Schriftliche Habilitationsleistung
bestehend aus einer Habilitationsschrift
entstanden im Zeitraum des Habilitationsverfahrens
mit dem Titel

Consistent Time-Integration Schemes
for Flexible Multibody Systems
with Friction and Impacts
von Dr.-Ing. Thorsten Schindler
geboren am 14.04.1981 in Kassel
zur Erlangung der Lehrbefähigung für das Fachgebiet Angewandte Mechanik
im April 2015 eingereicht
an der Fakultät für Maschinenwesen der Technischen Universität München
Lehrstuhl für Angewandte Mechanik
Fakultät für Maschinenwesen, Technische Universität München

Fachmentorat

1. Univ.-Prof. Dr. Daniel Rixen (Vorsitz)
   Lehrstuhl für Angewandte Mechanik
   Fakultät für Maschinenwesen, Technische Universität München

2. Univ.-Prof. Dr.-Ing. Wolfgang Wall
   Lehrstuhl für Numerische Mechanik
   Fakultät für Maschinenwesen, Technische Universität München

3. Univ.-Prof. Dr.-Ing. Dr.-Ing. habil. Christoph Glocker
   Institut für Mechanische Systeme
   Departement für Maschinenbau und Verfahrenstechnik, ETH Zürich

Garching, den 14.04.2015

Univ.-Prof. Dr. Daniel Rixen
Lehrstuhl für Angewandte Mechanik
Fakultät für Maschinenwesen
Technische Universität München

Univ.-Prof. Dr.-Ing. Wolfgang Wall
Lehrstuhl für Numerische Mechanik
Fakultät für Maschinenwesen
Technische Universität München

Univ.-Prof. Dr.-Ing. Dr.-Ing. habil. Christoph Glocker
Institut für Mechanische Systeme
Departement für Maschinenbau und Verfahrenstechnik, ETH Zürich

ETH Zürich
Acknowledgment

This thesis summarizes my work as a research group leader for ‘numerical simulation’ and habilitation candidate at the Institute of Applied Mechanics of the Technische Universität München 2011-2015 as well as my time as a PostDoc at INRIA Grenoble in 2011 focussing on time-integration schemes and constraint solution techniques for nonsmooth mechanical systems.

The whole time after my PhD in 2010 would not have been so fruitful and interesting without the support of many people. I would like to express my gratitude to my mentors, Prof. Daniel Rixen, Prof. Wolfgang Wall and Prof. Christoph Glocker. Prof. Daniel Rixen, the successor of Prof. Heinz Ulbrich as head of the institute, in particular, did not only allow me to independently evolve in my own research but has also influenced my path with a lot of challenging new ideas. I would like to thank everyone, who inspired and supported my work after the PhD. Prof. Heinz Ulbrich and Thomas Thümmel motivated me to come back to the Institute of Applied Mechanics in 2011; they have always offered me, during my PhD 2006-2010, habilitation 2011-2015 and up to now invaluable guidance and advice. I remember Prof. Martin Arnold describing the habilitation period as an adventure. Indeed it was, a great one. It all started after meeting Vincent Acary in 2009, my PostDoc advisor at INRIA Grenoble in the team of Prof. Bernard Brogliato.

I would like to thank Prof. Friedrich Pfeiffer. Regular discussions and our close collaboration on two text books on dynamics inspired my work substantially.

As research group leader and experienced researcher as well as lecturer, I have closely cooperated with many people, hence the following survey will never be complete. I am thankful to all members of the Institute of Applied Mechanics and the people I have met and discussed to on my way, in particular to all the students I supervised. A lot of them are now PhD students around the globe. Thanks to my PhD mentees: Thomas Cebulla, Kilian Grundl, Oliver Hofmann, Markus Filippi, Bastian Esefeld and Andreas Krinner. The whole MBSim-Stammtisch team with Martin Förg, Markus Friedrich, Markus Schneider, Kilian Grundl, Jan Clauberg has been always a creative area for mechanical modelling and still works on the implementation of mechanical simulation using efficient computer methods. The European Network of Nonsmooth Mechanics, headed by Vincent Acary and Prof. Remco Leine, has been a fruitful platform for meeting experts and for intensive exchange of ideas and knowledge. Johannes Mayet and Benedikt Huber supported me in teaching multibody simulation and dynamics. The time as a Multiplikator of the Prolehre team of the Technische Universität München and organizer of ‘Fit in die Lehre’ at the Faculty of Mechanical Engineering with Heiko Peuscher has been very successful.
and valuable. Last but not least, I would like to thank the indispensable institute assistants, Manuela Müller-Philipp and Rita Schneider.

I have experienced the whole team of the Institute of Applied Mechanics, of the Technische Universität München and the community of nonsmooth mechanics all over the world as one big family resulting in a lot of good friends - thank you.

Thanks to all of my friends for the balance in my free time and deep gratitude to my growing little family for the understanding and support in the past and the future.

Schriesheim, October 2017

Thorsten Schindler
To my family.
Contents

I. Nonsmooth Framework 1

1. Motivation 3
   1.1. Aim and Scope ......................................... 9
   1.2. Industrial Applications of Nonsmooth Mechanical Systems .......... 11
       1.2.1. Pushbelt Continuously Variable Transmission ............... 11
       1.2.2. Rotor Dynamics ....................................... 12
       1.2.3. Elastohydrodynamics .................................. 12
       1.2.4. Belt Drives and Timing-Drives .......................... 14
       1.2.5. Surgical Assistance Systems ............................. 14
       1.2.6. Robotics ............................................... 15
       1.2.7. Computer Graphics ...................................... 16
       1.2.8. Granular Material ...................................... 16
       1.2.9. Electromechanical Systems ............................... 17
       1.2.10. Biological Systems ..................................... 18
   1.3. Nonsmooth Dynamical Systems .................................. 19
       1.3.1. Equations of Motion .................................... 19
       1.3.2. Contact and Impact Laws ................................ 22
       1.3.3. Contour Description .................................... 26
       1.3.4. Abstract Framework .................................... 29
   1.4. Integration Methods ......................................... 33
       1.4.1. Event-Driven Schemes .................................. 34
       1.4.2. Classical Timestepping Schemes .......................... 36
       1.4.3. Discussion .............................................. 38

2. Constraint Representation and Solution Algorithms 41
   2.1. Complementarity Formulation ................................ 42
   2.2. 2-step proj Formulation ..................................... 49
       2.2.1. Projected Splitting Methods .............................. 52
       2.2.2. LCP and 2-step proj Formulations: Painleve’s Paradox .... 55

3. Nonsmooth Dynamical Simulation Software 61
   3.1. SICONOS .................................................. 61
       3.1.1. Numerics ................................................ 62
       3.1.2. Kernel .................................................. 63
       3.1.3. Mechanics .............................................. 65
       3.1.4. Control .................................................. 66
       3.1.5. Front End .............................................. 66
   3.2. MBSim ..................................................... 66
   3.3. Comparison ................................................ 68
B.2. Definition of the General Scheme .............................................. 180
  B.2.1. Velocity Representation .................................................... 181
  B.2.2. Impact Representation ...................................................... 181
  B.2.3. Weighting Integral Representation: Reduced Evaluation ........ 181
  B.2.4. Runge-Kutta Representation .............................................. 182
B.3. Local Error for ODE Methods .................................................. 184

C. Dense Output and Error Evaluation ............................................. 185

D. Generalized-α Method ............................................................... 187

E. Bathe-Method ........................................................................... 189

F. Floating Frame of Reference Description for a Slider-Crank Mechanism 191
  F.1. Finite Element Formulation .................................................... 191
  F.2. Floating Frame of Reference .................................................. 192
  F.3. Mass Matrix ........................................................................ 194
  F.4. Quadratic Velocities .............................................................. 199
  F.5. Stiffness Matrix ................................................................... 201
  F.6. External Forces ..................................................................... 202
  F.7. Unilateral Constraints ............................................................ 203
  F.8. Boundary Conditions ............................................................. 204

G. Modal Analysis ........................................................................ 205

Bibliography .................................................................................. 209

List of Figures ............................................................................... 225

List of Tables ............................................................................... 231
Abstract

This work deals with nonsmooth mechanical systems and their discretization for numerical simulation. The mechanical systems consist of rigid and flexible bodies, joints as well as contacts and impacts with dry friction. We present a framework, which consistently treats velocity jumps, e.g., due to impacts, and benefits from nice properties of established integration methods. In impulsive as well as non-impulsive intervals, constraints are implicitly formulated on velocity level in terms of an augmented Lagrangian technique. They are satisfied exactly without any penetration and solved by semi-smooth Newton methods.

We present time-discontinuous Galerkin methods, which embed classical timestepping schemes from nonsmooth mechanics and yield consistent higher-order time-integration methods including impulsive forces. They transform higher order trial functions of event-driven integration schemes into consistent timestepping schemes for nonsmooth mechanical systems with friction and impacts. Splitting separates the portion of impulsive contact forces from the portion of non-impulsive contact forces. Impacts are included within the discontinuity of the piecewise continuous trial functions, that is, with first-order accuracy. Non-impulsive contact forces are integrated with respect to the local order of the trial functions. As a consequence, the computing time can be significantly reduced.

A further abstraction shows that also time-integration methods from computational mechanics with high-frequency damping profit from impulsive corrections. We compare the generalized-α method, the Bathe method and the ED-α method. These implicit time-integration schemes are only chosen as illustrative examples. Each base integration scheme tailored for a specific application can be embedded in the concept of mixed timestepping schemes, like it is introduced and presented in the present work.

Mixed timestepping schemes are formulated on velocity level. For preservation of geometric constraints, we offer the extension to projected timestepping schemes. The Gear-Gupta-Leimkuhler method or stabilized index 2 formulation enforces constraints on position and velocity level at the same time. It yields a robust numerical discretization avoiding the drift-off effect. Adding the position level constraint to a timestepping scheme on velocity level maintains physical consistency of the impulsive discretization.

We demonstrate the capability of our formalism with various nonlinear and flexible multi-contact examples with friction and impacts. It is shown that the newly proposed integration schemes yield a unified behavior for the description of contact mechanical problems.
Part I.

Nonsmooth Framework
1. Motivation

Mechanical modeling is aimed to represent a real machine or at least its vision by a model. The objective is the mechanical understanding of the underlying processes or the phenomena, which we have encountered in practice. Modeling is goal-oriented. If our model is too complex, it will represent too many details and probably it will hide the reason, e.g., for some disturbing vibration in a machine tool. We are interested in a model, which explains this disturbing vibration. Hence, the model has to be complete in this sense but not more complex than necessary: "Make things as simple as possible, but not simpler" (Albert Einstein). The assumption that the mere application of automatic modeling tools will yield a solution is often illusive. A phenomenon may be reproduced but the solution may not be found because of complexity. It is even worse, if we identify the wrong solution or propose construction improvements not based on appropriate physical proportions. Modeling is not an easy task and requires expert knowledge and experience. Effective modeling is an iterative process and somehow can be considered as the art of a finally successful engineer or scientist.

We consider another idea, which makes mechanical or physical modeling the work of a versed artist who has to be full-fledged in plenty of different topics. The automatically generated mathematical and lately also numerical models may show artificial effects or even do not adequately represent the phenomena from experiments. A superposition of numerical artifacts may disturb the mechanical phenomenon, whereas mathematical properties may yield non-solvable problems or misleading reports. The mechanical or physical model has to be adapted to avoid such issues. A black box in the modeling process may cause severe troubles, even for the safety of the later operators of the machine or for the benefit within the whole product cycle (Fig. 1.1).

The development of mechanical models, as well as the automatic generation of reliable mathematical and numerical models and their efficient solution are the main tasks in computational mechanics. In this work, we deal with, both, formalisms, which produce mathematical and numerical models from mechanical models, and the respective solution procedures. In the last decades, lots of formalisms have been presented for specific application areas. In mechanics, multibody systems and finite element methods are the two most wide-spread techniques. They provide a language and a way of thinking for mechanical modeling, as well as an automatic generation and solution of mathematical and numerical models. These two subjects importantly developed in 1950s-1960s, being addressed both by engineers and by mathematicians. Historically, multibody systems are considered to yield more physical discretizations, that is a more physical language and thinking [147, 146]. Perhaps the reason is that the original area of application of multibody systems is the representation of
the large motion of agglomerations of mechanical objects. At the beginning, these were considered to be rigid, hence, relatively simple to model. Figure 1.2 shows the typical modeling entities for multibody systems. Each modeling entity hides a block of discretized mathematical equations. Engineers model a multibody system with Bodys, which carry inertia, and Links, which do not have inertia properties. An automatic modeling tool generates the numerical model and solves it. Thereby, Links are used to define force interactions between Bodys. Reaction forces may occur due to Joints, which are mathematically equivalent to equality constraints, or due to Contacts, which are mathematically equivalent to inequality constraints. The latter vary the models topology over time and are the root of impacting behavior. Spring/Damping entities define applied or impressed forces between Bodys. It has not been said how the Bodys are connected by specific Link entities. In particular, this depends on the mathematical representation of the Body and its independent Contour description. A Link originally was used to couple disjoint body-fixed points on a Rigid Body. The latter is equivalent to the discretized differential equation of a free flying rigid mechanical object. Flexible Bodys, however, include deformations, e.g., elasticity of the Body. Their mathematical representation may be various.
Also the finite element method can be used for the derivation of the mathematical equations. Hence, Links may also represent, e.g., Mortar couplings [196, 150] within the finite element method. From this point of view, multibody systems form a larger framework which includes Body discretizations based on the finite element method. Contrary, the finite element method is considered to yield more mathematical discretizations [147, 146]. Its original area of application is the representation of small vibrational deflections within structural mechanics. Using local trial functions, one splits the underlying partial differential equations within the domain of definition. This yields a grid which decomposes a mechanical object into finite elements. Each finite element represents part of the support of the trial functions and so the local residuum of the mechanical equations. Assembling the local residua to the global residuum yields a spatial discretization. Hence, one has to define the geometry and the material behavior, as well as a finite element including shape of the local domain and respective trial functions. The meshing, assembling and solution is done by the automatic modeling tool. Nowadays, finite element methods are not restricted to small vibrational deflections, but can be used to cover large motions. From this point of view, multibody systems are part of the finite element method.

Due to the mainly separated history of multibody systems and finite element methods, we may find similar solution strategies using different notation. We try to respect both points of view, though we shall start from the view of multibody systems.

Figure 1.3 shows a multibody system as an abstraction of a real world system. In this work, we emphasize the treatment of Contacts, contributing to the development of the disciplines computational contact mechanics and nonsmooth mechanics. Computational contact mechanics is a subdiscipline of computational mechanics, which is mainly based on the application of the finite element method. Hence, e.g., elasticity of the Bodys is usually considered. When the two Bodys in Fig. 1.3 get in contact, that is the contact closes, a wave travels through both Bodys due to elasticity, reflects and forces the Bodys to separate. Contact forces have a finite value in continuum descriptions.

Nonsmooth mechanics is based on rigidity. Contacting Rigid Bodys impact. The traveling wave effect has to be represented by an additional law, the impact law, which is valid only for the time of impact. Thereby, the time of impact is considered to be infinitesimally small. The position and the orientation of the Bodys stay constant, whereas the velocity and the angular velocity change. They are defined
**Figure 1.3.** A multibody system is an abstraction of a real world system [160, Figure 2.3].

Concerning the impact law just after the time of impact by an impulse, which has a finite value corresponding to the occurring velocity jump. The respective contact force is a Dirac-measure. This nonsmoothness describes ideal behavior in some sense. However, it makes the discipline nonsmooth mechanics to a difficult matter in theory and practice. Anyhow with the following example, we explain the need of nonsmooth modeling and solution techniques.

**Example 1.1 (Bouncing ball)** Let us have a look at Fig. 1.4. It depicts a classic example of nonsmooth mechanics, namely, the bouncing ball. The ball is considered to be a point mass \( m > 0 \). Its motion, with respect to time \( t \), is described by the position coordinate \( q \). We are interested in the motion of the ball from \( t = 0 \) to \( t = T > 0 \). Thereby, the ball is influenced by gravity \( \Gamma \), which contributes to the applied forces \( h < 0 \), and by the ground, which cannot be penetrated and "generates" reaction impulses in terms of interactions \( i \). The ball falls to the ground during a free flight period. The shape of its trajectory is the usual parabola. When the ball hits the ground at the time instant \( t_0 \), \( 0 < t_0 < T \), we apply an impact law. Newton's impact law states, that the velocity coordinate is reflected:

\[
    u^+(t_0) = -\epsilon_N u^-(t_0),
\]

**Figure 1.4.** Bouncing ball example [161, Figure 1, adapted].
where $\epsilon_N \in [0,1]$ is the restitution coefficient of the interaction between ball and ground in normal direction. We suppose that $\epsilon_N$ vanishes. Then, the right limit of the velocity coordinate at $t_0$, $u^+(t_0)$ also vanishes. The left limit of the velocity coordinate at $t_0$, $u^-(t_0)$, equals the final velocity of the ball due to its free flight. The reaction impulse can be calculated with the impact equation

$$m(u^+(t_0) - u^-(t_0)) = i(t_0).$$

After the impact, we ensure that the ball remains on the ground. This is achieved by the non-penetration condition

$$0 \leq q, \quad q \frac{di}{dt} = 0, \quad \frac{di}{dt} \geq 0.$$

Recall that the interaction $i$ is defined as an impulse. With the non-penetration condition, we assume non-negativity of the position coordinate $q$ and of the interaction push force $\frac{di}{dt}$, which prevents the penetration. The vanishing product $q \frac{di}{dt}$ explains their complementary behavior. For the overall calculation, we need the transformation equation and the momentum equation, namely,

$$\dot{q} = u,$$

$$m \dot{u} = h + \frac{di}{dt}.$$

Evaluating the non-penetration condition does not provide us with sufficient information to calculate $\frac{di}{dt}$ because the position coordinate is not negative after the impact. However, the non-penetration condition has to be satisfied for all times. It hides information of its derivative:

$$0 \leq u, \quad u \frac{di}{dt} = 0, \quad \frac{di}{dt} \geq 0, \quad if \, q = 0.$$

The post-impact velocity vanishes and we still cannot calculate $\frac{di}{dt}$. We have to analyze the situation for the accelerations, in addition:

$$0 \leq \dot{u}, \quad \dot{u} \frac{di}{dt} = 0, \quad \frac{di}{dt} \geq 0, \quad if \, q = 0 \land u = 0.$$

Now, we insert the momentum equation and come out with a linear complementarity problem (LCP) for the unknown interaction push force. We try all possible configurations:

1. We show, that $\dot{u} > 0$ is not possible.
   Because of complementarity, it is $\frac{di}{dt} = 0$. Inserting in the momentum equation gives the contradiction $\dot{u} = \frac{h}{m} < 0$.

2. We show, that $\dot{u} = 0 \land \frac{di}{dt} = 0$ is not possible.
   Indeed, inserting $\frac{di}{dt} = 0$ in the momentum equation gives the contradiction $\dot{u} = \frac{h}{m} < 0$. 


3. The relation $\dot{u} = 0 \land \frac{di}{dt} > 0$ is the only possibility.

It follows

$$\frac{di}{dt} = -h.$$ 

Nonsmooth mechanics nicely describes the ideal behavior of the ball. If the post-impact velocity does not vanish, as we have assumed, but the impact is partly elastic, bouncing and free flight alternate infinitely often in a sufficient large finite time interval. Such a phenomenon is called Zeno phenomenon, or an accumulation of impacts.

![Figure 1.5: Abstract bouncing ball example with finite elements.](image)

In computational contact mechanics, we would deal with an elastic or plastic ball. The ball may be discretized with, e.g., brick finite elements. Hence, the mass of the ball is distributed to the finite element nodes according to the interpolation defined by the trial functions. In particular, there are surface nodes with a finite mass, which hit the ground first. This is a similar situation as the bouncing ball example itself (Fig. 1.5).

An impact occurs and we need an impact law to describe the ideal post-impact situation. After the impact, a traveling wave effect is initiated by the elastic coupling of the finite element nodes. Impulsive forces from the impact transform to finite force distributions within the whole ball. However we recognize that only in the continuous limit of the refinement the effect of the primary impact vanishes. In each practical refinement situation, an impact occurs at a specific range of finite element surface nodes. Typically, this point of view is not considered in computational contact mechanics. Coming from the continuous case, finite contact forces are assumed and time-discretization schemes are designed upon this precondition. Classic time-discretization schemes from computational contact mechanics tend to suffer from stability issues in the local contact velocities because of this reason [112, 3]. Different stabilization strategies have been proposed to correct these artificial effects, e.g., by a subsequent treating [119, 51] or by mass re-distribution techniques [52, 196], which consistently set the surface node masses to zero. Speaking in the language of multibody systems, this means inserting a layer of well-defined springs on the surface of the ball and so replacing the original ideal constraints by regularizations.
1.1. Aim and Scope

This work deals with nonsmooth mechanical systems, in particular with semi-discretized dynamical equations from computational mechanics, and their consistent treatment for numerical simulation. The contribution of the manuscript is the discussion and significant extension of time integration methods from nonsmooth mechanics and computational mechanics. We present the concept of time-discontinuous Galerkin methods, which embeds classical timestepping schemes from nonsmooth mechanics and yields consistent higher-order time-integration methods including impulsive forces. As a consequence, the computing time can be significantly reduced. A further abstraction shows that also time-integration methods from computational mechanics with high-frequency damping profit from impulsive corrections. These implicit time-integration schemes are only chosen as illustrative examples. Each base integration scheme, tailored for a specific application, can be embedded in the concept of mixed timestepping schemes, as introduced and presented in the present work. Mixed timestepping schemes are formulated on velocity level. If one is interested in preserving geometric constraints on position level, we offer the extension to projected timestepping schemes.

In the present Chapter 1, we introduce the basic notation of nonsmooth mechanical systems. We give industrial examples for possible applications of nonsmooth modeling. We explain the equations of motion of nonsmooth mechanical systems and their derivation. Time-discretization methods used in nonsmooth mechanics, that is, the families of event-driven schemes and timestepping schemes, are presented. Both families apply impact laws consistently. Therefore, event-driven schemes resolve the time of the impact up to numerical precision. Between impacts, standard time-discretization schemes for differential algebraic equations are used. Timestepping schemes do not resolve the time of the impacts, but include their possible appearances during the simulation in the discretization of the total set of relevant equations. Because of their low approximation order, they may turn out to be inefficient when there are only a few impact events. Event-driven schemes may get inefficient for many impact events, in particular they cannot handle the case of infinite impacts in a finite time interval.

For each time-step of the time-discretization schemes, constraint equations have to be solved. We use a specific formulation for the constraint equations, that is, based on projections onto convex sets. In Chapter 2, we give an overview about different formulations for the constraint equations. We explain their application area and numerical solution methods. The goal of this chapter is to categorize the solution method, which is used in this work. The text is partly taken from the publication [165] of the author.

Chapter 3 completes the first part about the nonsmooth framework by presenting different software codes, where ideas about nonsmooth mechanical systems and their numerical representation have been implemented successfully. We discuss the codes MBSim and SICONOS, which have been developed in Munich and Grenoble. To some extent, the text is taken from the publication [162] of the author.
1. Motivation

The next part is about consistent time-integration schemes for nonsmooth mechanical systems. This is the main part of the work and consists of several chapters, including the appendix. The findings are original research and are partly cited from the publications [161, 171, 167, 154] of the author.

We start with the consistent and robust concept of timestepping schemes for semi-discrete mechanical systems. The main drawback is the lack of problem adaptive accuracy in non-impulsive intervals. We show how timestepping schemes can be embedded within the context of time-discontinuous Galerkin methods, when we choose piecewise constant trial functions for the velocity approximation (Chapter A). The use of higher-order but piecewise continuous trial functions for the velocity, together with the splitting of impulsive and non-impulsive contact reactions, offer the opportunity to both stay consistent and benefit from a higher-order integration of non-impulsive contact reactions. We explain the idea with a specific time-discretization scheme from the resulting concept (Chapter B), that is, based on piecewise linear velocity trial functions - concerning function evaluations and implementation complexity, it is the easiest case and called "forecasting trapezoidal rule". We start writing the constraint equations for non-impulsive reactions on acceleration level (Chapter 5). However for an automatic switching between non-impulsive and impulsive intervals, constraint equations for non-impulsive reactions are also formulated on velocity level, like the constraint equations for contact impulse and jumping velocity (Chapter 6). We focus on half-explicit methods and evaluate all magnitudes, but the constraints, explicitly (Chapter 7). Constraint equations are solved separately by semi-smooth Newton methods. Finally, we apply the schemes to nonlinear and flexible multi-contact examples with friction and explain their behavior (Chapters 8-9 and Chapter C).

In the case of finite element applications, the base integration scheme may get unstable without additional damping because of its half-explicit nature. For this reason, we derive different timestepping methods based on the general framework of impulsive corrections and non-impulsive base integration schemes but leave the time-discontinuous Galerkin setting. Often high artificial frequencies due to finite element discretizations disturb the interpretation and realization of results. These frequencies may not be removed by reduction methods because of variable contact geometries. We use base integration schemes like the generalized-α method [41], the Bathe-method [16] and the ED-α method [25] within mixed timestepping schemes [58, 57, 1]. These are accurate in the low-frequency domain but damp in the high-frequency domain. Hence, they act like a filter. Finally, we compare the schemes within nonlinear multi-contact examples with friction (Chapters 10-12 and Chapters D-G).

Within the time-discretization schemes discussed so far, the impact law and the non-penetration constraint are solved on the velocity level. This is a consistent approach, but results in a drift-off effect. We show exemplary, how to achieve not only compliance of the impact law but also of the non-penetration constraint. Therefore, we discuss coordinate and derivative projection methods, in particular the Gear-Gupta-Leimkuhler method. The latter is even energy-consistent and can
be interpreted as a symmetric projection method depending on the underlying
time-discretization approach (Chapters 13-15).

The new time-discretization schemes are introduced step-by-step. We use an impact-
sliding slider-crank mechanism to demonstrate their drawbacks and possible remedies
(Chapters 4, 10, F).

The manuscript ends with a conclusion and an outlook part including a summary of
the content (Chapter 16). Final remarks about the essence of the presented results
are given in Chapter 17. There, we also gain a perspective on open problems.

1.2. Industrial Applications of Nonsmooth Mechanical Systems

At the Institute of Applied Mechanics of the Technische Universität München (AM-
TUM), several questions from practical applications have been answered successfully
with methods from nonsmooth mechanics [143, 199]. Also at Inria Bipop, many
projects dealing with nonsmooth dynamical problems have been treated [5]. In this
section, we present some real world models, which involve nonsmooth techniques.

1.2.1. Pushbelt Continuously Variable Transmission

At AM-TUM, the modeling of pushbelt Continuously Variable Transmissions (CVT)
has been a major application in the last years. The objective is the simulation and
the understanding of the transmission system depicted in Fig. 1.6. The system
consists of two pulleys as well as a pushbelt with about 400 elements and two layers
of nine to twelve steel rings each. Power is transmitted from one pulley to the other
by friction forces between the pulley sheaves and the elements, push forces between
the elements and friction between the elements and the ring packages.

Figure 1.6.: Pushbelt variator and pushbelt with elements [33, Figure 1.1].

The cooperation with Bosch started with the doctoral dissertation of BULLINGER [33],
where the whole pushbelt is modeled as a beam like structure with time-depending
density because of the relative motion of the elements. The flexibility of the pulley sheaves is modeled with a force element taking into account the first quasi-static mode of the sheave and shaft deflections. An Eulerian view is used taking care of the periodicity for a fixed geometric transmission ratio. Hence, the detailed element motion and, in particular, its contact behavior are approximated in a rough but very elegant way. Beginning with Geier [79, 164], the interest in the local contact behavior and so the modeling depth increased. Each element is modeled as a planar Rigid Body. The rings are summarized to one virtual ring package, which is modeled with a planar co-rotational beam finite element method [195]. Frictional contacts are considered in-between elements, between elements and the virtual ring package and between elements and the pulley sheaves, which again are modeled with the force element from Bullinger [33, 88]. This is a highly nonlinear setting from nonsmooth mechanics with about 1500 degrees of freedom and 3500 contacts. In a third step, spatial effects have been considered by Schindler [159, 169]. Conceptionally, the path of detailed modeling from [79, 164] is canonically extended [159, Table 2.2]. About 3500 degrees of freedom and 5500 contacts create a challenging benchmark problem for nonsmooth mechanical systems [168, 170, 163]. For the preparation of an extended validation, some extensions of the models in [79, 164, 159, 169] have been described by Cebulla [36, 37]. The extended validation project with global and local measurement data of the project partner for planar and spatial analysis is addressed by Grundl [87]. Starting with [159, 169], the simulation framework MBSim [129, 162] (Section 3.2) has been used to set up the nonsmooth mechanical models.

1.2.2. Rotor Dynamics

The monitoring of rotor systems has been treated in recent projects at AM-TUM. The model of Ginzinger [80, 81] has also been built-up with MBSim [129, 162] (Section 3.2). It has been validated with rotor test rigs in Munich and Bath. A simplified rotor model is shown in Fig. 1.7. It is used for validation of time-stepping schemes in Chapter 9. For this model, backlash and friction occur in the journal bearing. It can be considered as a safety bearing or an auxiliary bearing, which is often used in rotor systems with magnetic bearings [80, 81].

1.2.3. Elastohydrodynamics

In elastohydrodynamics, we are interested in the detailed analysis of tribological effects, like friction, lubrication and wear. For appropriate modeling, we couple the mechanical system to a fluid system, which gives us the necessary tribological information. Usually we would like to focus on key effects. Hence, we have to think about efficient but complete coupling methods, which return one-way or two-way interactions. In some cases, we might want to derive more accurate contact models, sometimes we want to know more about the lubrication itself. However, the starting point is a fluid-structure interaction, which we have to model, e.g., like
it was done by Krinner et al. [113]. Figure 1.8 shows typical applications of elastohydrodynamic contacts, these are, lubricated journal bearings and bearing shells, even with elastic deformation effects. Both examples comprise conformal contact situations with a thick fluid film and moderate peak loads [188]. They are ongoing research at AM-TUM. Counterformal contact situations with thin fluid films and high peak loads naturally occur in chain/sprocket contacts within timing-drives (cf. the dissertation of Filippi [66, 65]). These contacts are also called Hertzian contacts [188]. Staying with this traditional thinking, nonsmooth modeling may be either used for the representation of the fluid pressure, which has to satisfy some non-negativity condition, or for the modeling of counterformal contact situations. Therefore, the techniques developed in [123, 124] may be useful to efficiently describe the wave propagations inside the mechanical systems.

Figure 1.7.: Model of a rotor test rig [167].

Figure 1.8.: Rigid rotor in elastic lubricated journal bearing (left) [113, Figure 2] and squeezing machine with elastic lubricated bearing shell (right) [113, Figure 6].
1.2.4. Belt Drives and Timing-Drives

If we seize the counterformal contact situations from the previous section, timing-drives will be a prime example to apply modeling techniques from nonsmooth mechanics. Figure 1.9 shows an Audi-V6 timing-drive with about 250 frictional contacts in the chain strands and frictional contacts between chains/sprockets as well as between chains/guides. Also the postprocessed treatment with acoustics tools builds up on these considerations. Zander et al. [198] model belt drives with MBSim [129, 162] (Section 3.2) on the one hand, and with SIMPACK\footnote{http://www.simpack.com/} on the other hand. It is shown, how nonsmooth contact modeling can be considered as the rigid limit of contact modeling with stiff springs, which is usually done in industrial software codes. Both applications have been undertaken at AM-TUM.

Figure 1.9.: Audi-V6 timing-drive [65, Figure 1].

1.2.5. Surgical Assistance Systems

In surgery, new techniques are often developed and analyzed in large university hospitals. The know-how is concentrated in few experts. It is difficult to transfer and offer these techniques to non-expert surgeons and patients all over the world. Telepresence robots may help to improve this process. The expert may do the surgery using a control desk or an interface at one end of the world, with a robot manipulator performing the real tasks at another end of the world. A robot manipulator, which is constructed and prepared by expert knowledge, may also assist non-expert surgeons...
to do complicated interventions. Then, the control desk, which is used by the non-expert surgeon, and the manipulator are installed at the same place. For both application cases, it is necessary to develop manipulators, which transfer motion and forces using a control desk. Transmission takes time, the transfer behavior itself may be difficult and there may be no (haptic) feedback to the operator. Figure 1.10 shows such a manipulator for gastrointestinal surgery. At AM-TUM, HORST has undertaken investigations in this area [96, 95]. The slave manipulators are actuated by bowden controls, which run inside the endoscope. Stick-slip phenomena occur at the slave manipulators, complicating the handling for the operator and requiring high expertise for this specific tool. Simulations have been performed to analyze and improve the transfer behavior, also using the framework of nonsmooth mechanics.

![Figure 1.10: Robot for gastrointestinal surgery [148, Figure 5].](image)

### 1.2.6. Robotics

In robotics, there are many applications which involve nonsmooth phenomena. For the understanding of walking and running of biped robots, frictional contact situations to the ground are analyzed. Many research teams deal with walking, running and beyond that. At AM-TUM, the investigation of human-like walking is a major topic: two biped robots, Johnnie and Lola, have been developed in the last years. Essential research results are summarized by BUSCHMANN ET AL. [34]. Passive walking was the topic of FÖRG [71] together with the Locomotion Laboratory (now in Darmstadt), and with the group for Optimization in Robotics and Biomechanics in Heidelberg. We mention also the Biorobotics and Locomotion Lab of the Cornell University and Inria Bipop. Following [122], the interest for simulation increased to gain knowledge about walking and running with prostheses as well as the symbiosis of robot and human.

Another example for methods from nonsmooth mechanics being helpful in robotics is grasping [194] (Fig. 1.11). Both normal contact forces and tangential contact forces
with stick-slip transitions are simulated carefully to allow reliable predictions for stable grasping.

The relevance of robotics research questions has been documented by the funding of the National Robotics Initiative\(^2\) in the United States and the IEEE Robotics & Automation Society Technical Committee on Model-Based Optimization for Robotics\(^3\).

### 1.2.7. Computer Graphics

Nonsmooth modeling techniques for computer graphics can be found in the work of Baraff\(^1\), perhaps for the first time. The author applied complementarity formulations to computer-animated films. Today, the techniques for computer-animation and for physics-based simulation converge. Computer-animations are oriented towards augmented-reality. Physics-based simulations should answer questions concerning construction and dimensioning within a reasonable and tolerable time. Real-time applications even increase the demand for a response within a strict time constraint. On the contrary, the realistic impression of computer-animations is achieved by simplified physics-based models. Figure 1.12 shows the simulation of hair dynamics at Inria Bipop, both, for computer-animation and for partners from the shampoo industry\(^2\). Each hair is modeled with beam finite elements. Hair may contact and "glue" together due to friction. Hence, hair dynamics is a nice example for modeling techniques from nonsmooth mechanics. Another example is crash simulation.

### 1.2.8. Granular Material

Carrying sand in the construction industry, driving on sand-like terrain, sintering powder, sediment transport (in porous media) are just some examples to show that the

---

\(^1\) Baraff

\(^2\) nsf.gov/nri

\(^3\) http://www.ieee-ras.org/model-based-optimization
1.2. Industrial Applications of Nonsmooth Mechanical Systems

Figure 1.12.: Hair dynamics [23, Figure 2].

Simulation of granular material or granular matter becomes more and more important. In modeling granular material, the shape of the single elements is kept simple, that is, they are represented by balls (Fig. 1.13). Billions of them are in contact and are simulated at the same time on high-performance clusters containing conventional multi-core processors but also graphical processing units [189, 151, 110]. At present, methods from nonsmooth mechanics and the discrete element method compete concerning efficiency and reliability. When applying methods from nonsmooth mechanics to large but conceptionally easy applications like granular material, usually we do not solve exact frictional problems. Approximations are solved to benefit from efficient formulations [151, 110, 128].

Figure 1.13.: Shaking bodies [189, Figure 7].

1.2.9. Electromechanical Systems

Due to the vicinity to Schneider Electric⁴ and EDF⁵, Inria Bipop has been working on a bunch of applications in the area of electromechanical systems [3]. Circuit breakers

⁴ http://www.schneider-electric.com/
⁵ http://www.edf.com/
detect faults and interrupt the current flow to protect an electrical circuit from damage. They are available for low-current (several Ampere) to high-current circuits (several Mega Ampere) also including arc flash protection. Different construction types exist depending on the area of application. Magnetic circuit breakers, for example, create a magnetic force proportional to the amount of current. For a threshold value, the magnetic force opens a mechanical contact and stops the current flow. This process includes electro-magnetic but also mechanical modeling. In particular, the accurate representation of mechanical contacts, which may open and close rapidly in time, is necessary.

Diodes are electrical switches which let pass current flow in one direction and block current flow in the other direction. Ideal diodes behave similar to non-penetration conditions in mechanics. Their resistance in one direction is zero and in the other direction is infinite. Figure 1.14 shows the nonsmooth characteristics of an ideal diode with voltage $v$ and current $i$. In electrical circuits, diodes are used, e.g., for over-voltage protection or to convert alternating current (AC) to direct current (DC).

![Characteristics of an ideal diode.](image)

As real systems are nonlinear by nature, one may introduce nonlinearity also in feedback control systems. Perhaps the easiest extension of a linear control system is a switching control system. The system dynamics changes depending on different sets in the state space. The sets are separated by a border. If the vector fields in the vicinity of the border direct to each other, a stable sliding mode on the sliding surface, that is, the border, results. Controlling the sliding mode is nonsmooth by definition [149].

1.2.10. Biological Systems

Gene regulatory networks are used to reproduce and explain the cell response with respect to the environment. There are many models available to represent the dynamics of gene regulatory networks. An easy approximate model, which covers regulatory effects depending on switching protein concentrations, is based on piecewise-linear differential equations (Fig. 1.15). ACARY [4] presents such a modeling of the regulation functions as step functions. The dynamics can still be treated
qualitatively, which allows the computation of steady states and stability. From a theoretical point of view, a nonsmooth treatment is necessary due to discontinuous right-hand sides similar to the sliding mode control from the previous section.

![Gene regulatory network](image)

**Figure 1.15.** Gene regulatory network [4, Figure 1].

### 1.3. Nonsmooth Dynamical Systems

So far, we have discussed the example of the bouncing ball in detail at the beginning of this chapter. We have also seen several application areas of nonsmooth dynamical systems, so that we know that a nonsmooth dynamical system involves discontinuities of its state variables or their derivatives. Depending on the type of dynamical system, these discontinuities concern the dynamics more or less. In the case of the gene regulatory networks, for example, only the right hand side is discontinuous. As a consequence, velocity and position variables, that is, the state, are still continuous. For a mechanical system, this means that the accelerations would be discontinuous. This happens for frictional mechanical systems with stick-slip and slip-stick transitions. One or the other time-integration scheme does not like such a phenomenon. Moreover in mechanics, we deal with more severe discontinuities, namely with impacts: the velocity variables jump, that is, they are discontinuous. Hence, there is no force which can compensate such a velocity jump, but a Dirac measure. No standard time-integration scheme will handle such a discontinuity. We have to introduce a second set of equations, the impact equations, which involve impulsive interactions as right-hand side contributions. We have discussed this for the bouncing ball example. Fortunately in mechanics and in all the application areas described above, there is no example for jumping position coordinates, to the best of the knowledge of the author. Hence, the bouncing ball (Fig. 1.4) is a typical example of a nonsmooth dynamical system. In this section, we want to generalize its underlying structure for arbitrary nonsmooth mechanical systems.

