0}$ (——) with $\lambda_{T,0} = -\mu_s \lambda_N \operatorname{Sign}(\gamma_T)$ (——) and $\lambda_{T,1} = (\mu_s - \mu_d)\lambda_N \frac{\pi}{2} \arctan(\varepsilon \gamma_T)$ (——).

Figure 7.4, the trajectory of the center of mass ${}_I\vec{\boldsymbol{r}}_{OS} = (x,\ y,\ 0)$ is visualized. Clearly, the penetration problem of Moreau's mid-point rule is visible, illustrating the importance of the position level constraints. For both alternative methods no such penetration is visible, clearly demonstrating their superiority.

7.4 Block on Belt System and Stribeck Curve

The block-on-belt system is a well known archetype for systems showing stick-slip motion. The system has been studied extensively in the literature, e.g. [64, 113, 114, 115, 147], and typically involves a Stribeck-type friction law. Consequently, this example demonstrates how such a behavior can be included into the presented nonsmooth mechanical systems of Chapter 3 and how the proposed nonsmooth time integration methods can deal with it. Such a friction law can simply be incorporated by decomposing the friction force into two parts, i.e., $\lambda_T = \lambda_{T,0} + \lambda_{T,1}$, see Figure 7.5 and [114]. Therein, $\lambda_{T,0} = -\mu_s \lambda_N \operatorname{Sign}(\gamma_T)$ describes classical set-valued Coulomb friction, i.e., $\gamma_T \in \mathcal{N}_{B_1(\mu_s \lambda_N)}(-\lambda_{T,0})$. A possible Stribeck effect is taken into account by the second part $\lambda_{T,1} = (\mu_s - \mu_d)\lambda_N \frac{\pi}{2} \arctan(\varepsilon \gamma_T)$, lowering the friction force depending on the relative tangent velocity γ_T and the relation between a static and dynamic friction coefficient μ_s and μ_d, respectively. Clearly, the second term is smooth and can be added to the smooth part of the mechanical system gathered in $\boldsymbol{h}(t, \boldsymbol{q}, \boldsymbol{u})$.

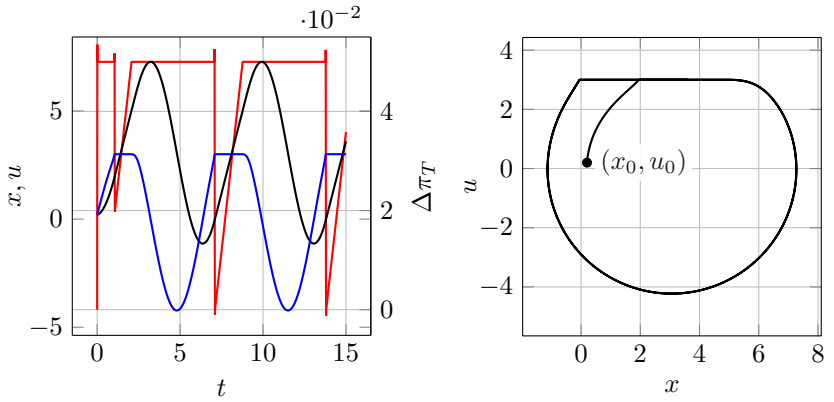

Figure 7.6: Time evolution of the position x (——), velocity u (——) and friction force $\Delta\pi_T$ (——) (left). Phase portrait (right).

Let a mass with $m = 1$ slide on a belt that runs with constant velocity $u_b = 0.2$. The position of the mass is described by the coordinate x, while its velocity is denoted u and satisfies $\dot{x} = u$, see Figure 7.5. The mass is connected to a linear spring with stiffness $c = 1$ that is relaxed for $x = 0$. The system is subjected to gravity with acceleration $g = 10$. Hence, a constant normal contact force $\lambda_N = mg = 10$ is obtained. The Stribeck-type friction law introduced above is characterized by $\mu_s = 0.5$, $\mu_d = 0.3$ and $\varepsilon = 3$. The relative contact velocity is given by $\gamma_T = u - u_\mathrm{b}$. Consequently, the system is described by

$$\begin{aligned} \dot{x} &= u \\ m\dot{u} &= -cx + \lambda_{T,0} + (\mu_s - \mu_d)\lambda_N \tfrac{\pi}{2} \arctan(\varepsilon\gamma_T) \\ \gamma_T &\in \mathcal{N}_{B_1(\mu_s\lambda_N)}(-\lambda_{T,0})\,. \end{aligned} \tag{7.14}$$

Using the two-stage nonsmooth Lobatto IIIA-IIIB method (RATTLE), the system is computed for $t \in [0, 15]$ with the initial conditions $x_0 = x(t_0) = 0.2$, $u_0 = u(t_0) = 0.2$ and a step-size $h = 10^{-2}$. The numerically obtained results are depicted in Figure 7.6. Clearly, a periodic motion is obtained after a short transition phase. Moreover, the phase portrait shows a nonsmooth transition from slip to stick, which is in line with the investigations of [113, 115].

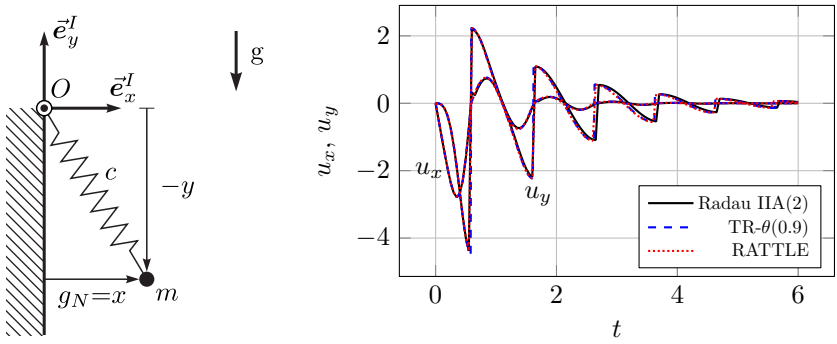

Figure 7.7: Sketch of the elastic pendulum system (left). Time evolution of the velocities $\boldsymbol{u} = (u_x,\ u_y)$ of the point mass (right).

7.5 Stiff Elastic Pendulum with Impacts

In what follows, a simple elastic but stiff Cartesian pendulum is investigated, which serves as a model problem for complex structural mechanics problems involving contacts. As depicted in Figure 7.7, the system consists of a point mass with mass $m = 1$ and a spring with high stiffness $c = 10^8$. The point mass is addressed by the Cartesian coordinates $\boldsymbol{q} = (x,\ y)$ and is subjected to gravitational forces with $g = 10$. The spring is considered relaxed for $\|\boldsymbol{q}\| = L = 1$. Whenever the time derivative of the generalized positions exists, they coincide with the generalized velocities $\boldsymbol{u} = (u_x,\ u_y)$, i.e., $\dot{\boldsymbol{q}} = \boldsymbol{u}$. The ingredients for the equality of measures (3.2) are given in terms of

$$\boldsymbol{M} = \begin{pmatrix} m & 0 \\ 0 & m \end{pmatrix} \quad \text{and} \quad \boldsymbol{h} = \begin{pmatrix} 0 \\ -mg \end{pmatrix} - c\frac{\boldsymbol{q}}{\|\boldsymbol{q}\|}(\|\boldsymbol{q}\| - L)\,. \tag{7.15}$$

The description of unilateral contact with the vertical wall, located at $x = 0$, is described by the contact distance

$$g_N = x\,, \tag{7.16}$$

together with a restitution coefficient $e_N = 0.5$.

Although the nonsmooth partitioned Runge–Kutta methods of Chapter 5 are well-suited for the already presented examples, serious step-size restrictions are encountered when stiff problems have to be solved. In order to prevent the numerical method to diverge, a step-size $h = 10^{-4}$ is required for the two-stage Lobatto IIIA-IIIB method (RATTLE). The same

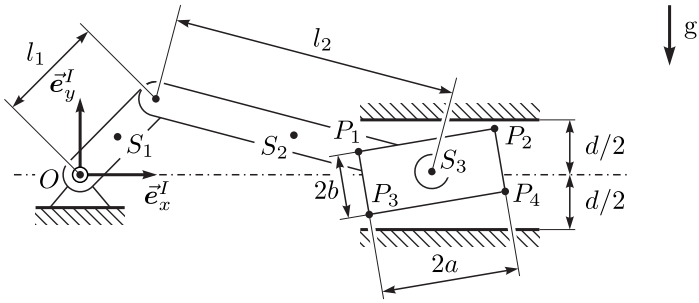

Figure 7.8: Sketch of the slider-crank mechanism.

small step-size is required if Moreau's mid-point rule should be applied. As the name suggests, the nonsmooth projected stiffly accurate Runge–Kutta methods from Chapter 4 do not lead to such restrictions. For the present problem, the two-stage Radau IIA method could successfully solve the problem with a step-size $h = 5 \cdot 10^{-2}$, which ultimately leads to 15 times faster solution in terms of runtime when compared with Moreau's mid-point rule. Similarly, the TR-$\theta(\rho_\infty)$ method with $\rho_\infty = 0.9$ can solve the problem with such a large time-step, although only very limited numerical damping is introduced by the method. The TR-$\theta(\rho_\infty)$ method solved the problem 10 times faster compared to Moreau's mid-point rule. As depicted in Figure 7.7, the numerical solutions between both class of methods can merely be distinguished, although their respective step-sizes differ by a factor of 500. Since the TR-$\theta(\rho_\infty)$ method introduces less numerical dissipation compared to the Radau IIA method, its solution is closer to the one found by RATTLE. In conclusion, it can be stated that in the presence of stiff springs, typically met in structural mechanics problems, the nonsmooth projected stiffly accurate Runge–Kutta methods are superior compared to the nonsmooth partitioned methods and should therefore be preferred. This is in line with the investigations of [99] for stiff systems described by differential algebraic equations.

7.6 Slider-Crank Mechanism

Inspired by the investigations of [63], this example examines a slider-crank mechanism subjected to both unilateral and bilateral constraints[2]. Hence,

[2]A similar problem, called the roller-crank mechanism, exhibits a much simpler contact geometry and is investigated in [123] and [156].

this example demonstrates the ability of the proposed nonsmooth Runge–Kutta methods to handle impulsive changes of bilateral constraint forces as well as a variety of high frequency contact patterns. The mechanism is described by three independent rigid bodies with masses m_1, m_2, m_3 and inertia Θ_{S_1}, Θ_{S_2}, Θ_{S_3} with respect to their centers of mass S_1, S_2, S_3. Each body is subjected to gravitational forces. In accordance with the notation introduced for the rotating bouncing ball example in Section 7.1, the generalized positions $\boldsymbol{q} = (x_1,\ y_1,\ \varphi_1,\ x_2,\ y_2,\ \varphi_2,\ x_3,\ y_3,\ \varphi_3)$ and generalized velocities $\dot{\boldsymbol{q}} = \boldsymbol{u} = (u_{x_1},\ u_{y_1},\ u_{\varphi_1},\ u_{x_2},\ u_{y_2},\ u_{\varphi_2},\ u_{x_3},\ u_{y_3},\ u_{\varphi_3})$ are introduced. The tuples ${}_I\vec{\boldsymbol{r}}_{OS_i} = (x_i,\ y_i,\ 0)$, $i = 1, 2, 3$, address the center of masses S_i with respect to the inertial basis I and φ_i, $i = 1, 2, 3$, describe the bodies orientations with respect to the $\vec{\boldsymbol{e}}_x^I$-axis. Consequently, the ingredients for the equality of measures (3.2) are given in terms of

$$\boldsymbol{M} = \operatorname{diag}(m_1,\ m_1,\ \theta_{S_1},\ m_2,\ m_2,\ \theta_{S_2},\ m_3,\ m_3,\ \theta_{S_3}) \tag{7.17}$$

and

$$\boldsymbol{h} = (0,\ -m_1 g,\ 0,\ 0,\ -m_2 g,\ 0,\ 0,\ -m_3 g,\ 0)\,. \tag{7.18}$$

The individual rigid bodies are interconnected by the bilateral constraints

$$\begin{aligned} \boldsymbol{g}_1 &= \begin{pmatrix} x_1 - \frac{l_1}{2}\cos\varphi_1 \\ y_1 - \frac{l_1}{2}\sin\varphi_1 \end{pmatrix}, \quad \boldsymbol{g}_2 = \begin{pmatrix} x_1 + \frac{l_1}{2}\cos\varphi_1 \\ y_1 + \frac{l_1}{2}\sin\varphi_1 \end{pmatrix} - \begin{pmatrix} x_2 - \frac{l_2}{2}\cos\varphi_2 \\ y_2 - \frac{l_2}{2}\sin\varphi_2 \end{pmatrix} \\ \boldsymbol{g}_3 &= \begin{pmatrix} x_2 + \frac{l_2}{2}\cos\varphi_2 \\ y_2 + \frac{l_2}{2}\sin\varphi_2 \end{pmatrix} - \begin{pmatrix} x_3 \\ y_3 \end{pmatrix}. \end{aligned} \tag{7.19}$$

Each vertex of the slider is subjected to frictional contact, described by the contact distances

$$\begin{aligned} g_{N_1} &= \frac{d}{2} - (y_3 - a\sin\varphi_3 + b\cos\varphi_3) \\ g_{N_2} &= \frac{d}{2} - (y_3 + a\sin\varphi_3 + b\cos\varphi_3) \\ g_{N_3} &= \frac{d}{2} + (y_3 - a\sin\varphi_3 - b\cos\varphi_3) \\ g_{N_4} &= \frac{d}{2} + (y_3 + a\sin\varphi_3 - b\cos\varphi_3) \end{aligned} \tag{7.20}$$

Table 7.1: Geometry and inertia properties of the slider-crank mechanism.

l_1	l_2	$a = 2b$	d	g
0.153	0.306	0.05	0.052	9.81
θ_{S_1}	θ_{S_2}	θ_{S_3}	$m_1 = m_2$	m_3
$7.4 \cdot 10^{-5}$	$5.9 \cdot 10^{-4}$	$2.7 \cdot 10^{-6}$	0.038	0.076

and the relative tangent velocities

$$\begin{aligned} \gamma_{T_1} &= u_{x_3} + u_{\varphi_3}(a \sin \varphi_3 - b \cos \varphi_3) \\ \gamma_{T_2} &= u_{x_3} - u_{\varphi_3}(a \sin \varphi_3 + b \cos \varphi_3) \\ \gamma_{T_3} &= u_{x_3} + u_{\varphi_3}(a \sin \varphi_3 + b \cos \varphi_3) \\ \gamma_{T_4} &= u_{x_3} - u_{\varphi_3}(a \sin \varphi_3 - b \cos \varphi_3)\,, \end{aligned} \tag{7.21}$$

together with $e_N = 0.4$, $e_F = 0$ and $\mu_T = 0.01$. Constraint derivatives and generalized force directions for both unilateral and bilateral constraints are obtained by differentiating (7.19) to (7.21) with respect to time t and generalized velocities $\boldsymbol{u}$, respectively.