### 1.3.1. Equations of Motion

The following text is based on [166]. For non-impulsive motion in time, a mechanical system is described by

\[
\dot{q} = Y v ,
\]  

(1.1)
and
\[
M \dot{v} = h + W_N \lambda_N + W_T \lambda_T .
\] (1.2)

These equations typically hold after semi-discretization in space of respective mathematical models, e.g., by the finite element method [52]. We want to calculate the generalized position and generalized velocity functions on the time interval \( I := [0,T] \) with \( T > 0 \):

\[
q : [0,T] \to \mathbb{R}^{N_d}, \ t \mapsto q(t) ,
\]

\[
v : [0,T] \to \mathbb{R}^{N_d}, \ t \mapsto v(t) .
\]

The generalized mass matrix satisfies
\[
M : \mathbb{R}^{N_d} \to \mathbb{R}^{N_d,N_d}, \ q \mapsto M(q)
\]

and the nonlinear generalized force has the structure
\[
h : \mathbb{R}^{N_d} \times \mathbb{R}^{N_d} \to \mathbb{R}^{N_d}, \ (q,v) \mapsto h(q,v) .
\]

With (1.1), we have the opportunity to choose generalized positions and generalized velocities in a physically intuitive way. In particular, the vector \( v \) may contain angular velocities, whereas the vector \( q \) may contain arbitrary parameters to describe rotations. Then,

\[
Y : \mathbb{R}^{N_d} \to \mathbb{R}^{N_d,N_d}, \ q \mapsto Y(q)
\]

is a transformation matrix, which in general may be rectangular. However, we assume it to be square, that is, we assume minimal parametrizations of rotational relationships. The matrices of generalized force directions

\[
W_N : \mathbb{R}^{N_d} \to \mathbb{R}^{N_d,N_c}, \ q \mapsto W_N(q) ,
\]

\[
W_T : \mathbb{R}^{N_d} \to \mathbb{R}^{N_d,N_c}, \ q \mapsto W_T(q)
\]

are sometimes also called wrench matrices. They define the normal and tangential directions for \( N_c \) constraint reactions concerning the generalized velocities. The Lagrange multipliers

\[
\lambda_N : [0,T] \to \mathbb{R}^{N_c}, \ t \mapsto \lambda_N(t)
\]

can be identified with normal contact forces in the equations of motion (1.2). The Lagrange multipliers

\[
\lambda_T : [0,T] \to \mathbb{R}^{2N_c}, \ t \mapsto \lambda_T(t)
\]

can be identified with spatial tangential contact forces in the equations of motion (1.2).
A nice example for the splitting in generalized force directions $W_N$ and $W_T$ is the sledge with mass $m$ in Fig. 1.16. We choose the Cartesian position $\mathbf{8}$ as generalized position and its absolute time-derivative as generalized velocity. As the sledge moves on a wavy hill due to gravity $\Gamma$, a constraint equation, $g_N(\mathbf{8}) = 0$, has to be satisfied all the time. Thereby, we assume that the sledge cannot lift off. The normal contact force is splitted into its generalized direction $W_N$ and a normal force parameter $\lambda_N$. We get a similar splitting for the tangential contact force into the generalized direction $W_T$ and the tangential force parameter $\lambda_T = \mu \lambda_N$, which involves the friction coefficient $\mu$.

With the initial conditions
\begin{align*}
q(0) &= q_0, \quad (1.3) \\
v(0) &= v_0, \quad (1.4)
\end{align*}
the evolution of $q$ and $v$ is non-impulsive almost everywhere due to (1.1)-(1.2). However if one does not use any classic concept of regularization by introducing contact stiffnesses or contact potentials [197, 64], for countable time instances $t_j$, the evolution of the mechanical system might get impulsive and we use
\[
M(q(t_j)) \left( v_j^+ - v_j^- \right) = W_N(q(t_j)) A_{N,j} + W_T(q(t_j)) A_{T,j} \quad (1.5)
\]
instead of (1.2). The impact equations (1.5) lead to jumps in the velocity variables. Their derivatives do not exist anymore in the classical sense. One has to define the left-hand and right-hand limits
\[
v_j^- := \lim_{t \uparrow t_j} v(t) , \quad v_j^+ := \lim_{t \downarrow t_j} v(t) .
\]
Then, the Lagrange multipliers

\[ \Lambda^N_j = \lim_{\delta \to 0^+} \int_{t_j}^{t_j + \delta} \lambda^N dt, \quad \Lambda^T_j = \lim_{\delta \to 0^+} \int_{t_j}^{t_j + \delta} \lambda^T dt \]  

describe the finite impulsive interaction of normal and tangential contacts in the sense of distributions.

1.3.2. Contact and Impact Laws

The contact reactions in (1.2) and (1.5) are unknown. Hence, we have just two equations for the calculation of the unknowns \( q, v, \lambda^N, \lambda^T \) in non-impulsive periods and two equations for the calculation of the unknowns \( q, v^+_j, \Lambda^N, \Lambda^T \) in impulsive periods. In this section, we state the missing contact and impact laws as additional relations. They meet the role of constitutive laws within the mechanical description.

We start with the non-impulsive motion. The nonlinear normal and tangential gap functions

\[ g^N : \mathbb{R}^N \to \mathbb{R}^N, \quad q \mapsto g^N(q), \] 
\[ g^T : \mathbb{R}^N \to \mathbb{R}^{2N}, \quad q \mapsto g^T(q) \]

describe normal distances and tangential displacements of contacting bodies in a mechanical system and are therefore constraining the overall motion. The path of the sledge in Fig. 1.16 describes such a constraint. In general, a constraint may also depend explicitly on time, that is, it is rheonomic. However, we assume constraints to be holonomic-scleronomic, that is, they depend only on the generalized position \( q \) [147]. The matrices of generalized force directions \( W^N \) and \( W^T \) can be calculated as the derivatives of the normal and tangential gap functions with respect to \( q \). They define normal and tangential directions to the constraint surfaces. We normalize these directions, which corresponds to scaling of the gap functions. Then, values of the Lagrange multipliers \( \lambda^N, \lambda^T \) match the values of the normal and tangential contact forces, whereas \( W^N, W^T \) give their directions. We want to avoid any penetration in normal direction of a contact:

\[ \mathbf{0} \leq g^N \perp \lambda^N \geq \mathbf{0}. \]  

Relation (1.9) is called non-penetration unilateral contact condition or Signorini-Fichera-condition. The normal gap functions \( g^N \) is not allowed to get negative, push forces \( \lambda^N \) originate from this constraint. The push forces vanish if there is an open or in-active contact, that is \( g^N > 0 \). If the push forces are positive, the contact has to be closed or active, that is \( g^N = 0 \). This complementarity relation is demanded by \( g^N \perp \lambda^N \), that is \( g^N \lambda^N = \mathbf{0} \) entry-by-entry (left side of Fig. 1.17). The vertical part of the relation in Fig. 1.17 defines a set-valued behavior. The normal contact force is not a function of the normal gap function, but there is just a
relation between both. The normal contact law (1.9) and the equations of motion (1.1), (1.2) have to be solved simultaneously to get a solution for $q$, $v$ and $\lambda_N$.

\[
\begin{align*}
\lambda_N &\geq 0, g_N = 0 \\
\lambda_N &= 0, g_N \geq 0
\end{align*}
\]

\[
\begin{align*}
\lambda_T &= \pm \mu \lambda_N \\
\dot{g}_T &= 0
\end{align*}
\]

Figure 1.17.: Contact laws [154, Figure 3].

In tangential direction of a contact, the relation (1.10) contains sticking and sliding conditions with the friction coefficient $\mu \geq 0$ and the time-derivative denoted by a dot:

\[
\begin{align*}
\|\lambda_T\| \leq \mu \lambda_N & \quad \text{for } \dot{g}_T = 0 \land g_N \leq 0, \\
\lambda_T = -\frac{\dot{g}_T}{\|\dot{g}_T\|} \mu \lambda_N & \quad \text{for } \dot{g}_T \neq 0 \land g_N \leq 0.
\end{align*}
\]

Coulomb’s friction law (1.10) (right side of Fig. 1.17, for the planar case) involves only local tangential velocities $\dot{g}_T$ and couples them with tangential contact forces $\lambda_T$. The vertical sticking part again describes a set-valued behavior. The corresponding sticking force $\lambda_T$ has to be determined together with the position $q$ and velocity $v$ in the equations of motion (1.1), (1.2). For the applicability of Coulomb’s friction law, we first have to check that the respective contacts are closed. We are allowed to take into account just the contacts which belong to the continuous index set

\[
\mathcal{I}_q^* = \{ k \in \mathcal{I}_0 : g_{N_k}(q) \leq 0 \}
\]

of the closed constraints. Thereby, the index set $\mathcal{I}_0$ contains all constraints on the system. Closed contacts behave similar to bilateral constraints, which satisfy $g_N = 0$ all the time (Fig. 1.18). However for bilateral constraints, also tensile forces $\lambda_N < 0$ may occur - there is no assumption for $\lambda_N \in \mathbb{R}$ being a push force. That is why for the example of the sledge, we postulate a unilateral contact, which is active all the time (Fig. 1.16).

When we have solved the bouncing ball example at the beginning of this chapter, we have seen that the constraints due to the normal and the tangential contact relations hide information. In particular, the equations of motion (1.2) include information on the acceleration and on the contact forces, whereas (1.9) and (1.10) include information on the contact forces, as well as on position and velocity. To combine these information, we check the possible motion from the position level to the velocity level until we finally arrive at the acceleration level of the contact laws.
For the contacts, which belong to the continuous index set of the closed constraints
\[ I_q^{v^+} = \{ k \in I_q : g_{N_k}(q,v^+) \leq 0 \} , \tag{1.12} \]
which stay closed, we can finally merge the equations of motion (1.2) and the normal contact law on acceleration level:
\[ 0 \leq \ddot{g}_N \perp \lambda_N \geq 0 . \]

Similarly, we argue for the tangential contacts. We just check closed constraints, which stay closed. Then, we check sticking and sliding on velocity level - this is simply an evaluation because we know the state of the mechanical system. As a result, we get the index set
\[ I_{q,v^+}^T = \{ k \in I_{q,v^+}^T : \|\dot{g}_{Tk}(q,v^+)\| \leq 0 \} \tag{1.13} \]
of the rolling/sticking constraints. On acceleration level, we calculate equivalent forces:
\[ \begin{cases} \|\lambda_T\| \leq \mu \lambda_N & \text{for } \dot{g}_T = 0 , \\ \lambda_T = -\frac{\dot{g}_T}{\|\dot{g}_T\|} \mu \lambda_N & \text{for } \dot{g}_T \neq 0 . \end{cases} \tag{1.14} \]

As we can see, we arrive at a large combinatorial problem. For a time-discretization, it would be of advantage to avoid some of the tests. We will see in Chapter 2, which strategies exist.

If \( g_{N_k} (q(t_j)) = 0 \) for some component \( k^* \), but \( g_{N_k^*} (q(t)) > 0 \), for \( t_j - \delta \leq t < t_j \) with an appropriate \( \delta > 0 \), the mechanical system gets impulsive and we have to solve the impact equations (1.5). As in the non-impulsive case, these are not sufficient to calculate the overall dynamics. We need some constitutive laws to complete the description. There are many impact laws in the literature, which describe the pre- and post-impact behavior. NEWTON’s impact law is the best-known impact law:
\[ \begin{cases} \|\lambda_T^+\| \leq \mu \lambda_N & \text{for } \dot{g}_T^+ = 0 , \\ \lambda_T^+ = -\frac{\dot{g}_T^+}{\|\dot{g}_T^+\|} \mu \lambda_N & \text{for } \dot{g}_T^+ \neq 0 . \end{cases} \tag{1.15} \]
whereby we usually use the abbreviations

\[
\begin{align*}
\dot{g}_{Nj}^+ &:= \dot{g}_{Nj}^- + \varepsilon_N \dot{g}_{Nj}^- , \\
\dot{g}_{Tj}^+ &:= \dot{g}_{Tj}^- + \varepsilon_T \dot{g}_{Tj}^- .
\end{align*}
\]

(1.16)  

(1.17)

It involves the coefficient of restitution \(\varepsilon_N \in [0,1]\) for the normal direction and the coefficient of restitution \(\varepsilon_T \in [0,1]\) for the tangential direction. Kinematically, it relates pre- and post-impact velocities. For \(\varepsilon_N = 0, \varepsilon_T = 0\), the impact is plastic, for \(\varepsilon_N = 1, \varepsilon_T = 1\), the impact is totally elastic. An impact leads to jumps in the velocity variables. Their derivatives do not exist anymore in the classical sense. One has to define the left-hand and right-hand limits

\[
\begin{align*}
\dot{g}_{Nj}^- &:= \lim_{t \downarrow t_j^-} \dot{g}_{Nj}^- (t) , \\
\dot{g}_{Tj}^- &:= \lim_{t \downarrow t_j^-} \dot{g}_{Tj}^- (t) , \\
\dot{g}_{Nj}^+ &:= \lim_{t \uparrow t_j^+} \dot{g}_{Nj}^+ (t) , \\
\dot{g}_{Tj}^+ &:= \lim_{t \uparrow t_j^+} \dot{g}_{Tj}^+ (t) .
\end{align*}
\]

The Lagrange multipliers \(\Lambda_N, \Lambda_T\) are defined as an integral of \(\lambda_N, \lambda_T\) according to (1.6). As Newton’s impact law represents just a subset of all post-impact velocities, extensions to multiple impacts exist with Fremond’s non-diagonal restitution matrix [74]. It couples pre- and post-impact velocities of different contacts. However, this algebraic approach might lead to wrong results in the non-plastic case [85]. Poisson’s impact law relates pre- and post-impact impulses by considering compression and extension phases also for multiple impacts [145]. Hence, it is of kinetic type. Poisson’s impact law is still relatively easy to apply and gives much better results than Newton’s impact law. However, it may also increase the energy for more than one frictional contact [86]. Recently, the LZB-law has been proposed by Liu, Zhao and Brogliato [123, 124]. It includes the evolution of energy in terms of compression and expansion phases similar to Poisson’s impact law. The impact law uses the setting of Darboux-Keller [109], that is it relates to the normal impulse during the impact like Poisson’s law. The LZB-law, like Poisson’s law, is based on the first order dynamics of the impact, which is solved by an event-driven integration scheme. In addition, it covers the dispersion of the impacting wave by considering stiffness and energy saving ratios. A general overview as well as an application and validation of the LZB-law to granular chains can be found in [137]. The book of Johnson [105] is also a good reference for contact and impact laws. In particular, it contains second order dynamics for impacts, that is, compliant models. Stronge’s book [184] introduces the energetic restitution coefficient by comparing the total energy before and after the impact.

When they are connected by set-valued force laws, not only the initiating contact has to be considered but all closed contacts and all bilateral constraints have to be evaluated. In Fig. 1.19, the impulsive force \(A^{(1)}\) is propagated to all Joints. However in Fig. 1.20, the spring between the first and the second Body only accepts single-valued and not-impulsive forces. It stops the impulsive force propagation and initiates a transformation to finite forces in the multibody system. Note that a damper leads to jumps in accelerations. To ensure the validity of the bilateral
constraint after collision time, the impact law states
\[ \dot{g}_{Nj}^+ = 0. \]

1.3.3. Contour Description

The equations of motion (1.1)-(1.2), as well as the contact laws (1.9) and (1.10), contain generalized variables \( q, v \) and local variables \( g_N, g_T, \dot{g}_N, \dot{g}_T, \lambda_N, \lambda_T \). The same observation is true for the impulsive equations: the overall problem is formulated as a mixed problem with local and generalized variables. An additional kinematics
law gives the interconnection between both types of variables. The kinematics law is formulated by assigning a contour to each Body. The Cartesian surface description of a Body is parametrized by generalized coordinates and contour parameters:

\[ \mathbb{N} : \mathbb{R}^{N_d} \times \mathbb{R}^2 \to \mathbb{R}^3, (q,s) \mapsto \mathbb{N}(q,s) . \]

Hence, a specific point on the surface of a Body changes due to its motion defined by the generalized coordinates \( q \). Further, specific points may move on the surface of a Body due to a change of the contour parameter \( s \).

![Figure 1.21: Contact distance problem for possible contact points [160, Figure 6.1].](image)

Regarding Fig. 1.21, we agree to the convention that each point \( P \) on the surface of a Body is accompanied by a right-handed trihedral with an outward pointing normal vector and two tangent vectors:

\[ \mathbf{n} : \mathbb{R}^{N_d} \times \mathbb{R}^2 \to \mathbb{R}^3, (q,s) \mapsto \mathbf{n}(q,s) , \]
\[ \mathbf{t}_1 : \mathbb{R}^{N_d} \times \mathbb{R}^2 \to \mathbb{R}^3, (q,s) \mapsto \mathbf{t}_1(q,s) , \]
\[ \mathbf{t}_2 : \mathbb{R}^{N_d} \times \mathbb{R}^2 \to \mathbb{R}^3, (q,s) \mapsto \mathbf{t}_2(q,s) . \]

These three vectors define a Cartesian coordinate system, which is floating on the surface due to the contour parameter \( s \). Recall that we want to avoid penetration of Bodys. That is why we look for nearest points on the two Bodys and are interested in finding pairs which minimize the gap distance:

\[ g_N = (\mathbf{n}^{(1)})^T \mathbb{N}_D . \]

For a minimal gap distance, we are necessarily in the situation of Fig. 1.22. We assume that the surfaces of the Bodys are as differentiable as it is necessary to calculate candidates according to the conditions of the contact distance problem:

\[ (\mathbf{t}_1^{(1)})^T \mathbb{N}_D = 0 , \]
\[ (\mathbf{t}_2^{(1)})^T \mathbb{N}_D = 0 , \]
Figure 1.22.: A solved contact distance problem [160, Figure 6.3].

\[ (t_1^{(2)})^T N_D = 0 , \]
\[ (t_2^{(2)})^T N_D = 0 . \]

After having solved the nonlinear equations, e.g., with NEWTON-type algorithms [49], we get pairs, \( \{ (s_i^{(1)}, s_i^{(2)}) \} \), which represent possible contact points. We have to select the pair which corresponds to the minimal gap distance. Consequently, we decide if the specific contact situation is active or not active, according to Fig. 1.23. If it is active, that is, closed, we solve for the specific contact forces or impacts. Hence, we consider the contact as a constraint in the mechanical equations according to Section 1.3.2. This also means that, at least, the normal gap velocity

\[ \dot{g}_N = (n^{(1)})^T (v_P^{(2)} - v_P^{(1)}) + (\dot{n}^{(1)})^T N_D = 0 \]

Figure 1.23.: A contact may be open, touching or penetrating [160, Figure 6.4].
and the tangential gap velocities

\[ \dot{g}_T = \left(t^{(1)}\right)^T \left(v_P^{(2)} - v_P^{(1)}\right) + \left(t^{(1)}\right)^T N_D = 0 \]

have to be calculated. At least for analytical calculations, also accelerations are necessary:

\[ \ddot{g}_N = \left(n^{(1)}\right)^T \left(\dot{v}_P^{(2)} - \dot{v}_P^{(1)}\right) + \left(n^{(1)}\right)^T \left(v_P^{(2)} - v_P^{(1)}\right), \]

\[ \ddot{g}_T = \left(t^{(1)}\right)^T \left(\dot{v}_P^{(2)} - \dot{v}_P^{(1)}\right) + \left(t^{(1)}\right)^T \left(v_P^{(2)} - v_P^{(1)}\right). \]

In the setting of these kinematic relationships, the wrench matrices \( W_N \) and \( W_T \) are projections of the Cartesian normal and tangential directions of the contact reactions in the space of generalized velocities of the involved Bodys by JACOBIAN matrices concerning the force application points [160].

1.3.4. Abstract Framework

Equations (1.1), (1.2), (1.9) and (1.10) as well as (1.5), (1.14) and (1.15), together with the kinematics equations from the very last section, describe impacting mechanical systems in general. The most important realization is the occurrence of a velocity jump due to the impact. The function describing the state of position and velocity contains non-impulsive and impulsive propagation episodes. Using a description based on classical function derivatives, one has to distinguish these two propagation episodes. However, with appropriate mathematical objects, namely measures, it is possible to enter the modern theory of nonsmooth dynamical systems. We pick the description from [161].

**Problem 1.2 (Measure differential inclusion)** Solve the initial value problem

\[ q(0) := q_0, \]

\[ v(0) := v_0, \]

\[ dq = Y \cdot \begin{array}{l} v \end{array} dt, \]

\[ dv = M^{-1}hd\tau + M^{-1}di \]

in terms of measures.

Problem 1.2 defines a consistent generalization, not distinguishing between non-impulsive and impulsive motion. Using this setting, existence and uniqueness of solutions may be analyzed. Thereby, MOREAU [131, 132] gives an exhaustive, not only historic, introduction. A description with the theory from functional analysis and measure theory may be found in the book of MONTEIRO MARQUES [126] and in the one of ZIEMER [200]. BALLARD [14] treats mechanical systems without
friction, but with impacts. More applied but still providing theoretical results is Glocker’s book [84] about general mechanical systems with impacts and friction. Leine books [120, 121] discuss bifurcation and stability of mechanical systems with impacts and friction. A review may be found in Brogliato’s publications [30, 31] and in Trinkle’s review paper [193]. Acary’s book [3] gives an overview about nonsmooth dynamical systems including their numerical treatment, also depending on the degree of nonsmoothness. Impacts are the most severe nonsmoothness inside mechanical systems: they force velocities, that is, part of the state variables, to jump. Stewart [179, 180] considers them as index 2 dynamical systems. Index 1 dynamical systems have discontinuous right-hand sides. Often piecewise systems or hybrid systems belong to this category. An example are mechanical systems with stick-slip-stick transitions, but without impacts. The solution theory is based on Filippov’s book [67]. Filippov solutions are discussed in the reviews of Cortes and diBernardo [45, 22]. Clarke [42] presents the relation to control theory. A reference for differential inclusions is Aubin’s book [13], the functional analytical background can be found in Deimling’s book [48]. Rockafellar’s book [156] gives the basics of convex analysis. Without going into detail in this work, the abstract framework for nonsmooth mechanical systems is based on the following assumptions:

- \( d \) is the Lebesgue measure of the time \( t \).
- \( q \in W^{1,1}(I) \) is the absolutely continuous position, provided with the measure \( dq = \dot{q}dt \) and the weak time derivative \( \dot{q} \), that is, the classical derivative almost everywhere according to Rademacher’s theorem.
- \( v \in LBV(I) \) is the velocity of locally bounded variation. Omitting the Cantor part of the singular measure, one can split its associated measure

\[
dv := \mathbf{Y} dt + \sum_j (v^+_j - v^-_j) \delta_{t_j}
\]

in a non-impulsive and an impulsive part with

- locally integrable accelerations \( \mathbf{Y} \in L^1_{loc}(I) \),
- countable velocity jumps \( v^+_j \) and Dirac measures \( \delta_{t_j} \).

- \( Y := Y(q) \in C^0(\mathbb{R}^{Nd}) \) is a transformation matrix.
- \( 0 < M^{-1} := M^{-1}(q) \in C^0(\mathbb{R}^{Nd}) \) is the inverse mass.
- \( h := h(q,v) \in L^1_{loc}(\mathbb{R}^{Nd} \times \mathbb{R}^{Nd}) \) is a locally integrable external force.
- \( i \in LBV(I) \) is the interaction (impulse) of locally bounded variation. Omitting the Cantor part of the singular measure, one can split its associated measure

\[
di := ri dt + \sum_j p_j \delta_{t_j}
\]
being part of

\[ (q,v,r) \in N_C, \quad (1.24) \]

countable impact relations \( (q_j, v_j, p_j) \in N_I \).

The inclusions (1.24), (1.25) are formal ways to state contact relations or other nonsmooth laws in terms of generalized variables. We can write the contact force \( r := r(q,v) \in L^1_{loc} (\mathbb{R}^{N_d} \times \mathbb{R}^{N_d}) \) as a locally integrable function. In practice, this might not always be obvious. If the contact relation is \textit{single-valued}, the contact force is a (compliant) function of position \( q \) and velocity \( v \). However, if the contact relation is \textit{set-valued}, one has to solve nonlinear/nonsmooth relations to gain \( r \). The set-valued contact relation may be bilateral, unilateral or may describe a dry friction behavior.

Problem 1.2 is a \textit{measure differential inclusion} (MDI). It uses a weak description of time derivatives in terms of measures. The connection to the separated notation for non-impulsive and impulsive periods seems to be obvious. Now, a measure differential equation in mathematics is defined in the space of distributions. This means that there is another interpretation behind Problem 1.2, which we would like to explain. To this purpose, we need some notation. Let \( \mathcal{D}(I) \) be the space of all \( C^\infty \)-functions with compact support. The set of linear functionals that maps \( \mathcal{D}(I) \) to the set of real numbers defines the dual space \( \mathcal{D}^*(I) \), which is called the space of distributions. For a distribution \( d \in \mathcal{D}^*(I) \), it is conventional to write

\[ d : \mathcal{D}(I) \to \mathbb{R}, \quad \varphi \mapsto \langle d, \varphi \rangle \]

where \( \langle \cdot, \cdot \rangle \) is the primal-dual pairing and \( \langle d, \cdot \rangle \) is the linear functional which defines \( d \). For \( f \in L^1_{loc}(I) \) (respectively a measure \( \mu \) in the set of measures \( \mathcal{M}(I) \)), a corresponding distribution \( T_f \) (respectively \( T_\mu \)) is associated such that

\[ \langle T_f, \varphi \rangle = \int_I f \varphi \, dt \quad \text{(respectively } \langle T_\mu, \varphi \rangle = \int_I \varphi \mu \text{)} \]

One makes abuse of notation by identifying \( T_f \) with \( f \), that is, \( \langle f, \varphi \rangle = \langle T_f, \varphi \rangle \) (respectively \( T_\mu \) with \( \mu \), \( \langle \mu, \varphi \rangle = \langle T_\mu, \varphi \rangle \)). The distributional derivative of a distribution \( d \) is symbolized by \( Dd \) and is usually defined by

\[ \langle Dd, \varphi \rangle := -\langle d, \dot{\varphi} \rangle, \quad \forall \varphi \in \mathcal{D}(I). \quad (1.26) \]

Hence for a function or a measure, there is an associated distribution. This allows us to re-write and interpret Problem 1.2 like in the modern theory of partial differential equations (PDE). The measure equation (1.20) is defined by

\[ \int \dot{q} \varphi_q \, dt = \int Y v \varphi_q \, dt \quad (1.27) \]
being valid for all \( \varphi_q \) such that the integrals make sense. In particular, Equation (1.27) holds for all \( \varphi_q \in \mathcal{D}(I) \). Hence, the function \( \dot{q} \) (and respectively \( Yv \)) can be identified with its distribution \( T_\dot{q} \) (respectively \( T_{Yv} \)) or with the linear functional \( \langle \dot{q}, \cdot \rangle = \int \dot{q} \cdot dt \) (respectively \( \langle Yv, \cdot \rangle = \int Yv \cdot dt \)). One interprets the constructors \( \dot{q} \) and \( Yv \) as elements of \( D^*(I) \) and writes

\[
\langle \dot{q}, \varphi_q \rangle = \langle Yv, \varphi_q \rangle \quad \forall \varphi_q \in \mathcal{D}(I),
\]

instead of (1.27).

Whereas for \( q \) there exists a weak time derivative \( \dot{q} \), the derivative \( Dv \) of \( v \) exists at least in a distributional sense. If \( v \) was absolutely continuous, the distributional derivative definition (1.26) would exactly characterize the integration by parts formula

\[
\int \dot{v} \varphi_v dt := - \int v \dot{\varphi}_v dt \quad \forall \varphi_v \in \mathcal{D}(I).
\]

Thereby, we have to take into consideration our interpretation of absolutely continuous functions as elements of \( D^*(I) \). Whereas a distributional derivative always exists, additional smoothness properties have to be proven. Then, a distributional derivative might be, e.g., a weak derivative for absolutely continuous functions interpreted as elements of \( D^*(I) \) or even a classical derivative of differentiable functions interpreted as elements of \( D^*(I) \). In fact, the distributional derivative \( Dv \) of \( v \), which is assumed to be a locally bounded variation function, is the differential measure \( dv \) and accordingly it has specific ‘smoothness’ properties, which are obviously not as strong as those of \( \dot{q} \). To enforce the notation of derivatives, we continue using the more general description \( Dv \) instead of \( dv \).

**Problem 1.3 (Distribution differential inclusion)** Solve

\[
\langle \dot{q}, \varphi_q \rangle = \langle Yv, \varphi_q \rangle \quad \forall \varphi_q \in \mathcal{D}(I), \tag{1.28}
\]

\[
\langle Dv, \varphi_v \rangle = \langle M^{-1}h, \varphi_v \rangle + \langle M^{-1}di, \varphi_v \rangle \quad \forall \varphi_v \in \mathcal{D}(I). \tag{1.29}
\]

In Problem 1.3, the involved distributions are on the one hand constructed by locally integrable functions \( L^1_{loc}(I) \), and on the other hand by measures \( \mathcal{M}(I) \). In this sense, \( L^1_{loc}(I) \) and \( \mathcal{M}(I) \) can be identified with subspaces of \( D^*(I) \). Actually, elements of these subspaces map functions to \( \mathbb{R} \) which are not elements of \( \mathcal{D}(I) \).

The test functions \( \varphi_q \) and \( \varphi_v \) do not have to be elements of \( \mathcal{D}(I) \) or even of \( C^\infty \) - only the occurring linear functionals have to be consistently defined for a suitable smoothness of \( \varphi_q \) and \( \varphi_v \). For the position, the test functions \( \varphi_q \) do not need to be continuous because each element of \( L^1_{loc}(I) \) naturally defines a measure as a density function with respect to the LEIBESGUE measure. Finite evaluations of the primal-dual pairings are not a problem in practice for position test functions. For the velocity, at least function evaluations of \( \varphi_v \) are necessary such that elements of
\( \mathcal{M} (I) \) can be consistently applied:

\[
\sum_j \langle [v_j] \rangle \varphi_v (t_j) = \sum_j M_j^{-1} p_j \varphi_v (t_j).
\]  

Hence, the test functions for the velocity \( \varphi_v \) must be continuous at the impact times.

Let us summarize the content of this section: the measure differential inclusion in Problem 1.2 somehow naturally comes from combining non-impulsive and impulsive periods of nonsmooth dynamical systems. We can write down increments in terms of measures. Problem 1.2, however, is nothing else, but the weak variational formulation in Problem 1.3. Hence, it is natural to apply Galerkin schemes for the time-discretization of nonsmooth dynamical systems as we know from the numerical treatment of PDEs. This means, that we can discretize Problem 1.3 by choosing finite dimensional subspaces of the continuous function spaces.

### 1.4. Integration Methods

We borrow some formulations from [167, 160] and recall the motivation at the beginning of the present Chapter 1. We study numerical integration schemes for the simulation of nonsmooth mechanical systems. Rigid and Flexible Bodys, Joints as well as Contacts and impacts with dry friction constitute the mechanical models. Thereby, we formulate the contact conditions as constraints and do not allow any penetration, e.g., due to penalty techniques. As a result, velocity jumps occur during the transient simulation of semi-discrete models and we have to be careful in the formulation of efficient and stable time-discretization schemes.

We distinguish two cases.

1. **non-impulsive contact forces** – For the contact between flexible bodies, the contact force is finite in continuum models, that is, in not semi-discrete models, although velocity jumps may arise. Classic (implicit) time integration schemes for computational mechanical problems, e.g., members of the Newmark family [136, 94, 41, 176, 19], have been adapted to these demands and extended with respect to contact/velocity updates (Laursen-Love scheme) [119, 52]. Another strategy to preferably get a well-posed problem is the application of energy-momentum paradigms like in energy-momentum schemes [178, 177, 19], which are modifications of the midpoint rule [176], to impact problems (Laursen-Chawla scheme) [119]. A contact-stabilized Newmark scheme is proposed in [51].

2. **impulsive contact forces** – For the contact between rigid bodies, the reaction forces are impulsive and the classic time integration schemes do not work anymore [3]. The application of mass redistribution techniques [52] is a procedure which reminds of penalty approaches with the benefit of having a theoretical foundation [196]. However in [112], it is shown that all these schemes suffer
from oscillations in the relative contact velocities. Event-driven schemes and
timestepping schemes are further concepts to simulate rigid multibody systems
or semi-discrete systems consistently by applying impact laws. Thereby, event-
driven schemes resolve impact events to a high precision. In-between impact
events, standard integration schemes are used. Classic timestepping schemes
do not resolve impact events, but include their possible existence directly in
the discretization. Thus, they have low accuracy in non-impulsive intervals,
whereas event-driven schemes may get inefficient and inconsistent for many
impact events [3].

We distinguish event-driven schemes, which are also called event-tracking schemes,
and timestepping schemes, which are also known as event-capturing schemes.

1.4.1. Event-Driven Schemes

An event-driven scheme discretizes the non-impulsive equations (1.1), (1.2), (1.9)
and (1.10) by a classic numerical rule [3, Chapter 8]. At the same time, it observes
the gap functions for open \((g_N > 0)\)-close \((g_N \leq 0)\) transitions or stick \((g_T = 0)\)-slip
\((g_T \neq 0)\) transitions. In such cases, the exact transition time is resolved by a
root-finding algorithm with respect to a given tolerance because the structure, that
is, the actual degree of freedom of the mathematical model, changes and therefore
also the numerical simulation has to be restarted. In the case of an open-close
transition, the impact equations (1.5), (1.14) and (1.15) are solved separately and
subsequently the non-impulsive integration is restarted. For the derivation of the
numerical model, standard integration schemes exist [10]. LSODAR\(^6\) or DASKR\(^7\)
are typical examples. Thereby, the relation of the kinematical representations of
the constraint equations (1.9), (1.10) in theory and numerics play a crucial part
for the discretization [73, 54, 92]. Depending on the numerical representation of
the constraints on position, velocity or acceleration level, the time-discretization is
rated in terms of an index. It may differ from the respective index of the continuous
equations and of the index for evaluating the severity of nonsmoothness according to
STEWART [180]. We discuss this concept for timestepping schemes in the following
subsection. As event-driven schemes resolve the exact transition times, on the one
side they cannot resolve Zeno phenomena, that is, an infinite number of impacts in a
finite time interval. On the other side, the resolution of the transition time itself may
be crucial as the used tolerance is depending on the specific mechanical problem, e.g.,
on penetration velocities. These are two main drawbacks of event-driven schemes for
their application to the mathematical description of impacting mechanical systems.
However for non-impulsive intervals, event-driven schemes are extremely efficient
because they are based on sophisticated numerical integration schemes with e.g.,
high accuracy for the numerical representation of the non-impulsive equations.

Event-driven integration schemes treat set-valued force laws by considering the
different possible cases. Therefore, they are based on the index set \(I_0\) of all constraints,
the index set $\mathcal{I}_q^1$ of the closed constraints (1.11), the index set $\mathcal{I}_q^{v,\pm}$ of the closed constraints which stay closed, that is of the persisting constraints (1.12), and the index set of the rolling/sticking constraints $\mathcal{I}_q^{v,\pm}$ (1.13). Exact transition times are resolved with root functions by modeling classical indicators

\[ g_{N,j} = 0 \ , \ j \in \mathcal{I}_0 \setminus \mathcal{I}_q^1 \] for closing contacts,
\[ \lambda_{N,j} = 0 \ , \ j \in \mathcal{I}_q^1 \] for opening contacts,
\[ \| \dot{g}_{T,j} \| = 0 \ , \ j \in \mathcal{I}_q^{v,\pm} \setminus \mathcal{I}_q^{v,\pm} \] for slip-stick transitions,
\[ \| \lambda_{T,j} \| = \mu \lambda_{N,j} \ , \ j \in \mathcal{I}_q^{v,\pm} \] for stick-slip transitions.

In-between such events, standard numerical integration schemes, e.g., Newmark schemes or Runge-Kutta schemes [10], can be applied to

\[ \dot{q} = Yv \ , \quad (1.31) \]
\[ M \dot{v} = h + V\lambda \ , \quad (1.32) \]
\[ g_{N,I_2} = 0 \ , \ \lambda_{N,I_2,U} \geq 0 \ , \quad (1.33) \]
\[ g_{T,I_3} = 0 \ , \ \| \lambda_{T,I_3} \| \leq \mu \lambda_{N,I_3} \quad (1.34) \]

with
\[ V = \left( W_{N,I_2} + W_{T,I_1,I_3} \frac{d\lambda_{T,I_3\setminus I_3}}{d\lambda_{N,I_2}} \quad W_{T,I_3} \right) \quad \lambda = \begin{pmatrix} \lambda_{N,I_2} \\ \lambda_{T,I_3} \end{pmatrix} \]

and $U$ symbolizing unilateral constraints. The generalized force parameters $\lambda$ and the generalized force directions $V$ are associated with bilateral constraints, persisting unilateral constraints and rolling/sticking friction. The generalized sliding friction force parameters depend on the generalized normal force parameters, row-by-row:

\[ \frac{d\lambda_{T,I_2\setminus I_3}}{d\lambda_{N,I_2}} = -\mu \frac{g_{T,I_3\setminus I_3}}{\| g_{T,I_3\setminus I_3} \|} \quad (1.35) \]

**Closing contacts**

We extend $\mathcal{I}_1$ with the respective contact and solve the impact equations (1.5),

\[ M \left( v_j^+ - v_j^- \right) = W_{N,I_1} A_{N,I_1} + W_{T,I_1} A_{T,I_1} \ , \]

and impact laws for all set-valued closed contacts rigidly connected to the impact source:

\[ \dot{g}_{N,I_1,B} = 0 \ , \quad 0 \leq g_{N,I_1,U}^+ + \varepsilon_N g_{N,I_1,U}^- \perp A_{N,I_1,U} \geq 0 \ , \]

\[ \begin{cases} \| A_{T,I_1} \| \leq \mu A_{N,I_1} & \text{for } \dot{g}_{T,I_1}^+ + \varepsilon_T \dot{g}_{T,I_1}^- = 0 \ , \\
A_{T,I_1} = -\frac{\dot{g}_{T,I_1}^+ + \varepsilon_T \dot{g}_{T,I_1}^-}{\| \dot{g}_{T,I_1}^+ + \varepsilon_T \dot{g}_{T,I_1}^- \|} \mu A_{N,I_1} & \text{for } \dot{g}_{T,I_1}^+ + \varepsilon_T \dot{g}_{T,I_1}^- \neq 0 \ . \end{cases} \]
For the bouncing ball example, we have tried all possible contact configurations until a solution is found. More general and more efficient methods are described in Chapter 2. Finally, the index sets as well as the acceleration level information are recalculated concerning the new situation and the integration is restarted.

**Opening contact or stick-slip transition**

We remove the respective contacts from $\mathcal{I}_1$ respectively $\mathcal{I}_3$. Then, we restart the integration.

**Slip-stick transitions**

We add the new rolling/sticking contact to $\mathcal{I}_3$. Then, we solve (1.31)-(1.35) for all set-valued persisting contacts to exclude a slip-slip transition. Appropriate solution methods can be found in Chapter 2. Finally, we restart the integration.

### 1.4.2. Classical Timestepping Schemes

Classic timestepping schemes do not distinguish impulsive and non-impulsive equations, that is they do not resolve exact transition times. Hence, no additional tolerances are required with respect to root-finding. Classic timestepping schemes discretize the equations of motion including the constraints, impact laws and impact equations in a physically consistent and uniform way [3, Chapter 10]. This means that possible impacts are the determining factor for the order of classic timestepping schemes. In comparison to the maximum possible order in non-impulsive phases, the actual order may be low. However, convergence results are available for classic timestepping schemes and not for event-driven schemes. Two mainstream concepts for classic timestepping schemes exist: Schatzman-Paoli [140, 141, 139, 182] and Moreau-Jean schemes [133, 101, 116, 77, 72]. The Schatzman-Paoli scheme discretizes normal gap functions on the level of positions and satisfies impact laws after several time steps. We prefer to formulate the impact law on the level of velocities in discrete time, which is the core of Moreau-Jean schemes. In comparison with the impact equations (1.5), this seems to be more natural.

To exemplary formulate some classic timestepping schemes on an arbitrary time interval $I_i := [t_{i-1}, t_i)$, with the time step-size $\Delta t_i := t_i - t_{i-1}$, we construct the approximations at time $t_i$

\[
q_i \approx q(t_i) , \quad v_i^- \approx \lim_{t \uparrow t_i} v(t), \quad v_i^+ \approx \lim_{t \downarrow t_i} v(t) ,
\]

\[
\Lambda_{N_i} \approx \sum_{j : t_j \in I_i} A_{N,j} , \quad \Lambda_{T_i} \approx \sum_{j : t_j \in I_i} A_{T,j} ,
\]
and use them in the abbreviations

\[ Y_i := Y(q_i), \]
\[ M_i^{-1} := M_i^{-1}(q_i), \]
\[ M_i^{-1}_N := M_i^{-1}\left(q_i + \frac{\Delta t_i}{2} v_i^+\right), \]
\[ h_i^+ := h(q_i, v_i^+), \]
\[ h_{M_i} := h\left(q_i + \frac{\Delta t_i}{2} v_i^+, v_i^+\right), \]
\[ W_{N_i} := W_N(q_i), \]
\[ W_{T_i} := W_T(q_i), \]
\[ W_{N,M_i} := W_N\left(q_i + \frac{\Delta t_i}{2} v_i^+\right), \]
\[ W_{T,M_i} := W_T\left(q_i + \frac{\Delta t_i}{2} v_i^+\right). \]

**Classic Moreau-Jean timestepping scheme**

The classic Moreau-Jean timestepping scheme is given by

\[ q_i = q_{i-1} + \Delta t_i \left[(1 - \theta) Y_{i-1} v_{i-1}^+ + \theta Y_i v_i^+\right], \]
\[ v_i^+ = v_{i-1}^+ + \Delta t_i \left[(1 - \theta) M_i^{-1} h_{i-1}^+ + \theta M_i^{-1} h_i^+\right] + M_i^{-1} [W_{N_i} A_{N_i} + W_{T_i} A_{T_i}] \]

(1.36) \hspace{1cm} (1.37)

together with the active normal impact equations on velocity level,

\[ g_{N_i,i,M}^+ = 0, \]
\[ 0 \leq g_{N_i,i,M}^+ + \epsilon_N g_{N_i,i-1,M}^+ 0 \geq 0, \]

(1.38) \hspace{1cm} (1.39)

and the active tangential impact equations on velocity level,

\[
\begin{cases}
\|A_{T_i,i,M}\| \leq \mu A_{N_i,i,M} \quad \text{for} \quad \dot{g}_{T_i,i,M}^+ + \epsilon_T \dot{g}_{T_i,i-1,M}^+ = 0, \\
A_{T_i,i,M} = -\frac{\dot{g}_{T_i,i,M}^+ + \epsilon_T \dot{g}_{T_i,i-1,M}^+}{\|A_{T_i,i,M}^+ + \epsilon_T \dot{g}_{T_i,i-1,M}^+\|} \mu A_{N_i,i,M} \quad \text{for} \quad \dot{g}_{T_i,i,M}^+ + \epsilon_T \dot{g}_{T_i,i-1,M}^+ \neq 0.
\end{cases}
\]

(1.40)

To reduce the degree of nonlinearity, usually one estimates in advance active contact equations by checking if a prediction is negative:

\[ I_{i,M} := \left\{ k \in I_0 : \tilde{g}_{N_k}^i := g_{N_k}(q_{i,k}) + \kappa \Delta t_i \dot{g}_{N_k}(q_{i-1}, v_{i-1}^+) \leq 0 \right\}. \]

(1.41)

Such a prediction is heuristic with \( \kappa \in [0,1] \). In this work, we use an explicit Euler step with \( \kappa = 1/2 \). The classic Moreau-Jean timestepping scheme is a representative timestepping scheme: it does not adapt the time step-size and naturally combines non-impulsive and impulsive equations, contacts as well as impacts by calculating implicitly the finite mean impulses \( A_{N_i,i,M} \) and \( A_{T_i,i,M} \) within \( I_i \) - their trajectory within \( I_i \) is in this sense an assumption. Equations (1.36)-(1.40) are solved for the unknowns \( q_i, v_i^+, A_{N_i,i,M} \) and \( A_{T_i,i,M} \). The solution of the nonlinear expressions
forms the kernel of the algorithm. We give solution schemes in Chapter 2. Without constraints, this algorithm is the well-known \(\theta\)-integrator, with \(\theta \in [0,1]\).

**Moreau’s midpoint rule**

Concerning [171], we summarize that timestepping schemes discretize the equations of motion, the impact equations, as well as the constraint and impact laws consistently with a fixed time step size, e.g., on impulsive level. An event-detection is not necessary. Another example is Moreau’s midpoint rule:

\[
q_i = q_{i-1} + \frac{Y_i v_i^+ + Y_i v_{i-1}^+}{2} \Delta t_i ,
\]

(1.42)

\[
v_i^+ = v_{i-1}^+ + M_{i-1}^{-1} \left( h_{M_{i-1}} \Delta t_i + W_{N,M_{i-1}} A_N + W_{T,M_{i-1}} A_T \right) ,
\]

(1.43)

with the impact laws (1.39), (1.40).

One might think that Moreau’s midpoint rule produces second order approximations for the generalized position and the generalized velocity. However, the approximate mean impulses \(A_N\), \(A_T\) are incorporated implicitly and of first order in (1.43). Hence, the scheme always produces first order generalized velocity approximations, when at least one gap function is considered active according to (1.41).

**1.4.3. Discussion**

Timestepping schemes are a consistent point of departure for the time-discretization of nonsmooth mechanical systems. However, several extensions are of possible interest and are current research topics.

**Constraint stabilization [171, 2, 32, 130]**

Moreau-Jean schemes in general are formulated on the level of velocities. Hence, a linear drift from the non-penetration invariant will occur [76, 10]. The Gear-Gupta-Leimkuhler formulation [78] treats the non-penetration invariant as an additional constraint and therefore inserts a derivative projection. Applied to Moreau’s timestepping interpretation of the midpoint rule [133], this can be understood as a symmetric projection [89, 176]. We analyze this phenomenon in Chapters 13-15 based on [171].

**High-frequency damping [38, 32, 154]**

Artificial high-frequency oscillations occur due to finite element discretizations and excitations within, e.g., contact problems. One possible remedy is applying the generalized-\(\alpha\) method [41] or other base integration schemes, which are tailored for this application, in the sense of timestepping schemes. We analyze this phenomenon in Chapters 10-12 and Chapters D-G based on [154].
1.4. Integration Methods

**Variational integration** [107]

Sophisticated integration schemes can often be derived from a discrete variational principle [153, 127, 90, 117, 104]. Efforts were made to extend the concept to timestepping schemes for impacting mechanical systems. However, some kind of splitting seems to be necessary because a symplectic method will not stay symplectic for an impact event [179]. We do not consider this kind of extension in the present work.

**Higher order timestepping and step-size adaptation**

There are two types of higher order timestepping schemes which can consistently deal with impacts.

- **Augmented timestepping schemes** [187, 97]
  Within an augmented timestepping scheme, one applies classic augmentation strategies like order extrapolation whenever one heuristically expects no impacts during an integration step. Extrapolation induces instabilities in closed contacts because of chattering in Aitken-Neville schemes or because of the absence of splitting of non-impulsive and impulsive force propagations. It is a serial process. That is why, usually one applies a fixed increased order when we anticipate a non-impulsive phase. Also from a practical point of view, that is not infinitely-differentiable input data, the application of an extrapolation-based augmentation strategy like in [125] seems to be over-engineered.

- **Mixed timestepping schemes** [1, 161, 58, 59]
  A mixed timestepping scheme couples standard integration methods for non-impulsive differential algebraic equations with timestepping schemes for impulsive phases usually by heuristics. We analyze these methods in Chapters 5-9 and Chapters A-C based on [161, 167]. Thereby, heuristics for switching between impulsive and non-impulsive phases can be avoided by embedding in discontinuous Galerkin methods. The synchronization of fixed increased order integration and consistent low order integration is automatic but perhaps not the most robust way for an implementation.

Step size adaptation can be applied for both higher order timestepping strategies. It extends the classic approach [50, 92] by heuristics which meet the timestepping idea. We mention the following approaches.

- **Foreseeing gap-estimations** [97, 59] or retrospective time step bisection [187, 1] ensure sufficiently exact detection of possible velocity jumps.

- **Time step-size switching** $\Delta t_{\text{impulsive}} = O \left( \Delta t_{\text{non-impulsive}}^{p+1} \right)$ couples non-impulsive and impulsive regions using the integration order $p$ of the non-impulsive propagation [187, 1].

- **Error estimation** is based on not adapted [97] or adapted [1] Richardson strategies with some additional heuristics, that is
– exclusion of the possibly jumping velocities in the error estimation [97, 1],
– discussion of appropriate norms [1],
– preferable interval-by-interval separation of possible velocity jumps [97, 1],
– dependence on penetration for closed constraints [68, 1].

Mainly because of missing smoothness, it is very difficult to derive an appropriate time step-size adaptation respecting the tolerance demands. To the knowledge of the author, all mentioned items, that is event prediction, norm selection, error estimation and time step-size selection, have not been theoretically analyzed in the literature.
2. Constraint Representation and Solution Algorithms

For better reading of this chapter, it will be convenient to recall the equations of motion of nonsmooth mechanical system with bilateral and unilateral contacts as well as friction:

\[
\dot{q} = Y(q)v, \quad (2.1)
\]
\[
M(q)\dot{v} = h(q,v,t) + W(q)\lambda, \quad (2.2)
\]
\[
(q,v,\lambda,t) \in \mathcal{N}. \quad (2.3)
\]

Here \(q\) is the generalized position and \(\dot{v}\) the weak time derivative of the generalized velocity \(v\). The mass matrix \(M\) is symmetric and positive definite, the matrix \(Y\) defines the linear relation between the derivative of generalized positions and generalized velocities, and the matrix \(W\) states the generalized directions of set-valued contact reactions. The vector \(h\) contains all smooth external, internal and gyroscopic forces, which are functions of \(q, v\) and explicitly of the time \(t\). It also includes forces that can be expressed as functions, e.g., compliant contacts. The Lagrange multipliers \(\lambda\) refer to contact reaction values. The set \(\mathcal{N}\) restricts the possible values of time, generalized position and generalized velocity such that bodies do not penetrate each other and that friction forces satisfy COULOMB’s law.

![Figure 2.1.: Constraint formulations and solution methods.](image)

The most commonly used local constraint representations and important solution algorithms are shown in Fig. 2.1. We have not shown optimization-based algorithms, as COULOMB friction is not integrable. In this work, we deal with 2-step proj formulations - we show the equivalence to complementarity formulations and the basic solution behavior with fixed-point, that is, splitting, schemes based on [165].
cone-complementarity, 1-step proj and Fischer-Burmeister formulations, also with the application of interior point methods (IPM), we refer to [157, 3, 47, 151, 110, 128].

2.1. Complementarity Formulation

A bilateral constraint

\[ B_C := \{(g_N, \lambda_N) \in \mathbb{R} \times \mathbb{R} \mid g_N = 0\} \quad (2.4) \]

models a contact which is always closed, e.g., a joint in mechanisms. Thereby, it is \( g_N \) the normal distance between the interacting bodies. The bilateral constraint is satisfied on position level at all times. That is why it includes so called hidden constraints on velocity level by replacing \( g_N = 0 \) in \( (2.4) \) with \( \dot{g}_N = 0 \) and on acceleration level by replacing \( g_N = 0 \) by \( \ddot{g}_N = 0 \).

Unilateral contacts can close and open. This is represented by SIGNORINI-FICHERA-conditions:

\[ U_C := \{(g_N, \lambda_N) \in \mathbb{R} \times \mathbb{R} \mid 0 \leq g_N \perp \lambda_N \geq 0\} . \quad (2.5) \]

Symbol \( \perp \) implies complementarity, that is, \( g_N \lambda_N = 0 \). These conditions also include hidden velocity and acceleration level relations by replacing \( g_N \) with its first or second time derivative.

The force laws for bi- and unilateral constraints are shown in Figs. 2.2a and 2.2b.

![Bilateral constraint](image)

(a) Bilateral constraint

![Unilateral constraint](image)

(b) Unilateral constraint

**Figure 2.2.:** Force laws for bilateral and unilateral contacts [165, Figure 1].

In order to formulate COULOMB’s law for both bi- and unilateral constraints, the force of a single contact is decomposed in a positive component \( \lambda_N \) normal to the contact plane and in case of three dimensional dynamics two tangential components \( \lambda_T \) in the friction directions:

\[
T_C (\lambda_N) := \{(\dot{g}_T, \lambda_T) \in \mathbb{R}^2 \times \mathbb{R}^2 \mid \\
\dot{g}_T = 0 \Rightarrow \|\lambda_T\| \leq \mu|\lambda_N| , \\
\dot{g}_T \neq 0 \Rightarrow \lambda_T = -\frac{\dot{g}_T}{\|\dot{g}_T\|}\mu|\lambda_N| \} . \quad (2.6)
\]
We denote $\mu > 0$ the coefficient of friction. In the degenerate case $\mu = 0$ or $\lambda_N = 0$, these conditions are superfluous. For sticking, that is, $\dot{g}_T = 0$, forces are calculated by using the hidden constraint at acceleration level by replacing $\dot{g}_T$ by $\ddot{g}_T$ in (2.6). In the non-degenerate cases, $T_C (\lambda_N)$ equals the following complementarity formulation:

$$ T_\sigma (\lambda_N) := \{ (\dot{g}_T, \lambda_T) \in \mathbb{R}^2 \times \mathbb{R}^2, \sigma \in \mathbb{R} \mid \begin{array}{l} 0 = \mu |\lambda_N| \dot{g}_T + \sigma \lambda_T , \\ 0 \leq \sigma \perp |\lambda_N| - ||\lambda_T|| \geq 0 \} \right. \} . $$  

(2.7)

**Proof** Let $(\dot{g}_T, \lambda_T) \in T_C (\lambda_N)$. Then, there are two cases:

1. $\dot{g}_T = 0$ and $|\lambda_T| \leq \mu |\lambda_N|$:
   
   Choose $\sigma := ||\dot{g}_T|| = 0$. Then, it is obviously $(\dot{g}_T, \lambda_T) \in T_C (\lambda_N)$.

2. $\dot{g}_T \neq 0$ and $\lambda_T = \frac{-\dot{g}_T}{||\dot{g}_T||} |\lambda_N|$:
   
   Choose $\sigma := ||\dot{g}_T||$. Then, it is
   
   $$ \begin{align*} 
   \mu |\lambda_N| \dot{g}_T - \sigma \frac{\dot{g}_T}{||\dot{g}_T||} |\lambda_N| &= 0 . 
   \end{align*} $$

The other conditions for $(\dot{g}_T, \lambda_T) \in T_\sigma (\lambda_N)$ are again obvious.

Let $(\dot{g}_T, \lambda_T) \in T_\sigma (\lambda_N)$. Then, there are two cases:

1. $|\lambda_T| = \mu |\lambda_N|$:
   
   It is
   
   $$ 0 = \sigma ||\lambda_T|| - \mu |\lambda_N| ||\dot{g}_T|| = (\sigma - ||\dot{g}_T||) \mu |\lambda_N| . $$

   If the friction setting is not degenerated ($|\lambda_N| > 0$), it follows $\sigma = ||\dot{g}_T||$. As the other case is obvious, it is only interesting to discuss $\sigma > 0$. Then it is
   
   $$ \lambda_T = \frac{-\dot{g}_T}{\sigma} \mu |\lambda_N| = \frac{-\dot{g}_T}{||\dot{g}_T||} \mu |\lambda_N| $$

   and finally $(\dot{g}_T, \lambda_T) \in T_C (\lambda_N)$.

2. $|\lambda_T| < \mu |\lambda_N|$:
   
   It is $\sigma = 0$ and $\mu |\lambda_N| \dot{g}_T = 0$. One obtains $\dot{g}_T = 0$ in the not degenerate case and $(\dot{g}_T, \lambda_T) \in T_C (\lambda_N)$.

In the non-degenerate cases, conditions (2.7) are equivalent to describing the friction losses as maximum power dissipation in the following sense [192]:

$$ T_M (\lambda_N) := \{ (\dot{g}_T, \lambda_T) \in \mathbb{R}^2 \times \mathbb{R}^2, \lambda_T \in \arg \max_{||\lambda_T|| \leq \mu |\lambda_N|} \left( -\dot{g}_T^T \lambda_T \right) \} . $$  

(2.8)

Originally the principle can be stated as follows: 'A quasi-static system chooses that motion, from among all motions satisfying the constraints, which minimizes the instantaneous power' [142]. This principle is known from material behavior and is not related to, e.g., the principle of minimum potential energy. It is only applicable in the quasi-static case for COULOMB friction if there are no velocity-depending
forces. Equation (2.8) can be used for illustration in the dynamic case. We assume that $\mu$, $\lambda_N$ are known and $\dot{g}_T$ is not implicitly depending on $\dot{\lambda}_T$ [3, Section 3.9.1.2].

**Proof** First, we notice that (2.8) can be equivalently written as

$$T_M(\lambda_N) = \{ (\dot{g}_T, \lambda_T) \in \mathbb{R}^2 \times \mathbb{R}^2, \lambda_T \in \arg \min_{\|\lambda_T\| - \mu |\lambda_N| \leq 0} \{ \dot{g}_T^T \lambda_T \} \} .$$

Both the objective $\min \{ \dot{g}_T^T \lambda_T \}$ and the inequality constraint $\|\lambda_T\| - \mu |\lambda_N| \leq 0$ are defined by convex functions of $\lambda_T$. We get an ordinary convex program according to [156] which satisfies the Slater condition, that is, strict feasibility, e.g., for $\lambda_T = 0$, and has a finite minimum due to the continuity of the inequality constraint function.

According to [156, Corollary 28.3.1], $\lambda_T$ is an optimal solution of the ordinary convex program if and only if there exists a Lagrange multiplier $\sigma$ which, together with $\lambda_T$, satisfies the Karush-Kuhn-Tucker conditions:

$$0 \in \partial [\dot{g}_T^T \lambda_T] + \sigma \partial [\|\lambda_T\| - \mu |\lambda_N|] \text{ Lagrange multiplier rule ,}$$

$$0 \leq \sigma \lambda_N - \|\lambda_T\| \geq 0 \text{ feasibility}$$

with the subdifferential $\partial$ of the respective convex functions. The subdifferential is a generalization of the ordinary derivative. Hence, the subdifferential [156],

$$\partial [\dot{g}_T^T \lambda_T] = \{ \dot{g}_T \} ,$$

$$\partial (\lambda_T) [\|\lambda_T\| - \mu |\lambda_N|] = \begin{cases} \{ \frac{\lambda_T}{\|\lambda_T\|} \} & \text{if } \lambda_T \neq 0 , \\ \{ 0 \} & \text{else .} \end{cases}$$

is the set of slopes of linear functions passing through the evaluation point below or on the graph of the original function. We just have to analyze the Lagrange multiplier rules because the feasibility of the Karush-Kuhn-Tucker conditions and of (2.7) is obviously identical.

Let $(\dot{g}_T, \lambda_T) \in T_{\sigma}(\lambda_N)$. Then, there are two cases:

1. $\dot{g}_T = 0$:
   - It is $\sigma \lambda_T = 0$. If $\sigma = 0$, the Lagrange multiplier rule of the maximum power dissipation is obviously satisfied.
   - If $\sigma > 0$, we get a contradiction because of $0 = \|\lambda_T\| = \mu |\lambda_N| > 0$.

2. $\dot{g}_T \neq 0$:
   - It is $\sigma \lambda_T = -\mu |\lambda_N| \dot{g}_T$. If $\sigma = 0$, this is a contradiction. If $\sigma > 0$, we get $\|\lambda_T\| = \mu |\lambda_N| \neq 0$ and finally $(\dot{g}_T, \lambda_T) \in T_M(\lambda_N)$.

Let $(\dot{g}_T, \lambda_T) \in T_M(\lambda_N)$. Then, there are two cases:

1. $\dot{g}_T = 0$:
   - If $\sigma = 0$, the relation $(\dot{g}_T, \lambda_T) \in T_{\sigma}(\lambda_N)$ is obvious. If $\sigma > 0$, we get $\|\lambda_T\| = \mu |\lambda_N| \neq 0$ which is a contradiction.

2. $\dot{g}_T \neq 0$:
   - If $\sigma = 0$, this is a contradiction. If $\sigma > 0$, it is $\|\lambda_T\| = \mu |\lambda_N| \neq 0$ and finally $(\dot{g}_T, \lambda_T) \in T_{\sigma}(\lambda_N)$.

For the complementarity formulation in global generalized coordinates with $N_c$ unilateral contact points, the terms in the normal and tangential directions are collected on acceleration level:

$$M \ddot{v} = h + W_N \lambda_N + W_T \lambda_T ,$$

$$0 \leq \lambda_N \perp W_N^T \dot{u} + \ddot{w}_N \geq 0 ,$$

$$0 = (U \lambda_N) \circ (W_T^T \ddot{v} + \dot{w}_T) + \lambda_T \circ \sigma ,$$
\[ 0 \leq \sigma \perp (U \lambda_N) \circ (U \lambda_N) - \lambda_T \circ \lambda_T \geq 0, \]

where \( \dot{g} = W^T \dot{v} + \bar{w} \) with \( \bar{w} \) contains the components of the relative Coriolis and centripetal accelerations at the contacts, \( U \) is \( \text{diag}(\ldots \mu_i \ldots) \), and \( \circ \) denotes the Hadamard product,

\[ \circ : \mathbb{R}^{2N_c} \times \mathbb{R}^{2N_c} \to \mathbb{R}^{2N_c}, \quad (u \circ w)_i = u_i w_i. \]

These equations represent a nonlinear differential complementarity problem (DCP). The discrete-time equations for simulation form a nonlinear complementarity problem (NCP) which can be solved, e.g., by the PATH algorithm \[ 62, 63 \]. It is a Newton-based solver, which generates internally a path of mixed linear complementarity problems, which guarantees sufficient progress.

For each subproblem of the PATH solver or for linearized friction cones, a mixed linear complementarity problem (MLCP) has to be solved:

\[
\begin{align*}
M \dot{v} &= h + W_N \lambda_N + W_f \lambda_f , \\
0 &\leq \lambda_N \perp W_N^T \dot{v} + \bar{w}_N \geq 0 , \\
0 &\leq \lambda_f \perp W_f^T \dot{v} + E \sigma + \bar{w}_T \geq 0 , \\
0 &\leq \sigma \perp U \lambda_N - E^T \lambda_f \geq 0 ,
\end{align*}
\]

where \( \lambda_{Ti} = \lambda_{fi1} - \lambda_{fi2} \) with \( \lambda_{fi1/2} \geq 0 \) and appropriate \( W_f, E \) (cf. (2.9)-(2.14) for the particle-on-plane-problem in Fig. 2.4). The relationship

\[ 0 \leq \begin{pmatrix} \lambda_N \\ \lambda_f \\ \sigma \end{pmatrix} \perp \begin{pmatrix} \dot{g}_N \\ \dot{g}_T \\ s \end{pmatrix} \geq 0 \]

with the matrix-vector notation

\[
\begin{pmatrix} M & -W_N & -W_f & 0 \\ W_N^T & 0 & 0 & 0 \\ W_f^T & 0 & 0 & E \\ 0 & U & -E^T & 0 \end{pmatrix}
\begin{pmatrix} \dot{v} \\ \lambda_N \\ \lambda_f \\ \sigma \end{pmatrix} + \begin{pmatrix} -h \\ \bar{w}_N \\ \bar{w}_T \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ \dot{g}_N \\ \dot{g}_T \\ s \end{pmatrix}
\]

is a compact summary. The MLCP can be also written as a linear complementarity problem (LCP)

\[
\begin{pmatrix} W_N^T M^{-1} W_N & W_N^T M^{-1} W_f & 0 \\ W_f^T M^{-1} W_N & W_f^T M^{-1} W_f & E \\ U & -E^T & 0 \end{pmatrix}
\begin{pmatrix} \lambda_N \\ \lambda_f \\ \sigma \end{pmatrix} + \begin{pmatrix} W_N^T M^{-1} h + \bar{w}_N \\ W_f^T M^{-1} h + \bar{w}_T \\ 0 \end{pmatrix} = \begin{pmatrix} \dot{g}_N \\ \dot{g}_T \\ s \end{pmatrix}
\]

by eliminating \( \dot{v} \) using

\[ \dot{v} = M^{-1} h + M^{-1} W_N \lambda_N + M^{-1} W_f \lambda_f . \]
The resulting LCP can always be solved with LEMKE’s algorithm when a solution exists [46, 20, 181] although it might not be unique in the case of friction as can be shown by PAINLEVE’s example [15].

A good overview about numerical methods for LCPs can be found in [106]. Here we focus on an explanation of LEMKE’s algorithm. We use the standard notation in Fraktur style:

\[
\mathbf{w} = q + \mathbf{M}\mathbf{z},
\]

\[\mathbb{R}^n \ni 0 \leq \mathbf{w} \perp \mathbf{z} \geq 0 \in \mathbb{R}^n\]

and look for the unknowns \( \mathbf{w} \) and \( \mathbf{z} \). Due to the complementarity in each of the \( n \) pairs \( \mathbf{w}_j \mathbf{z}_j = 0, j \in \{1, \ldots, n\} \), at least one unknown vanishes. This unknown is called nonbasic variable, the other basic variable. The idea is to run through all \( 2^n \) possible combinations of nonbasic variables \( \eta^c \) and to reduce the original but re-written system of linear equations

\[
(I - \mathbf{M}) \begin{pmatrix} \mathbf{w} \\ \mathbf{z} \end{pmatrix} = q
\]

to

\[
\mathbf{C}\eta = q,
\]

with

\[
\mathbf{C}_j = \begin{cases} I_j & \text{if } \mathbf{w}_j \text{ is basic} \\ -\mathbf{M}_j & \text{if } \mathbf{z}_j \text{ is basic} \end{cases}
\]

We solve for the respective basic variables \( \eta \). If \( \eta \geq 0 \) is feasible, a solution is found.

LEMKE’s algorithm defines a rule to run through the sets of possible nonbasic variables until a first solution is found, by changing just one variable from step to step. If \( q \geq 0 \), setting \( \mathbf{z} = 0 \) gives a solution of the original LCP because \( \mathbf{w} = q \). Otherwise, one considers an augmented LCP with a covering vector \( \mathbf{e} \) containing only one-entries and with a slack variable \( z_0 \geq 0 \):

\[
(I - \mathbf{M} - \mathbf{e}) \begin{pmatrix} \mathbf{w} \\ \mathbf{z} \\ z_0 \end{pmatrix} = q.
\]

We choose \( 0 = \eta^c = \mathbf{z} \) as nonbasic variables and \( z_0 \) as nonbasic driving variable. Working on the basic columns \( \mathbf{v} = I^{-1}q \), we find \( r := \arg \min \{v_j\} \) lexicographically in case of ambiguity and we set \( z_0 := -\mathbf{v}_r > 0 \). Then, it is

\[
\mathbf{w} = I^{-1}q + z_0I^{-1}e \geq 0
\]
2.1. Complementarity Formulation

the first possibility of \( w \) to be non-negative and blocking that \( z_0 \) increases to \( \infty \). To proceed Gaussian elimination, we consider two exchange rules exemplary for the next step:

Rule 1: The basic variable \( w_r \) blocked \( z_0 \) now it is nonbasic \( w_r = 0 \) and removed and we include \( z_0 > 0 \) as basic variable.

Rule 2: We choose \( z_r \) as the nonbasic driving variable of the next step, as it will not violate complementarity (\( z_r w_r = 0 \) due to \( w_r = 0 \)). In general if \( w_k \) is removed in one step, one has to bring in \( z_k \) in the next step - the other way around if \( z_k \) is removed in one step, one has to bring in \( w_k \) in the next step [180].

If \( z_0 = 0 \) is removed in one step, we have found a solution of the original LCP - however there might be more. If there is no blocking in one step, termination in a ray occurs. However, interpretation depends on \( \mathfrak{M} \).

In the special case of planar systems, COULOMB’s friction law is linear anyway (cf. Fig. 2.3). Linearizing the spatial friction cone results in unrealistic behavior because in general the friction force will not oppose the direction of sliding. Hence, we will work with LCPs and LEMKE’s algorithm in the following.

![Figure 2.3.](image)

**Figure 2.3.** Planar Coulomb friction [165, Figure 2].

Another reason to avoid LEMKE’s algorithm is the exponential calculation time. We consider the particle-on-plane-problem (Fig. 2.4), which is taken from [165].

![Figure 2.4.](image)

**Figure 2.4.** Configuration of a point mass in contact with a surface [165, Figure 5].

Let us replace \( \lambda_T \) with the sum of its positive and negative parts: \( \lambda_T = \lambda_f^1 - \lambda_f^2 \), where \( \lambda_f^1 \geq 0 \) is the friction force component acting in the positive \( x \)-direction and
\( \lambda_{f2} \geq 0 \) acts in the \((-x)\)-direction. Then, the equations of motion are linear in the unknown forces, accelerations and in the auxiliary variable \( \sigma \):

\[
\begin{align*}
    m \dot{v}_x &= h_x + \lambda_{f1} - \lambda_{f2}, \\
    m \dot{v}_y &= h_y + \lambda_N, \\
    0 &\leq \lambda_N \perp \dot{v}_y \geq 0, \\
    0 &\leq \lambda_{f1} \perp \dot{v}_x + \sigma \geq 0, \\
    0 &\leq \lambda_{f2} \perp -\dot{v}_x + \sigma \geq 0, \\
    0 &\leq \sigma \perp \mu \lambda_N - \lambda_{f1} - \lambda_{f2} \geq 0.
\end{align*}
\]

Also for a solution, we have \( \sigma = |\dot{v}_x| \) [193]. The above mixed LCP can be reformulated as a standard LCP of size four by eliminating \( \dot{v}_x \) and \( \dot{v}_y \).

The numerical solver has to test pivots until a solution is found. In the worst case, this is \( 2^n \) pivots where \( n \) is the number of unknowns. However, the average time required by \textsc{Lemke}'s algorithm is approximately cubic in \( n \). In the following, 16 cases have to be discussed whereby 8 of them degenerate to only one because of the missing tangential considerations.

- \( \lambda_N = 0 \) and \( \dot{v}_y > 0 \):
  It follows from \( \dot{v}_y = h_y/m \) that \( h_y > 0 \). The whole tangential part is degenerated which mathematically can be seen in \( \lambda_{f1} \geq 0 \) but also in \( -\lambda_{f1} - \lambda_{f2} \geq 0 \) and hence in \( \lambda_{f2} = 0 \). Altogether we have \( \dot{v}_x = h_x/m \).

- \( \lambda_N > 0 \) and \( \dot{v}_y = 0 \):
  It follows that \( \lambda_N = -h_y \). With this information, one starts the tangential calculations.

- \( \lambda_{f1} > 0 \) and \( \dot{v}_x + \sigma = 0 \):
  * \( \lambda_{f2} > 0 \) and \( -\dot{v}_x + \sigma = 0 \):
    * \( \sigma > 0 \) and \( -\mu h_y - \lambda_{f1} - \lambda_{f2} = 0 \):
      \( 2\sigma = \dot{v}_x + \sigma - \dot{v}_x + \sigma = 0 \) yields a contradiction.
    * \( \sigma = 0 \) and \( -\mu h_y - \lambda_{f1} - \lambda_{f2} > 0 \):
      It is \( \dot{v}_x = 0 \) and \( \lambda_{f2} = h_x + \lambda_{f1} \). This is a sticking case. Notice that the model cannot resolve the individual values of \( \lambda_{f1} \) and \( \lambda_{f2} \), which were used for mathematical convenience. However, the total friction force \( \lambda_T \) is uniquely determined.
  * \( \lambda_{f2} = 0 \) and \( -\dot{v}_x + \sigma > 0 \):
    * \( \sigma > 0 \) and \( -\mu h_y - \lambda_{f1} = 0 \):
      It holds \( \lambda_{f1} = -\mu h_y \) and we have \( \sigma = -\dot{v}_x = -h_x/m + \mu h_y/m \), with the prerequisite \( -h_x/m + \mu h_y/m > 0 \). This case corresponds to leftward sliding.
2.2. 2-step proj Formulation

\[ \sigma = 0 \text{ and } -\mu h_y - \lambda f_1 > 0: \]
As \( \dot{v}_x < 0 \) and \( \dot{v}_x = 0 \), this is a contradiction.

\[ \lambda f_1 = 0 \text{ and } \dot{v}_x + \sigma > 0: \]

* \( \lambda f_2 > 0 \text{ and } -\dot{v}_x + \sigma = 0: \)
  * \( \sigma > 0 \text{ and } -\mu h_y - \lambda f_2 = 0: \)
    It is \( \lambda f_2 = -\mu h_y \) and we have \( \sigma = \dot{v}_x = h_x/m + \mu h_y/m \), with
    the prerequisite \( h_x/m + \mu h_y/m > 0 \). This case corresponds to
    rightward sliding.
  * \( \sigma = 0 \text{ and } -\mu h_y - \lambda f_2 > 0: \)
    As \( \dot{v}_x > 0 \) and \( \dot{v}_x = 0 \), this is a contradiction.

* \( \lambda f_2 = 0 \text{ and } -\dot{v}_x + \sigma > 0: \)
  * \( \sigma > 0 \text{ and } -\mu h_y = 0: \)
    As \( h_y < 0 \) and \( h_y = 0 \), this is a contradiction.
  * \( \sigma = 0 \text{ and } -\mu h_y > 0: \)
    As \( \dot{v}_x < 0 \) and \( \dot{v}_x > 0 \), this is a contradiction.

\[ \sigma = 0 \text{ and } -\mu h_y - \lambda f_2 > 0: \]
As \( \dot{v}_x < 0 \) and \( \dot{v}_x > 0 \), this is a contradiction.

\[ \lambda f_2 > 0 \text{ and } -\dot{v}_x + \sigma = 0: \]

\[ \sigma > 0 \text{ and } -\mu h_y - \lambda f_2 = 0: \]

\[ \lambda f_2 = 0 \text{ and } -\dot{v}_x + \sigma > 0: \]

\[ \sigma = 0 \text{ and } -\mu h_y - \lambda f_2 > 0: \]

\[ \sigma > 0 \text{ and } -\mu h_y = 0: \]

\[ \sigma = 0 \text{ and } -\mu h_y > 0: \]

\[ \lambda f_2 = -\mu h_y \text{ and } \dot{v}_x = h_x/m + \mu h_y/m > 0. \]

2.2. 2-step proj Formulation

\[ \text{proj}_C(x_2) \]
\[ x_1 = \text{proj}_C(x_1) \]

Figure 2.5.: proj function for a convex set [72, Figure 3.6].

In this section, the content of which is in large part taken from [165], we develop a
formulation using nonsmooth equations to apply general numerical solution methods.
To this purpose, we use convex analysis [102, 156]. Define the projection to a convex
set, \( C \subset \mathbb{R}^n \), \( n \in \mathbb{N} \) as a special case of proximations (Fig. 2.5):

\[ \text{proj}_C(x) = \arg \min_{x^* \in C} \| x - x^* \|, \quad x \in \mathbb{R}^n. \]

We claim that the contact conditions can be formulated as

\[ B_p := \{ (g_N, \lambda_N) \in \mathbb{R} \times \mathbb{R} \mid f_B(\lambda_N, g_N) := \lambda_N - \text{proj}_{C_B}(\lambda_N - r g_N) = 0 \} , \]

(2.15a)
We show directly from the set definitions that

\[ U_P := \{ (g_N, \lambda_N) \in \mathbb{R} \times \mathbb{R} \mid f_U(\lambda_N, g_N) := \lambda_N - \text{proj}_{C_U}(\lambda_N - r g_N) = 0 \} \, , \tag{2.15b} \]

\[ T_P(\lambda_N) := \{ (\mathbf{g}_T, \lambda_T) \in \mathbb{R}^2 \times \mathbb{R}^2 \mid f_T(\lambda_T, \mathbf{g}_T) := \lambda_T - \text{proj}_{C_T(\lambda_T)}(\lambda_T - r \mathbf{g}_T) = 0 \} \, , \tag{2.15c} \]

where the corresponding convex sets are specified by

\[ C_B = \mathbb{R} \, , \]

\[ C_U = \{ x \in \mathbb{R} \mid x \geq 0 \} \, , \]

\[ C_T : \mathbb{R} \to \mathcal{P}(\mathbb{R}^2) : y \mapsto C_T(y) = \{ x \in \mathbb{R}^2 \mid \| x \| \leq \mu|y| \} \, , \]

with \( y \in \mathbb{R} \). The set \( C_T \) is the friction disk with the local sticking limit as radius. The auxiliary parameter \( r \in \mathbb{R}^+ \) is arbitrary from the mathematical viewpoint. It physically transforms units and we will see that \( r \) can be used to influence convergence and stability of the used numerical scheme. Notice that these equations are formulated on the level of the lowest-order derivatives, positions in the normal directions and velocities in the tangential directions. Hidden higher-order derivative formulations can be obtained simply by increasing the number of dots over the \( \mathbf{g} \)'s.

We show directly from the set definitions that

\[ B_C = B_P \, , \, U_C = U_P \, , \, T_C = T_P \, , \]

for the auxiliary parameter \( r \in \mathbb{R}^+ \).