The geometry and inertia properties of the mechanism are listed in Table 7.1. The initial conditions are chosen as $t_0 = 0$, $\boldsymbol{q}_0 = \boldsymbol{q}(t_0) = (0.0765,\ 0,\ 0,\ 0.306,\ 0,\ 0,\ 0.459,\ 0,\ 0.017)$ and $\boldsymbol{u}_0 = \boldsymbol{u}(t_0) = (0,\ 11.475,\ 150,\ 0,\ 11.475,\ -75,\ 0,\ 0,\ 0)$. In order to get high-resolution results, the simulation is performed using a constant step-size $h = 10^{-4}$, although a ten times larger value could solve the problem without any convergence issues. The estimated proximal point parameters are modified with $\alpha = 0.5$ in order to guarantee convergence.

Since in [29] and [42], the system was already computed with the nonsmooth RATTLE algorithm and the family of Lobatto IIIA-IIIB methods, it is subsequently investigated with the two-stage Radau IIA method as a representative of the family of nonsmooth projected stiffly accurate Runge–Kutta methods. Exemplary simulation results are depicted in Figure 7.9. Since the simulation started with a small perturbation of the sliders orientation ($\varphi_3 \approx 1°$), it is apparent from Figure 7.9(a), that the slider's orientation is stabilized after a small time lapse ($t \approx 0.01$). Afterwards, $\varphi_3 = u_{\varphi_3} \approx 0$ and the sliders center of mass moves according to Figure 7.9(b) without any rotation. The time evolution of the velocities u_{φ_1} and u_{φ_2} is shown in Figure 7.9(c). Neglecting dissipation, they show a periodic solution. An exemplary phase portrait of $(\varphi_2, u_{\varphi_2})$ is given in Figure 7.9(d). Moreover, Figure 7.9(e) and (f) show the evolution of two exemplary contact distances and their time derivatives. As enforced by

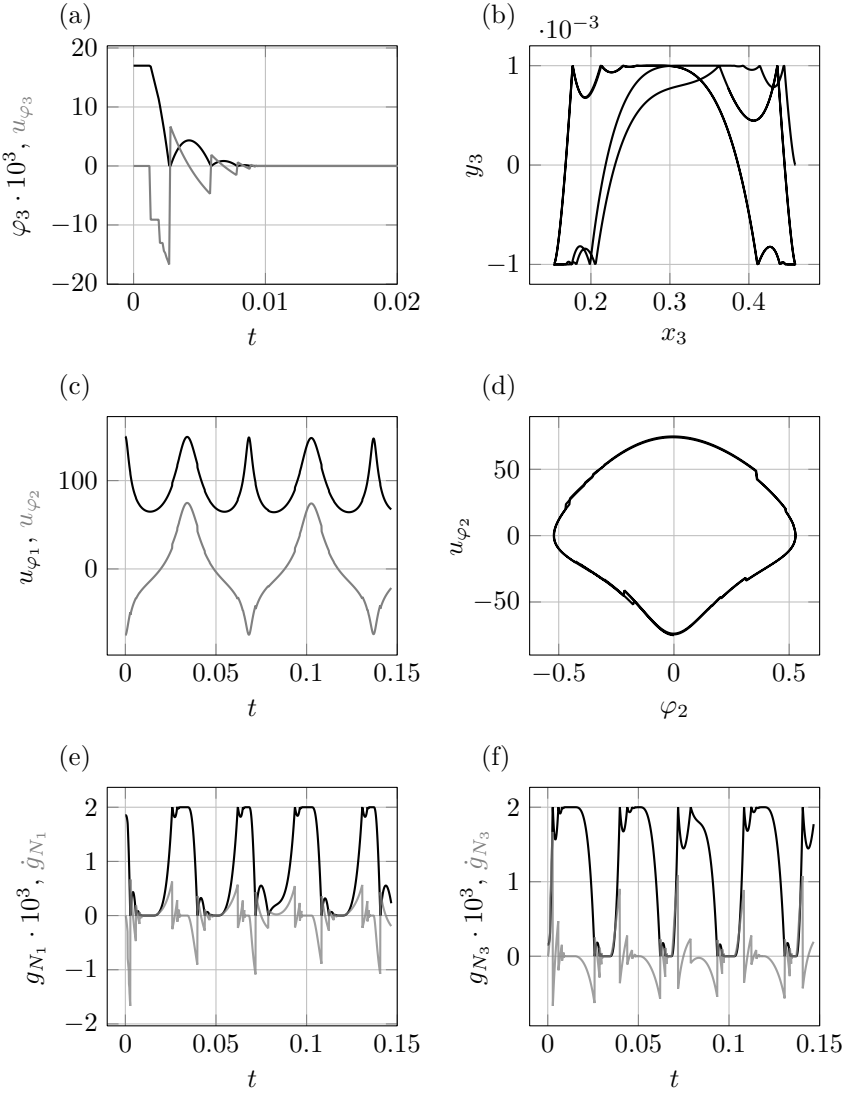

Figure 7.9: Exemplary results of the slider-crank mechanism solved with the nonsmooth two-stage Radau IIA method.

the method, no penetration is present and whenever a contact is closed, the corresponding contact velocity satisfies a generalized Newton's impact law. The results depicted in Figure 7.9 are in line with those obtained by [29] and [42] using the nonsmooth RATTLE algorithm and the family of Lobatto IIIA-IIIB methods. Due to the huge number of impact events, this example is only weakly influenced by the numerical integration method. Consequently, besides the numerical methods that do not enforce the position level constraints, all other presented methods are expected to be good choices for this type of problems. For Moreau's mid-point rule and the shifted Moreau method, drift in the bilateral constraints as well as contact penetration can be expected, depending on the chosen step-size.

7.7 Point Mass on Slope and Nonsmooth Convergence

Consider a point mass (mass $m = 1$, gravitational acceleration $g = 10$) falling and sliding on a slope described by the exponential function. The position of the mass is described by the Cartesian coordinates $\boldsymbol{q} = (x,\ y)$ with corresponding velocities $\boldsymbol{u} = (u_x, u_y)$. The curve $\boldsymbol{f}\colon \mathbb{R} \to \mathbb{R}^2$, defining the slope, and its first derivative are given by

$$\boldsymbol{f}(x) = \big(x,\ \exp(-x)\big)\,, \quad \boldsymbol{f}'(x) = \big(1,\ -\exp(-x)\big)\,, \tag{7.22}$$

as shown in Figure 7.10. The relative tangent velocity γ_T used to describe set-valued Coulomb friction between the point mass and the slope is computed using the tangent $\boldsymbol{t}$, i.e.,

$$\gamma_T = \boldsymbol{t}(x)^{\mathrm{T}}\boldsymbol{u} \quad \text{with} \quad \boldsymbol{t}(x) = \frac{\boldsymbol{f}'(x)}{\|\boldsymbol{f}'(x)\|}\,. \tag{7.23}$$

Using the outward pointing normal $\boldsymbol{n}$ of the slope, the normal contact distance is

$$g_N = \boldsymbol{n}(x)^{\mathrm{T}}\big(\boldsymbol{q} - \boldsymbol{f}(x)\big) \quad \text{with} \quad \boldsymbol{n}(x) = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \boldsymbol{t}(x)\,. \tag{7.24}$$

Inelastic impacts are assumed with $e_N = e_F = 0$ and the friction coefficient is given as $\mu_T = 0.3$. Depending on the chosen initial conditions, two cases are investigated, see Figure 7.10.

case 1: The point mass starts from rest and lies on the slope, i.e., it starts with a closed contact. With $t_0 = 0$, the initial conditions are chosen as $\boldsymbol{q}(t_0) = \boldsymbol{f}(0) = (0,\ 1)$ and $\boldsymbol{u}(t_0) = (0,\ 0)$. For this setup and the chosen friction coefficient, the point mass slides downhill and comes to rest at $t \approx 2.1$ due to friction.

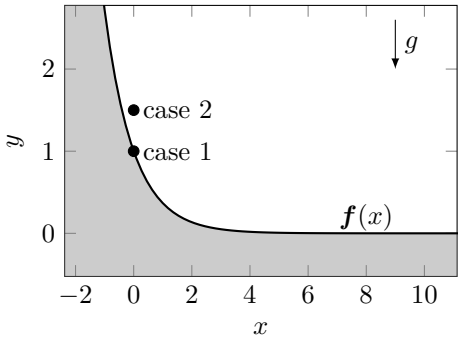

Figure 7.10: Sketch of the initial conditions for the point mass (falling and) sliding on a slope.

case 2: The point mass starts at rest with $\boldsymbol{q}(t_0) = (0,\ 1.5)$, i.e., it starts from a resting position situated above the slope. Hence, the point mass will fall down and eventually come into contact with the slope, where an inelastic impact occurs. Subsequently, the point mass starts sliding on the slope until coming to rest.

The trajectories for both cases are depicted in Figure 7.11 and were computed using the presented nonsmooth two-stage Radau IIA method with a constant step-size $h = 10^{-2}$. It can be seen that in case 1 the velocity of the point mass is continuous, whereas in case 2 there is a velocity jump at $t \approx 0.3$ due to the impact. Between $t = 2$ and $t = 2.5$, the motions of both cases show a kink in the velocity. At that time instant, the velocity is not differentiable. This behavior is the result of a slip-stick transition, which causes the acceleration to jump.

In analogy to Example 4.11 and 5.3, this section investigates the convergence of the presented numerical methods numerically. For that, the cases 1 and 2 are simulated with different step sizes $h \in \{8 \cdot 10^{-4},\ 1.6 \cdot 10^{-3},\ 3.2 \cdot 10^{-3},\ 6.4 \cdot 10^{-3},\ 1.28 \cdot 10^{-2},\ 2.56 \cdot 10^{-2},\ 5.12 \cdot 10^{-2},\ 1.024 \cdot 10^{-1}\}$ and the convergence behavior of the solutions with respect to an accurate numerical solution computed with $h^{\mathrm{ref}} = 2 \cdot 10^{-4}$ is studied. Thereto, the simulation error defined in (4.56) is utilized again. In order to guarantee very tight tolerances, the semi-smooth Newton method proposed in Section 6.2 is applied to solve each individual stage of the proposed nonsmooth time integration methods. Thereto, the relative and absolute tolerances $rtol = atol = 10^{-14}$ were used.

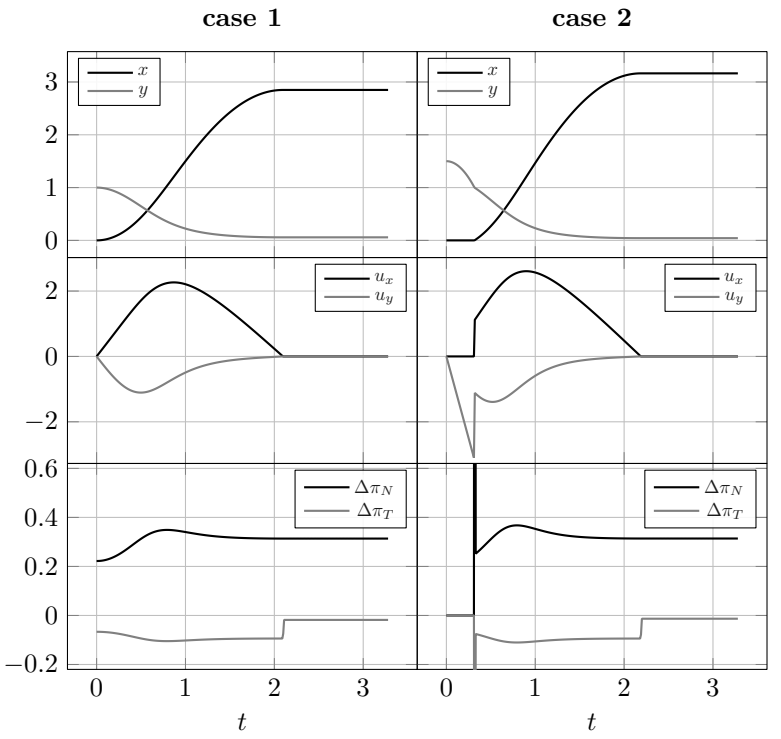

Figure 7.11: Time evolution of positions, velocities and percussion differences for the point mass on slope example solved with the presented nonsmooth two-stage Radau IIA method.

The convergence behavior of the discussed nonsmooth projected and partitioned Runge–Kutta methods is visualized in Figure 7.12. In both cases, the time-span $t \in [0, 0.8192]$ is analyzed only, i.e., the slip-stick transition is excluded from the convergence study. With this choice, the motion in case 1 is smooth and corresponds to the motion of a point mass, which moves along the slope and is subjected to sliding friction described by a single-valued force law.

For case 1, the outstanding convergence rates $2s - 1$ of the Radau IIA methods known from bilaterally constrained systems [98] are observed in Figure 7.12. This is no surprise, as for case 1, the contact is always closed and the discrete contact and friction laws are consistent with the discretization of bilateral constraints, i.e., the contact and friction laws

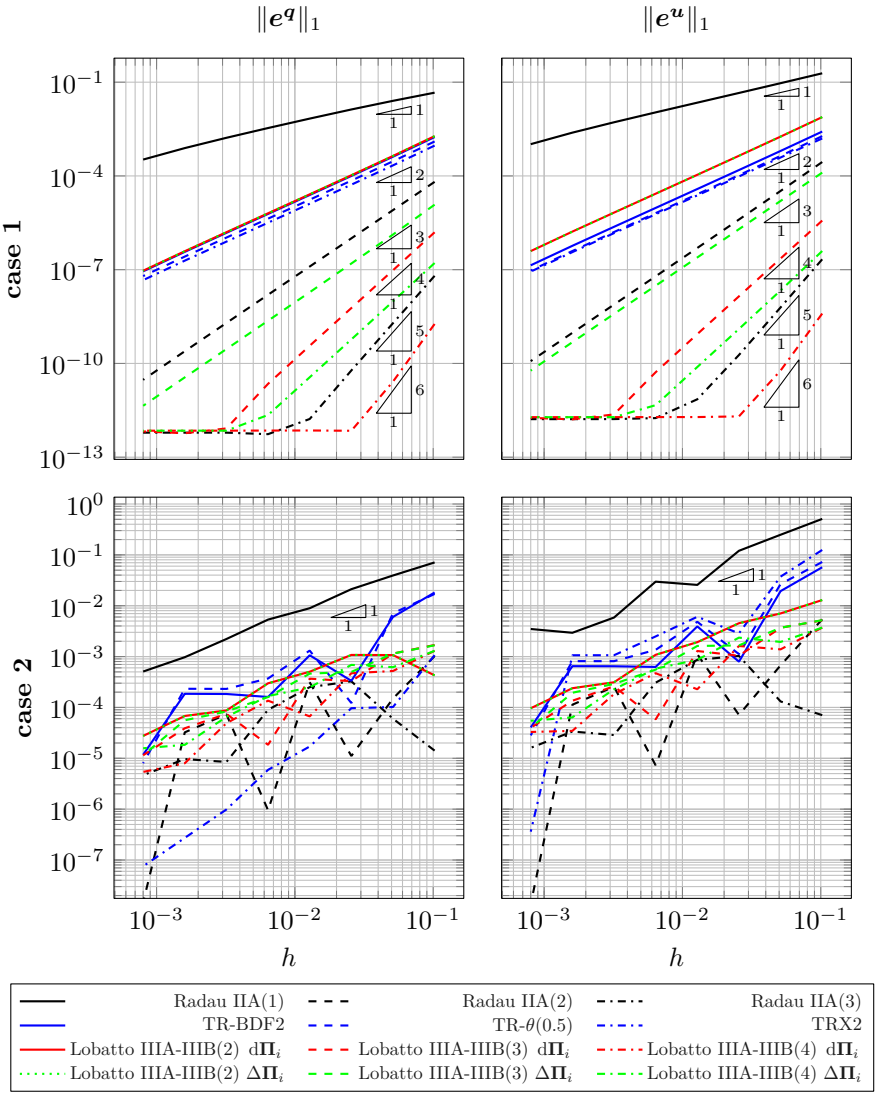

Figure 7.12: Convergence of selected nonsmooth projected and partitioned Runge–Kutta methods applied to the point mass on slope problem.

reduce to the discretization of bilateral constraints. This showcases the importance of discretizing the normal contact laws as well as the friction laws consistently with the bilateral constraints. Otherwise, the good convergence behavior could be lost.

The very same observations can be made for the nonsmooth partitioned Runge–Kutta methods. For $s \geq 2$, the nonsmooth Lobatto IIIA-IIIB methods indicate a rate of convergence of $2s-2$ as known from literature [99, 158], which is in line with the findings of [42]. When the discrete contact and friction laws of the proposed nonsmooth partitioned Runge–Kutta methods are enforced using the integrated percussion differences $\Delta\Pi_{N,i}$, $\Delta\boldsymbol{\Pi}_{F,i}$ with $i = 1, \ldots, s-1$ instead of the scaled discrete percussion measures $\mathrm{d}\Pi_{N,i}$ and $\mathrm{d}\boldsymbol{\Pi}_{F,i}$, the Lobatto IIIA-IIIB methods show an order reduction, see Figure 7.12. Consequently, the presented nonsmooth partitioned Runge–Kutta methods should be preferred compared to an alternative formulation using the integrated percussion differences.

For case 2, Figure 7.12 shows a convergence behavior that is not monotonous and appears to be of first-order. This convergence behavior is in line with the results for other event-capturing schemes [31, 65, 170]. As all event-capturing schemes, the presented methods smear the impact law over a whole time interval and use the velocity at the beginning and the end of the time-step as pre- and post-impact velocities, respectively. It is clear that this kind of discretization becomes exact for $h \to 0$. Hence, as observed, the presented methods are convergent. The apparent first-order convergence might be explained by the fact that the just mentioned smearing of the impact law resolves the exact impact time only to the order of the time-step h.

7.8 Painlevé Rod and the Frictional Catastrophe

It is well known that an episode of smooth motion with persistent frictional contact may end with a velocity jump without any collision occurring [128] and is therefore called *frictional catastrophe*. The most prominent example that exhibits such a behavior is called the Painlevé paradox, dedicated to Paul Painlevé [137]. Following the investigations of [70, 71, 116] this problem will be utilized to investigate how the proposed numerical methods can cope with the above described phenomenon.

Consider a rigid homogeneous slender rod of mass $m = 1$, length $2l$ and rotational inertia $\Theta_S = \frac{1}{3}ml^2$, where $l = 1$. As shown in Figure 7.13, the rod moves in the $\vec{\boldsymbol{e}}_x^I$-$\vec{\boldsymbol{e}}_y^I$-plane and is under the influence of gravity with gravitational acceleration $g = 10$. Describing the orientation of the

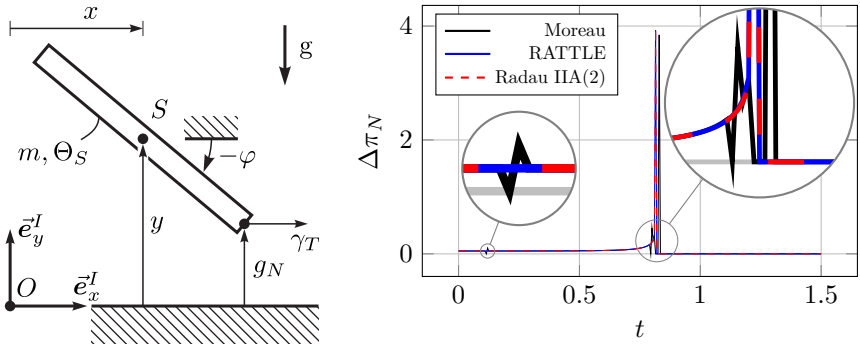

Figure 7.13: Sketch of the Painlevé rod system (left) and time evolution of the percussion difference $\Delta\pi_N$ (right).

rod by the absolute angle φ with respect to the $\vec{e}_x^I$-axis, the minimal coordinates $\boldsymbol{q} = (x,\ y,\ \varphi)$ are chosen, where the center of mass S of the rod is addressed by ${}_I\vec{r}_{OS} = (x,\ y,\ 0)$. Using the natural velocity coordinates $\boldsymbol{u}$ corresponding to $\dot{\boldsymbol{q}}$ almost everywhere, the mass matrix and the force tuple of the system have the form (7.1). The frictional contact of the rod's tip with the ground is described in terms of

$$g_N = y + l\sin\varphi \quad \text{and} \quad \gamma_T = u_x - u_\varphi l\sin\varphi\,. \tag{7.25}$$

The friction coefficient is set to $\mu_T = 5/3$ and $e_N = e_F = 0$ is used to model inelastic impact. With the initial conditions $t_0 = 0$, $\boldsymbol{q}(t_0) = (0,\ l\sin(\varphi_0),\ \varphi_0)$ and $\boldsymbol{u}(t_0) = (v,\ 0,\ 0)$, the rod's tip is initially in contact with an initial inclination of $\varphi_0 = -31°$. Moreover, it slides with initial horizontal velocity $v = 30$. As analyzed in [71], this sliding motion eventually results in detaching of the contact. This detaching comes with a finite time blowup of the accelerations and nonimpulsive contact forces, which is seen as an impact. The blowup of the accelerations leads to vertical asymptotes in the velocities, which constitute problems for some integration methods. However, since the asymptote is just an isolated singularity, event-capturing schemes with constant time-step might overcome these blowups, as they do not try to fully resolve it.

The system is simulated with a time-step $h = 5\cdot 10^{-3}$ and the estimated proximal point parameters are scaled by $\alpha = 0.5$. For both, fixed-point iterations and Newton iterations, relative and absolute tolerances of 10^{-8} are used. The simulation result in Figure 7.13 shows that the presented two-stage nonsmooth Radau IIA method as well as the two-stage nonsmooth Lobatto IIIA-IIB (RATTLE) method can indeed deal with the Painlevé

paradox. Moreover, the methods are compared with Moreau's mid-point rule [74, 126]. As indicated by the first magnification in Figure 7.13, after a short period of sliding, Moreau's mid-point rule performs an artificial impact. This can be explained by the fact that the method numerically allows unilateral constraint drift. Since such a drift is prohibited by the proposed nonsmooth Runge–Kutta methods, they do not show such a nonphysical behavior. More dramatically, the second magnifications makes apparent that Moreau's mid-point rule performs a sequence of impacts until the rod takes off the ground. In contrast to that, the nonsmooth projected and partitioned Runge–Kutta methods exactly resolve the asymptotic behavior of the normal contact force and result in a smooth transition of the contact percussion until a certain limit is taken that leads to the take-off.

7.9 Ball in Corner and Redundant Contacts

The following example demonstrates the capability of the proposed solution algorithms from the previous chapter to cope with redundant contacts. Thereto, the very simple example of a ball coming into contact with two inclined planes, proposed in [39, 41], is considered. As already discussed, the semi-smooth Newton method, proposed in Section 6.2, is not able to solve such a problem due to the singularity of the involved generalized gradient. As an alternative, Section 6.3 introduced the iterative projection method based on simple fixed-point iterations. Consequently, only the regular part of the system is solved by a (simplified) Newton method which leads to an algorithm suited to deal with redundant contacts.

Consider once again the planar dynamics of the ball described in Section 7.1. The two contact planes are assume to be inclined by the angles $\alpha = 45°$ and $\beta = 45°$, respectively, as depicted in Figure 7.14. The resulting contact distances are gathered in

$$\boldsymbol{g}_N = \begin{pmatrix} -x\sin\alpha + y\cos\alpha - R \\ x\sin\beta + y\cos\beta - R \end{pmatrix} . \tag{7.26}$$

Friction is described by the relative tangent velocities

$$\gamma_{T,1} = u_x\cos\alpha + u_y\sin\alpha + ru_\varphi\,, \quad \gamma_{T,2} = u_x\cos\beta - u_y\sin\beta + ru_\varphi\,, \tag{7.27}$$

together with the friction coefficients $\mu_T^1 = \mu_T^2 = 0.3$. The first plane realizes an elastic impact with restitution coefficient $e_N^1 = 0.5$, while the second one performs inelastic impacts with $e_N^2 = 0$. Tangential impacts are assumed to be inelastic with $e_F^1 = e_F^2 = 0$. Starting at rest with $t_0 = 0$

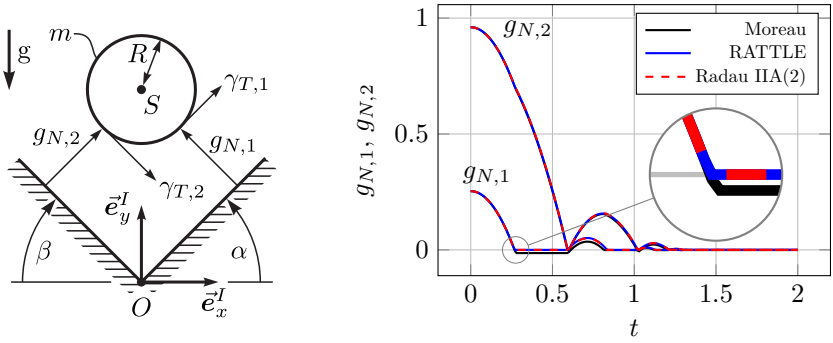

Figure 7.14: Sketch of the ball in corner system (left) and simulated time evolution of the gaps (right).

and $\boldsymbol{q}(t_0) = (-0.5,\ 1,\ 0)$, the ball will eventually come to rest with both contacts closed. Using a constant step-size $h = 10^{-2}$, the simulation is performed using the relative and absolute tolerances 10^{-8} for both fixed-point and Newton's method. The estimated proximal point parameters are scaled by $\alpha = 0.5$. As depicted in Figure 7.14, both families of the proposed nonsmooth time integration methods, together with the simple fixed-point iteration method, can solve the problem. Consequently, the proposed solution method is capable to deal with redundant contacts. As indicated by the magnification in Figure 7.14, the solution obtained by Moreau's mid-point rule [74, 126] shows contact penetration, which is prohibited by both alternative methods.