**Proof** We discuss the bilateral and the unilateral constraint as well as Coulomb friction separately.

- **Bilateral constraint**
  Starting with the 2-step proj formulation \((g_N, \lambda_N) \in B_P\), it is obvious that the argument \((\lambda_N - r g_N)\) of \(\text{proj}_{C_B(\lambda_N)}\) is always an element of the corresponding convex set \(C_B = \mathbb{R}\). This yields:

  \[ 0 = \lambda_N - \lambda_N + r g_N \]

  and finally \( g_N = 0 \) as well as \((g_N, \lambda_N) \in B_C\).

  Vice versa, it is \( \lambda_N \in \mathbb{R} \) and \( g_N = 0 \). The relations both \( \lambda_N = \text{proj}_{C_B}(\lambda_N - r g_N) \) and \((g_N, \lambda_N) \in B_P\) are obvious.

- **Unilateral constraint**
  Starting with the 2-step proj formulation \((g_N, \lambda_N) \in U_P\), one has to distinguish two cases.

  1. \( \lambda_N - r g_N \in C_U \) or \( \lambda_N - r g_N \geq 0 \):
     This yields \( 0 = \lambda_N - \lambda_N + r g_N \) and \( g_N = 0 \), as well as \( \lambda_N \geq 0 \).

  2. \( \lambda_N - r g_N \notin C_U \) or \( \lambda_N - r g_N < 0 \):
     This yields \( \lambda_N = \text{proj}_{C_U}(\lambda_N - r g_N) = 0 \) and \( g_N > 0 \).

  This is a representation of the unilateral corner law by a complementarity formulation \((g_N, \lambda_N) \in U_C\).

  Vice versa if one has a representation by a complementarity formulation \((g_N, \lambda_N) \in U_C\), again there are two cases for the branches of the unilateral corner law.

  1. \( g_N = 0 \) with \( \lambda_N \geq 0 \):
     This yields \( \lambda_N - r g_N = \lambda_N \geq 0 \) or \( \lambda_N - r g_N \in C_U \), and the validity of the 2-step proj formulation \((g_N, \lambda_N) \in U_P\) in the specific case.
2. \( g_N > 0 \) and \( \lambda_N = 0 \):
This yields \( \lambda_N - r g_N < 0 \) or \( \lambda_N - r g_N \not\in C_U \), and the validity of the 2-step proj formulation \((g_N, \lambda_N) \in U_p \) in the specific case.

- Coulomb friction
  Let \( \mu \geq 0 \) and \( \lambda_N \in \mathbb{R} \).

Starting with the 2-step proj formulation \((\dot{g}_T, \lambda_T) \in T_P (\lambda_N)\), one has to distinguish two cases.

1. \( \lambda_T - r \dot{g}_T \in C_T (\lambda_N) \) or \( \|\lambda_T - r \dot{g}_T\| \leq \mu |\lambda_N| \):
   This yields \( 0 = \lambda_T - \lambda_T^T + r \dot{g}_T \) and \( \dot{g}_T = 0 \). One has \( \|\lambda_T\| \leq \mu |\lambda_N| \) and finally \((\dot{g}_T, \lambda_T) \in T_C (\lambda_N)\).

2. \( \lambda_T - r \dot{g}_T \not\in C_T (\lambda_N) \) or \( \|\lambda_T - r \dot{g}_T\| > \mu |\lambda_N| \):
   This yields:
   \[
   \lambda_T = \frac{\lambda_T - r \dot{g}_T}{\|\lambda_T - r \dot{g}_T\|} \mu |\lambda_N| - \frac{r \|\dot{g}_T\|}{\|\lambda_T - r \dot{g}_T\|} \frac{\dot{g}_T}{\|\dot{g}_T\|} |\lambda_N| .
   \]
   In the case \( \dot{g}_T = 0 \), this would be a contradiction because
   \[
   \|\lambda_T\| > \mu |\lambda_N| \land \lambda_T = \frac{\lambda_T}{\|\lambda_T\|} \mu |\lambda_N| .
   \]
   In the case \( \dot{g}_T \neq 0 \), the contact is sliding and we must have \( \|\lambda_T\| = \mu |\lambda_N| \). Comparing the expressions, one sees that \( k \) must equal 1. Therefore, the only solution is given by \( \lambda_T = - \frac{\dot{g}_T}{\|\dot{g}_T\|} \mu |\lambda_N| \) and \((\dot{g}_T, \lambda_T) \in T_C (\lambda_N)\).

Vice versa if one has a representation by a complementarity formulation \((\dot{g}_T, \lambda_T) \in T_C (\lambda_N)\), again there are two cases.

1. \( \dot{g}_T = 0 \) and \( \|\lambda_T\| \leq \mu |\lambda_N| \):
   This yields \( \lambda_T - r \dot{g}_T \in C_T (\lambda_N) \) and, trivially \((\dot{g}_T, \lambda_T) \in T_P (\lambda_N)\).

2. \( \dot{g}_T \neq 0 \) and \( \lambda_T = - \frac{\dot{g}_T}{\|\dot{g}_T\|} \mu |\lambda_N| \):
   This yields:
   \[
   \|\lambda_T - r \dot{g}_T\| = \left| \frac{\dot{g}_T}{\|\dot{g}_T\|} \mu |\lambda_N| - r \frac{\dot{g}_T}{\|\dot{g}_T\|} \right| = |\lambda_N| + r \frac{\|\dot{g}_T\|}{\|\dot{g}_T\|} |\lambda_N| > \mu |\lambda_N| ,
   \]
   and \( \lambda_T - r \dot{g}_T \not\in C_T (\lambda_N) \). Finally:
   \[
   \lambda_T = - \frac{\dot{g}_T}{\|\dot{g}_T\|} \mu |\lambda_N| = - \frac{\mu |\lambda_N| + r \|\dot{g}_T\|}{\mu |\lambda_N| + r \|\dot{g}_T\|} \frac{\dot{g}_T}{\|\dot{g}_T\|} \mu |\lambda_N| = \frac{\lambda_T - r \dot{g}_T}{\|\lambda_T - r \dot{g}_T\|} \mu |\lambda_N| = \text{proj}_{C_T (\lambda_N)} (\lambda_T - r \dot{g}_T) ,
   \]
   which means \((\dot{g}_T, \lambda_T) \in T_P (\lambda_N)\).

Typical spatial plots of the involved nonsmooth functions \( f_B, f_U \) and \( f_T \) are given in Fig. 2.6 for \( r = 0.1 \), in Fig. 2.7 for \( r = 0.5 \) and in Fig. 2.8 for \( r = 1.0 \) always in the same sector and scaling. In the tangential cases, \( \mu = 0.1 \) and \( \lambda_N = 1 \) have been used.
2. Constraint Representation and Solution Algorithms

Figure 2.6.: proj functions for contacts and planar friction with \( r = 0.1 \).

Figure 2.7.: proj functions for contacts and planar friction with \( r = 0.5 \).

The intersection with the zero-constraint and the solution sets are reminiscent of the force laws from Figs. 2.2 and 2.3.

2.2.1. Projected Splitting Methods

We summarize all of the possible contact descriptions in one large fixed-point scheme using (2.15) block-by-block, that is, in a 2-step –TRESCA fixed-point– way first introducing normal force projections and after that tangential force projections on acceleration level:

\[
\lambda = \text{proj} (\lambda - r \ddot{g}) \tag{2.16}
\]

with

\[
\ddot{g} = W^T \dot{v} + \bar{w} \tag{2.17}
\]

Now, \( r \) is assumed to be a positive diagonal matrix. Solving (2.2) for

\[
\dot{v} = M^{-1} h + M^{-1} W \lambda
\]

and eliminating it from (2.17) yields

\[
\ddot{g} = G \lambda + b
\]

Figure 2.8.: proj functions for contacts and planar friction with \( r = 1.0 \) [165, Figure 4].
where \( G = W^T M^{-1} W \) is the Delassus matrix and \( b = W^T M^{-1} h + \bar{w} \). Thus, (2.16) becomes

\[
\lambda = \text{proj} \left( (I - rG) \lambda - rb \right),
\]

(2.18)

where \( I \) is the identity matrix.

We explain the formulation of fixed-point methods for the solution of (2.18). For this reason, we define contraction conditions based on the linear part

\[
\lambda = (I - rG) \lambda - rb
\]

\( \Leftrightarrow G\lambda = -b \).

(2.20)

Then, several classic splitting schemes based on the decomposition

\[
G = B + (G - B)
\]

are available for its solution [183] using the relaxation parameter \( \omega \in \mathbb{R}^+ \):

- Relaxed Richardson scheme \( B = I/\omega \),
- Relaxed Jacobi scheme \( B = \text{diag} (G)/\omega \),
- Relaxed GAUSS-SEIDEL scheme (SOR) \( B = \text{diag} (G)/\omega + \text{tril} (G) \).

The operators \( \text{diag} (G) \) and \( \text{tril} (G) \) define matrices of the same dimensions as \( G \), whereby the only elements not equal to zero are the extracted diagonal of \( G \) and the lower triangular elements without the diagonal, respectively. Comparing the canonic iteration scheme for the linear part

\[
\lambda^{k+1} = (I - B^{-1}G) \lambda^k - B^{-1}b,
\]

with (2.19) yields the strategy

\[
r := B^{-1},
\]

for the proj iteration

\[
\lambda^{k+1} = \text{proj} \left( (I - rG) \lambda^k - rb \right)
\]

which extends [72, 186, 165]. We have already discussed the example of a planar point mass sliding on some rigid horizontal surface (Fig. 2.9) in Section 2.1 concerning LCP formulations. The Delassus matrix for this system is given by

\[
G = \begin{pmatrix}
\frac{1}{m} & 0 \\
0 & \frac{1}{m}
\end{pmatrix}.
\]

It follows that one has an optimal convergence scheme for \( \omega = m \) in the case of a Richardson and \( \omega = 1 \) in the case of Jacobi and GAUSS-SEIDEL scheme.
According to [7] \( r \) is a stabilization within an augmented Lagrangian setting. However, what does \( r \) mean practically? To understand this, we first consider the normal component of the contact force in the example of the planar point mass (Fig. 2.9). The fixed-point equation is \( \lambda_N = \text{proj}_{C_{|\lambda_N|}}(\lambda_N - r\ddot{g}_N(\lambda_N)) \). Since the first term in the argument of the proj function is a force, \( r \) must have units of mass. Taking into consideration that \( \text{proj}_{C_{|\lambda_N|}} \) is zero for all negative arguments, one observes that the equation can be satisfied in only two ways. First if \( \ddot{g}_N \) is positive, then equality requires that \( \lambda_N = 0 \). Observe that the only way for \( \lambda_N \) to be positive is for \( \ddot{g}_N \) to be zero. Now consider some external force \( h_y < 0 \) pushing the point mass to the surface. If the first iterate \( \lambda^0_N < 0 \) and the value of the argument of the proj function is less than zero, the proj function sets the next value \( \lambda^1_N \) to zero. If, in this particular case, one wanted to compute \( \ddot{g}_N(\lambda^1_N) \), one would find its value to be negative which would lead to penetration. The proj function "knows" this, hence it increases the next estimate \( \lambda^2_N \) to \( -r\ddot{g}_N(\lambda^1_N) \). The value of \( r \) should be interpreted as the estimate of the mass, hence the proj function can be viewed as trying to correct the contact force with the exact amount necessary to prevent the penetration. However since the value is not known exactly, the proj function correction is not quite right, indeed it iterates. Finally note that if the estimate of the mass is too inaccurate, that is, outside the interval \((0, 2m)\), then the iterates diverge (Fig. 2.10).

For sliding contacts, the proj function is equivalent to \( \lambda_T = \mu\lambda_N \) or \( \lambda_T = -\mu\lambda_N \). This allows to exclude \( \lambda_T \) from our considerations. The more interesting case occurs for sticking contacts: \( \lambda_T = \text{proj}_{C_T}(\lambda_T - r\ddot{g}_T(\lambda_N, \lambda_T)) \). For such a case, each iteration can be thought of as making a guess for \( \lambda_T \) (with the current \( \lambda_N \)) and as checking for physical consistency. If the guess is outside the allowable friction bounds, the proj function projects it to the nearest boundary. The chosen value of \( \lambda_T \) causes sliding even though the friction force is within the bounds. The proj function attempts to correct the problem but since \( r \) is a relaxation or an over-estimate of \( m \), it over-corrects. This causes a spiral iteration (Fig. 2.11). In practice, the numerical solution proceeds by interleaving fixed-point iterations between the normal and tangential proj functions. For the particle-on-plane problem, the graph in Fig. 2.10 is unchanged, hence the normal force iterations are unchanged. However, the graph in Fig. 2.11 is dependent on the result of the preceding iteration for the normal force. The tangential iterations are guaranteed to converge to the correct value as
2.2. 2-step proj Formulation

\[ \lambda_0^N \rightarrow \lambda_1^N \rightarrow \lambda_2^N \rightarrow \ldots \lambda_k^N \rightarrow \ldots \]

Figure 2.10.: Progression of iterates for finding the normal contact force [165, Figure 6].

The normal direction iterations converge, because the iterations for the tangential component converge from any starting guess.

\[ \lambda_T^* = h_x \]

Figure 2.11.: Progression of iterates for finding the tangential contact force [165, Figure 7].

2.2.2. LCP and 2-step proj Formulations: Painleve’s Paradox

Painleve’s paradox refers to problems in multibody dynamics in which the solution of the instantaneous dynamics is not unique [175]. One such "paradox" arises in the planar case of a rigid object translating in contact with a rigid horizontal surface. The number of solutions of the acceleration-level formulation depends on the values of the mass, moment of inertia, coefficient of friction, and the angle and distance.
from the contact point to the center of mass of the object (Fig. 2.12). In the case considered here, a gravitational force is assumed to act downward through the center of mass perpendicular to the contact tangent, that is, \( h_y = -m g \). Since \( m \) and \( g \) are positive, \( h_y \) is negative. The lateral external force \( h_x \) is zero.

\[
\begin{align*}
\dot{q}_x &= v_x, \quad \text{(2.21)} \\
\dot{q}_y &= v_y, \quad \text{(2.22)} \\
\dot{q}_\theta &= v_\theta, \quad \text{(2.23)} \\
m \dot{v}_x &= \lambda_T, \quad \text{(2.24)} \\
m \dot{v}_y &= \lambda_N + h_y, \quad \text{(2.25)} \\
J \dot{v}_\theta &= \frac{L}{2} (\lambda_T \sin(q_\theta) - \lambda_N \cos(q_\theta)), \quad \text{(2.26)}
\end{align*}
\]

where \( J \) is the moment of inertia about the center of mass and about an axis perpendicular to the plane of motion and \( L/2 \) is the distance from the contact point to the center of mass.

Let us assume that at the current time instant, the object is translating toward the left and leaning toward the right, that is, \( v_x < 0, v_y = v_\theta = 0 \) and \( 0 < q_\theta < \pi/2 \). Using the velocity kinematics, the direction of the friction force is determined to be in the positive \( x \)-direction, because the contact point is initially sliding. The unilateral contact constraint must still be evaluated on acceleration level because of \( v_y = 0 \) and \( \dot{v}_y = 0 \). Thus, the contact constraints for this problem are:

\[
\begin{align*}
(\ddot{g}_N, \lambda_N) &\in U_{C/P}, \quad \text{(2.27)} \\
(\dot{g}_T, \lambda_T) &\in T_{C/P} (\lambda_N). \quad \text{(2.28)}
\end{align*}
\]

The kinematic quantities, \( g_N \) and \( g_T \), and their relevant time derivatives are:

\[
\begin{align*}
g_T &= q_x - \frac{L}{2} \cos(q_\theta), \quad \text{(2.29)} \\
g_N &= q_y - \frac{L}{2} \sin(q_\theta). \quad \text{(2.30)}
\end{align*}
\]
2.2. 2-step proj Formulation

\[ \dot{g}_T = v_x + \frac{L}{2} v_{\theta} \sin(q_{\theta}) \],
\[ \dot{g}_N = \dot{v}_y + \frac{L}{2} \left( v_{\theta}^2 \sin(q_{\theta}) - \dot{v}_{\theta} \cos(q_{\theta}) \right) \] .

Note that (2.28) with \( \dot{v}_x < 0 \) implies that \( \lambda_T = \mu \lambda_N \). This allows us to reduce the formulation of the problem to an LCP of size one:

\[ 0 \leq \lambda_N \perp G \lambda_N + b \geq 0 \],

where \( G = \frac{1}{m} + \frac{L^2 \cos(q_{\theta})}{4J} (\cos(q_{\theta}) - \mu \sin(q_{\theta})) \) and \( b = -\Gamma \). For physically plausible values of \( m, J, L, q_{\theta} \) and \( \mu \), the Delassus matrix \( G \) can be made negative. By choosing the sign of the gravitational acceleration \( \Gamma \), the parameter \( b \) can be made negative (the force acts downward) or positive (the force acts upward).

Figure 2.13.: Globally convergent solution: \( \Gamma, G > 0 \). Contact maintained [165, Figure 9].

Figure 2.14.: Globally convergent solution: \( \Gamma < 0, G > 0 \). Contact lost [165, Figure 10].

For this problem, it is known that when \( G \) is strictly positive (Figs. 2.13 and 2.14), there is a unique solution for any \( b \) [46]. Also if \( G \) is strictly negative then if \( b \) is strictly positive, there are two solutions (Fig. 2.15), and if \( b \) is strictly negative (Fig. 2.16), there is no solution. The 2-step proj formulation can be written as a
single scalar fixed-point problem:

\[ \lambda_N - \text{proj}_{C^U}(\lambda_N - r \ddot{g}_N) = 0. \quad (2.33) \]

Figs. 2.13-2.15 show that there is a unique solution when \( G > 0 \) and either zero or two solutions when \( G < 0 \). From the point of view of solution existence, the 2-step proj formulation completely agrees with complementarity theory. A difference appears when attempting to find a solution via the fixed-point iteration scheme. In the case with \( \Gamma \) and \( G \) negative, for any initial guess \( \lambda^0_N \) smaller than the second solution \( (\lambda^*_N)_2 \), the iterates will converge to the first solution \( (\lambda^*_N)_1 = 0 \). Iterations beginning with \( \lambda^0_N > (\lambda^*_N)_2 \) will diverge. While a pivoting algorithm solving the LCP formulation could terminate at either solution (or be modified to find both), the fixed-point scheme will only find \( (\lambda^*_N)_1 \). Another interesting difference is that in the case with \( \Gamma \) positive and \( G \) negative, fixed-point iterations are divergent from any starting guess. Thus, one sees that a pivoting method is superior in that it will be able to recognize the case of non-existence of solutions. However, the fixed-point scheme will not be able to determine if divergence is due to a poor initial guess or solution non-existence.

The simple Painleve’s problem discussed here shows that fixed-point iterations can diverge when a solution exists and that this solution will be found by a pivoting scheme. Similar results could also be found by coupled examples [100], like constrained bars [121, Figure 7.5] or rocking blocks [82, Figure 5.5]. However, fixed-point schemes are worth pursuing further to study the exploitation of fine-grained parallelism in the solution process. If on the one side regularization of complementarity formulations, that is, compliant modeling, allows us to exploit parallel implementation of quadratic program solvers, on the other side it does so at the expense of the introduction of a-physical effects. Parallel implementations of fixed-point schemes will not require such model adjustments.
Figure 2.16.: No solution: $\Gamma > 0$, $G < 0$. Acceleration-based model fails [165, Figure 11].
3. Nonsmooth Dynamical Simulation Software

In view of implementations, presented later in this work, the following description focuses on the open source tools SICONOS and MBSim. Besides them, Chrono [128], solfec [111] and daVinci [20] are also open source tools, which are worth to be mentioned and are based on the standard structure of multibody simulation frameworks, that is, the classification in bodies and links [158].

3.1. SICONOS

\begin{center}
\begin{tikzpicture}
  \node (siconos) at (0,0) {SICONOS};
  \node (geom) at (0,-1) {GEOM};
  \node (smesh) at (0,-2) {SMESH};
  \node (kernel) at (0,-3) {Kernel};
  \node (code_aster) at (3,-1) {Code_Aster};
  \node (open_cascade) at (3,-2) {OpenCascade};
  \node (lmgc90) at (3,-3) {LMGC90};
  \node (siconos_mechanics) at (0,-4) {SICONOS: Mechanics};
  \node (siconos_numerics) at (0,-5) {SICONOS: Numerics};
  \node (siconos_kernel) at (0,-6) {SICONOS: Kernel};
  \node (siconos_control) at (0,-7) {SICONOS: Control};
  \node (siconos_frontend) at (0,-8) {SICONOS: FrontEnd};
  \draw (siconos) -- (geom);
  \draw (geom) -- (smesh);
  \draw (smesh) -- (kernel);
  \draw (kernel) -- (open_cascade);
  \draw (open_cascade) -- (lmgc90);
  \draw (open_cascade) -- (siconos_mechanics);
  \draw (siconos_mechanics) -- (siconos_numerics);
  \draw (siconos_numerics) -- (siconos_kernel);
  \draw (siconos_kernel) -- (siconos_control);
  \draw (siconos_control) -- (siconos_frontend);
\end{tikzpicture}
\end{center}

Figure 3.1.: Simulation platform Salome.

SICONOS\(^1\) is a scientific computing software dedicated to SIMulation and CONtrol of NONsmooth dynamical Systems licensed under GNU General Public License and provides simulation and modeling tool. It is hosted by Inria Bipop and was initiated during a European project from 1st September 2002 until 31st August 2006. Nowadays, SICONOS is part of a global simulation network and it is continuously extended. The platform for integration of different simulation software programs is called Salome\(^2\) (Fig. 3.1). It offers itself among others, e.g., a tool for geometry and one for meshing. However due to its generic CORBA\(^3\) and lower level layers, that is, Kernel and Graphical User Interface, it is easy to include additional modules like SICONOS in Salome. OpenCascade\(^4\) can be used for CAD, Code_Aster\(^5\) for generic finite element calculations, LMGC90\(^6\) for the simulation of granular materials. Recently, SICONOS: Mechanics has extended the Salome modules. The project run

\begin{itemize}
  \item \(^1\) http://siconos.gforge.inria.fr
  \item \(^2\) http://www.salome-platform.org
  \item \(^3\) http://www.corba.org/
  \item \(^4\) http://www.opencascade.org
  \item \(^5\) http://www.code-aster.org
  \item \(^6\) http://transfert.lmgc.univ-montp2.fr/LMGC90/
\end{itemize}
from 1st January 2009 until 31st December 2012 and was dedicated to convenient modeling and simulation of mechanical systems in the nonsmooth dynamical system framework. Therefore, the specific plug-in mechanisms of SICONOS are used to define dynamical systems, relations and nonsmooth laws for rather simple kinematics and kinetics or BulletSpaceFilter\(^7\) for complicated collision detection. The final equations are solved by SICONOS.

In SICONOS, we distinguish five parts, Numerics, Kernel, Mechanics, Control and Front End [5].

### 3.1.1. Numerics

The *Numerics* part is written in C and Fortran. It is dedicated to low-level algorithms for the solution of complementarity problems, nonsmooth problems and variational inequalities (Chapter 2).

#### Complementarity Problems

For linear complementarity problems,

\[ \mathbf{w} = \mathbf{q} + M\mathbf{z} , \]
\[ \mathbb{R}^n \ni 0 \leq \mathbf{w} \perp \mathbf{z} \geq 0 \in \mathbb{R}^n , \]

direct solvers, like LEMKE’s algorithm with pivoting, an enumerative scheme, which tests all combinations, and an interface to the PATH solver, are implemented. Fixed-point algorithms contain GAUSS-SEIDEL or successive overrelaxation schemes, projected gradient schemes and LATIN schemes. Algorithms based on the proj or the Fischer-Burmeister formulation use NEWTON-like strategies. For mixed linear complementarity problems the offer for solution algorithms is similar.

For nonlinear complementarity problems,

\[ \mathbf{w} = \mathcal{F}(\mathbf{z}) , \]
\[ \mathbb{R}^n \ni 0 \leq \mathbf{w} \perp \mathbf{z} \geq 0 \in \mathbb{R}^n , \]
either the PATH solver or a direct NEWTON solver based on the proj or Fischer-Burmeister formulation are available.

#### Nonsmooth Problems

Nonsmooth problems from mechanics are frictional contact problems. They can be formulated as NCPs or with an equation based formulation using 1-step or 2-step
proj or Fischer-Burmeister techniques. Fixed-point schemes and Newton schemes are available in SICONOS.

**Variational Inequalities**

Variational inequalities

\[ \mathbf{F}^T (y - z) \geq 0, \quad \forall y \in \mathcal{Y}, \]

with a set \( \mathcal{Y} \) are rigorously treated in the books of STEWART [180] and FACCHINEI [60, 61]. The unilateral contact condition stated as a complementarity formulation

\[ 0 \leq g^{\perp} \lambda \geq 0 \]

can be transformed easily to the variational inequality

\[ g (y - \lambda) \geq 0, \quad \forall y \geq 0. \]

We may find similar applications within relay systems and motivate the implementation of fixed-point and Newton schemes, as well as an interface for the PATH solver for such problem formulations in SICONOS. However, we do not detail variational inequalities in this work.

### 3.1.2. Kernel

![Dynamical system classes in SICONOS](uml.png)

The SICONOS Kernel is written in C++. The Modeling Tools are used to build up a mathematical model from a physical model. We define first-order state variables \( x \), mass matrix \( \tilde{M} \), right-hand side \( \tilde{h} \), contact forces \( \tilde{r} \), generalized positions \( q \), generalized velocities \( v \), mechanical mass matrix \( M \), mechanical right-hand side \( h \), mechanical contact forces \( r \) and the mechanical transformation matrix \( Y \). A DynamicalSystem (Fig. 3.2) represents dynamical equations explicitly and interfaces

- **FirstOrderNonLinearDS**
  
  \[ \tilde{M} (x) \dot{x} = \tilde{h} (x, t) + \tilde{r}, \]
- **LagrangianDS**

\[ M(q) \ddot{q} = h(q,\dot{q},t) + r, \]

- **NewtonEulerDS**

\[ q = Y(q) v, \]
\[ M(q) \dot{v} = h(q,v,t) + r. \]

\[
\begin{array}{c}
\text{EqualityConditionNSL} \\
\text{RelayNSL} \\
\text{ComplementarityConditionNSL} \\
\text{MixedComplementarityConditionNSL}
\end{array}
\]

\[
\begin{array}{c}
\text{MultipleImpactNSL} \\
\text{NewtonImpactNSL} \\
\text{NewtonImpactFrictionNSL}
\end{array}
\]

**Figure 3.3.** Nonsmooth law classes in SICONOS (UML).

A **LagrangianDS** is stated in generalized coordinates and a **NewtonEulerDS** offers not projected equations with arbitrary rotational description. **Interactions** allow to connect several **DynamicalSystems** with consistent **NonsmoothLaws** (Fig. 3.3). For evaluating the **NonsmoothLaw** in its own local kinematic variables, \( \tilde{g} \) and \( g \), and kinetic variables \( \lambda \), a **Relation** (Fig. 3.4) to global kinematic variables, \( x, q, \dot{q} \) and \( v \), and kinetic variables, \( \tilde{r} \) and \( r \), of a **DynamicalSystem** is defined. It additionally belongs to the **Interaction**:

- **FirstOrderR**

\[ \tilde{g} = \tilde{g}(x,\lambda,t), \]
\[ \tilde{r} = \tilde{r}(x,\lambda,t), \]

- **LagrangianR**

\[ g = g(q,\lambda,t), \]
\[ r = r(q,\lambda,t), \]

- **NewtonEulerR**

\[ g = g(q), \]
\[ r = W(q) \lambda. \]

One tries to make use of the relative degree of the **Interaction**, which corresponds to the index of the differential algebraic equation. **NonSmoothDynamicalSystem**
represents the Topology of DynamicalSystems and Interactions by the Boost Graph Library\textsuperscript{8}.

The evaluation of the Topology is done within the Simulation Tools using Boost Basic Linear Algebra and Bindings Library as well as ATLAS\textsuperscript{9}. The first goal is to derive a numerical model for the Simulation out of the physical progress (Fig. 3.5). Two strategies are available, namely TimeStepping and EventDriven. Both of them decompose the desired time interval using NonSmoothDynamicalSystem and EventsManager, which itself schedules TimeDiscretisation and NonSmoothEvent. For each integration step a OneStepIntegrator depending on the chosen strategy generates the discretization of the OneStepNSProblem representing the Interaction. Different integration schemes are available: Moreau-Jean timestepping with projection, Schatzman-Paoli timestepping, Newmark schemes, half-explicit schemes or LSODAR. In a second step, the OneStepNSProblem interfaces the different possible solution schemes implemented in the Numerics part of SICONOS. Both, NonSmoothDynamicalSystem and Simulation build up the whole Model.

Exception handling, global testing and unit tests complete the Kernel.

### 3.1.3. Mechanics

The Mechanics part of SICONOS is written in C++ and offers a convenient description of mechanical systems with friction and impacts, which are an important

---

\textsuperscript{8} http://www.boost.org

\textsuperscript{9} http://math-atlas.sourceforge.net
subclass of dynamical systems. With the concept of bodies, joints and contacts, the nonsmooth relations of SICONOS are implicitly defined.

3.1.4. Control

The Control part of SICONOS is written in C++ and offers a ControlManager, Sensors and Actuators. By defining SensorEvent and ActuatorEvent, they are scheduled within the event handling of SICONOS.

3.1.5. Front End

We would like to mention that it is always possible to read and write XML structures of the Kernel components. Within the FrontEnd, a Python interface is automatically generated by SWIG\(^\text{10}\). A Scilab interface is manually added.

3.2. MBSim

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{mbsim_diagram.png}
\caption{MBSim overview diagram.}
\end{figure}

MBSim [129] is developed at AM-TUM and a private skill group. It is explained in detail in [162]. It is licensed under GNU Lesser General Public License and is written in C++. Figure 3.6 gives an overview of MBSim and its surrounding. Originally,

\(^{10}\) http://www.swig.org
MBSim has been set up for multibody simulation. Hydraulics, electronics, control and power train systems are considered in modules via plug-ins. Linear algebra routines are interfaced by FMatVec [70]. MBSim creates simulation result files in HDF5 file format\(^\text{11}\) by HDF5Serie [93]. H5PlotSerie uses these files for plotting whereas OpenMBV [138] does the visualization. Parallel Co-simulation [75] can be accomplished with MATLAB/Simulink\(^\text{12}\), HySim [144] and KetSim [108]. MBSim consists of a modeling part using C++, XML or a GUI and a simulation part.

![Class hierarchy diagram](#)

**Figure 3.7.** Dynamic system classes in MBSim (UML) [162, Figure 4, adapted].

The class hierarchy of MBSim is similar to the one of SICONOS. Concerning modeling, Objects surely belong to DynamicSystems but also Links, the Interactions of SICONOS, make up a DynamicSystem (Fig. 3.7):

- **Object**

  \[ \dot{q} = Y(q)v, \]
  \[ M(q) \dot{v} = h(q,v,t), \]

- **Link**

  \[ (q,v,\lambda,t) \in N_C, \]
  \[ W(q) \lambda. \]

Element is for data administration. A Graph structure is not available for modeling but is automatically constructed during simulation. Though, DynamicSystems can be ordered in Groups. The top-most DynamicSystem is called DynamicSystemSolver and is similar to SICONOS’ NonSmoothDynamicalSystem. One difference between mechanics and hydraulics is the internal kinematic description (Fig. 3.8). Also the linkage might follow other structural rules (Fig. 3.9). Basic mechanical types are summarized in the following:

- **SpringDampers** connect two Frames with a functional relationship.

- **KineticExcitations** define an excitation function to one Frame.

---

\(^{11}\) http://www.hdfgroup.org

\(^{12}\) http://www.mathworks.com
3. Nonsmooth Dynamical Simulation Software

**Figure 3.8.** Object classes in *MBSim* (UML). [162, Figure 5, adapted]

- Joints connect two Frames with force laws depending on the ideal normal relative kinematics and a constitutive law.
- Contacts are represented by constitutive laws and ContactKinematics on position level using Contour descriptions.

**Figure 3.9.** Link classes in *MBSim* (UML) [162, Figure 7, adapted].

We distinguish between kinematics and kinetics of links. Set-valued and single-valued GeneralizedForceLaws and GeneralizedImpactLaws can be selected like NonsmoothLaw in *SICONOS*. A *SICONOS* Relation is represented in *MBSim* by a Frame or by a Contour together with a ContactKinematics.

From the point of view of simulation, DynamicSystemSolver represents the interface to the Integrator and implements NEWTON and fixed point solvers based on the 2-step proj formulation. It performs tasks of *SICONOS* OneStepNSProblem and OneStepIntegrator. As the Integrator defines the strategy of evaluations, it can be interpreted as *SICONOS* Simulation.

### 3.3. Comparison

*SICONOS* and *MBSim* have a similar architecture. Nevertheless, the structure of *SICONOS* seems to be better concerning encapsulation, e.g., DynamicSystemSolver
seems to be overburdened. In general, \textit{SICONOS} makes more use of external library functionality and software development paradigms. Constraint solvers using nonsmooth complementarity and nonsmooth optimization formulations are available in addition to the 2-step proj formulation solvers of \textit{MBSim}. However, the mechanical development methodology in \textit{MBSim} seems to be more advanced concerning the implementation of the \textit{Newton Euler} formalism and frame recursions. The \textit{SICONOS Relation} is one example for a more general naming concept in \textit{SICONOS} to match applications in control and optimization theory. For only mechanical applications, it is quite abstract.
Part II.

Time-Discontinuous Galerkin Timestepping Schemes
4. Impacting Slider-Crank Mechanism

Figure 4.1.: Slider-crank mechanism with unilateral constraints and friction [167, Figure 1].

The slider-crank mechanism in Fig. 4.1 is a nonlinear benchmark example for an impacting mechanical system with dry friction. Using this example, we will discuss numerical results throughout the following chapters. We take the following text from [167] - the original example is from [69]. The generalized positions \( q \) consist of the angles \( \theta_1, \theta_2 \) and \( \theta_3 \). The generalized velocities \( v \) consist of corresponding angular velocities \( \omega_1, \omega_2 \) and \( \omega_3 \).

The generalized mass matrix satisfies

\[
M : \mathbb{R}^3 \rightarrow \mathbb{R}^{3,3},
q \mapsto M(q) = \begin{pmatrix}
J_1 + l_1^2 \left( \frac{m_1}{4} + m_2 + m_3 \right) & l_1 l_2 \cos(\theta_1 - \theta_2) \left( \frac{m_2}{2} + m_3 \right) & 0 \\
l_1 l_2 \cos(\theta_1 - \theta_2) \left( \frac{m_2}{2} + m_3 \right) & J_2 + l_2^2 \left( \frac{m_2}{4} + m_3 \right) & 0 \\
0 & 0 & J_3
\end{pmatrix}
\] (4.1)

and the nonlinear generalized force can be obtained by

\[
h : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}^3, \quad (q,v) \mapsto h(q,v)
\]

where

\[
h(q,v) = \begin{pmatrix}
-l_1 l_2 \sin(\theta_1 - \theta_2) \left( \frac{m_2}{2} + m_3 \right) \omega_2^2 - \Gamma l_1 \cos \theta_1 \left( \frac{m_1}{4} + m_2 + m_3 \right) \\
l_1 l_2 \sin(\theta_1 - \theta_2) \left( \frac{m_2}{2} + m_3 \right) \omega_1^2 - \Gamma l_2 \cos \theta_2 \left( \frac{m_2}{4} + m_3 \right) \\
0
\end{pmatrix}
\] (4.2)

Thereby, \( l_1 \) and \( l_2 \) are the length of the crank and of the connecting rod, respectively. The inertia of crank, connecting rod and slider consist of translational masses \( m_1, m_2 \) and \( m_3 \), as well as of rotational inertia values \( J_1, J_2 \) and \( J_3 \). The system is subject to gravitation \( \Gamma \) in negative \( y \)-direction.
The length and the height of the slider are given by Figure 4.2.\: generalized force directions are given by the notch is given by and tangential gap functions split up for each corner:

Considering the geometry of the slider according to Fig. 4.2, the nonlinear normal and tangential gap functions split up for each corner:

\[
g_{N_1}(q) = \frac{d}{2} - l_1 \sin \theta_1 - l_2 \sin \theta_2 + a \sin \theta_3 - b \cos \theta_3 ,
\]

\[
g_{N_2}(q) = \frac{d}{2} - l_1 \sin \theta_1 - l_2 \sin \theta_2 - a \sin \theta_3 - b \cos \theta_3 ,
\]

\[
g_{N_3}(q) = \frac{d}{2} + l_1 \sin \theta_1 + l_2 \sin \theta_2 - a \sin \theta_3 - b \cos \theta_3 ,
\]

\[
g_{N_4}(q) = \frac{d}{2} + l_1 \sin \theta_1 + l_2 \sin \theta_2 + a \sin \theta_3 - b \cos \theta_3 ,
\]

\[
g_{T_1}(q) = l_1 \cos \theta_1 + l_2 \cos \theta_2 - a \cos \theta_3 - b \sin \theta_3 ,
\]

\[
g_{T_2}(q) = l_1 \cos \theta_1 + l_2 \cos \theta_2 + a \cos \theta_3 - b \sin \theta_3 ,
\]

\[
g_{T_3}(q) = l_1 \cos \theta_1 + l_2 \cos \theta_2 - a \cos \theta_3 + b \sin \theta_3 ,
\]

\[
g_{T_4}(q) = l_1 \cos \theta_1 + l_2 \cos \theta_2 + a \cos \theta_3 + b \sin \theta_3 .
\]

The length and the height of the slider are 2\(a\) and 2\(b\), respectively. The height of the notch is given by \(d\) and the gap \(c\) is defined by \(d = 2b + 2c\). The matrices of generalized force directions are given by

\[
W_N : \mathbb{R}^3 \rightarrow \mathbb{R}^{3,4} , \quad q \mapsto W_N(q) = \begin{pmatrix}
-l_1 \cos \theta_1 & -l_2 \cos \theta_2 & a \cos \theta_3 + b \sin \theta_3 \\
-l_1 \cos \theta_1 & -l_2 \cos \theta_2 & -a \cos \theta_3 + b \sin \theta_3 \\
l_1 \cos \theta_1 & l_2 \cos \theta_2 & -a \cos \theta_3 + b \sin \theta_3 \\
l_1 \cos \theta_1 & l_2 \cos \theta_2 & a \cos \theta_3 + b \sin \theta_3
\end{pmatrix}^T
\]

\[
W_T : \mathbb{R}^3 \rightarrow \mathbb{R}^{3,4} , \quad q \mapsto W_T(q) = \begin{pmatrix}
-l_1 \sin \theta_1 & -l_2 \sin \theta_2 & a \sin \theta_3 - b \cos \theta_3 \\
-l_1 \sin \theta_1 & -l_2 \sin \theta_2 & -a \sin \theta_3 - b \cos \theta_3 \\
l_1 \sin \theta_1 & l_2 \sin \theta_2 & -a \sin \theta_3 + b \cos \theta_3 \\
l_1 \sin \theta_1 & l_2 \sin \theta_2 & a \sin \theta_3 + b \cos \theta_3
\end{pmatrix}^T
\]
Only the $i$-th column of the matrix $\frac{\partial W^T_N}{\partial q_i} \in \mathbb{R}^{4,3}$ does not vanish. We derive each column of $W^T_N$ with respect to the corresponding generalized coordinate and define

$$Q_N = \begin{pmatrix}
l_1 \sin \theta_1 & l_2 \sin \theta_2 & -a \sin \theta_3 + b \cos \theta_3 \\
l_1 \sin \theta_1 & l_2 \sin \theta_2 & a \sin \theta_3 + b \cos \theta_3 \\
-l_1 \sin \theta_1 & -l_2 \sin \theta_2 & a \sin \theta_3 + b \cos \theta_3 \\
-l_1 \sin \theta_1 & -l_2 \sin \theta_2 & -a \sin \theta_3 + b \cos \theta_3
\end{pmatrix}$$

as well as

$$v = \begin{pmatrix}v_1^2 & v_2^2 & v_3^2\end{pmatrix}^T.$$

Fixed characteristics, the parameter values and the initial values used in this work are given in Table 4.1.