7.10 The Tippedisk

The tippedisk - a tippetop without rotational symmetry is an important mechanical-mathematical archetype for friction-induced instability phenomena [154, 155]. Furthermore, it serves as an excellent benchmark example for the validation of nonsmooth time integration methods in the presence of highly dynamic finite rotations. Essentially, the tippedisk is an eccentric disk of mass $m = 0.435$, whose center of gravity S does not coincide with the geometric center G of the disk, see Figure 7.15. One way to construct such a tippedisk is to take a homogeneous disk of radius $r = 4.5 \cdot 10^{-2}$ and remove mass at the distance $b = 2 \cdot 10^{-2}$ from the geometric center by drilling a hole of radius $a = 1.5 \cdot 10^{-2}$. The eccentricity e then follows as $e = ba^2/(r^2 - a^2)$. Consequently, this choice of

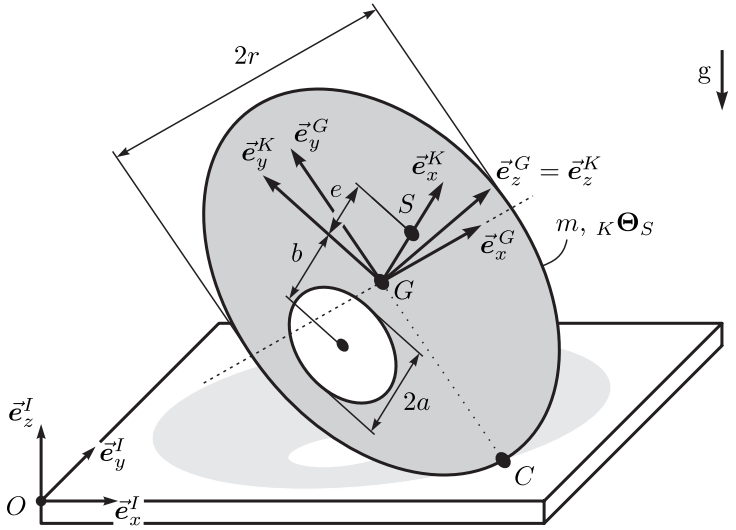

Figure 7.15: Sketch of the Tippedisk system.

parameters results in the inertia matrix in the body-fixed K-basis given by ${}_K\boldsymbol{\Theta}_S = \mathrm{diag}(2.49 \cdot 10^{-4},\ 2.2972 \cdot 10^{-4},\ 4.7072 \cdot 10^{-4})$. If the disk is placed on a flat support, like a table, and in the presence of gravity with acceleration $g = 9.81$, it is quite obvious that there are two stationary spinning solutions, where the gravitational force and the normal contact force balance each other. Namely, a solution where the center of gravity S is vertically below the geometric center G and one where S is vertically above G. The former one is subsequently referred as *non-inverted* solution, while the latter one is called *inverted* solution. If the non-inverted tippedisk is spun fast around an in-plane axis, the center of gravity rises until the disk reaches an inverted configuration. This process is called the *inversion phenomenon*. However, due to dissipation, the spinning velocity decreases slowly over time.

The position of the disk is described by the components ${}_I\vec{\boldsymbol{r}}_{OS} \in \mathbb{R}^3$ of the position vector of S with respect to the inertial basis I. To characterize the orientation of the disk, the body-fixed K-basis is introduced such that $\vec{\boldsymbol{e}}_z^{\,K}$ is the normal with respect to the face of the disk. The transformation matrix $\boldsymbol{A}_{IK} = ({}_I\vec{\boldsymbol{e}}_x^{\,K}\ {}_I\vec{\boldsymbol{e}}_y^{\,K}\ {}_I\vec{\boldsymbol{e}}_z^{\,K})$ is parametrized using a unit quaternion $\boldsymbol{p} \in \mathbb{R}^4$. Hence, the configuration of the disk is described by $\boldsymbol{q} = ({}_I\vec{\boldsymbol{r}}_{OS},\ \boldsymbol{p}) \in \mathbb{R}^7$. The generalized velocities $\boldsymbol{u} = ({}_I\vec{\boldsymbol{v}}_S,\ {}_K\vec{\boldsymbol{\omega}}_{IK}) \in \mathbb{R}^6$ are composed of the representations of the velocity $\vec{\boldsymbol{v}}_S \in \mathbb{E}^3$ of the center of gravity S and

the angular velocity $\vec{\boldsymbol{\omega}}_{IK} \in \mathbb{E}^3$ of the K-basis with respect to the I and K-basis, respectively. As introduced in Section 3.1, this choice leads to a mechanical system with the generalized kinematic equation (3.1). The relevant quantities $\boldsymbol{B}$, $\boldsymbol{\beta}$, $\boldsymbol{M}$, and $\boldsymbol{h}$, describing such a parametrized rigid body under the influence of gravity, are given in model 4 of [155].

The contact distance between the disk and the flat support is described by

$$g_N = \vec{\boldsymbol{r}}_{OC} \cdot \vec{\boldsymbol{e}}_z^I = (\vec{\boldsymbol{r}}_{OS} + \vec{\boldsymbol{r}}_{SC}) \cdot \vec{\boldsymbol{e}}_z^I \,. \tag{7.28}$$

Following [155], a grinding G-basis can be introduced as

$$\vec{\boldsymbol{e}}_z^G = \vec{\boldsymbol{e}}_z^K \,, \quad \vec{\boldsymbol{e}}_x^G = \frac{\vec{\boldsymbol{e}}_z^I \times \vec{\boldsymbol{e}}_z^G}{\|\vec{\boldsymbol{e}}_z^I \times \vec{\boldsymbol{e}}_z^G\|} \,, \quad \vec{\boldsymbol{e}}_y^G = \vec{\boldsymbol{e}}_z^G \times \vec{\boldsymbol{e}}_x^G \,, \tag{7.29}$$

from which follows $\vec{\boldsymbol{r}}_{SC} = -e\vec{\boldsymbol{e}}_x^K - r\vec{\boldsymbol{e}}_y^G$, see Figure 7.15. As discussed in [155], simple set-valued Coulomb friction is not sufficient to describe the experimentally observed inversion phenomenon. Hence, an approximation of set-valued Coulomb–Contensou friction is taken into account with $\mu = 0.3$ and by using the set transformations introduced in [126] with the radius of assumed circular contact area $R = 10^{-3}$ and $\boldsymbol{a} = (1, 1, \frac{3}{16}\pi R)$. Thereto, the contact velocities

$$\boldsymbol{\gamma}_F = \begin{pmatrix} \vec{\boldsymbol{v}}_C \cdot \vec{\boldsymbol{e}}_x^I \\ \vec{\boldsymbol{v}}_C \cdot \vec{\boldsymbol{e}}_y^I \\ \frac{3\pi R}{16} \vec{\boldsymbol{\omega}}_{IK} \cdot \vec{\boldsymbol{e}}_z^I \end{pmatrix} \,, \tag{7.30}$$

are used, where $\vec{\boldsymbol{v}}_C = \vec{\boldsymbol{v}}_S + \vec{\boldsymbol{\omega}}_{IK} \times \vec{\boldsymbol{r}}_{SC}$ denotes the velocity of the contact point. A closer look at the contact velocities (7.30) reveals that the first two components correspond to the relative tangent velocities and capture the effects of set-valued isotropic Coulomb friction. Moreover, the third component is a representative radial contact velocity, which accounts for drilling friction.

The initial conditions are taken as $t_0 = 0$, ${}_I\vec{\boldsymbol{r}}_{OS}(t_0) = (0,\ 0,\ r - e)$, $\boldsymbol{p}(t_0) = (0.5,\ 0.5,\ 0.5,\ -0.5)$, ${}_K\vec{\boldsymbol{\omega}}_{IK}(t_0) = (-60,\ 0,\ 1)$ and ${}_I\vec{\boldsymbol{v}}_S(t_0) = \big[\boldsymbol{A}_{IK}(\boldsymbol{p}(t_0)){}_K\vec{\boldsymbol{\omega}}_{IK}(t_0)\big] \times {}_I\vec{\boldsymbol{r}}_{OS}(t_0)$. The subsequent simulations are performed using a constant step-size $h = 10^{-3}$. For comparison, the system is also solved with Moreau's mid-point rule [74, 126] and the nonsmooth generalized-alpha method of [41]. Since the former one is a first-order method that does not prevent contact penetration, a very small step-size of $h = 10^{-5}$ has to be chosen in order to accurately represent the observed inversion phenomenon. For the second-order method of [41] a constant step-size $h = 10^{-3}$ together with $\rho_\infty = 0.9$ is applied. The same step size

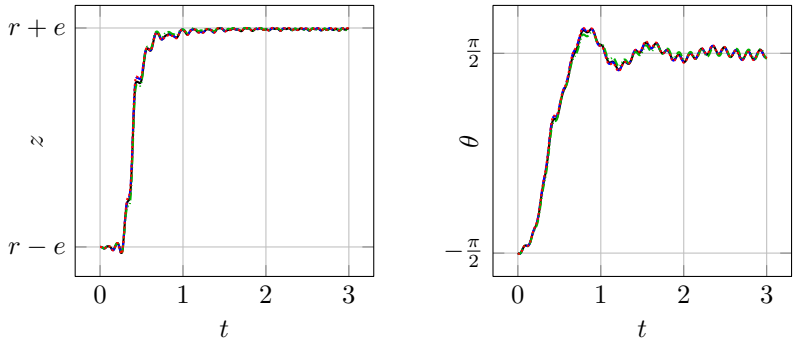

Figure 7.16: Simulation results of the tippedisk: Simulated time evolution of the $\vec{e}_z^I$-component of the center of gravity (left) and inclination angle θ corresponding to the last Euler angle with sequence "zxz" (right). (——): Moreau's mid-point rule [126], (- - - -): Nonsmooth generalized-alpha method [41], (·····): RATTLE and (-·-·-): two-stage Radau IIA.

is chosen for both the two-stage Lobatto IIIA-IIIB method (RATTLE) and the two-stage Radau IIA method, which represent the families of nonsmooth Runge-Kutta methods presented herein. Although the implementation of these methods is less optimized compared to Moreau's midpoint rule and partially relies on finite difference approximations of the involved Jacobians, they solve the problem 12 and 9 times faster, respectively.

It is apparent from Figure 7.16 that the simulation results obtained by the presented time integration methods are in line with the results obtained with the method of [126], although the applied step-size is a hundred times larger. This shows that the presented methods are well suited for mechanical systems with spatial friction as well as for systems with a generalized kinematic differential equation (3.1). Since a significantly larger step-size could be used, the number of required operations was reduced, resulting in substantial cost savings for the overall computation. Moreover, the results are in line with those computed with the method presented in [41]. Hence, for persistent frictional contact, the higher-order convergence of the presented methods as well as the absence of penetration is of crucial importance for the efficient solution of highly dynamical problems like the tippedisk.

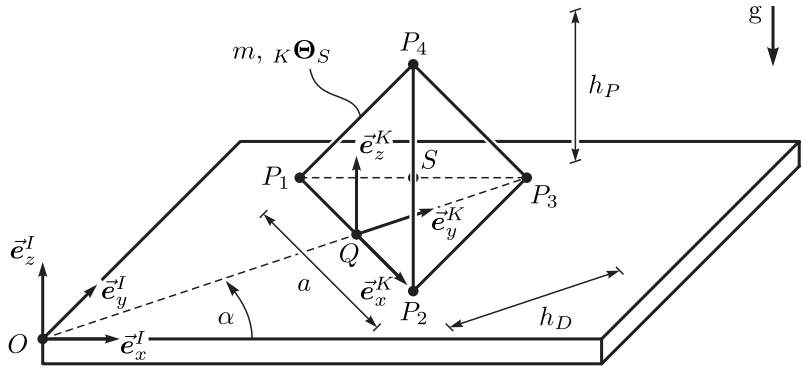

Figure 7.17: Sketch of the tetrahedron system.

7.11 Anisotropic Friction of Tetrahedron on Plane

This final example investigates once more redundant[3] frictional contacts, deals with anisotropic friction and requires again a non-trivial kinematic differential equation as introduced in (3.1). Moreover, it will be demonstrated that anisotropic Coulomb friction with elliptical force reservoir cannot adequately be approximated by a rectangular force reservoir. This can be explained by the fact that the rectangular for reservoir can be interpreted as the most simple polyhedral approximation proposed in [167]. Such an approximation is way to simple to adequately describe the elliptical force reservoir and introduces an approximation bias where frictional effects are stronger in some directions than others [120]. Even in the case of isotropic friction coefficients the friction forces are overestimated since the admissible force reservoir is not a circle but a square. Even worse, the rectangular force reservoir introduces wrong frictional force directions. Similar results are reported in [94] within a slightly different experiment.