**Table 4.1.** Characteristics of the slider-crank mechanism with unilateral constraints and friction [69].

<table>
<thead>
<tr>
<th>Geometrical characteristics</th>
<th>$l_1 = 0.1530$ m</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$l_2 = 0.3060$ m</td>
</tr>
<tr>
<td></td>
<td>$a = 0.0500$ m</td>
</tr>
<tr>
<td></td>
<td>$b = 0.0250$ m</td>
</tr>
<tr>
<td></td>
<td>$c = 0.0010$ m</td>
</tr>
<tr>
<td>Inertia properties</td>
<td>$m_1 = 0.0380$ kg</td>
</tr>
<tr>
<td></td>
<td>$m_2 = 0.0380$ kg</td>
</tr>
<tr>
<td></td>
<td>$m_3 = 0.0760$ kg</td>
</tr>
<tr>
<td></td>
<td>$J_1 = 7.4 \cdot 10^{-5}$ kgm$^2$</td>
</tr>
<tr>
<td></td>
<td>$J_2 = 5.9 \cdot 10^{-4}$ kgm$^2$</td>
</tr>
<tr>
<td></td>
<td>$J_3 = 2.7 \cdot 10^{-6}$ kgm$^2$</td>
</tr>
<tr>
<td>Force elements</td>
<td>$F = 9.81$ m/s$^2$</td>
</tr>
<tr>
<td>Contact parameters</td>
<td>$\epsilon_{N_1} = \epsilon_{N_2} = \epsilon_{N_3} = \epsilon_{N_4} = 0.4$</td>
</tr>
<tr>
<td></td>
<td>$\epsilon_{T_1} = \epsilon_{T_2} = \epsilon_{T_3} = \epsilon_{T_4} = 0.0$</td>
</tr>
<tr>
<td></td>
<td>$\mu_1 = \mu_2 = \mu_3 = \mu_4 = 0.01$</td>
</tr>
<tr>
<td>Initial conditions</td>
<td>$\theta_{10} = 0.0$</td>
</tr>
<tr>
<td></td>
<td>$\theta_{20} = 0.0$</td>
</tr>
<tr>
<td></td>
<td>$\theta_{30} = 0.0$</td>
</tr>
<tr>
<td></td>
<td>$\omega_{10} = 150.0$ 1/s</td>
</tr>
<tr>
<td></td>
<td>$\omega_{20} = -75.0$ 1/s</td>
</tr>
<tr>
<td></td>
<td>$\omega_{30} = 0.0$ 1/s</td>
</tr>
</tbody>
</table>
5. Half-Explicit Timestepping Schemes on Acceleration Level

The main drawback of classic timestepping schemes (Section 1.4.2) is the lack of problem adaptive accuracy in non-impulsive intervals. However, first, using higher-order but piecewise continuous trial functions for the velocity, and, second, splitting of impulsive and non-impulsive contact reactions, offer the opportunity to both stay consistent and benefit from a higher-order integration of non-impulsive contact reactions. The 'forecasting trapezoidal rule' is a special case for piecewise linear velocity trial functions. This is the simplest scheme, in terms of function evaluations and implementation complexity. We explain and discuss the algorithm step-by-step. The proposed (intermediate) schemes are directly evaluated using the impacting slider-crank mechanism from Chapter 4. Table 5.5 summarizes the timestepping scheme. Due to the acceleration level approach, contact forces are calculated in a natural way. However, we show that drift-off effects may occur. The text is taken from [167].

5.1. Proposed Timestepping Scheme

We propose half-explicit 'forecasting' trapezoidal rules with linear trial functions for the velocity discretization not regarding friction:

\[ q_{i-1,0} = q_{i-1}, \]  \hspace{1cm} (5.1)

\[ q_{i-1,1} = q_{i-1} + \frac{\Delta t_i}{2} \{ Y_{i-1,0} v_{i-1,0} + Y_{i-1,1} v_{i-1,1} \}, \]  \hspace{1cm} (5.2)

\[ q_i = q_{i-1} + \frac{\Delta t_i}{2} \{ Y_{i-1,0} v_{i-1,0} + Y_{i-1,1} v_{i-1,1} \}, \]  \hspace{1cm} (5.3)

\[ v_{i-1,0} = v^+_{i-1}, \]  \hspace{1cm} (5.4)

\[ v_{i-1,1} = v^+_{i-1} + \Delta t_i M_{i-1}^{-1} \left[ h^+_{i-1} + W_{N_{i-1}} \lambda^+_N \right], \]  \hspace{1cm} (5.5)

\[ v^+_{i} = v^+_{i-1} + \frac{\Delta t_i}{2} M_{i-1}^{-1} \left[ h^+_{i-1} + W_{N_{i-1}} \lambda^+_N \right] \]  
\[ + \frac{\Delta t_i}{2} M_{i-1,1}^{-1} \left[ h^+_{i-1,1} + W_{N_{i-1,1}} \lambda^+_N \right] + M_{i}^{-1} W_{N_{i}} \Lambda_{N_{i}}, \]  \hspace{1cm} (5.6)

together with the contact equations on acceleration level

\[ \lambda^+_{N_{i-1}, x_{2(i-1),0}} - \text{proj}_{R_{i}^+} \left[ \lambda^+_{N_{i-1}, x_{2(i-1),0}} - r \bar{g}_{N_{i-1}, x_{2(i-1),0}}^+ \right] = 0, \]  \hspace{1cm} (5.7)
5. Half-Explicit Timestepping Schemes on Acceleration Level

\[
\lambda_{N_i, \mathcal{I}_2^{(i-1),1}}^- - \text{proj}_{\mathbb{R}_+^d} \left[ \lambda_{N_i, \mathcal{I}_2^{(i-1),1}}^- - r \tilde{g}_{N_i, \mathcal{I}_2^{(i-1),1}}^- \right] = 0 .
\]  

Moreover, we need impact equations on velocity level similar to (1.39). However, these shall be considered with respect to the discrete index set of the closed constraints

\[ \mathcal{I}_1^{l} = \left\{ k \in \mathcal{I}_0 : g_{N_k}(q_{i,l}) \leq 0 \right\} , \]

and to the discrete index set of the closed constraints, which stay closed,

\[ \mathcal{I}_2^{l \pm} = \left\{ k \in \mathcal{I}_1^{l} : \dot{g}_{N_k}(q_{i,l}, v_{i,l}^{\pm}) \leq 0 \right\} . \]

For the contact forces we have

\[ \lambda_{N_i}^- \approx \lim_{t \uparrow t_i} \lambda_N(t) , \quad \lambda_{N_i}^+ \approx \lim_{t \downarrow t_i} \lambda_N(t) . \]

The proposed algorithm is the implicit trapezoidal rule with an explicit \textsc{Euler} forecast for the second stage \( v_{i-1,1} \) of the velocity. The procedure is similar to the classic Moreau-Jean timestepping scheme for \( \theta = 1/2 \) in Section 1.4.2, and to the two stage Lobatto schemes, IIIA for position and III for velocity [35, 91, 92].

### 5.1.1. Active Sets

![Figure 5.1: Interpretation of velocity jumps](image)

We consider Fig. 5.1. The generalized position \( q_{i-1} \) and the right-hand limit of the generalized velocity \( v_{i-1}^+ \) are known at \( t_{i-1} \). These values are set to the first stage of the generalized position (5.1) and of the generalized velocity (5.4), respectively. The generalized velocity is assumed to be continuous in the interior of the time interval. In the specific case of linear trial functions, there is a linear velocity propagation until its left-hand limit \( v_{i-1,1} \) at \( t_i \). Hence, the second stage (5.5) of the generalized velocity is calculated with a step of the explicit \textsc{Euler} method. Everything is known but for the right-hand limit of the contact force \( \lambda_{N_i}^+ \). We postpone the question whether to incorporate (5.7) for its calculation. In general, the second stage of the generalized position at \( t_i \) is calculated with the trapezoidal rule (5.2). As the jump at \( t_i \) occurs in such a way that the generalized velocity at \( t_i \) is right-continuous, the left-hand limit of the generalized velocity \( v_i^- \) is corrected by a velocity jump leading
to the right-hand limit of the generalized velocity $v_i^+$. The respective trapezoidal rule (5.6) involves the unknown left-hand limit of the contact force $\lambda_{N_i}$ and the impulse $A_{N_i}$ as well as (5.8) and (1.39) if necessary. The effects of all impulses that would technically occur in the interior of the time interval are summarized at the end of the time interval.

### 5.1.2. Bouncing Ball Example

The procedure of incorporating (5.7)-(5.8) and (1.39) can be implemented straightforward for a decoupled bouncing ball example (Chapter 1 for gravitational acceleration in negative vertical direction) because there is only one single contact possibility [161]. We have two possibilities depending on the sign of $g_{N_{i-1}}$:

- If $g_{N_{i-1}} \leq 0$ at the beginning of the time interval, we do not consider the impulse $A_{N_i} = 0$ at the end of the time interval. This is done in order to take advantage of higher order integration possibilities. If also $\dot{g}_{N_{i-1}} > 0$, then we have $\lambda_{N_{i-1}}^+ = 0$. Otherwise, Equation (5.7) is active for the explicit evaluation of $\lambda_{N_{i-1}}^+$ with a nonlinear equation solver. Thereby, we substitute the circumstantial local acceleration because of (1.2), obtaining
  \[
  \ddot{g}_{N_{i-1}}^+ := W_{N_{i-1}}^T v_{i-1}^+ = W_{N_{i-1}}^T M^{-1}_{i-1} h_{i-1}^+ + W_{N_{i-1}}^T M^{-1}_{i-1} W_{N_{i-1}} \lambda_{N_{i-1}}^+.
  \]

  If additionally $g_{N_{i-1,1}} > 0$ or $\dot{g}_{N_{i-1,1}} > 0$, it is $\lambda_{N_i}^- = 0$. Otherwise, Equation (5.8) is active for the explicit evaluation of $\lambda_{N_i}^-$ with a nonlinear equation solver and the local acceleration is
  \[
  \ddot{g}_{N_i}^- := W_{N_{i-1,1}}^T v_{i-1}^- = W_{N_{i-1,1}}^T M^{-1}_{i-1,1} h_{i-1,1} + W_{N_{i-1,1}}^T M^{-1}_{i-1,1} W_{N_{i-1,1}} \lambda_{N_i}^-.
  \]

- If $g_{N_{i-1}} > 0$ at the beginning of the time interval, then clearly $\lambda_{N_{i-1}}^- = 0$. However, we will not consider the contact force $\lambda_{N_i}^- = 0$ at the end of the time interval because analytically it does not exist during an impact. Concerning the impulse, the additional condition $g_{N_{i-1,1}} > 0$ yields $A_{N_i} = 0$. If $g_{N_{i-1,1}} \leq 0$, Equation (1.39) is active for the implicit solution of both $v_{i}^+$ and $A_{N_i}$ with a nonlinear equation solver and the local velocity
  \[
  \dot{g}_{N_i}^+ := W_{N_i}^T v_{i}^+ = \dot{g}_{N_i}^- + W_{N_i}^T M^{-1}_{i} W_{N_i} A_{N_i}
  \]
  according to (1.5).

For the decoupled bouncing ball example, the proposed half-explicit trapezoidal rule works quite well and shows the expected characteristics in numerical experiments, that is, first order accuracy in impulsive phases and second order accuracy in non-impulsive phases [161]. Estimations have been calculated with the dense output formulas in Chapter C of the appendix.
Problem 5.1 (Bouncing ball: free flight) Discuss the scalar initial value problem

\[ q(0) := 1, \quad v(0) := 0, \quad \dot{q} = v, \quad \dot{v} = -10t^2 \]

in terms of measures.

The analytical solution is given by

\[ q(t) = 1 - \frac{5}{6} t^4, \quad v(t) = -\frac{10}{3} t^3, \quad i(t) = 0. \]

During free flight, the state is of order two for the half-explicit trapezoidal rule and of order one for the Moreau-Jean timestepping scheme with \( \theta = 1 \) (Fig. 5.2). The interaction is zero and resolved exactly. It is not shown in the figure.

![Figure 5.2.: Experimental convergence analysis: free flight [161, Figure 5, adapted].](image)

Problem 5.2 (Bouncing ball: rest phase) Discuss the scalar initial value problem

\[ q(0) := 0, \quad v(0) := 0, \]
\[ \dot{q} = v, \quad \dot{v} = -10t^2 + r, \]
\[ 0 \leq q^\perp r \geq 0 \]

in terms of measures.

The analytical solution is given by
\[ q(t) = 0, \quad v(t) = 0, \quad i(t) = \frac{10}{3} t^3. \]

During rest phase, the interaction is of order two for the half-explicit trapezoidal rule and of order one for the Moreau-Jean timestepping scheme with \( \theta = 1 \) (Fig. 5.3). The state is zero and resolved exactly. It is not shown in the figure.

![Graph](image)

**Figure 5.3.** Experimental convergence analysis: rest phase [161, Figure 6, adapted].

**Problem 5.3 (Bouncing ball: combined analysis)** Given the Newton restitution coefficient \( \epsilon_N = 0.5 \), discuss the scalar initial value problem
\[ q(0) := 1, \quad v(0) := 0, \]
\[ \dot{q} = v, \quad \dot{v} = -2, \]
\[ v^+ = v^- + \max \left\{ 0, - (1 + \epsilon_N) v^- \right\} \text{ if } q = 0 \]

in terms of measures.
The analytical solution is given by

**free flight** $- 0 \leq t < 1$

$q(t) = 1 - t^2, \quad v(t) = -2t, \quad i(t) = 0$

**Zeno state** $- \forall n \in \mathbb{N}_0 : 3 - \frac{1}{2^{n+1}} \leq t < 3 - \frac{1}{2^n}$

$q(t) = -(t - 3)^2 - \frac{3}{2^n} (t - 1) + \frac{1}{2^n - 1} \left(3 - \frac{1}{2^n}\right),$

$v(t) = -2(t - 3) - \frac{3}{2^n},$

$i(t) = \sum_{k=0}^{n} \frac{3}{2^k}.$

For the **combined analysis**, the global error of state and interaction is of order one for both the half-explicit trapezoidal rule and the Moreau-Jean timestepping scheme with $\theta = 1$ (Fig. 5.4).

![Figure 5.4.](image-url) Experimental convergence analysis: combined analysis [161, Figure 7, adapted].

For multi-contact situations it is not clear how to decide which contacts shall be considered active concerning the three different constraint equations (5.7)-(5.8) and (1.39). The half-explicit trapezoidal rule needs further interpretation for a consistent extension to simultaneous contact occurrences and nonlinear dynamics.
5.2. Contact Forces on Acceleration Level

It seems good to use (1.9) together with (1.2) for the calculation of a contact force $\lambda_N$ at a certain time also for more general nonlinear mechanical systems. We again interpret (1.9) on acceleration level

$$\lambda_N = \begin{cases} 0 & \text{if } g_N > 0 \lor \dot{g}_N > 0 , \\ \text{proj}_{\mathbb{R}_0^+} [\lambda_N - r\ddot{g}_N] & \text{else ,} \end{cases}$$

(5.9)

using the 2-step proj formulation row-by-row. Often, we deal with holonomic-scleronomic constraints:

$$\dot{g}_N = W_N^T v ,$$

(5.10)

$$\ddot{g}_N = \sum_i \frac{\partial W_T^N}{\partial q_i} v_i + W_N^T \dot{v} .$$

(5.11)

We can simplify (5.11) to

$$\ddot{g}_N = Q_N v + W_N^T \dot{v}$$

(5.12)

with $v$ containing squared generalized velocity combinations. In comparison to the decoupled bouncing ball example, nonlinearities arose in (5.12) and in the right-hand side $h$. As before, we eliminate $\dot{v}$ in (5.12) by using (1.2):

$$\ddot{g}_N = Q_N v + W_N^T M^{-1} h + G_N \lambda_N$$

with the Delassus matrix [121], or equivalently the mass action matrix [143],

$$G_N = W_N^T M^{-1} W_N .$$

We focus on active contacts, $\mathcal{I}_2$ on acceleration level, and transform (5.9) formally using row-by-row interpretation:

$$\lambda_{N,\mathcal{I}_2} = \text{proj}_{\mathbb{R}_0^+} \left[ \lambda_{N,\mathcal{I}_2} - r (Q_{N,\mathcal{I}_2} v + W_{N,\mathcal{I}_2}^T M^{-1} h + G_{N,\mathcal{I}_2} \lambda_{N,\mathcal{I}_2}) \right] .$$

(5.13)

In the multi-contact case, active contacts might be depending. Hence if we decide to use, e.g., a semi-smooth NEWTON method as nonlinear equation solver for (5.13), we have to switch to semi-smooth variants of the GAUSS-NEWTON method [7, 40, 152].

This means that we systematically have to choose an approximate root $\tilde{\lambda}_{N,\mathcal{I}_2}$ of the
5. Half-Explicit Timestepping Schemes on Acceleration Level

function
\[ f : \mathbb{R}^{|\mathcal{I}_2|} \rightarrow \mathbb{R}^{|\mathcal{I}_2|}, \lambda_{N,\mathcal{I}_2} \mapsto f(\lambda_{N,\mathcal{I}_2}) = \lambda_{N,\mathcal{I}_2} - \text{proj}_{\mathbb{R}^+_{\mathcal{I}_2}}[\lambda_{N,\mathcal{I}_2} - r\hat{g}_{N,\mathcal{I}_2}(\lambda_{N,\mathcal{I}_2})]. \]

With the Moore-Penrose pseudoinverse operator \( \text{pinv} \), the Gauss-Newton algorithm reads
\[
\begin{align*}
\bar{\lambda}_{N,\mathcal{I}_2} &= 0, \quad \bar{f} = f(\bar{\lambda}_{N,\mathcal{I}_2}) \\
\text{while} \quad \|\bar{f}\| > \text{TOL} \\
\nabla f(\bar{\lambda}_{N,\mathcal{I}_2}) &= I - \Theta [\bar{\lambda}_{N,\mathcal{I}_2} - r\hat{g}_{N,\mathcal{I}_2}(\bar{\lambda}_{N,\mathcal{I}_2})] (I - rG_{N,\mathcal{I}_2}) \\
\bar{\lambda}_{N,\mathcal{I}_2} &= \bar{\lambda}_{N,\mathcal{I}_2} - \text{pinv}(\nabla f(\bar{\lambda}_{N,\mathcal{I}_2})) \bar{f} \\
\bar{f} &= f(\bar{\lambda}_{N,\mathcal{I}_2}) \\
\text{end} \\
\end{align*}
\]

The Heaviside function
\[
\Theta : \mathbb{R} \rightarrow \mathbb{R}, \quad x \mapsto \Theta(x) = \begin{cases} 0 & \text{if } x < 0, \\ 1 & \text{else,} \end{cases}
\]
is interpreted row-by-row. Moreover, we use a fixed parameter \( r \) [165].

5.3. Smooth Position and Velocity Prediction

After having calculated the right-hand limit of the contact force \( \lambda_{N,-1}^+ \) at \( t_{i-1} \) according to Section 5.2, we predict the left-hand limit of the generalized velocity at \( t_i \), that is, the second stage of the generalized velocity:
\[
v_{i-1,1} = v_{i-1}^+ + \Delta t_i M_{i-1}^{-1} \left[ h_{i-1}^+ + W_{N_{i-1}} \lambda_{N_{i-1}}^+ \right].
\]
Subsequently, we calculate the second stage of the generalized position:
\[
q_{i-1,1} = q_{i-1} + \frac{\Delta t_i}{2} \left\{ Y_{i-1,0} v_{i-1,0} + Y_{i-1,1} v_{i-1,1} \right\}.
\]

5.4. Assignment of Contact Forces and Impulses

Now, we have a pair of generalized position and generalized velocity at the left-hand limit of \( t_i \) that allows us to calculate the gap function \( g_{N_i} \). We have two possibilities.
• If there is no gap function that has been inactive at \( t_{i-1} \) \((g_{N_k}, (q_{i-1,0}) > 0)\) and becomes active at \( t_i \) \((g_{N_k}, (q_{i-1,1}) \leq 0)\), then no impact has to be considered. We set \( A_{N_i} = 0 \) to fully profit of higher integration possibilities and calculate the left-hand limit of the contact force \( \lambda_{N_i} \) at \( t_i \) according to Section 5.2. Knowing \( \lambda_{N_i} \), we improve the predicted left-hand limit of the generalized velocity at \( t_i \):

\[
v_i^- := v_i^{+} + \frac{\Delta t_i}{2} M_i^{-1} \left[ h_i^{+} + W_{N_i} \lambda_{N_i}^{+} \right] + \frac{\Delta t_i}{2} M_{i-1,1}^{-1} \left[ h_{i-1,1}^{+} + W_{N_i-1,1} \lambda_{N_i}^{-} \right]. \tag{5.16}
\]

• If there is at least one inactive gap function \((g_{N_k}, (q_{i-1,0}) > 0)\) at \( t_{i-1} \) that becomes active \((g_{N_k}, (q_{i-1,1}) \leq 0)\), then an impact occurs in the whole rigidly connected component of the impact source. Hence in this rigidly connected component, the calculation of \( \lambda_{N_i} \) is not consistent and does not improve the accuracy of the scheme. For the ease of description, we set \( \lambda_{N_i} = 0 \) and also \( \lambda_{N_i-1} = 0 \) everywhere in the multibody system and calculate the left-hand limit of the generalized velocity at \( t_i \), that is, the free velocity:

\[
v_i^- := v_i^{+} + \frac{\Delta t_i}{2} \left\{ M_i^{-1} h_i^{+} + M_{i-1,1}^{-1} h_{i-1,1} \right\}. \tag{5.17}
\]

This global consequence does not occur in the decoupled bouncing ball example. For the computation of \( v_i^{+} \) and \( A_{N_i} \), we write (1.14) on velocity level row-by-row:

\[
A_{N_i} = \begin{cases} 
0 & \text{if } g_N(q_{i-1,1}) > 0, \\
\text{proj}_{\mathbb{R}_0^+} \left[ A_{N_i} - r (\dot{\theta}_{N_i}^+ + \epsilon_N \dot{\theta}_{N_i}) \right] & \text{else}.
\end{cases}
\tag{5.18}
\]

Equation (1.5) can be used to eliminate \( \dot{\theta}_{N_i}^+ \), row-by-row resulting in

\[
A_{N_i, x_i^{-1,1}} = \text{proj}_{\mathbb{R}_0^+} \left[ A_{N_i, x_i^{-1,1}} - r \dot{\theta}_{N_i}^+ \right],
\]

with \( G_N := G_N(q_i) \) and

\[
\dot{\theta}_{N_i}^+ \mathcal{X}_i^{-1,1} = G_{N_i,x_i^{-1,1}} A_{N_i,x_i^{-1,1}} + (I + \epsilon_{N,x_i^{-1,1}}) W_{N_i,x_i^{-1,1}}^T v_i^-.
\]

We look for the roots of

\[
f : \mathbb{R}^{\mathcal{X}_i^{-1,1}} \rightarrow \mathbb{R}^{\mathcal{X}_i^{-1,1}},
\]

\[
f(A_{N_i,x_i^{-1,1}}) = A_{N_i,x_i^{-1,1}} - \text{proj}_{\mathbb{R}_0^+} \left[ A_{N_i,x_i^{-1,1}} - r \dot{\theta}_{N_i}^+ \right],
\]
with a semi-smooth Gauss-Newton method as in (5.14). The derivative of $f$ at $X_{N_i}^{\xi-1,1}$ is given by

$$\nabla f(X_{N_i}^{\xi-1,1}) = I - \Theta \left[ X_{N_i}^{\xi-1,1} - r \dot{g}^+_N \right] \left( I - r G_N \right).$$

We have to use structures from classic timestepping schemes and event-driven schemes to define the half-explicit trapezoidal rule.

### 5.5. State Increment

Finally, we update the generalized velocity:

$$v_i^+ = v_i^- + M_i^{-1} W_i X_N,$$  \hspace{1cm} (5.19)  

Hence in the impact-free case, we have $v_i^+ = v_i^-$ as expected. For the generalized position, it is

$$q_i = q_i^- + \frac{\Delta t}{2} \left\{ Y_i \cdot v_i^- + Y_{i-1,1} v_i^{1,1} \right\},$$  \hspace{1cm} (5.20)

### 5.6. Overview

The overall algorithm can be summarized as shown in Fig. 5.5.

### 5.7. First Analysis of Simulation Results

Simulation results for the slider-crank mechanism of Chapter 4 with $\varepsilon_N = 0.4$ and $\Delta t = 10^{-4}$ s are depicted in Figs. 5.6 and 5.7, respectively. Curves are symmetric because of the symmetric initial setting from Table 4.1. Impulsive and non-impulsive periods are separated. Hence, higher-order trial functions can be used in non-impulsive periods whereas consistency is preserved for the impacts. Nevertheless, we improve the curves concerning the following observations. The penetration of the bordering wall occurs because timestepping schemes in general do not detect but only capture events, e.g., the time instant when a gap function closes. Contact activity means $g_{N_i} \leq 0$ and $\dot{g}^+_N \leq 0$ and not $g_{N_i} = 0$ and $\dot{g}^+_N = 0$. We will see that a velocity level discretization of the contact forces is a remedy for the drift-off effect and opens the path to include friction naturally. On acceleration level, the implementation of the friction transitions would also be rather complicated. On velocity level, the nonlinearity in the equations of motion, e.g., $v$, will be reduced. Nonlinearity in general induces the not always decreasing energy trend of the trapezoidal rule which differs from the midpoint rule in the nonlinear regime [176]. On acceleration level,
Specify characteristics, start time \( t = 0 \), end time \( T \), time step size \( \Delta t_i \).

Initialize \( i = 1 \)

For \( t < T \)

**Evaluate left-hand side**
- Evaluate \( q_{i-1,0} = q_{i-1} \) (5.1)
- Evaluate \( v_{i-1,0} = v_{i-1}^+ \) (5.4)
- Evaluate \( \lambda^+_{N_{i-1}} \) on acceleration level \( (I_2(i-1),0) \)

**Predict right-hand side**
- Compute \( v_{i-1,1} = v_{i-1}^+ + \Delta t_i M_{i-1}^{-1} \left[ h_{i-1}^+ + W_{N_{i-1}} \lambda^+_{N_{i-1}} \right] \) (5.5)
- Compute \( q_{i-1,1} = q_{i-1} + \frac{\Delta t_i}{2} \left( Y_{i-1,0} v_{i-1,0} + Y_{i-1,1} v_{i-1,1} \right) \) (5.2)

**Decide strategy**
- If \( \exists k^* : g_{N_k^*} \left( q_{i-1,0} \right) > 0 \lor g_{N_k^*} \left( q_{i-1,1} \right) \leq 0 \) (1.7)
  - Set \( \Lambda_{N_i} = 0 \)
  - Evaluate \( \lambda^-_{N_i} \) on acceleration level \( (I_2(i-1),1) \)
  - Compute \( v_i^- = v_{i-1}^- + \frac{\Delta t_i}{2} M_{i-1}^{-1} \left[ h_{i-1}^- + W_{N_{i-1}} \lambda^-_{N_{i-1}} \right] \)
  - \( + \frac{\Delta t_i}{2} M_{i-1,1}^{-1} \left[ h_{i-1,1}^- + W_{N_{i-1},1} \lambda^-_{N_{i-1}} \right] \) (5.16)
- If \( \exists k^* : g_{N_k^*} \left( q_{i-1,0} \right) > 0 \lor g_{N_k^*} \left( q_{i-1,1} \right) \leq 0 \) (1.7)
  - Set \( \lambda^+_{N_{i-1}} = 0 \) and \( \lambda^-_{N_i} = 0 \)
  - Compute \( v_i^- = v_{i-1}^- + \frac{\Delta t_i}{2} \left( M_{i-1}^{-1} h_{i-1}^- + M_{i-1,1}^{-1} h_{i-1,1} \right) \) (5.17)
  - Compute \( \Lambda_{N_i} \) on velocity level \( (I_1(i-1),1) \)

**State increment**
- Compute \( v_i^+ = v_i^- + \Delta t W_{N_i} \Lambda_{N_i} \) (5.19)
- Compute \( q_i = q_{i-1} + \frac{\Delta t_i}{2} \left( Y_{i-1,0} v_{i-1,0} + Y_{i-1,1} v_{i-1,1} \right) \) (5.20)

**Write result of time step, e.g., \( q_i \) and \( v_i^+ \)**

**Update** \( t \) by \( \Delta t_i \) and increment \( i \) by 1

---

**Figure 5.5.** Half-explicit timestepping scheme on acceleration level [167, Figure 3].

Contacts and impacts are defined by different activity rules using \( I_2 \) and \( I_1 \). As a compromise between acceleration and position level discretizations, the velocity level discretization will unify contact and impact activation rules in Chapter 6.
Figure 5.6.: Normal gap functions of the slider for half-explicit timestepping on acceleration level [167, Figure 4].

Figure 5.7.: Total energy of the slider-crank mechanism for half-explicit timestepping on acceleration level [167, Figure 5].
6. Intermediate: Explicit Timestepping Schemes on Velocity Level

In this chapter, we introduce a velocity level discretization. It is an intermediate step to reduce drawbacks of the half-explicit timestepping schemes on acceleration level (Chapter 5), such as the degree of nonlinearity and the drift-off effect. The method equations are given by

\[ q_{i-1,0} = q_{i-1} , \]  
\[ q_{i-1,1} = q_{i-1} + \frac{\Delta t_i}{2} \{ Y_{i-1,0} v_{i-1,0} + Y_{i-1,1} v_{i-1,1} \} , \]  
\[ q_i = q_{i-1} + \frac{\Delta t_i}{2} \{ Y_{i-1,0} v_{i-1,0} + Y_{i-1,1} v_{i-1,1} \} , \]  
\[ v_{i-1,0} = v_{i-1} , \]  
\[ v_{i-1,1} = v_{i-1} + \Delta t_i M_{i-1}^{-1} \left[ h_{i-1} + W_{N_{i-1}} \lambda_{N_{i-1}} + W_{T_{i-1}} \lambda_{T_{i-1}} \right] , \]  
\[ v_i^+ = v_{i-1} + \frac{\Delta t_i}{2} M_{i-1}^{-1} \left[ h_{i-1} + W_{N_{i-1}} \lambda_{N_{i-1}} + W_{T_{i-1}} \lambda_{T_{i-1}} \right] + \frac{\Delta t_i}{2} M_{i-1}^{-1} \left[ h_{i-1} + W_{N_{i-1}} \lambda_{N_{i}} + W_{T_{i-1}} \lambda_{T_{i}} \right] \]  
\[ + M_{i-1}^{-1} \left[ W_{N_i} \lambda_{N_i} + W_{T_i} \lambda_{T_i} \right] , \]  

\[ \lambda_{N_{i-1},x_{i-1,0}}^+ - \text{proj}_{\mathbb{R}_0^+} \left( \lambda_{N_{i-1},x_{i-1,0}}^+ - r g_{N_{i-1},x_{i-1,0}}^+ \right) = 0 , \]  
\[ \lambda_{N_{i},x_{i-1,1}}^- - \text{proj}_{\mathbb{R}_0^+} \left( \lambda_{N_{i},x_{i-1,1}}^- - r g_{N_{i-1},x_{i-1,1}}^- \right) = 0 , \]

and the active normal impact equations on velocity level (5.18). Then, we can easily include friction, which is naturally stated on velocity level. For the active tangential contact equations on velocity level, we get

\[ \lambda_{t_{i-1},x_{i-1,0}}^+ - \text{proj}_{\mathbb{R}_0^+} \left( \lambda_{t_{i-1},x_{i-1,0}}^+ - r g_{t_{i-1},x_{i-1,0}}^+ \right) = 0 , \]  
\[ \lambda_{T_{i},x_{i-1,1}}^- - \text{proj}_{\mathbb{R}_0^+} \left( \lambda_{T_{i},x_{i-1,1}}^- - r g_{T_{i-1},x_{i-1,1}}^- \right) = 0 , \]

with

\[ \lambda_{T_i} \approx \lim_{t \to t_i} \lambda_T (t) , \quad \lambda_{T_i}^+ \approx \lim_{t \to t_i} \lambda_T^+ (t) . \]
The active tangential impact equations on velocity level are similar to (1.40).

6. Intermediate: Explicit Timestepping Schemes on Velocity Level

6.1. Contact Forces on Velocity Level

As in Section 5.2, we use (1.9) and (1.10) together with (1.2) for the calculation of contact forces $\lambda_N$ and $\lambda_T$. We interpret (1.9) and (1.10) on velocity level

$$\lambda_N = \begin{cases} 0 & \text{if } g_N > 0, \\ \text{proj}_{\mathbb{R}_0^+} [\lambda_N - r g_N] & \text{else}, \end{cases} \quad (6.11)$$

$$\lambda_T = \begin{cases} 0 & \text{if } g_N > 0, \\ \text{proj}_{C_T(\lambda_N)} [\lambda_T - r g_T] & \text{else}, \end{cases} \quad (6.12)$$

using the 2-step proj formulation row-by-row. Thereby now in contrast to Chapter 5, the necessary local kinematic relationships

$$\dot{g}_N = W_T^T v, \quad \dot{g}_T = W_T^T v$$

are already known and we cannot directly calculate the contact forces with (6.11), (6.12). Hence, we start calculating equivalent forces $\lambda_N$ and $\lambda_T$ such that the given local velocities are projected into their respective admissible space, e.g., during one time step — which is an assumption of an explicit representation. That is why, we substitute

$$\dot{g}_{N,\text{proj}} = \dot{g}_N + \Delta t \left\{ W_T^T M^{-1} \tilde{h} + G_N \lambda_N + W_N^T M^{-1} W_T \lambda_T \right\}, \quad (6.13)$$

$$\dot{g}_{T,\text{proj}} = \dot{g}_T + \Delta t \left\{ W_T^T M^{-1} \tilde{h} + W_T^T M^{-1} W_N \lambda_N + G_T \lambda_T \right\}, \quad (6.14)$$

in (1.9) and (1.10) using

$$\tilde{h} : \mathbb{R}^{N_d} \times \mathbb{R}^{N_d} \to \mathbb{R}^{N_d}, \ (q,v) \mapsto \tilde{h}(q,v) = \begin{cases} h(q,v) & 1: \text{more implicit}, \\ 0 & 2: \text{less implicit}. \end{cases}$$

Thereby, it is

$$G_N = W_N^T M^{-1} W_N, \quad G_T = W_T^T M^{-1} W_T.$$ 

We again focus on active contacts and transform (6.11) and (6.12) formally using row-by-row interpretation:

$$\lambda_{N,I_i} = \text{proj}_{\mathbb{R}_0^+} \left[ \lambda_{N,I_i} - r \dot{g}_{N,\text{proj},I_i}(\lambda_{N,I_i},\lambda_{T,I_i}) \right], \quad (6.15)$$

$$\lambda_{T,I_i} = \text{proj}_{C_T(\lambda_{N,I_i})} \left[ \lambda_{T,I_i} - r \dot{g}_{T,\text{proj},I_i}(\lambda_{N,I_i},\lambda_{T,I_i}) \right]. \quad (6.16)$$
As in Chapter 5, we systematically select an approximate root \((\bar{\lambda}_{N,I_1},\bar{\lambda}_{T,I_1})\) of the function

\[
f : \mathbb{R}^{[I_1]} \times \mathbb{R}^{2[I_1]} \to \mathbb{R}^{[I_1]} \times \mathbb{R}^{2[I_1]}, \quad (\lambda_{N,I_1},\lambda_{T,I_1}) \mapsto f(\lambda_{N,I_1},\lambda_{T,I_1})\]

where

\[
f(\lambda_{N,I_1},\lambda_{T,I_1}) = \left(\lambda_{N,I_1} - \text{proj}_{\mathbb{R}_+}^{N} \left[\lambda_{N,I_1} - r\bar{g}_{N,\text{proj,I}_1}(\lambda_{N,I_1},\lambda_{T,I_1})\right]\right)
\]

\[
\quad \left(\lambda_{T,I_1} - \text{proj}_{C_T(\lambda_{N,I_1})}^{T} \left[\lambda_{T,I_1} - r\bar{g}_{T,\text{proj,I}_1}(\lambda_{N,I_1},\lambda_{T,I_1})\right]\right)
\]

with the MOORE-PENROSE pseudoinverse operator \(\text{pinv}\). The GAUSS-NEWTON algorithm reads

\[
(\bar{\lambda}_{N,I_1},\bar{\lambda}_{T,I_1}) = (0,0), \quad \bar{f} = f(\bar{\lambda}_{N,I_1},\bar{\lambda}_{T,I_1})
\]

while \(\|\bar{f}\| > \text{TOL}\)

\[
\nabla f \left(\bar{\lambda}_{N,I_1},\bar{\lambda}_{T,I_1}\right) = \cdots
\]

\[
\quad \cdots = \left( I - \Theta_N \left( I - r\Delta t G_{N,I_1} \right) \right) \left( \Theta_N \left( -r\Delta t \bar{W}_{N,I_1} \lambda^{-1} \bar{W}_{T,I_1} \right) \right)
\]

\[
\quad \left( \Theta_T (r\Delta t \bar{W}_{T,I_1} M^{-1} \bar{W}_{N,I_1}) I - \Theta_T (I - r\Delta t G_{T,I_1}) \right)
\]

\[
\bar{f} = f(\bar{\lambda}_{N,I_1},\bar{\lambda}_{T,I_1})
\]

The Heaviside functions

\[
\Theta_N : \mathbb{R}^{[I_1]} \times \mathbb{R}^{2[I_1]} \to \text{diag}^{[I_1],[I_1]},
\]

\[
\Theta_{N,k} (\lambda_{N,I_1},\lambda_{T,I_1}) = \begin{cases} 0 & \text{if } \lambda_{N,k} - r\bar{g}_{N,k,\text{proj}} < 0, \\ 1 & \text{else}, \end{cases}
\]

\[
\Theta_T : \mathbb{R}^{[I_1]} \times \mathbb{R}^{2[I_1]} \to \text{diag}^{[I_1],[I_1]},
\]

\[
\Theta_{T,k} (\lambda_{N,I_1},\lambda_{T,I_1}) = \begin{cases} 0 & \text{if } \|\lambda_{T,k} - r\bar{g}_{T,k,\text{proj}}\| > \mu \lambda_{N,k}, \\ 1 & \text{else} \end{cases}
\]

are interpreted row-by-row. We use a fixed parameter \(r\).