Let the tetrahedron in Figure 7.17 be defined by the edges $a = 1$ and an isotropic density $\rho = 1$. Consequently, it has the mass $m = \rho e^3\sqrt{2}/12$ and the inertia matrix ${}_K\boldsymbol{\Theta}_S = \mathbf{1}_{3\times3}e^2/20$ with respect to the body-fixed K-basis. Once more, such a rigid body can be described by the identical components $\boldsymbol{B}$, $\boldsymbol{\beta}$, $\boldsymbol{M}$, and $\boldsymbol{h}$, already used for the previous example. The description of frictional contact is more involved since all four corner

[3]When all three corners of some tetrahedron surface come into and remain in contact with the floor, the system reduces to a planar problem with three independent position degrees. During stick, six frictional contact forces have to be determined, clearly leading to redundant constraints.

points of the tetrahedron can come into contact with the flat ground. Let $h_D = \frac{\sqrt{3}}{2}e$ and $h_P = \frac{\sqrt{2}}{3}e$ be given as depicted in Figure 7.17. It is convenient to introduce the body-fixed tuple

$$ {}_K\vec{\boldsymbol{r}}_{QS} = \begin{pmatrix} 0 \\ \frac{h_D}{3} \\ \frac{h_P}{4} \end{pmatrix} , \tag{7.31} $$

addressing the center of gravity S with respect to the point Q. For $k = 1, \ldots, 4$, the relative distance of the corner points P_k with respect to Q are described by the tuples

$$ {}_K\vec{\boldsymbol{r}}_{QP_{1,2}} = \begin{pmatrix} \mp\frac{a}{2} \\ 0 \\ 0 \end{pmatrix} , \quad {}_K\vec{\boldsymbol{r}}_{QP_3} = \begin{pmatrix} 0 \\ h_D \\ 0 \end{pmatrix} \quad \text{and} \quad {}_K\vec{\boldsymbol{r}}_{QP_4} = \begin{pmatrix} 0 \\ \frac{h_D}{3} \\ h_P \end{pmatrix} . \tag{7.32} $$

As in the previous example, the position of the tetrahedron is addressed by the components ${}_I\vec{\boldsymbol{r}}_{OS} \in \mathbb{R}^3$ of the position vector of S with respect to the inertial basis I. Hence, the absolute positions of the corner points can conveniently be addressed by

$$ \vec{\boldsymbol{r}}_{OP_k} = \vec{\boldsymbol{r}}_{OS} + \vec{\boldsymbol{r}}_{SP_k} \quad \text{with} \quad \vec{\boldsymbol{r}}_{SP_k} = \vec{\boldsymbol{r}}_{QP_k} - \vec{\boldsymbol{r}}_{QS} . \tag{7.33} $$

Recognizing the contact normal $\vec{\boldsymbol{n}} = \vec{\boldsymbol{e}}_z^I$, the contact distances can be computed by

$$ g_N^k = \vec{\boldsymbol{n}} \cdot \vec{\boldsymbol{r}}_{OP_k} . \tag{7.34} $$

Computing the velocity of the corner points

$$ \vec{\boldsymbol{v}}_{P_k} = \vec{\boldsymbol{v}}_S + \vec{\boldsymbol{\omega}}_{IK} \times \vec{\boldsymbol{r}}_{SP_k} , \tag{7.35} $$

the relative tangent velocities are obtained in a similar manner. Specifically, with the tangents $\vec{\boldsymbol{t}}_1 = \vec{\boldsymbol{e}}_x^I$ and $\vec{\boldsymbol{t}}_2 = \vec{\boldsymbol{e}}_y^I$ they read

$$ \boldsymbol{\gamma}_T^k = \begin{pmatrix} \vec{\boldsymbol{t}}_1 \cdot \vec{\boldsymbol{v}}_{P_k} \\ \vec{\boldsymbol{t}}_2 \cdot \vec{\boldsymbol{v}}_{P_k} \end{pmatrix} . \tag{7.36} $$

Frictional contact is assumed between the corners of the tetrahedron and the floor. Depending on the chosen friction model, two different cases emerge. Both cases are computed with the same initial conditions given by $t_0 = 0$, $\boldsymbol{q}(t_0) = (0,\ 0,\ \frac{e}{2},\ 1,\ 0,\ 0,\ 0)$ and $\boldsymbol{u}(t_0) = v_0(\cos\alpha,\ \sin\alpha,\ 0,\ 0,\ 0,\ 0)$, where $v_0 = 4$ and $\alpha = \frac{\pi}{5}$. That is, the tetrahedron starts above the ground with a constant initial velocity v_0 in the direction defined by

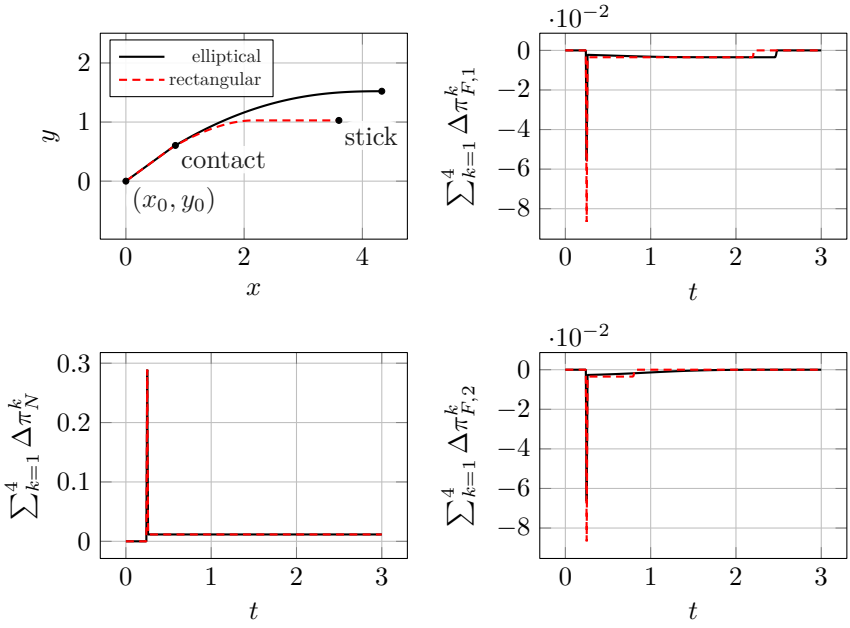

Figure 7.18: Simulation results of the tetrahedron example solved by the nonsmooth two-stage Radau IIA method.

the angle α. The contacts are assumed to be inelastic with restitution coefficients $e_N^k = e_F^k = 0$. For all contacts the same friction coefficients $\mu_{T_1}^k = 0.15$ and $\mu_{T_2}^k = 0.3$ are used. Consequently, the elliptical force reservoir can be transformed to a ball-like set as discussed in Section 3.5 by introducing $\mu_T^k = 0.3$ and $\boldsymbol{a}^k = (\frac{1}{2}, 1)$, which leads to $\boldsymbol{A}^k = \operatorname{diag}(\boldsymbol{a}^k)$ and $\boldsymbol{\gamma}_F^k = (\boldsymbol{A}^k)^{\mathrm{T}} \boldsymbol{\gamma}_T^k$. For the first case, classically set-valued anisotropic Coulomb friction with elliptical force reservoir is used, as described by (3.31) and (3.32). This is in a second case compared to the introduced rectangular force reservoir, see around (3.33).

The resulting systems are computed using the nonsmooth two-stage Radau IIA method with a constant step-size $h = 10^{-2}$. The numerically obtained projected trajectories as well as the resultant percussion differences $\sum_{k=1}^{4} \Delta\pi_N^k$ and $\sum_{k=1}^{4} \Delta\pi_{T,i}^k$, with $i = 1, 2$, are depicted in Figure 7.18. After a short period of free flight, the tetrahedron comes into contact with all three corner points $k = 1, 2, 3$ of the bottom plane at $t \approx 0.26$. Subsequently, a sliding motion is performed until stick. Obviously, the obtained trajectories differ, depending on the used friction model. The

friction model with rectangular force reservoir overestimates the friction forces, although the normal forces of both models coincide. This can be explained by the fact that in contrast to the friction model with elliptical force reservoir its force reservoir is given by a rectangle. This rectangle is the bounding box of the desired ellipse, hence the distance from the boundary to the origin is overestimated in most cases. Consequently, the absolute value of the friction forces as well as the corresponding generalized forces differ. This can be seen in Figure 7.18 and explains the observed differences.

Chapter 8

Conclusions and Outlook

This monograph gives an in-depth discussion on the numerical time integration of mechanical systems with frictional contact and impacts. Specifically, implicit Runge–Kutta methods, known for their excellent capabilities in solving differential algebraic equations, are extended to measure differential inclusions with mechanical structure. Two families of nonsmooth implicit Runge–Kutta methods are introduced:

- nonsmooth projected stiffly accurate Runge–Kutta methods
- nonsmooth partitioned Runge–Kutta methods

The subsequent section summarizes the properties of the proposed nonsmooth Runge–Kutta methods and gives recommendations for the choice of method depending on the specific application. Finally, the chapter closes by giving an outlook on future research topics in the field of numerical methods for nonsmooth mechanical systems.

8.1 Summary and Recommendations

As its name suggests, *nonsmooth projected stiffly accurate Runge–Kutta methods* are particularly effective for solving stiff measure differential inclusions. As the investigations of Chapter 4 have shown, the unique solvability of the resulting discrete equations require the enforcement of the unilateral constraint conditions using integrated percussion differences instead of the discrete percussion measures itself. Moreover, it was shown that solely enforcing the non-penetration conditions is not sufficient for higher-order methods. The resulting methods realize an arbitrary (possibly nonphysical)

impact law, depending on the applied step-size and the chosen initial conditions. In order to circumvent this ambiguity, a subsequent projection stage ensures that the desired impact laws are additionally satisfied over a time-step, even in the case of an inelastic impact. Different representatives of this family are introduced throughout this monograph that can be distinguished by their order of convergence and their possibility to add user-defined numerical dissipation. Due to their outstanding convergence rates, combined with excellent numerical damping characteristics, the proposed nonsmooth Radau IIA methods are a good starting point when no further information of the system under consideration is available. It was demonstrated by the numerical experiments of Section 7.7 that their superior convergence order, observed when the methods are applied to differential algebraic equations, are inherited to the case of nonsmooth mechanical systems with constant contact states. Moreover, it was demonstrated in Section 7.5 that the method is an excellent choice when stiff measure differential inclusions have to be integrated. Alternatively, if user-defined numerical dissipation is of crucial importance, the proposed TR-$\theta(\rho_\infty)$ method should be considered since it is second-order accurate and shares similar damping properties as the generalized-alpha method, well-known for the solution of structural mechanics problems, but can be applied to problems with generalized kinematic relations. In conclusion, Chapter 4 has shown that the family of nonsmooth projected stiffly accurate Runge–Kutta methods merits recognition for effectively addressing all outlined research objectives of this thesis. Therefore, they are an excellent choice for general-purpose numerical time integration methods tailored for nonsmooth mechanical systems with frictional contact and impacts. In addition, the merit of this chapter is the presentation of a novel ESDIRK method, called TR-$\theta(\rho_\infty)$, which allows for optional user-defined numerical dissipation.

The specific merit of Chapter 5 lies in the generalization of the recently proposed Lobatto-type variational integrators [42] to *nonsmooth partitioned Runge–Kutta methods*. Thereby, enabling the time integration of nonsmooth mechanical systems that exhibit a generalized kinematic relation. The family of partitioned Runge–Kutta methods originates from the solution of constrained symplectic mechanical systems with Hamiltonian structure and is well-known for their excellent long-term energy behavior, required for the solution of molecular dynamics simulations or the prediction of orbit motions of the outer solar system [85]. All in common, these methods share some sort of structure preservation that naturally contradicts strong stability requirements. Hence, they should not be considered when stiff differential equations or stiff measure differential inclusions are encountered [80, 99]. The most prominent representatives of this family are

the Lobatto IIIA-IIIB methods. When applied to bilaterally constrained mechanical systems, they simultaneously enforce the constraint conditions on position and velocity level [99]. This turned out to be an important property when the methods are extended to measure differential inclusions with mechanical structure. Specifically, this allows to simultaneously enforce the non-penetration conditions as well as desired, possibly inelastic, impact laws. Thereby, contact penetration and bilateral constraint drift is naturally prohibited by these methods. The unique solvability of the resulting discrete equations requires the introduction of an index shift between the kinematic quantities and the Lagrange multipliers involved in the unilateral constraint conditions. Furthermore, and in contrast to the methods proposed in Chapter 4, an order reduction can be observed when the integrated percussion differences are used to enforce the unilateral constraints. If the discrete percussion measures are chosen instead, this problem is completely eliminated. Once more, it was demonstrated by numerical experiments that their superior convergence orders, observed when the methods are applied to differential algebraic equations, are inherited to the case of nonsmooth mechanical systems with constant contact states. Consequently, this family of methods addresses all outlined research objectives of this thesis, with the exception of **Objective 6**, due to the limited stability properties. Finally, it was shown that a variant of the recently proposed nonsmooth RATTLE algorithm [29] and a variant of Moreau's mid-point rule [40] are naturally embedded into this family of methods, indicating that these methods are natural extensions of the very early nonsmooth time integration methods.

Selecting an appropriate numerical time integration method is a complex and critical task in computational simulations. To aid in this process, Figure 8.1 presents a comprehensive decision tree in the form of a flowchart. This flowchart is derived from the detailed recommendations and guidelines discussed so far, offering a systematic approach to choose the most suitable method for a given problem.

The application of the proposed families of nonsmooth Runge–Kutta methods requires the solution of a system of nonlinear equations subject to normal cone inclusions. For this purpose, two different solution algorithms are presented in Chapter 6, completing the proposed time integration methods. Since both algorithms involve a set of arbitrary positive parameters, the chapter concludes with a simple but sufficient method to estimate these parameters.

Besides the introduction of these two families of nonsmooth Runge–Kutta methods together with appropriate solution algorithms, Chapter 2 and 3 of this monographs give a concise and self-contained introduction to

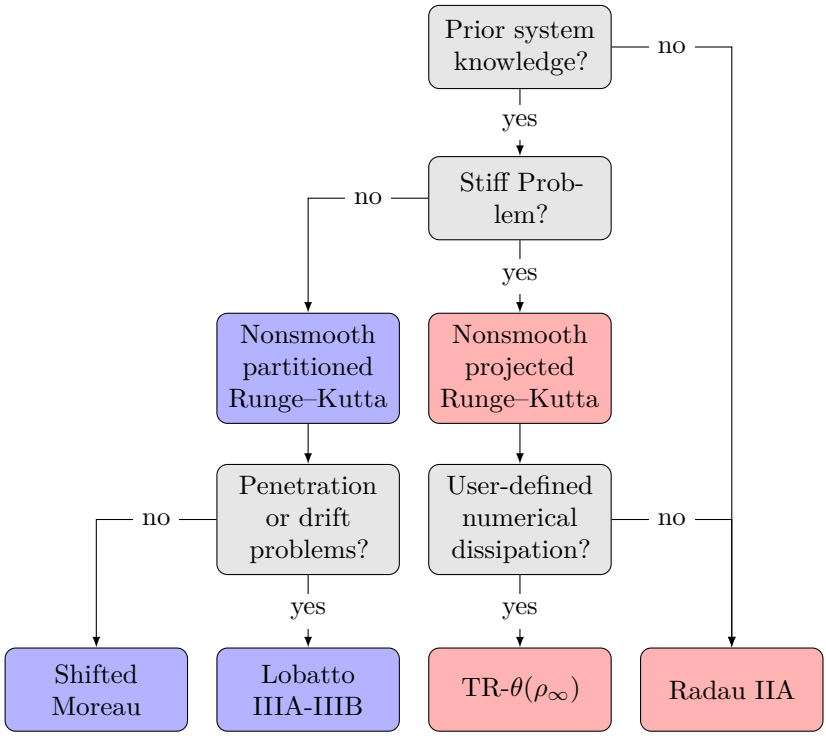

Figure 8.1: Decision tree for choosing an appropriate nonsmooth time integration method.

the mathematical theory of nonsmooth mechanical systems with frictional contact and impacts. Therein, the most commonly encountered friction phenomena are presented, which can be described in a unified way by nonlinear normal cone inclusions. Hence, no polyhedral approximation of the desired friction laws is considered.

This thesis is completed by an in-depth presentation of eleven selected benchmark examples that validate the most important properties of the proposed methods. Moreover, they meet the needs for a common baseline to the validate and compare the performance of the proposed methods with future time integration methods for such class of systems. It can finally be concluded that for all presented examples, the proposed nonsmooth Runge–Kutta methods are at least on par or even superior compared with existing nonsmooth time integration methods, thereby taking into account the formulated research objectives.

8.2 Future Works

Within this monograph it was demonstrated how two prominent families of implicit Runge–Kutta methods can be extended to integrate measure differential inclusions with mechanical structure. Besides these fully implicit Runge–Kutta methods, also half-explicit methods are frequently used for the integration of moderately stiff differential algebraic equations of differentiation index two and three [26, 136]. Preliminary investigations on the application of the index two methods to mechanical systems with frictional contact can be found in [112]. Future research should continue these investigations and evaluate if the proposed methodology of Chapter 4 can be applied to the half-explicit methods designed for the integration of differential algebraic equations of index three proposed in [136].

In the preparation for this monograph, preliminary investigations were made for the extension of linear multi-step methods like BDF to measure differential inclusions with mechanical structure. As already discussed in the introduction, this causes serious problems since the history of positions and velocities is required in order to compute a new time-step. Therefore, a similar approach as outlined for the nonsmooth projected stiffly accurate Runge–Kutta methods from Chapter 4 is not applicable, since the history will not be aware of the change of velocities that is enforced by a discrete version of the impact equation (4.111). A possible remedy for this issue might be the enforcement of the bilateral and unilateral constraints on acceleration level, as proposed in [41]. Thereby splitting the motion of the system into smooth and nonsmooth parts. The former one will be discretized using a linear multi-step method like BDF and the latter by an implicit Euler method, including a possibly inelastic impact law. Since this approach contradicts to **Objective 5** it is not considered herein, but definitely deserve further investigations.

Within this thesis, the proposed nonsmooth Runge–Kutta methods are applied with constant step-sizes h. In the context of differential equations, Runge–Kutta methods are so popular and successfully applied to a variety of different applications precisely due to their very good suitability for adaptive time-step implementations. The gains of such a procedure are concisely described by [144]: *A good ODE integrator should exert some adaptive control over its own progress, making frequent changes in its stepsize h. Usually the purpose of this adaptive stepsize control is to achieve some predetermined accuracy in the solution with minimum computational effort. Many small steps should tiptoe through treacherous terrain, while a few great strides should speed through smooth uninteresting countryside. The resulting gains in efficiency are not mere tens of percents or factors of 2; they can*

sometimes be factors of 10, 100, or more. As the excerpt of [144] indicates, such an adaptive step-size implementation is mandatory to leverage the usability of nonsmooth time integration methods for large scale industrial applications. Since classically, such adaptive implementations are based on monitoring the local truncation error of the numerical methods, a similar quantity has to be found for measure differential inclusions, which is closely related to the subsequent discussion. A first approach in this direction is already presented in [1, 2], where Runge–Kutta methods are combined with Moreau's mid-point rule. Preliminary investigations conducted during the preparation of this monograph indicate that monitoring the error of the $(s-1)$th-order collocation polynomial for the positions appears to be a promising approach. Specifically, the error estimate for the step $t_n \mapsto t_n + h$ is given by $\mathrm{err}_q = \boldsymbol{q}_n - \boldsymbol{q}(t_n)$. Here, the collocation polynomial is defined as $\boldsymbol{q}(t_n + \vartheta h) = \sum_{i=1}^{s} l_i(\vartheta)\boldsymbol{Q}_i$, expressed in terms of the Lagrange polynomials $l_i(\vartheta)$ for the nodes $c_1, c_2, \ldots, c_s$ and the stage positions $\boldsymbol{Q}_i$. A similar strategy was proposed by Guglielmi and Hairer [78, 79], in the context of stiff delay differential equations, differential algebraic delay equations and neutral problems with discontinuous solution using Radau IIA methods. Pursuing the path of finding a suitable step-size control mechanism is considered essential for leveraging the potential of the proposed numerical methods to meet industrial standards.

As the numerical experiments of Section 7.7 indicate, the proposed nonsmooth Runge–Kutta methods show the expected orders of convergence when applied to frictional contact problems with constant contact state that can alternatively be described by differential algebraic equations. During motions where the contact state changes, the convergence drops to a nonmonotone order of one. Although this is not a very satisfactory observation, such higher-order methods seem to outperform classical first-order methods as observed in Section 7.10, when applied to the Tippedisk example. Future investigations have to find rigorous mathematical proofs for the observed orders of convergence and should include explanations to the observed superiority of higher-order methods applied to measure differential inclusions with mechanical structure.

Bibliography

[1] V. Acary. Toward higher order event-capturing schemes and adaptive time-step strategies for nonsmooth multibody systems. Research Report RR-7151, INRIA, 2009.

[2] V. Acary. Higher order event capturing time-stepping schemes for nonsmooth multibody systems with unilateral constraints and impacts. *Applied Numerical Mathematics*, 62(10):1259–1275, 2012. Selected Papers from NUMDIFF-12.

[3] V. Acary. Projected event-capturing time-stepping schemes for nonsmooth mechanical systems with unilateral contact and Coulomb's friction. *Computer Methods in Applied Mechanics and Engineering*, 256:224–250, 2013.

[4] V. Acary and F. Bourrier. Coulomb friction with rolling resistance as a cone complementarity problem. *European Journal of Mechanics - A/Solids*, 85:104046, 2021.

[5] V. Acary and B. Brogliato. *Numerical Methods for Nonsmooth Dynamical Systems: Applications in Mechanics and Electronics*, volume 35 of *Lecture Notes in Applied and Computational Mechanics*. Springer, Berlin, Heidelberg, 2008.

[6] V. Acary, M. Brémond, and O. Huber. *On Solving Contact Problems with Coulomb Friction: Formulations and Numerical Comparisons*, pages 375–457. Springer International Publishing, Cham, 2018.

[7] V. Acary, P. Armand, H. M. Nguyen, and M. Shpakovych. Second-order cone programming for frictional contact mechanics using interior point algorithm. *Optimization Methods and Software*, pages 1–30, 2024.

[8] R. Alexander. Diagonally Implicit Runge–Kutta Methods for Stiff O.D.E.'s. *SIAM Journal on Numerical Analysis*, 14(6):1006–1021, 1977.

[9] J. Allard, S. Cotin, F. Faure, P.-J. Bensoussan, F. Poyer, C. Duriez, H. Delingette, and L. Grisoni. SOFA–an open source framework for medical simulation. *Studies in Health Technology and Informatics*, 125:13–18, 2007.

[10] H. C. Andersen. Rattle: A "Velocity" Version of the Shake Algorithm for Molecular Dynamics Calculations. *Journal of Computational Physics*, 52 (1):24–34, Oct. 1983.

[11] M. Anitescu and F. A. Potra. Formulating Dynamic Multi-Rigid-Body Contact Problems with Friction as Solvable Linear Complementarity Problems. *Nonlinear Dynamics*, 14(3):231–247, Nov 1997.

[12] S. S. Antman. *Nonlinear Problems of Elasticity*, volume 107 of *Applied Mathematical Sciences*. Springer, 2nd edition, 2005.

[13] C. Armanini, F. Boyer, A. T. Mathew, C. Duriez, and F. Renda. Soft Robots Modeling: A Structured Overview. *IEEE Transactions on Robotics*, 39(3):1728–1748, 2023.

[14] B. Armstrong-Hélouvry, P. Dupont, and C. C. De Wit. A survey of models, analysis tools and compensation methods for the control of machines with friction. *Automatica*, 30(7):1083–1138, 1994.

[15] M. Arnold and O. Brüls. Convergence of the generalized-α scheme for constrained mechanical systems. *Multibody System Dynamics*, 18(2):185–202, Sep 2007.

[16] M. Arnold, A. Cardona, and O. Brüls. *A Lie Algebra Approach to Lie Group Time Integration of Constrained Systems*, pages 91–158. Springer International Publishing, Cham, 2016.

[17] U. M. Ascher and L. R. Petzold. Projected Implicit Runge-Kutta Methods for Differential-Algebraic Equations. *SIAM Journal on Numerical Analysis*, 28(4):1097–1120, 1991.

[18] R. Bank, W. Coughran, W. Fichtner, E. Grosse, D. Rose, and R. Smith. Transient Simulation of Silicon Devices and Circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 4(4):436–451, 1985.

[19] J. Bender, M. Müller, M. A. Otaduy, M. Teschner, and M. Macklin. A Survey on Position-Based Simulation Methods in Computer Graphics. *Computer Graphics Forum*, 33(6):228–251, 2014.

[20] R. Bendfeld and C. D. Remy. Contact Force Control with Continuously Compliant Robotic Legs. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5256–5262, 2023.

[21] M. Benzi, G. H. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta Numerica*, 14:1–137, 2005.

[22] F. Bernicot and J. Venel. Existence of solutions for second-order differential inclusions involving proximal normal cones. *Journal de Mathématiques Pures et Appliquées*, 98(3):257–294, 2012.

[23] R. Berthold, M. N. Bartholdt, M. Wiese, S. Kahms, S. Spindeldreier, and A. Raatz. A Preliminary Study of Soft Material Robotic Modelling: Finite Element Method and Cosserat Rod Model. In *2021 9th International Conference on Control, Mechatronics and Automation (ICCMA)*, pages 7–13, 2021.

[24] T. A. Bickart. An Efficient Solution Process for Implicit Runge–Kutta Methods. *SIAM Journal on Numerical Analysis*, 14(6):1022–1027, 1977.

[25] A. Bosten, A. Cosimo, J. Linn, and O. Brüls. A mortar formulation for frictionless line-to-line beam contact. *Multibody System Dynamics*, 54(1): 31–52, Jan 2022.

[26] V. Brasey and E. Hairer. Half-Explicit Runge–Kutta Methods for Differential-Algebraic Systems of Index 2. *SIAM Journal on Numerical Analysis*, 30(2):538–552, 1993.

[27] K. E. Brenan and L. R. Petzold. The Numerical Solution of Higher Index Differential/Algebraic Equations by Implicit Methods. *SIAM Journal on Numerical Analysis*, 26(4):976–996, 1989.

[28] K. E. Brenan, S. L. Campbell, and L. R. Petzold. *Numerical Solution of Initial-value Problems in Differential-algebraic Equations*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, 1996.

[29] J. Breuling, G. Capobianco, S. R. Eugster, and R. I. Leine. A nonsmooth RATTLE algorithm for mechanical systems with frictional unilateral constraints. *Nonlinear Analysis: Hybrid Systems*, 52:101469, 2024.

[30] B. Brogliato. *Nonsmooth Mechanics: Models, Dynamics and Control*. Communications and Control Engineering. Springer International Publishing, 2016.

[31] O. Brüls, V. Acary, and A. Cardona. Simultaneous enforcement of constraints at position and velocity levels in the nonsmooth generalized-α scheme. *Computer Methods in Applied Mechanics and Engineering*, 281: 131–161, 2014.

[32] O. Brüls, V. Acary, and A. Cardona. On the Constraints Formulation in the Nonsmooth Generalized-α Method. In *Advanced Topics in Nonsmooth Dynamics*, pages 335–374. Springer, 2018.

[33] O. Brüls and M. Arnold. The Generalized-α Scheme as a Linear Multistep Integrator: Toward a General Mechatronic Simulator. *Journal of Computational and Nonlinear Dynamics*, 3(4):041007, 08 2008.

[34] J. C. Butcher. Implicit Runge–Kutta processes. *Mathematics of computation*, 18(85):50–64, 1964.

[35] J. C. Butcher. On Runge–Kutta processes of high order. *Journal of the Australian Mathematical Society*, 4(2):179–194, 1964.