### 6.2. Smooth Position and Velocity Prediction

As for the acceleration level case, we predict the left-hand limit of the generalized velocity at \(t_i\). This also includes the friction contribution:

\[
v_{i-1,1} = v_{i-1}^{+} + \Delta t_i \lambda_{i-1}^{-1} \left[ h_{i-1}^{+} + \bar{W}_{N,i-1} \lambda_{N,i-1}^{+} + \bar{W}_{T,i-1} \lambda_{T,i-1}^{+} \right].
\]
Subsequently, we calculate the second stage of the generalized position with (6.2).

### 6.3. Assignment of Contact Forces and Impulses

With the gap function \( g_{N_{i-1,1}} \), we can distinguish non-impulsive and impulsive phases.

- For non-impulsive phases, we have

\[
\mathbf{v}_i^- = \mathbf{v}_i^+ + \frac{\Delta t_i}{2} M_{i-1}^{-1} \left[ \mathbf{h}_{i-1} + \mathbf{W}_{N_{i-1},1} \lambda_{N_{i-1}}^+ + \mathbf{W}_{T_{i-1},1} \lambda_{T_{i-1}}^+ \right] \\
+ \frac{\Delta t_i}{2} M_{i-1,1}^{-1} \left[ \mathbf{h}_{i-1,1} + \mathbf{W}_{N_{i-1,1},1} \lambda_{N_{i,1}}^- + \mathbf{W}_{T_{i-1,1},1} \lambda_{T_{i,1}}^- \right].
\]  

(6.18)

- For impulsive phases, we remind of the free velocity (5.17) and of (1.14) in its row-by-row description (5.18). Then, we write also (1.15) on velocity level row-by-row:

\[
\mathbf{A}_{T_i} = \begin{cases}
0 & \text{if } g_{N_{i-1,1}} > 0, \\
\text{proj}_{C_T}(-A_{N_{i}}) - \mathbf{r} \left( \dot{g}_{N_{i}}^+ + \varepsilon_T \dot{g}_{T_i}^- \right) & \text{else}.
\end{cases}
\]  

(6.19)

Elimination of \( \dot{g}_{N_i}^- \) and \( \dot{g}_{T_i}^+ \) for active contacts on position level includes friction contributions yields

\[
\mathbf{A}_{N_{i,i-1,1}} = \text{proj}_{\mathbb{R}_0^+} \left[ \mathbf{A}_{N_{i,i-1,1}}^- - \mathbf{r} \dot{g}_{N_{i,i-1,1}}^+ \right],
\]

\[
\mathbf{A}_{T_{i,i-1,1}} = \text{proj}_{C_T}(-\mathbf{A}_{N_{i,i-1,1}}) \left[ \mathbf{A}_{T_{i,i-1,1}}^- - \mathbf{r} \dot{g}_{T_{i,i-1,1}}^+ \right].
\]

We solve for \((\mathbf{A}_{N_{i,i-1,1}}, \mathbf{A}_{T_{i,i-1,1}})\) as roots of a summarizing function

\[
\mathbf{f} : \mathbb{R}^{\lvert T_i \rvert} \times \mathbb{R}^{2\lvert T_i \rvert} \rightarrow \mathbb{R}^{\lvert T_i \rvert} \times \mathbb{R}^{2\lvert T_i \rvert}
\]

with a semi-smooth GAUSS-NEWTON method as in (6.17).

### 6.4. State Increment

Finally, we update the generalized velocity by

\[
\mathbf{v}_i^+ = \mathbf{v}_i^- + \mathbf{M}_{i}^{-1} \left( \mathbf{W}_{N_i} \mathbf{A}_{N_i} + \mathbf{W}_{T_i} \mathbf{A}_{T_i} \right)
\]  

(6.20)

and the generalized position by (6.3).
6.5. Overview

Specify characteristics, start time $t = 0$, end time $T$, time step size $\Delta t_i$

Initialize $i = 1$
For $i < T$

Evaluate left-hand side
Evaluate $q_{i-1,0} = q_{i-1}$ (6.1)
Evaluate $v_{i-1,0} = \dot{v}_{i-1}$ (6.4)
Evaluate $\lambda_{N,i-1}^+, \lambda_{T,i-1}^+$ on velocity level $(I_i^{i-1,0})$

Predict right-hand side
Compute $v_{i-1,1} = v_{i-1}^* + \Delta t_i M_{i-1}^{-1} \left[ h_{i-1}^+ + W_{N,i-1} \lambda_{N,i-1}^+ + W_{T,i-1} \lambda_{T,i-1}^+ \right]$ (6.5)
Compute $q_{i-1,1} = q_{i-1}^* + \frac{\Delta t_i}{2} \left( Y_{i-1,0} v_{i-1,0} + Y_{i-1,1} v_{i-1,1} \right)$ (6.6)

Decide strategy
if $\exists k^* : g_{N,i}^* \left( q_{i-1,0} \right) > 0 \land g_{N,i}^* \left( q_{i-1,1} \right) \leq 0$ (1.7)
Set $A_N = 0$, $A_T = 0$
Evaluate $\lambda_{N,i}^+, \lambda_{T,i}^+$ on velocity level $(I_i^{i-1,1})$
Compute $v_i^- = v_{i-1}^* + \frac{\Delta t_i}{2} \left( h_{i-1}^* + W_{N,i-1} \lambda_{N,i-1}^+ + W_{T,i-1} \lambda_{T,i-1}^+ \right)$
Compute $v_i^+ = v_{i-1}^* + \frac{\Delta t_i}{2} \left( M_{i-1}^{-1} h_{i-1}^* + M_{i-1,1}^{-1} h_{i-1,1} \right)$ (5.17)

State increment
Compute $v_i^+ = v_i^- + M_{i}^{-1} \left( W_{N,i} A_N + W_{T,i} A_T \right)$ (6.20)
Compute $q_i = q_{i-1} + \frac{\Delta t_i}{2} \left( Y_{i-1,0} v_{i-1,0} + Y_{i-1,1} v_{i-1,1} \right)$ (6.3)

Write result of time step, e.g., $q_i$ and $v_i^+$

Update $t$ by $\Delta t_i$ and increment $i$ by 1

Postprocessing

Figure 6.1.: Explicit timestepping scheme on velocity level [166, Figure 7].

The overall algorithm can be summarized as shown in Fig. 6.1.

6.6. Second Analysis of Simulation Results

Simulation results of the slider-crank mechanism in Fig. 4.1 of Chapter 4 with $\varepsilon_N = 0.4$, $\Delta t = 10^{-4}$ s and $\mathbf{h} \equiv \mathbf{0}$ are depicted in Fig. 6.2. The drift-off effect is quite strong, even stronger than for the half-explicit acceleration level approach in Chapter 5. We observe that the choice $\mathbf{h} \equiv \mathbf{h}$ induces bad oscillatory behavior into the nonlinear slider-crank mechanism, that is, loss of contact. This cannot be
Figure 6.2.: Normal gap functions of the slider for explicit timestepping on velocity level [166, Figure 8].

resolved even by keeping the activation rule from the left to the right interval end ($I_{1}^{i-1,1} = I_{1}^{i-1,0}$, not shown). An implicit calculation of the contact forces is proposed as a remedy in Chapter 7.
7. Half-Explicit Timestepping Schemes on Velocity Level

In the following, we introduce the final version of the proposed schemes by correctly applying velocity level discretizations in an implicit way. With this method, it is possible to reduce the degree of nonlinearity and the drift-off effect from half-explicit timestepping schemes on acceleration level (Chapter 5).

7.1. Proposed Timestepping Scheme

Using an implicit evaluation of constraints [27, 28, 134, 135, 12, 91, 92], we do not have a projection anymore as in (6.13) and (6.14) for calculating both $\lambda^+_{N_{i-1}}$, $\lambda^+_{T_{i-1}}$ and $\lambda^-_{N_{i}}$, $\lambda^-_{T_{i}}$. Instead, we solve the following equations for the second stage of the velocity

$$0 = \lambda^+_{N_{i-1}, T^{-1}_{i-0}} - \text{proj}_{R^0}^{+} \left( \lambda^+_{N_{i-1}, T^{-1}_{i-0}} - rW^T_{N_{i}, T_{i-1}, 0} (q_{i-1} + \Delta t_i, v^+_{i-1}) v_{i-1, 1} \right),$$

$$0 = \lambda^+_{T_{i-1}, T^{-1}_{i-0}} \cdots - \text{proj}_{C_T}^{+} (\lambda^+_{N_{i-1}, T^{-1}_{i-0}} - rW^T_{T_{i}, T_{i-1}, 0} (q_{i-1} + \Delta t_i, v^+_{i-1}) v_{i-1, 1}).$$

Moreover assuming that the output equation of the trapezoidal rule defines a virtual third stage, we solve the equations

$$0 = \lambda^-_{N_{i-1}, T^{-1}_{i-1}} - \text{proj}_{R^0}^{+} \left( \lambda^-_{N_{i-1}, T^{-1}_{i-1}} - rW^T_{N_{i}, T_{i-1}, 1} v^-_{i} \right),$$

$$0 = \lambda^-_{T_{i}, T^{-1}_{i-1}} - \text{proj}_{C_T}^{+} (\lambda^-_{N_{i}, T^{-1}_{i-1}} - rW^T_{T_{i}, T_{i-1}, 1} v^-_{i}).$$

It is important to evaluate $W$ on the right-hand side of the interval for a correct calculation of the next local velocity. As we do not know its value for the second stage of the velocity, we have to calculate a prediction, which, however, is not used for the calculation of any index set. The virtual third stage of the velocity coincides with the output equation of the trapezoidal rule. Hence, the velocity level constraint is automatically satisfied also in the successive time step. The overall algorithm can be summarized as shown in Fig. 7.2.
Figure 7.1.: Normal gap functions of the slider for half-explicit timestepping on acceleration (dashed) and velocity level (solid) [167, Figure 7].

7.2. Third Analysis of Simulation Results

With the half-explicit evaluation of the constraints, we get less drift of the velocity level discretization in comparison to the acceleration level discretization (Fig. 7.1) discussing the slider-crank mechanism in Fig. 4.1 of Chapter 4 with $\varepsilon_N = 0.4$ and $\Delta t = 10^{-4}$ s.

7.3. Half-Explicit Timestepping Schemes on Velocity Level in the Sense of Brasey

We can interpret the half-explicit trapezoidal rule with the notation of [27]

\[
q_{i-1,0} = q_{i-1},
\]

\[
q_{i-1,1} = q_{i-1} + \Delta t_i \mathbf{y}_{i-1,0} \mathbf{v}_{i-1}^+,
\]

\[
v_{i-1,0} = v_{i-1}^+,
\]

\[
v_{i-1,1} = v_{i-1}^+ + \Delta t_i M_{i-1}^{-1} \left[ \mathbf{h}_{i-1}^+ + W_{N_i-1}^+ \lambda_{N_i-1}^+ + W_{T_i-1}^+ \lambda_{T_i-1}^+ \right],
\]

\[
0 = \lambda_{N_i-1,1}^+ - \text{proj}_{B_0^+} \left( \lambda_{N_i-1,1}^+ - r W_{N_i,1}^T (q_{i-1,1}) v_{i-1,1} \right),
\]

\[
0 = \lambda_{T_i-1,1}^+ - \text{proj}_{C_T} (\lambda_{T_i-1,1}^+ - r W_{T_i,1}^T (q_{i-1,1}) v_{i-1,1}),
\]
7.4. Interpretation and Generalization

for the first stage unknowns $\mathbf{v}_{i-1,1}$, $\mathbf{x}_{N_{i-1}}^{+}$ and $\mathbf{x}_{T_{i-1}}^{+}$. For this interpretation, we have to change (7.2), e.g., to a forward Euler step. Then, we proceed:

$$\mathbf{q}_{i} = \mathbf{q}_{i-1} + \frac{\Delta t_i}{2} \left\{ \mathbf{Y}_{i-1,0} \mathbf{v}_{i-1,0} + \mathbf{Y}_{i-1,1} \mathbf{v}_{i-1,1} \right\},$$

$$\mathbf{v}_{i}^{+} = \mathbf{v}_{i-1}^{+} + \frac{\Delta t_i}{2} \mathbf{M}_{i-1}^{-1} \left[ \mathbf{h}_{i-1}^{+} + \mathbf{W}_{N_{i-1}} \mathbf{x}_{N_{i-1}}^{+} + \mathbf{W}_{T_{i-1}} \mathbf{x}_{T_{i-1}}^{+} \right]$$

$$+ \frac{\Delta t_i}{2} \mathbf{M}_{i-1}^{-1} \left( \mathbf{h}(\mathbf{q}_{i}, \mathbf{v}_{i-1,1}) + \mathbf{W}_{N_{i}} \mathbf{x}_{N_{i}}^{+} + \mathbf{W}_{T_{i}} \mathbf{x}_{T_{i}}^{+} \right),$$

$$0 = \mathbf{x}_{N_{i},\mathbf{I}}^{+} - \text{proj}_{\text{IR}_{i}^{+}} \left( \mathbf{x}_{N_{i},\mathbf{I}} - r \mathbf{W}_{N_{i},\mathbf{I}}^{T} \mathbf{v}_{i}^{-} \right),$$

$$0 = \mathbf{x}_{T_{i},\mathbf{I}}^{+} - \text{proj}_{C_{r}(\mathbf{C}_{N_{i}})} \left( \mathbf{x}_{T_{i},\mathbf{I}} - r \mathbf{W}_{T_{i},\mathbf{I}}^{T} \mathbf{v}_{i}^{-} \right).$$

The unknowns for the second stage are $\mathbf{v}_{i}^{+}$, $\mathbf{x}_{N_{i}}^{+}$ and $\mathbf{x}_{T_{i}}^{+}$. The impact equations add only missing contributions, because they are also stated on velocity level and, as the case may be, automatically satisfied. Concerning efficient evaluations, however, we implement a robust procedure as shown in Fig. 7.2 with

$$\mathbf{v}_{i}^{+} = \mathbf{v}_{i}^{+} + \mathbf{M}_{i}^{-1} \left[ \mathbf{W}_{N_{i}} \mathbf{A}_{N_{i}} + \mathbf{W}_{T_{i}} \mathbf{A}_{T_{i}} \right],$$

$$0 = \mathbf{A}_{N_{i},\mathbf{I}}^{+} - \text{proj}_{\text{IR}_{i}^{+}} \left( \mathbf{A}_{N_{i}} - r \mathbf{\dot{g}}_{N_{i},\mathbf{I}}^{+} \right),$$

$$0 = \mathbf{A}_{T_{i},\mathbf{I}}^{+} - \text{proj}_{C_{r}(\mathbf{C}_{N_{i}})} \left( \mathbf{A}_{T_{i}} - r \mathbf{\dot{g}}_{T_{i},\mathbf{I}}^{+} \right).$$

The unknowns are $\mathbf{v}_{i}^{+}$, $\mathbf{A}_{N_{i}}$ and $\mathbf{A}_{T_{i}}$.

Problem B.2 in the appendix is a generalization of the "forecasting trapezoidal rule" (Fig. 7.2) within time-discontinuous Galerkin methods [161]. The constraints for non-impulsive interactions $\left( \mathbf{q}_{i-1,1}^{h}, \mathbf{v}_{i-1,1}^{h}, \mathbf{r}_{i}, \mathbf{t}_{i} \right) \in \mathcal{N}_{C}$, and for impulsive interactions $\left( \mathbf{q}_{i}, \mathbf{v}_{i}^{h+}, \mathbf{p}_{i}, \mathbf{t}_{i} \right) \in \mathcal{N}_{I}$ may be calculated on velocity level. Because of the evaluation of the interpolation polynomials in Problem B.2, the internal stages for position and velocity as well as their corresponding constraints are usually coupled. For a half-explicit time-discretization, adaptations are necessary, possibly disconnecting from time-discontinuous Galerkin methods.

The trapezoidal rule as a locally second order scheme suffices for most practical considerations and industrial simulations. Applying the half-explicit time-discretization with the notation of [27] as base-integration scheme results in the following generalization:

$$\mathbf{q}_{i-1,1} = \mathbf{q}_{i-1} + \Delta t_i \sum_{k<l} \mathbf{a}_{k,l} \mathbf{Y}_{i-1,k} \mathbf{v}_{i-1,k},$$

$$\mathbf{v}_{i-1,1} = \mathbf{v}_{i-1}^{+} + \Delta t_i \sum_{k<l} \mathbf{a}_{k,l} \mathbf{M}_{i-1,k}^{-1} \left[ \mathbf{h}_{i-1,k} + \mathbf{W}_{N_{i-1,k}} \mathbf{x}_{N_{i-1,k}} + \mathbf{W}_{T_{i-1,k}} \mathbf{x}_{T_{i-1,k}} \right],$$

$$\mathbf{q}_{i} = \mathbf{q}_{i-1} + \frac{\Delta t_i}{2} \left\{ \mathbf{Y}_{i-1,0} \mathbf{v}_{i-1,0} + \mathbf{Y}_{i-1,1} \mathbf{v}_{i-1,1} \right\},$$

$$\mathbf{v}_{i}^{+} = \mathbf{v}_{i-1}^{+} + \frac{\Delta t_i}{2} \mathbf{M}_{i-1}^{-1} \left[ \mathbf{h}_{i-1}^{+} + \mathbf{W}_{N_{i-1}} \mathbf{x}_{N_{i-1}}^{+} + \mathbf{W}_{T_{i-1}} \mathbf{x}_{T_{i-1}}^{+} \right],$$

$$0 = \mathbf{x}_{N_{i},\mathbf{I}}^{+} - \text{proj}_{\text{IR}_{i}^{+}} \left( \mathbf{x}_{N_{i},\mathbf{I}} - r \mathbf{W}_{N_{i},\mathbf{I}}^{T} \mathbf{v}_{i}^{-} \right),$$

$$0 = \mathbf{x}_{T_{i},\mathbf{I}}^{+} - \text{proj}_{C_{r}(\mathbf{C}_{N_{i}})} \left( \mathbf{x}_{T_{i},\mathbf{I}} - r \mathbf{W}_{T_{i},\mathbf{I}}^{T} \mathbf{v}_{i}^{-} \right).$$
This first set of equations works as follows. For step $l$, we know or easily calculate the current and all past positions - we also know all past velocities. The current velocity and the past Lagrange multiplier are unknown. That is why, we define the constraint in such a way that it is satisfied for the current velocities but the past Lagrange multiplier. Here, we have already included one improvement in comparison to Section 7.3. As we know the current position, we can exploit this for the current index set as also shown in Section 9.2. At the end of the stage iteration, we know all position and velocity stages, but the last Lagrange multiplier stage is missing. For this reason, we use the output equation to solve for it: 

$$q_i = q_{i-1} + \Delta t_i \sum_k b_k Y_{i-1,k} v_{i-1,k},$$

$$v_i^- = v_{i-1}^- + \Delta t_i \sum_k b_k M_{i-1,k}^{-1} \left[h_{i-1,k} + W_{N_i-1,k} \lambda_{N_i-1,k} + W_{T_{i-1,k}} \lambda_{T_{i-1,k}} \right],$$

$$0 = \lambda_{N_i-1,s,t_i}^i - \text{proj}_{R_0^+} \left( \lambda_{N_i-1,s,t_i}^i - r W_{N_i,t_i}^T \right),$$

$$0 = \lambda_{T_i-1,s,t_i}^i - \text{proj}_{C_T} \left( \lambda_{T_i-1,s,t_i}^i - r W_{T_i,t_i}^T \right).$$

The evaluations are meant with respect to $t_{i-1,l} = t_{i-1} + c_l \Delta t_i$. Coefficients $a_{lk}$, $b_l$ and $c_l$ can be defined as usual [91, 92]. An example for a set of coefficients is given in [27]. At the end of the algorithm, we correct impulses according to (7.7)-(7.9) if necessary.
Specify characteristics, start time $t = 0$, end time $T$, time step size $\Delta t$

Initialize $i = 1$

For $t < T$

*Evaluate left-hand side*

Evaluate $q_{i-1,0} = q_{i-1}$ (6.1)

Evaluate $v_{i-1,0} = v_{i-1}^+$ (6.4)

*Predict right-hand side*

Compute Lagrange multipliers on velocity level

\[
\lambda_{N_i-1,Z_i}^+ - \text{proj}_{R_T^+} \left( \lambda_{N_i-1,Z_i}^+- rW^T_{N_i,Z_i} (q_{i-1} + \Delta t, v_{i-1}) v_{i-1,1} \right) = 0
\]

\[
\lambda_{T_i-1,Z_i}^+ - \text{proj}_{C_T} (\lambda_{N_i-1,Z_i}^+ - rW^T_{T_i,Z_i} (q_{i-1} + \Delta t, v_{i-1}) v_{i-1,1}) = 0
\]

with $v_{i-1,1} = v_{i-1}^+ + \Delta t M_{i-1}^{-1} \left[ h_{i-1}^{-1} + W_{N_i-1} \lambda_{N_i-1}^+ + W_{T_i-1} \lambda_{T_i-1}^+ \right]$ (6.5)

Compute $q_{i-1,1} = q_{i-1} + \Delta t \left\{ Y_{i-1,0}v_{i-1,0} + Y_{i-1,1}v_{i-1,1} \right\}$ (6.2)

*Decide strategy*

if $\exists k^* : g_{N_k^*} (q_{i-1,0}) > 0 \land g_{N_k^*} (q_{i-1,1}) \leq 0$ (1.7)

Set $A_{N_i} = 0, A_{T_i} = 0$

Compute Lagrange multipliers on velocity level

\[
\lambda_{N_i-1,Z_i}^- - \text{proj}_{R_T^-} \left( \lambda_{N_i-1,Z_i}^- - rW^T_{N_i,Z_i} v_{i}^- \right) = 0
\]

\[
\lambda_{T_i-1,Z_i}^- - \text{proj}_{C_T} (\lambda_{N_i-1,Z_i}^- - rW^T_{T_i,Z_i} v_{i}^-) = 0
\]

with $v_{i}^- = v_{i-1}^+ + \Delta t \left[ M_{i-1}^{-1} \left[ h_{i-1}^- + W_{N_i-1} \lambda_{N_i-1}^- + W_{T_i-1} \lambda_{T_i-1}^- \right] \right]$

\[
\lambda_{N_i-1}^+ = 0, A_{T_i} = 0
\]

if $\exists k^* : g_{N_k^*} (q_{i-1,0}) > 0 \land g_{N_k^*} (q_{i-1,1}) \leq 0$ (1.7)

Set $\lambda_{N_i-1}^+ = 0, A_{T_i-1} = 0$ and $\lambda_{N_i}^+ = 0, A_{T_i} = 0$

Compute $v_{i}^- = v_{i-1}^+ + \Delta t \left\{ M_{i-1}^{-1} \left[ h_{i-1}^- \right] + M_{i-1}^{-1} \left[ h_{i-1}^- \right] \right\}$ (5.17)

Compute $A_{N_i}, A_{T_i}$ on velocity level ($Z_i^{-1,1}$)

*State increment*

Compute $v_{i}^+ = v_{i}^- + M_{i}^{-1} \left( W_{N_i} A_{N_i} + W_{T_i} A_{T_i} \right)$ (6.20)

Compute $q_{i} = q_{i-1} + \Delta t \left\{ Y_{i-1,0}v_{i-1,0} + Y_{i-1,1}v_{i-1,1} \right\}$ (6.3)

Write result of time step, e.g., $q_i$ and $v_i^+$

Update $t$ by $\Delta t$, and increment $i$ by 1

*Postprocessing*

**Figure 7.2.** Half-explicit time-stepping scheme on velocity level [167, Figure 6].
8. Multi-Contact Examples

In this chapter, we discuss two examples to show the benefits of the proposed schemes based on [167]. The slider-crank mechanism is a nonlinear example and the bouncing ball in a box is a linear example. We compare computational results, convergence and computing time of the half-explicit timestepping schemes on acceleration and velocity level, with those of the classic explicit Moreau-Jean timestepping scheme. Computations were done with Matlab. Thereby index sets are calculated without any additional tolerances. Constraints are calculated with a tolerance of $10^{-12}$ to satisfy the constraint equations.

8.1. Slider-Crank Mechanism

Figure 8.1.: Movement of the center of gravity of the slider for different coefficients of restitution and half-explicit timestepping on velocity level [166, Figure 11].
For the slider-crank mechanism in Chapter 4, we give some additional simulation results without regarding friction. Calculations have been performed with the half-explicit timestepping scheme on velocity level. Figure 8.1 shows the slider’s movement in the notch for different coefficients of restitution. The trajectories of the three generalized coordinates and generalized velocities are depicted in Fig. 8.2. Figure 8.3 shows the splitting of the different Lagrange multipliers and therefore the possibility to achieve locally a higher order discretization. Curves are symmetric because of the symmetric setting from Table 4.1.

\[ \text{Figure 8.2.: Angles and angular velocities of the slider-crank mechanism for half-explicit timestepping on velocity level [166, Figure 12].} \]

### 8.1.1. Convergence and Computing Time for the Bilateral Case

For a bilateral slider-crank mechanism without friction \((c = 0 \text{ m})\), we compare the half-explicit timestepping scheme on velocity level with the classic explicit Moreau-Jean timestepping scheme. We analyze computing time and convergence of the schemes in comparison with a corresponding SIMPACK\(^1\) model, which is integrated with the SODASRT2 solver and a high tolerance \(10^{-12}\).

\[ \text{Table 8.1.: Bilateral slider-crank example: comparison of the error of the explicit Moreau-Jean and the half-explicit timestepping scheme on velocity level [167, Table 2].} \]

<table>
<thead>
<tr>
<th>(\Delta t [\text{s}])</th>
<th>(10^{-3})</th>
<th>(10^{-4})</th>
<th>(10^{-5})</th>
</tr>
</thead>
<tbody>
<tr>
<td>error of the Moreau-Jean timestepping scheme</td>
<td>(6.9 \cdot 10^{-1})</td>
<td>(2.1 \cdot 10^{-1})</td>
<td>(6.6 \cdot 10^{-2})</td>
</tr>
<tr>
<td>error of the half-explicit timestepping scheme</td>
<td>(4.2 \cdot 10^{-1})</td>
<td>(2.4 \cdot 10^{-2})</td>
<td>(8.4 \cdot 10^{-4})</td>
</tr>
</tbody>
</table>

\(^1\) [http://www.simpack.com/](http://www.simpack.com/)
8.1. Slider-Crank Mechanism

Independent of different time step-sizes, the relative overhead of the half-explicit timestepping scheme on velocity level in comparison with the classic explicit Moreau-Jean timestepping scheme is about 1.2. Hence, the computation time per time step is about 1.2 times larger for the half-explicit timestepping scheme on velocity level. We calculate the error with respect to the reference SIMPACK solution for different time step-sizes. Thereby, we compute the differences of the generalized coordinates for the considered time instances and arrange them in a matrix. The 2-norm of this matrix results in Table 8.1. The overall relative cost to achieve a certain tolerance is given by the relation of the fractions of the relative overhead and the necessary time step-size. For a given tolerance $10^{-2}$, we have a relative cost $1.2 \cdot 10^{-1}$ for the half-explicit timestepping scheme on velocity level in comparison to the classic explicit Moreau-Jean timestepping scheme. Thus, it makes sense to apply higher-order approximations for appropriate examples.

8.1.2. Convergence and Computing Time for the Unilateral Case

Regarding a unilateral slider-crank mechanism with friction, we compare the half-explicit timestepping scheme on velocity level with the classic explicit Moreau-Jean timestepping scheme. Thereby, we analyze computing time and convergence of the schemes assuming a reference solution given by a simulation of the half-explicit timestepping on velocity level with $\Delta t = 10^{-7}$ s.

The relative overhead of the half-explicit timestepping scheme on velocity level is about 1.15. Errors are indicated in Table 8.2. The results for the half-explicit timestepping scheme on velocity level are better perhaps because of the possibility to achieve a higher order, locally. For a given tolerance $10^{-2}$, the overall relative cost is
Table 8.2.: Unilateral slider-crank example: comparison of the error of the explicit Moreau-Jean and the half-explicit timestepping scheme on velocity level [167, Table 3].

<table>
<thead>
<tr>
<th>$\Delta t$ [s]</th>
<th>$10^{-3}$</th>
<th>$10^{-4}$</th>
<th>$10^{-5}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>error of the Moreau-Jean timestepping scheme</td>
<td>$7.1 \cdot 10^{-1}$</td>
<td>$1.5 \cdot 10^{-1}$</td>
<td>$4.5 \cdot 10^{-2}$</td>
</tr>
<tr>
<td>error of the half-explicit timestepping scheme</td>
<td>$4.7 \cdot 10^{-1}$</td>
<td>$1.8 \cdot 10^{-2}$</td>
<td>$3.3 \cdot 10^{-3}$</td>
</tr>
</tbody>
</table>

$1.15 \cdot 10^{-1}$ for the half-explicit timestepping scheme on velocity level in comparison to the classic explicit Moreau-Jean timestepping scheme. Hence, the relative overhead could be compensated by an adaptive time-step size and a control of the accuracy.

8.2. Bouncing Ball in a Box

The linear bouncing ball in a box is an example to discuss the drift-off effect and to explain the energy behavior of half-explicit timestepping schemes based on the trapezoidal rule.

A bouncing ball of radius $R$ in a planar box of width $a$ and height $b$ (Fig. 8.4) is described by the two translational coordinates, $x$ and $y$, of its center of gravity. The rigid ball with mass $m$ is subject to gravitation $\Gamma$ in $-\left(\cos(\pi/6) \sin(\pi/6)\right)^T$ direction. The normal gap functions $g_N$ are shown in Fig. 8.4. The vector containing the coefficients of restitution is $\epsilon_N$. Friction is not considered. Fixed characteristics are given in Table 8.3.

The generalized mass matrix satisfies

$$M : \mathbb{R}^2 \to \mathbb{R}^{2 \times 2}, \, q \mapsto M(q) = \begin{pmatrix} m & 0 \\ 0 & m \end{pmatrix},$$

(8.1)
Table 8.3.: Characteristics of the bouncing ball example [167, Table 4].

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geometrical characteristics</td>
<td></td>
</tr>
<tr>
<td>(a)</td>
<td>4.0 m</td>
</tr>
<tr>
<td>(b)</td>
<td>4.0 m</td>
</tr>
<tr>
<td>(R)</td>
<td>1.0 m</td>
</tr>
<tr>
<td>Inertia properties</td>
<td></td>
</tr>
<tr>
<td>(m)</td>
<td>1.0 kg</td>
</tr>
<tr>
<td>Force elements</td>
<td>(\Gamma = 9.81 \text{ m/s}^2)</td>
</tr>
<tr>
<td>Contact parameters</td>
<td>(\epsilon_{N_1} = \epsilon_{N_2} = \epsilon_{N_3} = \epsilon_{N_4} = 0.3)</td>
</tr>
<tr>
<td>Initial conditions</td>
<td></td>
</tr>
<tr>
<td>(x_0)</td>
<td>2.0 m</td>
</tr>
<tr>
<td>(y_0)</td>
<td>2.0 m</td>
</tr>
<tr>
<td>(\dot{x}_0)</td>
<td>0.0 m/s</td>
</tr>
<tr>
<td>(\dot{y}_0)</td>
<td>0.0 m/s</td>
</tr>
</tbody>
</table>

and the linear generalized force can be obtained by

\[
h: \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}^2, \quad (q,v) \mapsto h(q,v) = -\Gamma \begin{pmatrix} \cos(\pi/6) \\ \sin(\pi/6) \end{pmatrix}.
\] (8.2)

The normal gap functions are

\[
g_{N_1}(q) = y - R,
g_{N_2}(q) = a - x - R,
g_{N_3}(q) = b - y - R,
g_{N_4}(q) = x - R,
\]

resulting in a Boolean matrix for the generalized force directions.

Figure 8.5.: Trajectory of the bouncing ball in a box for half-explicit timestepping on velocity level [166, Figure 15].

Fig. 8.5 shows the trajectory of the ball starting from its initial position and ending in the lower left corner. The position and velocity curves are depicted in Fig. 8.6.
and clearly show the nonsmooth, that is, impulsive behavior.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figures/multi_contact_examples.png}
\caption{Position and velocity of the bouncing ball in a box for half-explicit timestepping on velocity level [166, Figure 16].}
\end{figure}

For the application of the half-explicit timestepping methods based on the trapezoidal rule to the slider-crank mechanism, we have mentioned a not always decreasing energy trend (Fig. 5.7). As we see in Fig. 8.7, the energy behavior for the linear bouncing ball in a box shows the expected curve, without any increasing periods. The previous non-decreasing trend is traced back to the nonlinear nature of the example. In Fig. 8.8, we see the normal gap functions. They coincide for both discretization schemes, because of the linearity of the example. All calculations have been performed with $\Delta t = 5 \cdot 10^{-3}$ s.
8.2. Bouncing Ball in a Box

Figure 8.7.: Total energy of the bouncing ball in a box for half-explicit timestepping on velocity level [167, Figure 10].

Figure 8.8.: Normal gap functions of the bouncing ball in a box for half-explicit timestepping on acceleration (dashed) and velocity level (solid) [167, Figure 11].
9. Flexible Examples

In this chapter, we discuss the behavior of waves and damping for half-explicit
timestepping schemes on velocity level. Examples from practice are the elastic bar
and the rotor, both linear and flexible. The text is based on [167].

9.1. Elastic Bar

\[ v_0 \]

\[ L \]

\[ x \]

Figure 9.1.: Linear elastic impacting bar on a rigid obstacle [167, Figure 12].

In this section, we consider the classical example of a linear elastic bar that impacts
a rigid obstacle at constant velocity \( v_0 \). The example is depicted in Fig. 9.1. The bar
has a length \( L \), a section area \( S \), a density \( \rho \) and a Young modulus \( E \). If we denote
by \( c_0 = \sqrt{E/\rho} \) the longitudinal wave speed, the time of contact is twice the time of
the traveling of elastic waves in the bar, that is, \( T = 2L/c_0 \). For the parameters in
Table 9.1, the contact time is small and the contact force is approximately constant
equaling \( \tau = ESv_0/c_0 \).

Table 9.1.: Characteristics of the elastic bar example [167, Table 5].

| Geometrical characteristics | \( L = 1 \text{ m} \)  
|                           | \( S = \pi \cdot 10^{-4} \text{ m}^2 \) |
| Material properties       | \( \rho = 7800 \text{ kg/m}^3 \)  
|                           | \( E = 2.1 \cdot 10^{11} \text{ N/m}^2 \) |
| Contact parameters        | \( \epsilon_N = 0 \) |
| Initial conditions        | \( v_0 = 0.1 \text{ m/s} \) |
| Solution characteristics  | \( c_0 = 5.188 \cdot 10^4 \text{ m/s} \)  
|                           | \( T = 3.854 \cdot 10^{-4} \text{ s} \)  
|                           | \( \tau = 1.271 \cdot 10^3 \text{ N} \) |
The bar is discretized by $N$ linear rod finite elements. The elementary mass and stiffness matrices are

\[ M_e = \frac{1}{6} \rho S l_e \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}, \quad K_e = \frac{1}{l_e} E S \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, \]

where $l_e = L/N$ is the length of an element. The first simulation results with the half-explicit timestepping scheme on velocity level for a time-step size $\Delta t = 10^{-7}$ s and 50 finite elements are reported in Fig. 9.2. This simulation has been carried out with SICONOS (Chapter 3). The expected solution for the elastic bar is determined exactly with the help of LEMKE’s algorithm (Chapter 2). The time of contact of the bar is well-approximated and the velocity within the contact interval is maintained at zero. The violation of the constraint is equal to $5 \cdot 10^{-9}$ m. The presence of the impulse is a consequence of the space discretization by finite elements. The finite mass of the node in contact is instantaneously stopped at the impact. This generates an impulse. This impulse, associated with a plastic impact, induces a loss of the total energy at impact, which can be observed in the energy plot. Finally, we remark that the reaction force is well approximated in the first half of the contact period before the development of an instability, which is inherent to standard finite element approximations of a traveling velocity jump through the mesh. Nevertheless, it is noteworthy that the contact is never lost due to artificial numerical chattering.

**Figure 9.2.:** Elastic bar simulation with the half-explicit timestepping scheme on velocity level $\Delta t = 1 \cdot 10^{-7}$ s, $N = 50$ [167, Figure 13].
In Fig. 9.3, a similar simulation is reported, though the number of elements is increased to 1000. We observe that the reaction impulse is reduced due to the decreased mass of the node in contact. The energy loss is also reduced. The instability is still visible, associated with the discretization. One of the usual ways to circumvent this problem is to introduce an artificial damping in the simulation. Since the discretization family presented in this paper does not include numerical damping, we have added a small amount of Rayleigh damping (proportional to the stiffness $C_e = \delta K_e$) to damp out this numerical artifact. Numerical damping is discussed in Chapter 11. In Fig. 9.4, the results with the viscous damping are plotted. The stiffness proportional damping coefficient is equal to $\delta = 10^{-3}/E s = 4.761 \cdot 10^{-6} s$.

We observe that the instability is nearly damped out and the post velocity is smooth. Only a remaining impulse travels through the bar. The decay of the total energy can be seen in the energy plot.

Comparing our results with those in [2] for the standard Moreau-Jean timestepping scheme, we observe the same type of instability of the reaction forces when there is no numerical damping in the Moreau-Jean algorithm. Nevertheless, the proposed scheme is mainly interesting for two reasons. First, it enables a splitting between the reaction force and the artificial reaction impulse, due to the space discretization. The value of the reaction force and the stress in the bar can be used as a good approximation of the forces in the space continuous bar. The magnitude of the
reaction impulse gives a measure of the quality of the mesh with respect to the contact representation. Second, the scheme is half-explicit and thus requires less computational effort for the evaluation of the constitutive behavior in the nonlinear setting. However, a CFL (Courant-Friedrich-Levy) type condition has to be satisfied, since the internal forces are explicitly evaluated.

(a) position, velocity and reaction impulse at the contact point
(b) position, velocity at the tip and velocity in the middle
(c) reaction forces $\lambda$ [N]
(d) kinetic, potential and total energy [Nm]

Figure 9.4.: Damped elastic bar simulation with the half-explicit timestepping scheme on velocity level $\Delta t = 1 \cdot 10^{-7}$ s, $N = 1000$ [167, Figure 15].

9.2. Rotor

A rotor test rig at AM-TUM (Fig. 9.5) is modeled in the multibody simulation framework MBSim [129, 162]. The idea of this example is the implementation of a variant of the half-explicit timestepping scheme on velocity level in MBSim and the application to an industrial example. Thereby, we test a simple and heuristic adaptation of the time-step size.

The rotor in Fig. 9.5 consists of a drive system with a driving torque about the rotor’s longitudinal axis $T_D$ and a failure given by a torque about the vertical axis $T_F$. The drive system is mounted by a spring-damper element with constants $c_D, d_D$. The rotor’s axle is modeled with $N_A$ beam elements in a floating frame of reference
framework. Each finite element node has five degrees of freedom, two small out-of-plane translational deflections, two small out-of-plane rotational deflections and the rotation about the rotor’s longitudinal axis. The length of the axle is $l_A$, its radius is $r_A$, the density is $\rho_A$, the Young’s modulus is $E_A$, and the shear modulus is $G_A$. A mass proportional damping $d_{1A}$ and a torsional damping $d_{2A}$ are used. The fly-wheel is a rigid body with originally six degrees of freedom, which is mounted bilaterally at position $x_F$ from the axle’s left end. The fly-wheel has the mass $m_F$ and the rotational inertia $J_F$ about the rotor’s longitudinal axis. The journal bearing consists of a rigid frustum with six degrees of freedom mounted bilaterally at its center at the left end of the rotor’s axle but with negligible inertia values. This frustum of radius $r_{JF}$ and half width $b_{JF}$ moves freely in a circular bearing with radius $r_{JB}$, that is, backlash and dry friction $\mu$. The bearing is a rigid body with six degrees of freedom, mass $m_{JB}$ and negligible rotational inertia values. The bearing is mounted at its center by a spring-damper element with constants $c_{JB}$, $d_{JB}$. The actual contact may only occur at position $x_{JB}$ from the axle’s left end. We consider the bearing, e.g., as an auxiliary bearing occurring in many rotor systems with active magnetic bearings [81] or as an example for a safety bearing. The whole system is subject to gravitation $\Gamma$ in negative vertical direction. The characteristics are summarized in Table 9.2. Damping is added because of the experience reported in Section 9.1.

### Table 9.2.: Characteristics of the rotor example [167, Table 6].

<table>
<thead>
<tr>
<th>rotor axle</th>
<th>journal bearing</th>
<th>drive system</th>
</tr>
</thead>
<tbody>
<tr>
<td>$l_A = 0.59$ m</td>
<td>$r_{JB} = 1.01 \cdot 10^{-2}$ m</td>
<td>$T_D = 100$ Nm for 0.05 s</td>
</tr>
<tr>
<td>$r_A = 12.5 \cdot 10^{-3}$ m</td>
<td>$m_{JB} = 2.3$ kg</td>
<td>$T_F = 10$ Nm</td>
</tr>
<tr>
<td>$\rho_A = 7.85 \cdot 10^4$ kg/m$^3$</td>
<td>$x_{JB} = b_{JF}/2$</td>
<td>$c_D = 6 \cdot 10^6$ N/m</td>
</tr>
<tr>
<td>$E_A = 2.1 \cdot 10^{11}$ N/m$^2$</td>
<td>$\mu = 0.01$</td>
<td>$d_D = 10$ Ns/m</td>
</tr>
<tr>
<td>$G_A = 0.81 \cdot 10^{11}$ N/m$^2$</td>
<td>$c_{JB} = 10^6$ N/m</td>
<td></td>
</tr>
<tr>
<td>$d_{1A} = 601/s$</td>
<td>$d_{JB} = 10$ Ns/m</td>
<td></td>
</tr>
<tr>
<td>$d_{2A} = 0.001$ kgm$^2$/s</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>fly wheel</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$m_F = 4.98$ kg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$J_F = 0.01$ kgm$^2$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x_F = 1.8 \cdot 10^{-1}$ m</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>journal frustum</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_{JF} = 10^{-2}$ m</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$b_{JF} = 19 \cdot 10^{-3}$ m</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>gravitation</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Gamma = 9.81$ m/s$^2$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
We adapt the interpretation of the contact activity for the first stage unknowns. Instead of (7.5)-(7.6), we solve

\[ 0 = \lambda^+_{N_{i-1}x_i^{-1},1} - \text{proj}_C \left( \lambda^+_{N_{i-1}x_i^{-1},1} - rW_T^{N_{i-1},1}(q_{i-1,1})v_{i-1,1} \right), \quad (9.2) \]

\[ 0 = \lambda^+_{T_{i-1}x_i^{-1},1} - \text{proj}_C \left( \lambda^+_{T_{i-1}x_i^{-1},1} - rW_T^{T_{i-1},1}(q_{i-1,1})v_{i-1,1} \right). \quad (9.3) \]

An overview of the method is depicted in Fig. 9.7. The time-step size for impulsive periods \( \Delta t \) is heuristically adapted concerning the experiences with the slider-crank example (Table 8.2): \( \Delta t = \frac{\Delta t}{10} \). As we do not know the error constant, we prefer this heuristic in comparison to \( \Delta t = \Delta t^2 \), which uses the non-impulsive order of the schemes.

\[ x \times 10^{-4} \]

\[ z \times 10^{-4} \]

\[ y \times 10^{-4} \]

\[ x \text{-axis} \]

\[ y \text{-axis} \]

\[ z \text{-axis} \]

\[ \text{Figure 9.6.: Curve of the journal frustum in the journal bearing [166, Figure 25].} \]

The typical forward whirl phenomenon, due to the external loadings and the dry friction contact, can be seen in Fig. 9.6. We are interested in the general behavior of the variant of the half-explicit timestepping scheme on velocity level. In MBSim, the tolerances define tubes around the corner laws [72]. Starting from the impulsive tolerances \( tol_A = 10^{-10} \text{ Ns} \), \( tol_g = 10^{-10} \text{ m/s} \), we use \( tol_{\lambda} = \frac{tol_A}{\Delta t} \), \( tol_{\dot{g}} = \frac{tol_g}{\Delta t} \) for the calculation of the constraint equations with a single-step fixed-point method [72]. The index sets are detected with a tolerance of \( 10^{-8} \text{ m} \). The variant of the half-explicit timestepping scheme on velocity level uses a fraction of impulsive integration steps of 1.3\%. The overall computing time of the variant of the half-explicit timestepping scheme on velocity level in comparison with the half-explicit variant of the Moreau-Jean timestepping scheme in [129] is 0.32. Hence, the new proposed scheme offers huge computing time saving potentials. In the calculations, we have used \( \Delta t = 5 \times 10^{-6} \text{ s} \).
Specify characteristics, start time $t = 0$, end time $T$, time step size $\Delta t_i$

Initialize $i = 1$

For $t < T$

Evaluate left-hand side

Evaluate $q_{i-1,0} = q_{i-1}$

Evaluate $v_{i-1,0} = v_{i-1}^+$

Predict right-hand side

Compute $q_{i-1,1} = q_{i-1,0} + \Delta t_i Y_{i-1,0} v_{i-1,0}$

Decide strategy

if $\exists k^*: g_{N_k^+} \left( q_{i-1,0} \right) > 0 \wedge g_{N_k^+} \left( q_{i-1,1} \right) \leq 0$

Set $A_{N_i} = 0$, $A_{T_i} = 0$

Compute Lagrange multipliers on velocity level

$$\lambda_{N_i,T_i}^{+} - \text{proj}_{R_{N_i}^+} \left( \lambda_{N_i,T_i}^{+} - r W^T_{N_i,T_i} v_{i-1,1} \right) = 0$$

$$\lambda_{T_i,T_i}^{+} - \text{proj}_{C_T} (\lambda_{T_i,T_i}^{+} - r W^T_{T_i,T_i} v_{i-1,1}) = 0$$

with $v_{i-1,1} = v_{i-1,0} + \Delta t_i M_{i-1,0}^{-1} \left[ h_{i-1,0} + W_{N_i} \lambda_{N_i}^{+} + W_{T_i} \lambda_{T_i}^{+} \right]$.

Compute $q_i = q_{i-1,0} + \frac{\Delta t_i}{2} \left[ Y_{i-1,0} v_{i-1,0} + Y_{i-1,1} v_{i-1,1} \right]$.

Compute Lagrange multipliers on velocity level

$$\lambda_{N_i,T_i}^{-} - \text{proj}_{R_{N_i}^+} \left( \lambda_{N_i,T_i}^{-} - r W^T_{N_i,T_i} v_{i}^{-} \right) = 0$$

$$\lambda_{T_i,T_i}^{-} - \text{proj}_{C_T} (\lambda_{T_i,T_i}^{-} - r W^T_{T_i,T_i} v_{i}^{-}) = 0$$

with $v_i^{-} = v_i^{-} = v_{i-1,0} + \frac{\Delta t_i}{2} M_{i-1,0}^{-1} \left[ h_{i-1,0} + W_{N_i} \lambda_{N_i}^{-} + W_{T_i} \lambda_{T_i}^{-} \right]$.

if $\exists k^*: g_{N_k^+} \left( q_{i-1,0} \right) > 0 \wedge g_{N_k^+} \left( q_{i-1,1} \right) \leq 0$

Set $\lambda_{N_i}^{-} = 0$, $\lambda_{T_i}^{-} = 0$ and $\lambda_{N_i}^{+} = 0$, $\lambda_{T_i}^{+} = 0$.

Compute $q_{i-1,1} = q_{i-1,0} + \Delta t_i Y_{i-1,0} v_{i-1,0}$

Compute $v_{i-1,1} = v_{i-1,0} + \Delta t_i M_{i-1,0}^{-1} h_{i-1,0}$.

Compute $q_i = q_{i-1,0} + \Delta t_i \left[ Y_{i-1,0} v_{i-1,0} + Y_{i-1,1} v_{i-1,1} \right]$.

Compute $v_i^{-} = v_i^{-} = v_{i-1,0} + \Delta t_i \left[ M_{i-1,0}^{-1} h_{i-1,0} + M_i^{-1} h(q_i, v_i^{-}) \right]$.

Compute Lagrange multipliers on velocity level

$$A_{N_i,T_i}^{-} - \text{proj}_{R_{N_i}^+} \left( A_{N_i} - r \tilde{g}_{N_i,T_i}^+ \right) = 0$$

$$A_{T_i,T_i}^{-} - \text{proj}_{C_T(A_{N_i})} \left( A_{T_i} - r \tilde{g}_{T_i,T_i}^+ \right) = 0$$

with $v_i^{-} = v_i^{-} + M_i^{-1} (W_{N_i} A_{N_i} + W_{T_i} A_{T_i})$.

Write result of time step, e.g., $q_i$ and $v_i^+$.

Update $t$ by $\Delta t_i$ respectively $\Delta t_i$ and increment $i$ by 1.

Postprocessing

Figure 9.7.: Half-explicit timestepping scheme on velocity level, variant [166, Figure 24].
Part III.

Frequency-Damped Mixed Timestepping Schemes
10. Mixed Timestepping Schemes

The proposed timestepping schemes based on time-discontinuous Galerkin methods according to Figs. 5.5, 7.2 and 9.7 are very efficient and consistent concerning the representation of nonsmoothness. However, flexible examples (Section 9.1) show that the base integration scheme may get unstable without additional damping because of its half-explicit nature. For this reason, we derive different timestepping methods based on the general framework of impulsive corrections and non-impulsive base integration schemes dropping the time-discontinuous Galerkin setting. We focus on implicit base integration schemes to damp high artificial frequencies due to finite element discretizations. Often even for rubber material, implicit base integration schemes are superior to explicit base integration schemes concerning results and computing time. We use base integration schemes like the generalized-α method [41], the Bathe-method [16] and the ED-α method [25] within mixed timestepping schemes [58, 1]. We evaluate both the non-impulsive forces and the impulsive reactions on the advanced velocity level using the projection formulation for convex sets together with semi-smooth Newton methods [7, 40, 152, 165]. The text is based on [154].

10.1. Flexible Slider-Crank Mechanism

\[ l_1, l_2 \]

\[ m_1, m_2, m_3, J_1, J_2, J_3 \]

\[ \Gamma \]

Figure 10.1.: Flexible slider-crank mechanism [154, Figure 19].

Similar to Chapter 4, we consider the flexible slider-crank mechanism shown in Fig. 10.1 for the comparison of the proposed timestepping schemes. Thereby, \( l_1 \) is the length of the crank and \( l_2 \) the initial length of the connecting rod for the undeformed state. The inertia values of rigid crank, flexible connecting rod and rigid slider consist of translational masses \( m_1, m_2, m_3, \) as well as of rotational inertia values \( J_1, J_2, J_3 \) for the undeformed state. For the flexible connecting rod, we
consider $\rho$ and $E$ as density and Young’s modulus, respectively. The cross-sectional area is given by $A = HD$, that is, the product of height $H = m_2/(\rho l_2 D)$ and thickness $D = \sqrt{12J_2/m_2 - l_2^2}$ of the rod, and the second moment of area is given by $I = \frac{1}{12}HD^3$. The system is subject to gravitation $\Gamma$ [69].

In order to describe the flexible system, the local coordinate system of the connecting rod is located tangentially in the joint between the crank and the connecting rod. Thus, the basis for a floating frame of reference formulation is accomplished. Such a description is characterized by a separation of the coordinates of an elastic body into reference and elastic coordinates. The reference coordinates delineate the rigid body movement and consist of the translational coordinates describing the absolute position of the local coordinate system and the rotational coordinates describing the orientation by angles. The elastic coordinates capture the flexible movement. Crank and slider are described by minimal coordinates. For the evaluation of the equations of motion, we consider the inertia coupling between the different sets of coordinates. The derivation of the system matrices for the given slider-crank mechanism is studied in Chapter F.

### 10.2. Summary of Computational Algorithm

The idea is to combine both the non-impulsive motion and the impulsive motion within one consistent integration scheme. We want that, whenever necessary, this integration can automatically model impacts and velocity jumps. Moreover, it should be able to switch to effective higher order integration with all kinds of nice benefits of sophisticated integration schemes for differential algebraic equations. We introduce a framework, derived from a time-discontinuous Galerkin setting as proposed in Figs. 5.5, 7.2 and 9.7. It includes integration schemes for differential algebraic equations as base integration schemes on velocity level for one time-step from $t_i$ to $t_{i+1}$. With this propagation at the end of the time-step, we can check by the same activity rules, that is, index set calculations on velocity level with $\mathcal{I}_i$, if new contacts have been closed:

$$\exists k^* : g_{N_k^*}(q_i) > 0 \land g_{N_k^*}(q_{i+1}) \leq 0 \, .$$

In this case, we correct the solution for the velocity variables and calculate $v_{i+1}^+$ with the impulsive forces $A_{N_{i+1}}$ and $A_{T_{i+1}}$. If

$$\exists k^* : g_{N_k^*}(q_i) > 0 \land g_{N_k^*}(q_{i+1}) \leq 0 \, ,$$

we set $v_{i+1}^+ = v_{i+1}^-$. The overall algorithm can be summarized as shown in Fig. 10.2.

As an example, Fig. 10.3 shows the slider of the slider-crank mechanism in Chapter 4 at different time-steps before and after an impact. The dotted line represents the normal gap velocity before an impact. Going from $t_{i+5}$ to $t_{i+6}$, there is no new active contact point. Hence, we just have to solve for non-impulsive forces on velocity level
10.3. Assignment of Impulses

The assignment of impulses is explained in detail in Sections 5.4 and 6.3. We repeat the basics. If there is at least one inactive gap function \( (g_{N_k}(q_i) > 0) \) at \( t_i \) that becomes active \( (g_{N_k}(q_i+1) \leq 0) \), an impact occurs in a rigidly connected component of its source. Hence, we update the velocity at \( t_{i+1} \) using (1.5):

\[
v_{i+1}^+ = v_{i+1}^- + M_{i+1}^{-1}W_{N_{i+1}}A_{N_{i+1}} + M_{i+1}^{-1}W_{T_{i+1}}A_{T_{i+1}}.
\] (10.1)

Figure 10.2.: Flowchart of computational algorithm [154, Figure 4].

To avoid further penetration. We notice the usual drift-off effect \( g_{N_k}(q_{i+6}) \leq 0 \) after imposing the contact forces \( \lambda_N \) on velocity level which only constrains \( g_{N_{i+6}} \geq 0 \).
For the computation of $u_{i+1}^+$, $A_N$, and $A_T$, we write (1.14) and (1.15) on velocity level, and eliminate $\dot{g}_{N,i+1}^+$ and $\dot{g}_{T,i+1}^+$ with (1.5):

$$A_{N,i+1,z_i^{i+1}} = \text{proj}_{0}^+ \left[ A_{N,i+1,z_i^{i+1}} - r \dot{g}_{N,i+1,z_i^{i+1}} \right] ;$$

$$A_{T,i+1,z_i^{i+1}} = \text{proj}_{C_T}(A_{N,i+1,z_i^{i+1}}) \left[ A_{T,i+1,z_i^{i+1}} - r \dot{g}_{T,i+1,z_i^{i+1}} \right] ;$$

cf. definitions (1.16) and (1.17). These equations can be solved with a nonsmooth GAUSS-NEWTON method. We use $r = 0.1 \Delta t$ without adaptations.

In the next chapter, we explain the generalized-α method as a base integration scheme. The BATHE-method, as well as the ED-α method are further base integration schemes, which can be used instead of the generalized-α method. We explain them and compare the properties of all three base integration schemes.
11. Base Integration Schemes

The following text is from [154]. Timestepping methods for structural dynamics are evaluated among other criteria based on two well understood and appreciated characteristics: unconditional stability and high frequency dissipation [136, 94, 41, 176, 19]. The first criterion ensures that the stability of the method does not depend on the time-step size. The second requirement regards the algorithmic dissipation and damping properties for non-physical high-frequency modes. In structural dynamics, large scale movements and lower modes are of interest, whereas high frequencies are usually not considered, or only appear as a consequence of discretization. However, high-frequency modes affect numerical and convergence properties of the system especially in the nonlinear regime. Hence, it is desirable to reduce these frequencies by numerical damping. The intensity of the damping should be controllable and should affect the lower modes as less as possible. In order to meet these criteria, a variety of integration methods has been developed and optimized (with regard to the desired properties).

The HHT-formalism [94] and the generalized-α-method [41] are two well known methods based on the Newmark-scheme [136]. Having a control parameter for high frequency damping, as well as second-order accuracy for the generalized coordinates and the Lagrange multipliers, the generalized-α method is a good choice for numerical integration [11]. An extended version of the generalized-α method for nonsmooth problems is presented in [38, 32]. An undesirable feature is that the damping parameters have to be selected and acceptable values depend on the characteristics of the problem being solved. In Newmark-based integrators, the damping parameter plays a very important role, as high frequency oscillations may effect results and even convergence of the finite element solver [55, 26, 166, 167]. Recently, new attempts have been made to overcome such problems. Among those, we focus on two schemes, which use extra information in each time interval, the BATHE-method and the ED-α method.

The BATHE-method combines the use of the trapezoidal rule and the Euler backward method [16]. It has been shown that the additional midpoint information can result in a more robust simulation in the nonlinear analysis and can avoid nonphysical large contact forces. The BATHE-method is also effective in the linear analysis [17]. The ED-α (Energy Decaying) scheme is proved to be one of the most effective methods in solving stiff nonlinear finite element problems. The ED-α method is based on a variational interpretation of Runge-Kutta methods and time-discontinuous Galerkin (TDG) methods [25]. It has been shown that the ED-α scheme benefits from unconditional stability in the nonlinear regime, and also damps out the unresolved and spurious high frequencies using tunable parameters [115, 26].
11.1. Generalized-α Method

According to [11], the generalized-α method for flexible multibody systems can be summarized as follows:

\[ M_{i+1}a_{i+1} + C_{i+1}v_{i+1}^{-} + K_{i+1}q_{i+1} = h_{i+1} + W_{N_{i+1}}\lambda_{N_{i+1}} + W_{T_{i+1}}\lambda_{T_{i+1}}, \]
(11.1)

\[(1 - \alpha_m) A_{i+1} + \alpha_m A_i = (1 - \alpha_f) a_{i+1} + \alpha_f a_i, \]
(11.2)

\[ q_{i+1} = q_i + \Delta t Y_i v_i^+ + \Delta t^2 [(0.5 - \beta) Y_i A_i + \beta Y_{i+1} A_{i+1}] , \]
(11.3)

\[ v_{i+1}^{-} = v_i^+ + \Delta t [(1 - \gamma) A_i + \gamma A_{i+1}] , \]
(11.4)

\[ q_0 = q(0), \quad v_0^+ = v(0), \]
(11.5)

\[ a_0 = M_0^{-1} \left( h_0 - K_0 q_0 - C_0 v_0^+ \right), \quad A_0 = a_0 , \]
(11.6)

where \( q_i \) is the vector of generalized coordinates, \( v_i \) is the vector of generalized velocities, \( a_i \) is the vector of generalized accelerations, \( A_i \) is the vector of acceleration-like auxiliary variables, \( C_i \) is the damping matrix and \( K_i \) is the stiffness matrix.

11.1.1. Characteristics

\[ \lambda_3^1 = -1 \]
\[ ||\lambda_3^1|| = ||\lambda_{1,2}^1|| \]
\[ \lambda_{1,2}^1 = -1 \]

\[ \rho_{\infty} = [0, 1] \]

**Figure 11.1.** Generalized-α stability region in \( \alpha_m - \alpha_f \) space, different test case \( A(-1,1/2), B(-1,0), C(-1, -1/2), D(1/2,1/2), E(0,1/3) \) [154, Figure 6].

The displacement and velocity updates (11.3) and (11.4) are identical to those of the Newmark algorithm. The structure of these update equations is obtained using Taylor series expansion about \( t_i \). The crucial task is to determine the relationship between the algorithmic parameters, \( \alpha_m, \alpha_f, \gamma, \) and \( \beta \). With appropriate expressions
for \( \gamma \) and \( \beta \), and if \( \alpha_m = 0 \), the algorithm reduces to the HHT method. With parametric values

\[
\begin{align*}
\alpha_m & = \frac{2\rho_\infty - 1}{\rho_\infty + 1}, \\
\alpha_f & = \frac{\rho_\infty}{\rho_\infty + 1}, \\
\gamma & = \frac{1}{2} - \alpha_m + \alpha_f, \\
\beta & = \frac{1}{4} \left(1 - \alpha_m + \alpha_f\right)^2,
\end{align*}
\]

(11.7)

where \( \rho_\infty \in [0,1] \) is the spectral radius of the amplification matrix at the high frequency limit, the generalized-\( \alpha \) method is unconditionally stable for linear problems, second order accurate possessing an optimal combination of high-frequency and low-frequency dissipation. The stability region is indicated by the shaded area in Fig. 11.1 (Chapter D).

![Spectral radii and Relative period errors](image)

**Figure 11.2:** Properties of the HHT-method for different \( \alpha = \alpha_f, \alpha_m = 0 \) [154, Figure 7].

In order to have a better insight on the effect that different parameters have on the eigenvalues of the amplification matrix, we plot the spectral radius and the relative period error with respect to the frequency \( \Delta t/T \), where the period is \( T = 2\pi/\omega \).

First, we consider a special case of the generalized-\( \alpha \) method by setting \( \alpha_m = 0 \) (HHT method). From Fig. 11.1, we are allowed to choose \( \alpha_f \) between 0 and 0.5. We see a cusp after we pass the point \( E \) in Fig. 11.2. Using optimum parameter values, we observe the behavior of the method for different \( \rho_\infty \) in Fig. 11.3. These vary from the no-dissipation case (\( \rho_\infty = 1 \)) to the so-called asymptotic annihilation case (\( \rho_\infty = 0 \)), or moving along the red dotted line from point \( D \) to point \( B \). In order to observe the properties of different regions in Fig. 11.1, we plot the spectral radius for point \( A \) and point \( C \) (Fig. 11.4). If we select \( \alpha_m \) and \( \alpha_f \) values away from the dotted line (\( \|\lambda^\infty_3\| = \|\lambda^\infty_1\| \)), we may expect a cusp. Figure 11.4 shows how we can modify \( \alpha_m \) and \( \alpha_f \) for point \( A \) and point \( C \) in order to have the same \( \rho_\infty \) but with smooth transition instead. Further properties can be found in Chapter D.
11.1.2. Contact Forces on Velocity Level

Using (1.9) and (1.10) together with (1.2), we calculate contact forces $\lambda_{N_{i+1}}$ and $\lambda_{T_{i+1}}$ on velocity level (Sections 5.2 and 6.1):

$$
\lambda_{N_{i+1},x_1}^i = \text{proj}_{R^+_N} \left[ \lambda_{N_{i+1},x_1}^i - r \dot{g}_{N_{i+1},x_1}^i \right],
$$

$$
\lambda_{T_{i+1},x_1}^i = \text{proj}_{C_T(\lambda_{N_{i+1},x_1}^i)} \left[ \lambda_{T_{i+1},x_1}^i - r \dot{g}_{T_{i+1},x_1}^i \right].
$$

In the multi-contact case, active contacts might be depending. Hence, we cannot use a nonsmooth NEWTON method. Instead, we solve (11.8), (11.9) by a nonsmooth variant of the GAUSS-NEWTON method with $r = 0.1$ without adaptations.

In case of no impact, applying the fixed point iteration method, we obtain unknowns at $t_{i+1}$ using the generalized-$\alpha$ algorithm for each time step like in Fig. 11.5.
11.2. Bathe-Method

The Bathe-method [17] is an effective implicit time integration scheme, proposed for the finite element solution of nonlinear problems in structural dynamics. Various important attributes have been demonstrated. In particular, it has been shown that the scheme remains stable without the use of adjustable parameters. For this method, the complete time step Δt is subdivided into two equal sub-steps. In the first sub-step, the trapezoidal rule is used, whereas in the second sub-step, the 3-point Euler backward method is applied.

As suggested in [17], we consider the dynamic equilibrium for time $t + Δt$ and $t + Δt/2$ which is indicated using index $i + 1$ and index $i + 1/2$, respectively,

\[
M_{i+1/2} a_{i+1/2} + C_{i+1/2} v_{i+1/2} + K_{i+1/2} q_{i+1/2} = h_{i+1/2} + W_{N_{i+1/2}} \lambda_{N_{i+1/2}} + W_{T_{i+1/2}} \lambda_{T_{i+1/2}},
\]

\[
M_{i+1} a_{i+1} + C_{i+1} v_{i+1}^- + K_{i+1} q_{i+1} = h_{i+1}^- + W_{N_{i+1}} \lambda_{N_{i+1}} + W_{T_{i+1}} \lambda_{T_{i+1}},
\]

\[
v_{i+1/2} = v_{i+1}^+ + \frac{Δt_i}{4} (a_i + a_{i+1/2}),
\]

\[
q_{i+1/2} = q_{i} + \frac{Δt_i}{4} (Y_i v_{i+1}^+ + Y_{i+1/2} v_{i+1/2}).
\]
We typically interpret the stages vibrations. Total annihilation is important to avoid high frequency noises generated 11.3. ED-

is that classical algorithms that are unconditionally stable and high frequency dissi-

nonlinear finite element problems in time. The basic motivation behind these schemes

In energy decaying (ED) schemes, we develop robust algorithms for integrating stiff

by the time integration algorithm. The BATHE-method shows a better behavior for the period error in comparison to the generalized-α method and the classic

Further properties can be found in Chapter E. As contact forces are calculated in the same way as for the generalized-α method, we summarize directly Figs. 11.6, 11.7.

Specify characteristics, time step size $\Delta t_i$

Initialize $k = 0$

$q_{i+1/2}^0 = q_{i}, v_{i+1/2}^0 = v_{i}, a_{i+1/2}^0 = a_{i}$

$M_{i+1/2}^0 = M_{i}, h_{i+1/2}^0 = h_{i}, W_{N+1/2}^0 = W_{N_{i}}, W_{T+1/2}^0 = W_{T_{i}}, K_{i+1/2}^0 = K_{i}$

while true

calculate $\lambda_{N_{i+1/2}}^k, \lambda_{T_{i+1/2}}^k$ with GAUSS-NEWTON algorithm

$\hat{R}_1^k = h_{i+1/2}^k + M_{i+1/2}^k \left( \frac{\Delta t}{2} Y_{i+1/2}^{-1} q_{i} + \frac{\Delta t}{2} v_{i} + a_{i} \right) + C_{i+1/2}^k \left( \frac{1}{4} \lambda_{N_{i+1/2}}^k - \lambda_{T_{i+1/2}}^k \right)$

$q_{i+1/2}^{k+1} = Y_{i+1/2}^{-1} \left( \hat{R}_1^k + W_{N_{i+1/2}}^k \lambda_{N_{i+1/2}}^k + W_{T_{i+1/2}}^k \lambda_{T_{i+1/2}}^k \right)$

$v_{i+1/2}^{k+1} = -v_{i} + \frac{4}{\Delta t} \left( Y_{i+1/2}^{-1} q_{i+1} - Y_{i}^{-1} q_{i} \right)$

$a_{i+1/2}^{k+1} = -a_{i} + \frac{4}{\Delta t} \left( v_{i+1/2}^{k+1} - v_{i} \right)$

if $\|v_{i+1/2}^{k+1} - v_{i+1/2}^k\| < \text{tol}$ break

update $M_{i+1/2}^{k+1}, h_{i+1/2}^{k+1}, W_{N_{i+1/2}}^{k+1}, W_{T_{i+1/2}}^{k+1}$

$k = k + 1$

Figure 11.6.: BATHE time integration scheme: first half step [154, Equation (64)].

11.3. ED-α Method

In energy decaying (ED) schemes, we develop robust algorithms for integrating stiff nonlinear finite element problems in time. The basic motivation behind these schemes is that classical algorithms that are unconditionally stable and high frequency dissipating in the linear regime, usually lose their properties in the nonlinear regime [26]. We typically interpret the stages $q_j, v_j, a_j$ as field variables associated with the time
11.3. ED-α Method

Specify characteristics, time step size $\Delta t_i$

Initialize $k = 0$

$q_i^0 = a_i + \Delta t / 2$, $v_i^0 = v_i + \Delta t / 2$, $a_i^0 = a_i + \Delta t / 2$

$M_{i+1} = M_i + \Delta t / 2$, $h_{i+1} = h_i + \Delta t / 2$, $W_{N_i} = W_{N_{i+1}}$, $W_{i+1}^0 = W_{i+1}^0$, $K_{i+1}^0 = K_{i+1}^0$

while true

$\text{calculate } \lambda_{N_{i+1}}^k, \lambda_{i+1}^k$ with GAUSS-NEWTON algorithm

$\tilde{K}_k = \frac{0}{2} M_{i+1} + \frac{9}{3} C_{i+1} + K_{i+1} Y_{i+1}^k$

$\tilde{R}_2 = h_{i+1} + M_{i+1} \left( \frac{12}{3} Y_{i+2}^{-1} q_{i+1} - \frac{3}{3} Y_{i}^{-1} q_i + \frac{4}{3} v_{i+1} - \frac{1}{3} v_i \right)$

$h_{i+1}^k = \left[ \tilde{K}_k + \tilde{R}_2 + W_{N_{i+1}} \lambda_{N_{i+1}}^k + W_{i+1} \lambda_{i+1}^k \lambda_{i+1}^k \right]$

$a_i^{k+1} = \frac{1}{9} Y_{i+1} q_i - \frac{4}{3} Y_{i+2} q_{i+1} + \frac{3}{3} Y_{i+1}^{-1} q_{i+1}$

$a_i^{k+1} = \frac{4}{3} v_i - \frac{4}{3} v_{i+1} + \frac{3}{3} v_{i+1}^k$

if $\|v_{i+1} - v_i^{k+1}\| < \text{tol}$, $v_{i+1} = v_i^{k+1}$ break

update $M_{i+1}^{k+1}$, $h_{i+1}^{k+1}$, $W_{N_{i+1}}^{k+1}$, $W_{i+1}^{k+1}$

$k = k + 1$

end

Figure 11.7.: Bathe time integration scheme: second half step [154, Equation (65)].

$t_i^+$. In this sense, the unknown fields are allowed to create a jump discontinuity at the beginning of the time step that is, responsible for the high frequency damping behavior of the scheme [25]:

$$M_j a_j + C_j v_j + K_j q_j = h_j + W_{N_j} \lambda_{N_j} + W_{T_j} \lambda_{T_j},$$

$$M_{i+1} a_{i+1} + C_{i+1} v_{i+1} + K_{i+1} q_{i+1} = h_{i+1} + W_{N_{i+1}} \lambda_{N_{i+1}} + W_{T_{i+1}} \lambda_{T_{i+1}},$$

$$v_j = v_j^+ + \Delta t_i \alpha_{AR} \left[ \alpha (a_j - a_i) - a_{i+1} + a_i \right],$$

$$v_{i+1} = v_i^+ + \frac{\Delta t_i}{2} (a_j + a_{i+1}),$$

$$q_j = q_i + \Delta t_i \alpha_{AR} \left[ \alpha (Y_j v_j - Y_i v_i^+) - Y_{i+1} v_{i+1} + Y_i v_i^+ \right],$$

$$q_{i+1} = q_i + \frac{\Delta t_i}{2} (Y_j v_j + Y_{i+1} v_{i+1})$$

$$a_0 = M_0^{-1} (h_0 - K_0 q_0 - C).$$

In this work for practical implementation of the scheme, the displacements $q_j$ and $q_{i+1}$ are eliminated, leaving a velocity-based iteration scheme in the $2 \times 4$ dof unknowns $v_j$ and $v_{i+1}$. However, the overall procedure is more expensive than other one-stage schemes like the generalized-α method, the matrices being twice as large.

Note that for $\alpha_{AR} = 0$ or $\alpha = 0$, we recover a conserving scheme. The parameter $\alpha_{AR}$ does not control the asymptotic value of the spectral radius but only controls the cut-off frequency of the scheme and so relative period errors (Fig. 11.9). The
11. Base Integration Schemes

**Figure 11.8.** Comparison: Newmark method, generalized-\(\alpha\) method and Bathe method [154, Figure 10].

**Figure 11.9.** Comparison: ED-\(\alpha\) method with \(\rho_\infty = 0\) and optimal \(\alpha\) (11.24) and Bathe method [154, Figure 11].

Minimum period elongation is obtained for \(\alpha_{AR} = 1/6\). The method is second order accurate for arbitrary \(\alpha_{AR} \geq 0\) and arbitrary ordinary differential equations. Third order accuracy is obtained for the scalar linear model problem in Section 11.4 and the special value \(\alpha_{AR} = 1/6\). The parameter \(\alpha\) is responsible for the asymptotic value of the spectral radius. An optimal choice turns out to be (11.24) [26] (Fig. 11.10):

\[
\alpha = \frac{1 - \rho_\infty}{1 + \rho_\infty}, \quad \alpha_{AR} = \frac{1}{6}. \tag{11.24}
\]

In case of no impact using (11.17) to (11.22), we have the method in Fig. 11.12. It is visualized in Fig. 11.11. The matrices \(\overline{M}_c, \overline{R}_c, \overline{W}_{Nc}, \overline{W}_{Tc}\) and vectors \(\overline{v}_c, \overline{\lambda}_c\) are defined as

\[
\overline{M}_c = \begin{pmatrix}
c_4 M_{i+1} + c_3 K_{i+1} Y_{i+1} & c_6 M_{i+1} + c_2 K_{i+1} Y_{i+1} \\
c_{10} M_j + c_8 K_j Y_j & c_{12} M_j + c_{10} K_j Y_j
\end{pmatrix},
\]
11.4. Comparison of Base Integration Schemes

Our next goal is to present the solution of a simple linear system as a model problem to represent the stiff and flexible parts. We compare the presented base integration

\[
\begin{align*}
\bar{R}_c &= \begin{pmatrix}
    h_{i+1} - c_5 M_{i+1} v_i - c_7 M_{i+1} a_i - c_1 K_{i+1} q_i \\
    h_j - c_{13} M_j v_i - c_{14} M_j a_i - c_9 K_j q_i - c_{11} K_j Y_i v_i
\end{pmatrix}, \\
W_{N_c} &= \begin{pmatrix} W_{N_i+1} & 0 \\ 0 & W_{N_j} \end{pmatrix}, \quad W_{T_c} = \begin{pmatrix} W_{T_i+1} & 0 \\ 0 & W_{T_j} \end{pmatrix}, \\
v_c &= \begin{pmatrix} v_{i+1} \\ v_j \end{pmatrix}, \quad \lambda_c = \begin{pmatrix} \lambda_{i+1} \\ \lambda_j \end{pmatrix}.
\end{align*}
\]

The coefficients \(c_1\) to \(c_{15}\) are

\[
\begin{align*}
c_1 &= 1, \\
c_2 &= \frac{\Delta t}{2}, \\
c_3 &= \frac{\Delta t}{2}, \\
c_4 &= \frac{2 \alpha_{AR} \alpha}{\Delta t \alpha_{AR} (\alpha + 1)}, \\
c_5 &= \frac{2 (0.5 - \alpha_{AR} \alpha)}{\Delta t \alpha_{AR} (\alpha + 1)}, \\
c_6 &= \frac{-1}{\Delta t \alpha_{AR} (\alpha + 1)}, \\
c_7 &= \frac{\alpha_{AR} (1 - \alpha)}{\Delta t \alpha_{AR} (\alpha + 1)}, \\
c_8 &= -\Delta t \alpha_{AR}, \\
c_9 &= 1, \\
c_{10} &= \Delta t \alpha_{AR} \alpha, \\
c_{11} &= \Delta t \alpha_{AR} (1 - \alpha), \\
c_{12} &= \frac{1}{\Delta t \alpha_{AR} (\alpha + 1)}, \\
c_{13} &= \frac{-2 (0.5 + \alpha_{AR})}{\Delta t \alpha_{AR} (\alpha + 1)}, \\
c_{14} &= \frac{-\alpha_{AR} (1 - \alpha)}{\Delta t \alpha_{AR} (\alpha + 1)}, \\
c_{15} &= \frac{2 \alpha_{AR}}{\Delta t \alpha_{AR} (\alpha + 1)}.
\end{align*}
\]

**Figure 11.10:** Comparison: ED-\(\alpha\) method with \(\alpha_{AR} = 1/6\) and optimal \(\alpha\) (11.24) and generalized-\(\alpha\) method [154, Figure 12].
We use the parameters \( k \) with stiffness \( \Gamma \) in case (a) for example, an almost rigid connection [17], while the other springs represent the flexible parts of the structural model.

\[ \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{pmatrix} = \begin{pmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix} + \begin{pmatrix} k_1 + k_2 & -k_2 & 0 \\ -k_2 & k_2 + k_3 & -k_3 \\ 0 & -k_3 & k_3 + k_0 \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix} = \begin{pmatrix} m_1 \Gamma \\ m_2 \Gamma \\ m_3 \Gamma \end{pmatrix} \]

(11.25)

We use the parameters \( k_1 = k_2 = k_3 = 1 \text{ N/m}, k_0 = 10^7 \text{ N/m}, m_1 = m_2 = m_3 = 1 \text{ kg}, \Gamma = 9.81 \text{ m/s}^2 \). In the left part of Fig. 11.13, a stiff spring \( k_0 \) is used to represent, for example, an almost rigid connection [17], while the other springs represent the flexible parts of the structural model.