[36] J. C. Butcher. A stability property of implicit Runge–Kutta methods. *BIT Numerical Mathematics*, 15(4):358–361, Dec 1975.

[37] J. C. Butcher. On the implementation of implicit Runge–Kutta methods. *BIT Numerical Mathematics*, 16(3):237–240, Sep 1976.

[38] J. C. Butcher and G. Wanner. Runge–Kutta methods: some historical notes. *Applied Numerical Mathematics*, 22(1):113–151, 1996. Special Issue Celebrating the Centenary of Runge–Kutta Methods.

[39] G. Capobianco. *Mechanical systems with frictional contact: geometric theory and time discretization methods*. PhD thesis, University of Stuttgart, 2021.

[40] G. Capobianco and S. R. Eugster. Time finite element based Moreau-type integrators. *International Journal for Numerical Methods in Engineering*, 114(3):215–231, 2018.

[41] G. Capobianco, J. Harsch, S. R. Eugster, and R. I. Leine. A Nonsmooth Generalized-alpha Method for Mechanical Systems with Frictional Contact. *International Journal for Numerical Methods in Engineering*, 122(22):6497–6526, Nov. 2021.

[42] G. Capobianco, J. Harsch, and S. Leyendecker. Lobatto-type variational integrators for mechanical systems with frictional contact. *Computer Methods in Applied Mechanics and Engineering*, 418:116496, 2024.

[43] J. R. Cash. Diagonally Implicit Runge–Kutta Formulae with Error Estimates. *IMA Journal of Applied Mathematics*, 24(3):293–301, 11 1979.

[44] Q.-Z. Chen, V. Acary, G. Virlez, and O. Brüls. A nonsmooth generalized-α scheme for flexible multibody systems with unilateral constraints. *International Journal for Numerical Methods in Engineering*, 96(8):487–511, 2013.

[45] J. Chung and G. M. Hulbert. A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-α method. *Journal of applied mechanics*, 60(2):371–375, 1993.

[46] A. Ciallella, I. Giorgio, E. Barchiesi, G. Alaimo, A. Cattenone, B. Smaniotto, A. Vintache, F. D'Annibale, F. dell'Isola, F. Hild, and F. Auricchio. A 3D pantographic metamaterial behaving as a mechanical shield: Experimental and numerical evidence. *Materials & Design*, 237:112554, 2024.

[47] A. Cosimo, J. Galvez, F. J. Cavalieri, A. Cardona, and O. Brüls. A robust nonsmooth generalized-α scheme for flexible systems with impacts. *Multibody System Dynamics*, 48(2):127–149, 2020.

[48] A. Cosimo, F. J. Cavalieri, J. Galvez, A. Cardona, and O. Brüls. A General Purpose Formulation for Nonsmooth Dynamics With Finite Rotations: Application to the Woodpecker Toy. *Journal of Computational and Nonlinear Dynamics*, 16(3), 2021.

[49] R. W. Cottle, J. S. Pang, and R. E. Stone. *The Linear Complementarity Problem*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, 2009.

[50] E. Coumans and Y. Bai. PyBullet, a Python module for physics simulation for games, robotics and machine learning, 2016–2021.

[51] A. Doménech, T. Doménech, and J. Cebrián. Introduction to the study of rolling friction. *American Journal of Physics*, 55(3):231–235, 03 1987.

[52] D. Durville. Simulation of the mechanical behaviour of woven fabrics at the scale of fibers. *International Journal of Material Forming*, 3(2):1241–1251, Sep 2010.

[53] D. Durville. Contact-friction modeling within elastic beam assemblies: an application to knot tightening. *Computational Mechanics*, 49(6):687–707, Jun 2012.

[54] D. Durville, I. Baydoun, H. Moustacas, G. Périé, and Y. Wielhorski. Determining the initial configuration and characterizing the mechanical properties of 3D angle-interlock fabrics using finite element simulation. *International Journal of Solids and Structures*, 154:97–103, 2018. Multiscale Modelling of Fibrous and Textile Materials.

[55] L. Eberhardt, R. I. Leine, J. Harsch, S. R. Eugster, P. Bartelt, and H. Lanter. Application of nonsmooth dynamics to rockfall protection ring net simulation, 2023. International Nonlinear Dynamics Conference (NODYCON), Rome, Italy.

[56] O. Egeland and J. T. Gravdahl. *Modeling and Simulation for Automatic Control*. Marine Cybernetics, 2002.

[57] B. Ehle. *On Padé Approximations to the Exponential Function and A-stable Methods for the Numerical Solution of Initial Value Problems.* Report (University of Waterloo. Department of Applied Analysis and Computer Science). University of Waterloo, 1969.

[58] T. Erez, Y. Tassa, and E. Todorov. Simulation tools for model-based robotics: Comparison of Bullet, Havok, MuJoCo, ODE and PhysX. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4397–4404, 2015.

[59] K. Erleben. Rigid Body Contact Problems Using Proximal Operators. In *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, SCA '17, New York, NY, USA, 2017. Association for Computing Machinery.

[60] S. R. Eugster, J. Harsch, M. Bartholdt, M. Herrmann, M. Wiese, and G. Capobianco. Soft Pneumatic Actuator Model Based on a Pressure-Dependent Spatial Nonlinear Rod Theory. *IEEE Robotics and Automation Letters*, 7(2):2471–2478, 2022.

[61] B. Fabien. *Analytical System Dynamics: Modeling and Simulation.* Springer US, Boston, MA, 2009.

[62] F. Facchinei and J. Pang. *Finite-Dimensional Variational Inequalities and Complementarity Problems.* Springer Series in Operations Research and Financial Engineering. Springer New York, 2007.

[63] P. Flores, R. I. Leine, and Ch. Glocker. Modeling and analysis of planar rigid multibody systems with translational clearance joints based on the non-smooth dynamics approach. *Multibody System Dynamics*, 23(2):165–190, Feb 2010.

[64] U. Galvanetto, S. Bishop, and L. Briseghella. Mechanical Stick-Slip Vibrations. *International Journal of Bifurcation and Chaos*, 05(03):637–651, 1995.

[65] J. Galvez, F. J. Cavalieri, A. Cosimo, O. Brüls, and A. Cardona. A nonsmooth frictional contact formulation for multibody system dynamics. *International Journal for Numerical Methods in Engineering*, 121(16):3584–3609, 2020.

[66] C. W. Gear. Simultaneous Numerical Solution of Differential-Algebraic Equations. *IEEE Transactions on Circuit Theory*, 18(1):89–95, 1971.

[67] C. W. Gear. Differential-Algebraic Equation Index Transformations. *SIAM Journal on Scientific and Statistical Computing*, 9(1):39–47, 1988.

[68] C. W. Gear, B. Leimkuhler, and G. K. Gupta. Automatic integration of Euler–Lagrange equations with constraints. *Journal of Computational and Applied Mathematics*, 12:77–90, 1985.

[69] C. Gehring, R. Diethelm, R. Siegwart, G. Nützi, and R. I. Leine. An Evaluation of Moreau's Time-Stepping Scheme for the Simulation of a Legged Robot. volume Volume 6: 10th International Conference on Multibody Systems, Nonlinear Dynamics, and Control of *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 08 2014.

[70] F. Génot and B. Brogliato. New results on Painlevé paradoxes. *European Journal of Mechanics-A/Solids*, 18(4):653–677, 1999.

[71] Ch. Glocker. *Dynamik von Starrkörpersystemen mit Reibung und Stößen*. PhD thesis, TU München, 1995.

[72] Ch. Glocker. *Set-Valued Force Laws*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[73] Ch. Glocker. An Introduction to Impacts. In H. J. and S. G.E., editors, *Nonsmooth Mechanics of Solids* , volume 485 of *CISM International Centre for Mechanical Sciences*. Springer, 2006.

[74] Ch. Glocker. Simulation of hard contacts with friction: an iterative projection method. In *Recent Trends in Dynamical Systems*, pages 493–515. Springer, 2013.

[75] Ch. Glocker, E. Cataldi-Spinola, and R. Leine. Curve squealing of trains: Measurement, modelling and simulation. *Journal of Sound and Vibration*, 324(1):365–386, 2009.

[76] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Philadelphia, PA, 2013.

[77] E. Griepentrog. Gemischte Runge–Kutta-Verfahren für steife Systeme. *In: Seminar-bericht Nr. 11, Sekt. Math., Humboldt-Univ. Berlin*, 11:19–29, 1978.

[78] N. Guglielmi. Open issues in devising software for the numerical solution of implicit delay differential equations. *Journal of Computational and Applied Mathematics*, 185(2):261–277, 2006. Special Issue: International Workshop on the Technological Aspects of Mathematics.

[79] N. Guglielmi and E. Hairer. Implementing Radau IIA Methods for Stiff Delay Differential Equations. *Computing*, 67(1):1–12, Jul 2001.

[80] E. Hairer and L. O. Jay. *Implicit Runge–Kutta Methods for Higher Index Differential-Algebraic Systems*, pages 213–224. WORLD SCIENTIFIC, 1993.

[81] E. Hairer and G. Wanner. Stiff differential equations solved by Radau methods. *Journal of Computational and Applied Mathematics*, 111(1): 93–111, 1999.

[82] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II*. Springer, second edition, 2002.

[83] E. Hairer, M. Roche, and C. Lubich. *The Numerical Solution of Differential-Algebraic Systems by Runge–Kutta Methods*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1989.

[84] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I*. Springer, 1993.

[85] E. Hairer, C. Lubich, and G. Wanner. *Geometric numerical integration*, volume 31 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, second edition, 2006.

[86] J. Harsch and S. R. Eugster. Nonunit quaternion parametrization of a Petrov–Galerkin Cosserat rod finite element. *PAMM*, 23(4):e202300172, 2023.

[87] J. Harsch, G. Ganzosch, E. Barchiesi, A. Ciallella, and S. R. Eugster. Experimental analysis, discrete modeling and parameter optimization of SLS-printed bi-pantographic structures. *Mathematics and Mechanics of Solids*, 27(10):2201–2217, 2022.

[88] M. D. Hersey. Rolling Friction, I—Historical Introduction. *Journal of Lubrication Technology*, 91(2):260–263, 04 1969.

[89] H. M. Hilber, T. J. R. Hughes, and R. L. Taylor. Improved numerical dissipation for time integration algorithms in structural dynamics. *Earthquake Engineering & Structural Dynamics*, 5(3):283–292, 1977.

[90] M. Hintermüller, K. Ito, and K. Kunisch. The Primal-Dual Active Set Strategy as a Semismooth Newton Method. *SIAM Journal on Optimization*, 13(3):865–888, 2002.

[91] E. Hofer. A Partially Implicit Method for Large Stiff Systems of ODEs with Only Few Equations Introducing Small Time-Constants. *SIAM Journal on Numerical Analysis*, 13(5):645–663, 1976.

[92] P. C. Horak and J. C. Trinkle. On the Similarities and Differences Among Contact Models in Robot Simulation. *IEEE Robotics and Automation Letters*, 4(2):493–499, 2019.

[93] M. Hosea and L. Shampine. Analysis and implementation of TR-BDF2. *Applied Numerical Mathematics*, 20(1):21–37, 1996.

[94] T. A. Howell, S. L. Cleac'h, J. Brüdigam, J. Z. Kolter, M. Schwager, and Z. Manchester. Dojo: A Differentiable Physics Engine for Robotics, 2023.

[95] S. Hüeber, G. Stadler, and B. I. Wohlmuth. A Primal-Dual Active Set Algorithm for Three-Dimensional Contact Problems with Coulomb Friction. *SIAM Journal on Scientific Computing*, 30(2):572–596, 2008.

[96] K. E. Jansen, C. H. Whiting, and G. M. Hulbert. A generalized-α method for integrating the filtered Navier–Stokes equations with a stabilized finite element method. *Computer Methods in Applied Mechanics and Engineering*, 190(3):305–319, 2000. ISSN 0045-7825.

[97] L. Jay. Convergence of a class of Runge–Kutta methods for differential-algebraic systems of index 2. *BIT Numerical Mathematics*, 33(1):137–150, Mar 1993.

[98] L. O. Jay. Convergence of Runge-Kutta methods for differential-algebraic systems of index 3. *Applied Numerical Mathematics*, 17(2):97–118, 1995.

[99] L. O. Jay. Symplectic Partitioned Runge–Kutta Methods for Constrained Hamiltonian Systems. *SIAM Journal on Numerical Analysis*, 33(1):368–387, 1996.

[100] L. O. Jay. *Lobatto Methods*, pages 817–826. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.

[101] L. O. Jay and D. Negrut. *A Second Order Extension of the Generalized–α Method for Constrained Systems in Mechanics*, pages 143–158. Springer Netherlands, Dordrecht, 2009.

[102] M. Jean. The non-smooth contact dynamics method. *Computer Methods in Applied Mechanics and Engineering*, 177(3):235–257, 1999. ISSN 0045-7825.

[103] M. Jean and J. J. Moreau. Unilaterality and dry friction in the dynamics of rigid body collections. In *1st Contact Mechanics International Symposium*, pages 31–48, Lausanne, Switzerland, 1992.

[104] C. T. Kelley. *Iterative Methods for Linear and Nonlinear Equations*. Society for Industrial and Applied Mathematics, 1995.

[105] C. T. Kelley. *Solving Nonlinear Equations with Newton's Method*. Fundamentals of Algorithms. Society for Industrial and Applied Mathematics, 2003.

[106] J. Kleinert and B. Simeon. *Differential-Algebraic Equations and Beyond: From Smooth to Nonsmooth Constrained Dynamical Systems*, pages 73–132. Springer International Publishing, Cham, 2022.

[107] B. M. Kwak. Complementarity Problem Formulation of Three-Dimensional Frictional Contact. *Journal of Applied Mechanics*, 58(1):134–140, 03 1991.

[108] C. Le Saux, R. I. Leine, and Ch. Glocker. Dynamics of a rolling disk in the presence of dry friction. *Journal of Nonlinear Science*, 15(1):27–61, 2005.

[109] J. Lee, M. X. Grey, S. Ha, T. Kunz, S. Jain, Y. Ye, S. S. Srinivasa, M. Stilman, and C. K. Liu. DART: Dynamic Animation and Robotics Toolkit. *Journal of Open Source Software*, 3(22):500, 2018.

[110] B. J. Leimkuhler and R. D. Skeel. Symplectic Numerical Integrators in Constrained Hamiltonian Systems. *J. Comput. Phys.*, 112(1):117–125, may 1994.

[111] R. I. Leine. Experimental and theoretical investigation of the energy dissipation of a rolling disk during its final stage of motion. *Archive of Applied Mechanics*, 79(11):1063–1082, Nov 2009.

[112] R. I. Leine and Ch. Glocker. A set-valued force law for spatial Coulomb–Contensou friction. *European Journal of Mechanics – A/Solids*, 22:193–216, 2003.

[113] R. I. Leine and H. Nijmeijer. *Dynamics and Bifurcations of Non-Smooth Mechanical Systems*, volume 18 of *Lecture Notes in Applied and Computational Mechanics*. Springer, Berlin, Heidelberg, 2004.

[114] R. I. Leine and N. van de Wouw. *Stability and Convergence of Mechanical Systems with Unilateral Constraints*, volume 36 of *Lecture Notes in Applied and Computational Mechanics*. Springer-Verlag, Berlin, 2008.

[115] R. I. Leine, D. H. van Campen, A. de Kraker, and L. van den Steen. Stick-Slip Vibrations Induced by Alternate Friction Models. *Nonlinear Dynamics*, 16(1):41–54, May 1998.

[116] R. I. Leine, B. Brogliato, and H. Nijmeijer. Periodic motion and bifurcations induced by the Painlevé paradox. *European Journal of Mechanics – A/Solids*, 21(5):869–896, 2002.

[117] R. I. Leine, A. Schweizer, M. Christen, J. Glover, P. Bartelt, and W. Gerber. Simulation of rockfall trajectories with consideration of rock shape. *Multibody System Dynamics*, 32(2):241–271, Aug 2014.

[118] Q. L. Lidec, W. Jallet, L. Montaut, I. Laptev, C. Schmid, and J. Carpentier. Contact Models in Robotics: a Comparative Analysis, 2023.

[119] Ch. Lubich. On projected Runge–Kutta methods for differential-algebraic equations. *BIT Numerical Mathematics*, 31(3):545–550, Sep 1991.

[120] M. Macklin, K. Erleben, M. Müller, N. Chentanez, S. Jeschke, and V. Makoviychuk. Non-smooth Newton Methods for Deformable Multi-body Dynamics. *ACM Trans. Graph.*, 38(5), oct 2019.

[121] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State. Isaac Gym: High Performance GPU-Based Physics Simulation For Robot Learning, 2021.

[122] O. Mangasarian. Global error bounds for monotone affine variational inequality problems. *Linear Algebra and its Applications*, 174:153–163, 1992.

[123] F. Marques, P. Flores, J. C. P. Claro, and H. M. Lankarani. Modeling and analysis of friction including rolling effects in multibody dynamics: a review. *Multibody System Dynamics*, 45(2):223–244, Feb 2019.

[124] C. Meier, W. A. Wall, and A. Popp. A unified approach for beam-to-beam contact. *Computer Methods in Applied Mechanics and Engineering*, 315: 972–1010, 2017.

[125] M. Möller. *Consistent Integrators for Non-Smooth Dynamical Systems*. PhD thesis, ETH Zurich, 2011.

[126] M. Möller, R. I. Leine, and Ch. Glocker. An efficient approximation of orthotropic set-valued force laws of normal cone type. In *7th Euromech Solid Mechanics Conference, Lisbon*, pages 7–11, 2009.

[127] J. J. Moreau. Dynamique de systèmes à liaisons unilatérales avec frottement sec éventuel ; essais numériques. Research report, Université des Sciences et Techniques du Languedoc, 1986.

[128] J. J. Moreau. Unilateral Contact and Dry Friction in Finite Freedom Dynamics. In J. J. Moreau and P. D. Panagiotopoulos, editors, *Non-Smooth Mechanics and Applications. CISM Courses and Lectures*, pages 1–82. Springer, Wien, 1988.

[129] J. J. Moreau. Bounded Variation in Time. In J. J. Moreau, P. D. Panagiotopoulos, and G. Strang, editors, *Topics in Nonsmooth Mechanics*, pages 1–74. Birkhäuser Verlag, Basel, 1988.

[130] J. J. Moreau. Numerical aspects of the sweeping process. *Computer Methods in Applied Mechanics and Engineering*, 177:329 – 349, 1999.

[131] K. W. Müller, C. Meier, and W. A. Wall. Resolution of sub-element length scales in Brownian dynamics simulations of biopolymer networks with geometrically exact beam finite elements. *Journal of Computational Physics*, 303:185–202, 2015.

[132] N. M. Newmark. A Method of Computation for Structural Dynamics. *Journal of the Engineering Mechanics Division*, 85(3):67–94, 1959.

[133] G. E. Nützi. *Non-Smooth Granular Rigid Body Dynamics with Applications to Chute Flows*. PhD thesis, ETH Zurich, 2016.

[134] NVIDIA Corporation. PhysX. URL `https://github.com/NVIDIA-Omniverse/PhysX`.

[135] J. Ortega and W. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, 1970.

[136] A. Ostermann. A class of half-explicit Runge–Kutta methods for differential-algebraic systems of index 3. *Applied Numerical Mathematics*, 13(1):165–179, 1993.

[137] P. Painlevé. Sur les lois du frottement de glissement. *Comptes Rendu des Séances de l'Academie des Sciences*, 121:112–115, 1895.

[138] L. Paoli and M. Schatzman. A numerical scheme for impact problems I: The one-dimensional case. *SIAM Journal on Numerical Analysis*, 40(2): 702–733, 2002.

[139] L. Paoli and M. Schatzman. A numerical scheme for impact problems II: The multidimensional case. *SIAM Journal on Numerical Analysis*, 40(2): 734–768, 2002.

[140] L. R. Petzold. Description of DASSL: a differential/algebraic system solver. 1982.

[141] F. Pfeiffer and Ch. Glocker. *Multibody Dynamics with Unilateral Contacts*. John Wiley & Sons, Ltd, 1996.

[142] F. G. Pfeiffer and Ch. Glocker. On frictionless impact models in rigid-body systems. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 359(1789):2385–2404, 2001.

[143] A. Popp and I. Steinbrecher. *Finite Element Formulations for Beam-to-Solid Interaction–from Embedded Fibers Towards Contact*, pages 425–434. Springer International Publishing, Cham, 2022.

[144] W. H. Press and S. A. Teukolsky. Adaptive Stepsize Runge-Kutta Integration. *Computer in Physics*, 6(2):188–191, 03 1992.

[145] A. Prothero and A. Robinson. On the Stability and Accuracy of One-Step Methods for Solving Stiff Systems of Ordinary Differential Equations. *Mathematics of Computation*, 28(125):145–162, 1974.

[146] L. Qi and J. Sun. A nonsmooth version of Newton's method. *Mathematical Programming*, 58(1):353–367, Jan 1993.

[147] E. Rabinowicz. Stick and slip. *Scientific American*, pages 109–119, 1956.

[148] J. Rang. Apdative timestep control for fully implicit Runge-Kutta methods of higher order. *Informatik-Berichte der Technischen Universität Braunschweig*, 2014-03, 2014.

[149] R. T. Rockafellar. *Convex Analysis*. Princeton Mathematical Series. Princeton University Press, 1970.

[150] R. T. Rockafellar and R.-B. Wets. *Variational Analysis*. Springer, Berlin, 2009.

[151] D. C. Rucker. Integrating Rotations Using Nonunit Quaternions. *IEEE Robotics and Automation Letters*, 3(4):2979–2986, 2018.

[152] D. C. Rucker and R. J. Webster III. Statics and Dynamics of Continuum Robots With General Tendon Routing and External Loading. *IEEE Transactions on Robotics*, 27(6):1033–1044, 2011.

[153] N. Rudin, D. Hoeller, P. Reist, and M. Hutter. Learning to Walk in Minutes Using Massively Parallel Deep Reinforcement Learning, 2022.

[154] S. Sailer. *Nonlinear dynamics of the tippedisk: a holistic analysis*. PhD thesis, University of Stuttgart, 2023.

[155] S. Sailer, S. R. Eugster, and R. I. Leine. The Tippedisk: a Tippetop Without Rotational Symmetry. *Regular and Chaotic Dynamics*, 25(6): 553–580, Nov 2020.

[156] E. Sánchez, A. Cosimo, O. Brüls, A. Cardona, and F. J. Cavalieri. Non-smooth numerical solution for Coulomb friction, rolling and spinning resistance of spheres applied to flexible multibody system dynamics. *Multibody System Dynamics*, 59(1):69–103, Sep 2023.

[157] J. M. Sanz-Serna. *On Solving Contact Problems with Coulomb Friction: Formulations and Numerical Comparisons*, pages 437–449. Clarendon Press, Oxford, 1992.

[158] R. Sato Martín de Almagro. *Discrete Mechanics for Forced and Constrained Systems*. PhD thesis, Universidad Complutense Madrid, 2019.

[159] S. Schoeder, H. Ulbrich, and T. Schindler. Discussion of the Gear–Gupta–Leimkuhler method for impacting mechanical systems. *Multibody System Dynamics*, 31(4):477–495, Apr 2014.

[160] A. P. Schweizer. *Ein nichtglattes mechanisches Modell für Steinschlag*. PhD thesis, ETH Zürich, 2015.

[161] F. Seely and N. Ensign. *Analytical Mechanics for Engineers*. Wiley, 1921.

[162] B. Simeon. *On the History of Differential-Algebraic Equations*, pages 1–39. Springer International Publishing, Cham, 2017.

[163] L. M. Skvortsov. Diagonally implicit Runge–Kutta methods for differential algebraic equations of indices two and three. *Computational Mathematics and Mathematical Physics*, 50(6):993–1005, Jun 2010.

[164] R. L. Smith. The Open Dynamics Engine (ODE). URL `https://www.ode.org`.

[165] I. Steinbrecher, M. Mayr, M. J. Grill, J. Kremheller, C. Meier, and A. Popp. A mortar-type finite element approach for embedding 1D beams into 3D solid volumes. *Computational Mechanics*, 66(6):1377–1398, Dec 2020.

[166] I. Steinbrecher, A. Popp, and C. Meier. Consistent coupling of positions and rotations for embedding 1D Cosserat beams into 3D solid volumes. *Computational Mechanics*, 69(3):701–732, Mar 2022.

[167] 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(15):2673–2691, 1996.

[168] M. Stilz, J. Breuling, S. R. Eugster, M. Pawlikowski, and R. Grygoruk. Chirality effects in panto-cylindrical structures. *Mathematics and Mechanics of Solids*, 29(4):818–830, 2024.

[169] C. Studer. *Augmented time-stepping integration of non-smooth dynamical systems*. PhD thesis, ETH Zurich, Switzerland, 2008.

[170] C. Studer. *Numerics of Unilateral Contacts and Friction: Modeling and Numerical Time Integration in Non-smooth Dynamics*, volume 47 of *Lecture Notes in Applied and Computational Mechanics*. Springer, Berlin, Heidelberg, 2009.

[171] C. Studer, R. I. Leine, and Ch. Glocker. Step size adjustment and extrapolation for time-stepping schemes in non-smooth dynamics. *International Journal for Numerical Methods in Engineering*, 76(11):1747–1781, 2008.

[172] Y. B. Suris. Hamiltonian methods of Runge–Kutta type and their variational interpretation. *Mathematical Modelling*, 2(4):78–87, 1990.

[173] J. Till, V. Aloi, and D. C. Rucker. Real-time dynamics of soft and continuum robots based on Cosserat rod models. *The International Journal of Robotics Research*, 38(6):723–746, 2019.

[174] E. Todorov, T. Erez, and Y. Tassa. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.

[175] A. A. Transeth, R. I. Leine, Ch. Glocker, and K. Y. Pettersen. Non-smooth 3D Modeling of a Snake Robot with Frictional Unilateral Constraints. In *2006 IEEE International Conference on Robotics and Biomimetics*, pages 1181–1188, 2006.

[176] A. A. Transeth, R. I. Leine, Ch. Glocker, and K. Y. Pettersen. 3-D Snake Robot Motion: Nonsmooth Modeling, Simulations, and Experiments. *IEEE Transactions on Robotics*, 24(2):361–376, 2008.

[177] S. V. Walker and R. I. Leine. Set-valued anisotropic dry friction laws: formulation, experimental verification and instability phenomenon. *Nonlinear Dynamics*, 96:885–920, 2019.

[178] T. Winandy, M. Baumann, and R. I. Leine. Variational analysis of inequality impact laws for perfect unilateral constraints. In *Advanced Topics in Nonsmooth Dynamics*, pages 47–92. Springer, 2018.

[179] M. Wösle and F. Pfeiffer. Dynamics of Multibody Systems Containing Dependent Unilateral Constraints with Friction. *Journal of Vibration and Control*, 2(2):161–192, 1996.

[180] W. Ying, C. S. Henriquez, and D. J. Rose. Composite backward differentiation formula : an extension of the TR-BDF 2 scheme. 2011.