The highest frequency in case (a) is \( f_{\text{max}} = 503.3 \text{ Hz} \), which is due to the stiff spring with stiffness \( k_0 \). The exact continuous trajectory for the 3 masses is shown in

---

**Figure 11.11.** ED-\( \alpha \) method in time [154, Figure 13].

**Figure 11.12.** ED-\( \alpha \) time integration scheme [154, Equation (74)].
Figs. 11.14, 11.15. The time-step size for comparison of different base integration schemes is chosen as $\Delta t = 10^{-3}$ s. Figure 11.14 shows the response of the first two degrees of freedom, which are connected with the soft springs. As we observe, the response from different base integration schemes follows the reference solution coming from modal analysis. The response of mass number 3, which is connected to the rigid wall through a stiff spring, is shown in Fig. 11.15. The generalized-$\alpha$ method with $\rho_\infty = 0.8$ is able to represent the high frequency vibration with some phase shift, however all other methods damp out this high frequency mode in a few time steps. Using $\Delta t = 10^{-3}$ s, corresponding to a sampling frequency $f_s = 1000$ Hz, we have to make sure that all frequencies higher than $f_s/2 = 500$ Hz are damped out according to the Nyquist-Shannon criterion, in order to avoid chattering in the response.
Figure 11.15.: Displacement of mass 3 for various integration methods (case (a)) [154, Figure 16].

Now, we consider case (b). Figure 11.16 shows the difference between simulation results if we try to satisfy the constraint on acceleration and velocity level. We conclude that using the acceleration level results in a constant residual velocity for mass 3, and of course a linear drift-off effect and violation of the constraint. The calculation of contact forces is straightforward and follows the reference solution (Fig. 11.17). Using the velocity level, the contact force needs to be calculated with an additional integration. This calculation results in a constant position error for this model problem. Using the velocity level results in an oscillatory behavior for the acceleration of mass 3 which means that we expect the same oscillatory behavior for the calculation of contact forces (Fig. 11.17). Inserting additional damping for the numerical integration, we obtain better approximations of contact forces and, at the same time, avoid those artificial high frequencies. The results yielded by the BATHE-method or the ED-α method are similar to the results obtained with the generalized-α method with $\rho_\infty = 0.0$. It is worth to mention that the highest frequency in the system is $f_{\text{max}} = 0.8717 \text{ Hz}$, that is, $\frac{2\pi}{T} = 0.0436$, which means that the oscillatory behavior of the contact force in case of no damping comes from the structure of the generalized-α method and has nothing to do with the poor representation of high frequencies in the system (model (a)). We conclude that the unsymmetrical structure of the BATHE-method or the ED-α method helps to improve the calculation of contact forces in case of model (b). Similar observations were done in Chapter 9 based on the results of the application of half-explicit timestepping schemes on velocity level to an impacting elastic bar and a rubbing rotor.
11.4. Comparison of Base Integration Schemes

Figure 11.16.: Displacement, velocity and acceleration for mass 1, 2 and 3 (case (b)) [154, Figure 17].

Figure 11.17.: Comparison of acceleration and velocity level for the calculation of contact forces (case (b)) [154, Figure 18].
12. Results

The application of the generalized-α method, the Bathe-method and the ED-α method as base integration schemes of the overall framework to nonsmooth problems with unilateral contacts with friction is discussed in this chapter. In this chapter, we show the spatial convergence of the schemes and validate the results comparing to the rigid case. Moreover, we discuss the time integration schemes concerning different aspects.

12.1. Validation for a Rigid Slider-Crank Mechanism

Figure 12.1.: Mass center motion for different numbers of elements [154, Figure 20].

For the validation of the results, we compare an almost rigid system \((E = 10^{15} \text{N/m}^2)\) with the rigid multibody system with 3 degrees of freedom (Chapter 4). Specific characteristics are given in Table 4.1. Figure 12.1 shows the convergence for the position of the slider mass center when we increase the number of elements and run the simulation with \(\Delta t = 10^{-5} \text{s}\), using the Bathe-method. The convergence results are comparable for all base integration schemes. Figure 12.2 shows the results for the position of the slider mass center compared to the same simulation with rigid bodies [69].

The generalized-α method and the Bathe-method are both second order accurate. In order to show this feature in a simulation, we consider the specific case of a bilateral contact by setting the gap \(c = 0 \text{m}\). Figure 12.3 shows the convergence when we decrease \(\Delta t\). The Slope of the line \(m\) shows the accuracy of the calculations in logarithmic scale. The reference solution is calculated with SIMPACK\(^1\).

\(^1\) http://www.simpack.com/
12. Results

Figure 12.2.: Mass center motion - comparison to a rigid body simulation [154, Figure 21].

12.2. Generalized-α and Bathe-Method

Figure 12.3.: Relative errors of position, velocity and contact force [154, Figure 22].

In order to have a better insight on closed gap situations and the calculation of contact forces, we change (Fig. 12.4) some initial properties of the slider-crank mechanism. New values are indicated in Table 12.1, all the other characteristics are set according to Table 4.1. The time-step size is $\Delta t = 10^{-5}$ s.

In Fig. 12.5, we see the comparison of the schemes for the angular displacements $\theta_1$ and $\theta_2$, that is, the inclinations of crank and connecting rod, ($\theta_3 = 0$). Both algorithms with different damping values behave almost the same, before and after the impact.
12.3. **ED-α and Bathe-Method**

Using the data in Table 12.1 and time-step size $\Delta t = 10^{-4}$ s, we run the same simulations up to $t = 0.5$ s to compare the stability and robustness of the ED-α method and the Bathe-method. Figure 12.9 shows the comparison of angular velocities, matching perfectly. The general behavior and amplitude of the contact force is in good agreement for energy decaying methods (Fig. 12.10).

### Table 12.1.

Modified characteristics of the slider-crank mechanism [154, Table 2].

<table>
<thead>
<tr>
<th>Geometrical characteristics</th>
<th>$c = 0.0005$ m</th>
</tr>
</thead>
<tbody>
<tr>
<td>Driven torque for crank</td>
<td>$T = 1$ N/m</td>
</tr>
<tr>
<td>Contact parameters for slider corners</td>
<td>$\epsilon_{N_1} = \epsilon_{N_2} = \epsilon_{N_3} = \epsilon_{N_4} = 0.1$</td>
</tr>
<tr>
<td></td>
<td>$\mu_1 = \mu_2 = \mu_3 = \mu_4 = 0.1$</td>
</tr>
<tr>
<td>Initial conditions</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\omega_{1o} = 0.0$ rad/s</td>
</tr>
<tr>
<td></td>
<td>$\omega_{2o} = 0.0$ rad/s</td>
</tr>
<tr>
<td></td>
<td>$\omega_{3o} = 0.0$ rad/s</td>
</tr>
</tbody>
</table>

**Figure 12.5.** Comparison between generalized-α method and Bathe-method for the angles $\theta_1$ and $\theta_2$ [154, Figure 24].

Figures 12.6 and 12.7 show the comparison for angular velocity and torque at the beam root. If we do not consider enough damping for the time integration schemes, high frequency oscillations corrupt the system response, especially in the velocity and stress fields. In this particular case, the results for the generalized-α method with $\rho_\infty = 0.8, 0.5$ get unstable soon after $t = 0.05$ s. The generalized-α method transfers energy from the higher (artificial) to the lower meaningful modes. Figure 12.8 shows the comparison of the contact force using the velocity level approach. The Bathe-method is able to represent the vibration as the contact is closed, whereas the generalized-α method results in an additional vibration for the velocity which causes the contact condition to go on and off repetitively in time.
12.4. Comparison of Computing Time

Based on the setting in Table 4.1, we analyze the relative central processing unit (CPU) time for the computation of a rigid slider-crank mechanism. Thereby, we compare the generalized-\(\alpha\) method, the BATHE-method, the ED-\(\alpha\) method, Moreau’s midpoint rule (Section 1.4.2) and the half-explicit timestepping scheme "forecasting trapezoidal rule" (HETS, Chapter 7).

Table 12.2.: Relative CPU time for \(\Delta t = 10^{-5}\) s (bilateral) [154, Table 3].

<table>
<thead>
<tr>
<th>Method</th>
<th>Moreau</th>
<th>HETS</th>
<th>generalized-(\alpha)</th>
<th>Bathe</th>
<th>ED-(\alpha)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rel. CPU time</td>
<td>1.0</td>
<td>1.35</td>
<td>4.15</td>
<td>4.35</td>
<td>5.00</td>
</tr>
</tbody>
</table>

We consider the relative CPU time per time-step, exemplary for \(\Delta t = 10^{-5}\) s in Table 12.2 for the bilateral case. We compare the time-step sizes \(\Delta t\) and their corresponding relative error with respect to the reference SIMPACK solution in Table 12.3. To give an explicit idea of the errors, we compute, e.g., the relative error...
for the connecting rod inclination $\theta_2$ for a set of considered time instances $\{t_k\}_{k=1}^M$:

$$\text{err} = \left\| \left( \cdots, \left| \frac{\theta_2(t_k) - \theta_{2\text{ref}}(t_k)}{\theta_{2\text{ref}}(t_k)} \right|, \cdots \right) \right\|_2.$$ 

For a given time-step size, the computational effort is minimized by the generalized-$\alpha$ method as there is no additional midpoint calculation. For the Bathe-method and the ED-$\alpha$ method, we use one additional point in the time integration algorithm which explains the increase in the computing time. In case of the ED-$\alpha$ method, the additional point can be interpreted as a jump at the beginning of the interval. According to Fig. 11.12, we increase the unknowns which have to be solved simultaneously by a factor of 2. The Bathe-method selects the additional point in the middle of the interval. The unknowns are solved independently from the unknowns at $t_{i+1}$. Smaller relative time-steps ($\Delta t$) result in smaller changes of the mass matrix, stiffness matrix and force vector, and thus in less calculation time for the unknowns compared to the ED-$\alpha$ method. Indeed, we need less iterations to find the unknowns at the middle of the interval compared to the end of the interval. Moreau’s midpoint
Figure 12.10.: Comparison between ED-α method with $\rho_\infty = 0$ and generalized-α method for the normal contact force [154, Figure 29].

Table 12.3.: Comparison of the error $\epsilon$ ($T = 0.05$ s, bilateral) [154, Table 3].

<table>
<thead>
<tr>
<th>$\Delta t$</th>
<th>$10^{-4}$ s</th>
<th>$10^{-5}$ s</th>
<th>$10^{-6}$ s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moreau</td>
<td>$1.3 \cdot 10^4$</td>
<td>$1.3 \cdot 10^0$</td>
<td>$1.3 \cdot 10^{-1}$</td>
</tr>
<tr>
<td>HETS</td>
<td>$0.1 \cdot 10^0$</td>
<td>$0.1 \cdot 10^{-2}$</td>
<td>$0.1 \cdot 10^{-4}$</td>
</tr>
<tr>
<td>generalized-α</td>
<td>$0.1 \cdot 10^0$</td>
<td>$0.1 \cdot 10^{-2}$</td>
<td>$0.1 \cdot 10^{-4}$</td>
</tr>
<tr>
<td>Bathe</td>
<td>$0.7 \cdot 10^{-1}$</td>
<td>$0.7 \cdot 10^{-3}$</td>
<td>$1.6 \cdot 10^{-5}$</td>
</tr>
<tr>
<td>ED-α</td>
<td>$1.3 \cdot 10^{-1}$</td>
<td>$1.3 \cdot 10^{-3}$</td>
<td>$1.6 \cdot 10^{-5}$</td>
</tr>
</tbody>
</table>

For the unilateral case, the solution of the ED-α method with $\rho_\infty = 0.0$, $\Delta t = 10^{-7}$ s is chosen as the reference solution. Results are given in Tables 12.4 and 12.5. These confirm the results of the bilateral case. Moreover, we observe an increased stability for the implicit methods.

Table 12.4.: Relative CPU time for $\Delta t = 10^{-5}$ s (unilateral) [154, Table 5].

<table>
<thead>
<tr>
<th></th>
<th>Moreau</th>
<th>HETS</th>
<th>generalized-α</th>
<th>Bathe</th>
<th>ED-α</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rel. CPU time</td>
<td>1.0</td>
<td>1.15</td>
<td>2.72</td>
<td>3.15</td>
<td>3.40</td>
</tr>
</tbody>
</table>

A major advantage of the floating frame of reference formulation is that the finite element nodal coordinates can be easily reduced using modal analysis techniques, based on a reduced set of eigenvectors (Chapter G).
Table 12.5.: Comparison of error $\epsilon$ ($T = 0.05\, s$, unilateral) [154, Table 6].

<table>
<thead>
<tr>
<th>$\Delta t$</th>
<th>$10^{-4}, s$</th>
<th>$10^{-5}, s$</th>
<th>$10^{-6}, s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moreau</td>
<td>$6.7 \cdot 10^{-2}$</td>
<td>$6.4 \cdot 10^{-3}$</td>
<td>$6.4 \cdot 10^{-4}$</td>
</tr>
<tr>
<td>HETS</td>
<td>$1.1 \cdot 10^{-3}$</td>
<td>$2.8 \cdot 10^{-5}$</td>
<td>$2.3 \cdot 10^{-6}$</td>
</tr>
<tr>
<td>generalized-$\alpha$</td>
<td>$2.0 \cdot 10^{-3}$</td>
<td>$1.2 \cdot 10^{-4}$</td>
<td>$1.4 \cdot 10^{-5}$</td>
</tr>
<tr>
<td>Bathe</td>
<td>$1.4 \cdot 10^{-3}$</td>
<td>$7.8 \cdot 10^{-5}$</td>
<td>$8.9 \cdot 10^{-6}$</td>
</tr>
<tr>
<td>ED-$\alpha$</td>
<td>$9.4 \cdot 10^{-4}$</td>
<td>$2.8 \cdot 10^{-5}$</td>
<td>$2.3 \cdot 10^{-6}$</td>
</tr>
</tbody>
</table>

The spatial convergence is plotted in Fig. 12.11 when we increase the numbers of mode shapes. Thereby, the simulation is based on Table 12.1, the time-step size is $\Delta t = 10^{-5}\, s$, the boundary condition is considered as clamped-free (tangential), and we use the BATHE-method for time integration. After an impact occurred, depending on the boundary condition and impact points, we need more number of modes to describe the vibration behavior than in the non-impulsive period (before the impact). As far as we deal with nonlinear problems, slightly different conditions before the impact may result in large changes in time after the impact. This fact explains the differences in the results. Spectral analysis for the BATHE-method (Fig. 11.8) suggests that there will be no effective mode in the response after $f_c = \frac{\Delta t}{T} = 10^2$. We calculate a sufficient number of modes considering the high frequency dissipation of the BATHE-method. All frequencies up to $f_c = \frac{10^2}{10^{-5}\, s} = 10^7\, 1/s$ (which covers the first 52 modes out of total 63 modes) can be effective in the final solution. Results are in good agreement between modal and full FEM simulation for the calculation of normal contact forces.

Figure 12.11.: Normal contact force convergence for modal approach [154, Figure 31].

Figure 12.12 shows the comparison between modal solutions for different boundary conditions. Different boundary conditions result in different mode shapes and frequencies. Assuming the same impact point and condition we get different vibration behavior, but the general pattern looks similar for all cases.
Figure 12.12.: Comparison between modal solutions for different boundary conditions [154, Figure 32].
Part IV.

Projected Timestepping Schemes
13. Motivation of Constraint Projection

Artificial oscillations during the simulation of mechanical systems disturb the interpretation of results. There are several possible reasons for such oscillations. Impulses may excite oscillations. However, also position level discretizations of constraints change the eigenvalues of the discrete dynamical system (numerical model), in comparison to the continuous counterpart (mathematical model) [185, 54]. Model reduction, regularization and stabilization are techniques to improve the behavior in general, in particular for thin structures. To this purpose, one considers, e.g., the energy error, the integration error and the constraint error at the same time. We focus on geometric constraints and on the way we can represent them consistently during the simulation of impacting mechanical systems.

13.1. Impacting Slider-Crank Mechanism

![Graphs showing movement of the center of gravity of the slider for different coefficients of restitution and Moreau’s midpoint rule.](image)

Figure 13.1.: Movement of the center of gravity of the slider for different coefficients of restitution and Moreau’s midpoint rule [171, Figure 3].

The following text is based on [171]. We simulate the slider-crank mechanism from Chapter 4 using Moreau’s midpoint rule. Figure 13.1 shows the results for four
different coefficients of restitution. The curves presented here, as well as the graphs depicted in the original literature [69], show the violation of the non-penetration condition (1.9), especially in the simulations with a low coefficient of restitution. In Fig. 13.1 (a), the center of gravity of the slider exceeds $10^{-3}$ m and in the later course it falls below the value $-10^{-3}$ m. With the clearance $c = 10^{-3}$ m of the notch (Table 4.1), this corresponds to a pervasion of the slider with the bordering wall. Also for higher coefficients of restitution, the gap functions fall below zero, but for shorter time periods.

Figure 13.2: Gap functions and their time derivatives for $\epsilon = 0.1$ and Moreau’s midpoint rule [171, Figure 4].

Figure 13.2 shows the time curve of the gap functions and their time derivatives for a coefficient of restitution $\epsilon = 0.1$. As the initial configuration satisfies $\theta_{30} = 0.0$, $\omega_{30} = 0.0$ 1/s and as the support at the center of gravity prevents the revolution of the slider, pairs of gap functions on opposite sides appear almost symmetric. Again, the non-penetration condition is violated. A detailed view shows the drift-off effect: the gap drifts approximately linearly to negative values while the gap velocity is slightly negative. However, the drift does not have a dominant effect for this configuration because the negative gap functions remain comparatively small in contrast to the geometric dimensions. The drift-off effect will not be negligible anymore, if contacts stay closed for long time periods.

13.2. Solution Strategy

For systems with only persistent contacts, one could consider the constraints on position level to overcome the drift-off effect. For systems which undergo impacts, we explicitly decided not to apply the Schatzman-Paoli timestepping method on position level, but use a variant of the Moreau-Jean timestepping schemes, the Moreau’s midpoint rule, to gain a physically consistent approximation of the impulsive phenomena. When both impacts and longer periods of closed contacts occur, neither
the approach based on position level nor the one based on velocity level seems to be satisfactory. GEAR, GUPTA and LEIMKUHLER [78] proposed a solution to a related problem for persistent contacts: they enforce constraints on position and velocity level at the same time. The additional constraint equation is compensated by a second set of Lagrange multipliers. In the next chapter, we apply the Gear-Gupta-Leimkuhler method or stabilized index 2 formulation to systems with unilateral constraints to overcome the drift-off effect for closed contacts.
14. Gear-Gupta-Leimkuhler Method for Persistent Contacts

We explain the effect of the Gear-Gupta-Leimkuhler method or stabilized index 2 formulation [78] on the numerical solution of a unilaterally constrained mechanical system with the help of a bilaterally constrained slider-crank mechanism, which is adapted from [69]. The drift-off effect is analyzed for constraint formulations on position, velocity and acceleration level, as well as for the Gear-Gupta-Leimkuhler formulation [76, 29]. The text is based on [171].

14.1. Slider-Crank Mechanism with Bilateral Constraints

Figure 14.1: Slider-crank mechanism with bilateral constraints [171, Figure 5].

Figure 14.1 shows a bilaterally constrained slider-crank mechanism. The angles $q = (\theta_1, \theta_2)^T$ are chosen as generalized coordinates, the angular velocities $v = (\omega_1, \omega_2)^T$ as generalized velocities. With the same notation as in Chapter 4, we gain equations of motion which are more specific: unilateral contacts condense to bilateral constraints and impacts never occur. The generalized mass matrix reads

$$M = \begin{pmatrix}
J_1 + l_1^2 \left( \frac{m_1}{4} + m_2 + m_3 \right) & l_1 l_2 \cos (\theta_1 - \theta_2) \left( \frac{m_2}{2} + m_3 \right) \\
l_1 l_2 \cos (\theta_1 - \theta_2) \left( \frac{m_2}{2} + m_3 \right) & J_2 + l_2^2 \left( \frac{m_2}{4} + m_3 \right)
\end{pmatrix},$$

and the vector of generalized forces is given by

$$h = \begin{pmatrix}
-l_1 l_2 \sin (\theta_1 - \theta_2) \left( \frac{m_2}{2} + m_3 \right) \omega_2^2 - \Gamma l_1 \cos \theta_1 \left( \frac{m_1}{2} + m_2 + m_3 \right) \\
l_1 l_2 \sin (\theta_1 - \theta_2) \left( \frac{m_2}{2} + m_3 \right) \omega_1^2 - \Gamma l_2 \cos \theta_2 \left( \frac{m_2}{2} + m_3 \right)
\end{pmatrix}.$$
The bilateral constraint holds the slider at a fixed $y$-position

$$g = l_1 \sin \theta_1 + l_2 \sin \theta_2 = 0$$

by causing a constraint force in direction of

$$W = 
\begin{pmatrix}
  l_1 \cos \theta_1 \\
  l_2 \cos \theta_2
\end{pmatrix}.$$ 

### 14.2. Simulation Results

The simulations were run with the characteristics of Table 4.1. A backward Euler method which approximates constraint forces of first order as Moreau’s midpoint rule is implemented in MATLAB with

$$q_{i+1} = q_i + Y_{i+1} v_{i+1} \Delta t,$$

$$v_{i+1} = v_i + \Delta t M^{-1}_{i+1} \left[ h_{i+1} + W_{i+1} \lambda_{i+1} \right].$$

The fixed time step size is $\Delta t = 10^{-4}$ s. Considering the constraint $g_{i+1} = 0$ on position level yields a differential algebraic system of index 3 (Fig. 14.2c). This is known to be badly conditioned and, e.g., scaling of the constraint equation yields an improvement concerning the stability of the numerical integration scheme [92, 19]. Replacing the constraint by its respective time derivatives, improves the robustness of numerical solvers and is equivalent to the original problem from an analytic point of view when we use consistent initial values. Arnold [10] mentions this strategy in the context of index reduction. A drawback of index reduction is the drift-off effect [10]. Figure 14.2b shows the roughly linear development of the $y$-position of the slider for a long-time simulation of the index 2 system, that is, considering the constraint $W^T_{i+1} v_{i+1} = 0$ on velocity level. Figure 14.2a also displays the drift-off effect for the constraint formulation on acceleration level, that is, for the index 1 system. As presented in [76], the drift-off is expected to be parabolic and indeed the $y$-position increases with rising gradient. As a compromise of both robust simulation and asymptotically small drift-off, one usually considers the constraints on velocity level. To overcome the linear drift-off effect in the index 2 system, Gear, Gupta and Leimkuhler proposed a formulation which considers the constraints on position and velocity level simultaneously [78]. Concerning the mathematical model, the index 2 system extends to

$$\dot{q} = Y v + \left( \frac{\partial g}{\partial q} \right)^T \psi,$$

$$M \dot{v} = h + W \lambda,$$

$$g = 0,$$

$$g = 0.$$
The Lagrange multipliers $\psi$ compensate the added equation and the constraint is satisfied on position as well as on velocity level, maintaining the stability of the index 2 formulation (Fig. 14.2d).

Figure 14.2.: Constraint satisfaction of the slider-crank mechanism with bilateral constraints for different kind of kinematic levels [160, Figure 5.2].
15. Gear-Gupta-Leimkuhler Method for Unilateral Contacts

We analyze two extensions of Moreau’s midpoint rule (Section 1.4.2), a decoupled approach and a unified approach. The text is based on [171], though here we significantly improve the decoupled approach.

15.1. Decoupled Approach

We adapt Moreau’s midpoint rule by adding a correction term enforcing the non-penetration constraint at the end of each time step:

\[ v_{i+1} = v_i + M_i^{-1} (h_{M_i} \Delta t + W_{N,M_i} A_{N,i+1}) \]  
\[ A_{N,i+1} = \text{proj}_{R_0^+} \left( A_{N,i+1} + \left( g_{N,i+1} - r \left( \frac{\partial g_N}{\partial q} \right)^T_{M_i} \Psi_{i+1} \right) \right) \]  
\[ q_{i+1} = q_i + Y_{M_i} \frac{v_{i+1} + v_i}{2} \Delta t + \left( \frac{\partial g_N}{\partial q} \right)^T_{M_i} \Psi_{i+1} \]  
\[ \Psi_{i+1} = \text{proj}_{R_0^+} \left( \Psi_{i+1} - rg_{N,i+1} \right) . \]

In contrast to [171], we do not use any first order approximations of the gap velocities or of the gaps. In this way, we avoid chattering of the gap velocities and obtain a better energetic behavior of the time-stepping scheme. Moreover, a decoupled calculation of generalized position and velocity approximations is possible. As in Moreau’s midpoint rule, the calculation of the generalized velocities \( v_{i+1} \) is achieved by using the approximate average Lagrange multiplier \( A_{N,i+1} \). Consecutively, the generalized velocities \( v_{i+1} \) are used to determine an explicit forecast:

\[ q_{n+1} = q_i + Y_{M_i} \frac{v_{i+1} + v_i}{2} \Delta t . \]

The generalized position \( q_{n+1} \) is used as an initial value for the iterative computation of \( q_{i+1} \) and of the approximate average Lagrange multiplier \( \Psi_{i+1} \approx \int_{I_{i+1}} \psi dt \). Whereas the projection subspace is known for the velocities, it has to be determined for the positions by iteration.

The equations of motion are derived via an energy principle, the impact law results from NEWTON’s kinematic considerations. However, the additional term \( \left( \frac{\partial g_N}{\partial q} \right)^T_{M_i} \Psi_{i+1} \) does not correspond to any physical principle but can be interpreted...
as part of the Karush-Kuhn-Tucker conditions for a *position coordinate projection* at the end of each time step concerning the Euclidean metric:

\[
\min_{q_{i+1}} \|q_{i+1} - \bar{q}_{i+1}\|^2 ,
\]

\[
g_N(q_{i+1}) \geq 0 .
\]

Equations (15.1)-(15.4) do not include a *derivative projection* like the Gear-Gupta-Leimkuhler method [53, 54]. The negative consequence of a not suitable position coordinate projection is a bad energetic behavior. Also concerning easy nonsmooth mechanical examples this is well-known [187]. As a workaround in [187], it is suggested to introduce a penetration tolerance depending on the specific setting. Alishenas [9] outlines an error analysis for persistent contacts. Working with the underlying ordinary differential equations, constraints on position and velocity level are interpreted as invariants that have to be preserved. It turns out that the error in the generalized velocity is dominant, and that a sequential position-velocity coordinate projection may be the most effective way to obtain an improvement of the energetic behavior. Although we already work on velocity level, we try to complete the position coordinate projection (a half projection method) with an adaptation based on Alishenas’ idea [9]. Other computed quantities remain unchanged. The governing equations are

\[
v_{i+1}^a = v_{i+1} + W_{N,M_i} \Psi_{i+1}^a ,
\]

\[
\Psi_{i+1}^{a,i} = \text{proj}_{\mathbb{R}^2} \left( \Psi_{i+1}^{a,i} - r \left( g_{N,i+1,Z_i^{i+1}} + \epsilon g_{N,i+1,Z_i^{i+1}} \right) \right) .
\]

This projection is known as full projection method and may also change the energy of the system. Such a phenomenon is known for coordinate projection methods and shown in Fig. 15.1 for the simulation of the slider-crank mechanism from Chapter 4, with a coefficient of restitution \( \epsilon_N = 0.1 \) and a time step size \( \Delta t = 10^{-4} \) s.

### 15.2. Unified Approach

In our opinion, the generalized constraint directions have to couple (15.1) and (15.3) to ensure a physically accurate behavior in general and from the early beginning. Our proposition is presented in the next sections.

#### 15.2.1. Discretization Scheme

A possible coupling is the implicit evaluation of the constraint matrices

\[
W_{N,M_{i+1}} = W_N \left( \frac{q_{i+1} + q_i}{2} \right) , \quad \left( \frac{\partial g_N}{\partial q} \right)_{M_i} = \frac{\partial g_N}{\partial q} \left( \frac{q_{i+1} + q_i}{2} \right) .
\]
As this strategy already forces the solution of a nonlinear system of equations, we additionally implicitly evaluate the generalized force vector

\[ \tilde{h}_{M,i+1} = h\left(\frac{q_{i+1} + q_i}{2}, \frac{v_{i+1} + v_i}{2}\right) \]

to benefit from a more stable discretization of important stiffness contributions. The generalized mass matrix comprises geometric nonlinearities and its implicit evaluation needs quite large effort. For this reason, we choose the explicit evaluation \( M_M \) :

\[
q_{i+1} = q_i + \frac{Y_{i+1}v_{i+1} + Y_i v_i}{2} \Delta t + \left(\frac{\partial g_N}{\partial q}\right)_M^T \Psi_{i+1}, \tag{15.5}
\]

\[
v_{i+1} = v_i + M^{-1}_M \left[ \tilde{h}_{M,i+1} \Delta t + \tilde{W}_{N,M,i+1} A_{N,i+1} \right]. \tag{15.6}
\]

We write the implicit evaluations

\[ \tilde{W}_{N,M} : \mathbb{R}^{N_d} \to \mathbb{R}^{N_d \times N_e}, \tilde{q} \mapsto \tilde{W}_{N,M}(\tilde{q}) = W_N\left(\frac{\tilde{q} + q_i}{2}\right), \]

\[ \left(\frac{\partial g_N}{\partial q}\right)_M : \mathbb{R}^{N_d} \to \mathbb{R}^{N_e \times N_d}, \tilde{q} \mapsto \left(\frac{\partial g_N}{\partial q}\right)_M(\tilde{q}) = \frac{\partial g_N}{\partial q}\left(\frac{\tilde{q} + q_i}{2}\right) \]

\[ \tilde{h}_M : \mathbb{R}^{N_d} \times \mathbb{R}^{N_d} \to \mathbb{R}^{N_d}, (\tilde{q}, \tilde{v}) \mapsto \tilde{h}_M(\tilde{q}, \tilde{v}) = h\left(\frac{\tilde{q} + q_i}{2}, \frac{\tilde{v} + v_i}{2}\right) \]

as functions depending on the unknown quantities. It is \( \tilde{W}_{N,M,i+1} = \tilde{W}_{N,M}(q_{i+1}), \)

\[ \left(\frac{\partial g_N}{\partial q}\right)_M = \left(\frac{\partial g_N}{\partial q}\right)_M(q_{i+1}) \]

and \( \tilde{h}_{M,i+1} = \tilde{h}_M(q_{i+1}, v_{i+1}) \). We define a nonlinear system of equations

\[ 0 = \varphi_{T^*}(x_{i+1}, T^*) \]
with an index set $I^*$, which is defined later, and depending on the unknown variables

$$x_{i+1,I^*} = (q_{i+1}^T \; v_{i+1}^T \; A_{N,i+1,I^*}^T \; \Psi_{N,i+1,I^*}^T)^T. \tag{15.8}$$

Solving the reduced system of equations (15.7) is equivalent to computing the roots of the function $\varphi_{I^*}(x_{I^*})$. This is accomplished with Newton’s method. Thereby, the derivative of $\varphi_{I^*}(x_{I^*})$ with respect to $x_{I^*}$ can be deduced by eliminating rows and columns corresponding to inactive contacts from the following matrix

$$\frac{\partial \varphi}{\partial x} \bigg|_{x_{I^*}^m} = \begin{pmatrix}
I - \frac{\partial}{\partial q} \left( \frac{\partial g_N}{\partial q} \right)_M^T \Psi - \frac{\partial}{\partial q} \left( \frac{\partial g_N}{\partial q} \right)_M^T \left( \frac{\partial g_N}{\partial q} \right)_M \Delta t - Y \frac{\Delta t}{2} & \cdots \\
- M_{N,i}^{-1} \frac{\partial h_M}{\partial q} \Delta t + \frac{\partial W_{N,M}}{\partial q} \Lambda & I - M_{N,i}^{-1} \frac{\partial h_M}{\partial q} \Delta t & \cdots \\
\frac{\partial}{\partial x} \left( A_{N,I^*} - \text{proj}_{R_0^+} \left( A_{N,I^*} - r \left( g_{N,I^*} + \varepsilon N g_{N,I^*} \right) \right) \right) & \cdots \\
\frac{\partial}{\partial x} \left( \Psi_{I^*} - \text{proj}_{R_0^+} \left( \psi_{I^*} - r g_{N,I^*} \right) \right) & \cdots \\
\cdots & 0 & \cdots \\
- M_{N,i}^{-1} \frac{\partial W_{N,M}}{\partial q} & 0 & \cdots \\
\cdots & \cdots & \cdots 
\end{pmatrix} \bigg|_{x_{I^*}^m} \tag{15.9}$$

In order to turn the Newton iteration into a nonsmooth Newton method, hence applicable to projection formulations, we choose an element of the corresponding subdifferential, that is, a subgradient [156]. With an appropriate function $f(x)$, we distinguish two cases:

$$\frac{\partial}{\partial x} \left( \text{proj}_{R_0^+} \left( f(x) \right) \right) \bigg|_{x_{I^*}^m} = \begin{cases} 
\frac{\partial}{\partial x} f(x) \bigg|_{x_{I^*}^m}, & \text{if } f(x_{I^*}^m) > 0, \\
0, & \text{else}.
\end{cases}$$

For an example, we analyze the if-case of the proj function. We have that the third row of $\varphi$ is given by

$$\frac{\partial \varphi_3}{\partial q} = r \frac{\partial}{\partial q} \left( W_{N}^T v \right),$$

$$\frac{\partial \varphi_3}{\partial v} = r W_{N}^T,$$

$$\frac{\partial \varphi_3}{\partial \Lambda} = 0,$$

$$\frac{\partial \varphi_3}{\partial \Psi} = 0.$$
Specify characteristics, start time \( t = 0 \), end time \( T \), time step size \( \Delta t \)

Initialize \( i = 0 \)
For \( t < T \)
   Evaluate \( M_{M_i} \)
   Evaluate \( g_{N,M_i} \) and define active set \( I_{1,M}^{i+1} \)
   Evaluate \( \dot{g}_{N_i} \)
While nonsmooth \( \text{NEWTON} \) iteration not converged
   Compute residuum (15.7)
   Compute Jacobian (15.9)
   Update \( x^{n+1} \)
Write result of time step, e.g., \( x_{i+1} \)
Update \( t \) by \( \Delta t \) and increment \( i \) by 1

Postprocessing

**Figure 15.2.:** Flowchart of the unified timestepping scheme [171, Figure 7].

For the fourth row, it is

\[
\frac{\partial \varphi_4}{\partial q} = \frac{\partial g_N}{\partial q},
\]
\[
\frac{\partial \varphi_4}{\partial v} = 0,
\]
\[
\frac{\partial \varphi_4}{\partial \Lambda} = 0,
\]
\[
\frac{\partial \varphi_4}{\partial \Psi} = 0.
\]

As long as the active set is empty, all Lagrange multipliers are equal to zero and the system of equations is solved without the constraint part. As soon as the active set is not empty, the system of equations is extended by \( \text{NEWTON} \)'s impact law and the non-penetration constraint, respectively. Thereby, the index set \( I^* = I_{1,M}^{i+1} \) is sufficient for the example of the slider-crank mechanism with the characteristics given in Table 4.1 because of the resulting symmetric trajectory of the center of gravity of the slider. However, the index set \( I^* = I_{1,M}^{i+1} \) does not consider constraints which are inactive at position \( q_i + \frac{\Delta t}{2} v_i \). The (heuristic) choice of the predictor seems to be important for the accurate stabilization of all gap functions which will be active at \( t_{i+1} \). Recently, Acary [2] discussed several different predictors. However as the only reliable way to include all possibly active gap functions at \( t_{i+1} \), it is suggested to extend the active set of constraints iteratively by adding additional unilateral constraints in case there is a violation. Starting from the empty set, the necessary calculations in each time step get more extensive but are important for general examples.

Our algorithm is set up as shown in Fig. 15.2 and is implemented in MATLAB.
15.2.2. Simulation Results

![Graphs showing movement of the center of gravity of the slider for different coefficients of restitution and the unified timestepping method.](image)

**Figure 15.3:** Movement of the center of gravity of the slider for different coefficients of restitution and the unified timestepping method [171, Figure 8].

In Fig. 15.3, the results concerning the slider-crank mechanism with unilateral constraints and characteristics as in Table 4.1 are presented using a time step size $\Delta t = 10^{-5}$ s. The qualitative behavior is similar to the behavior for Moreau’s midpoint rule shown in Fig. 13.1. Especially for high coefficients of restitution, the patterns look similar. The change in the theoretical framework mainly affects persistent contacts, which rarely occur for $\epsilon > 0.5$. In contrast, for $\epsilon = 0.1$, the drift-off effect has a comparatively high influence. The proposed unified timestepping scheme yields a distinct change in the system’s behavior. The drift-off effect does no longer occur and the non-penetration condition is satisfied improving the physical accuracy in comparison to Fig. 13.1.

The development of the gap functions and their time derivatives for $\epsilon = 0.1$ is presented in Fig. 15.4. The drift-off effect has vanished and the gap functions are not negative anymore. The gap velocities are still slightly smaller than zero in time periods where the drift-off effect occurred in the previous simulations. However, this slow trend to increasing permeation is compensated by the second set of approximate average Lagrange multipliers enforcing the non-penetration constraint.

To further investigate the physical accuracy of the method, the qualitative development of the total energy for Moreau’s midpoint rule and the unified approach is shown in Fig. 15.5. The total energy of simulation differs slightly after four seconds. As Moreau’s midpoint rule does not ensure the compliance of the constraints, a reference line is shown based on the same algorithm as the unified approach but
Figure 15.4.: Gap functions and their time derivatives for \( \epsilon = 0.1 \) and the unified timestep-method [171, Figure 9].

Figure 15.5.: Total energy of the slider-crank mechanism for different simulation methods [171, Figure 10, adapted].

only forcing the impact law like Moreau’s midpoint rule and neglecting the non-penetration condition. The reference line is based on (15.7) without the last row and without the term \( \left( \frac{\partial h\alpha}{\partial q} \right)_M^T (q_{i+1}) \Psi_{i+1} \) in the first row. The total energy development for the reference system is slightly smaller than for the unified approach and a bit higher than for Moreau’s midpoint rule. The proposed approach leads to the same qualitative behavior of the total energy and to slightly different quantitative results.

Due to the implicit discretization, the computing time increases by a factor of ten in contrast to the explicit Moreau’s midpoint rule when using the same time step size. Nevertheless, we gain a stable discretization.
Part V.

Conclusion and Outlook
16. Summary

We presented a framework for the simulation of nonsmooth mechanical systems with friction and impacts as well as rigid and flexible bodies with many possible interactions. The formalism consistently treats velocity jumps, e.g., those due to impacts, and benefits from nice properties of established integration methods.

We introduced the nonsmooth framework in the first three chapters. We motivated nonsmooth mechanics and computational contact mechanics and presented various examples from industrial applications which emphasize the relevance of nonsmooth modeling techniques. The abstract framework of nonsmooth mechanics involves measure theory and functional analysis. It is necessary to delve into the formulation of timestepping schemes. We outlined solution methods for nonsmooth constraint equations. These equations have to be solved in each integration step and build-up the kernel of the algorithms presented. At the end, open-source software for nonsmooth mechanical system simulation was presented.

In the second part of the work, timestepping schemes based on time-discontinuous Galerkin methods were presented (Chapters 5-9 and Chapters A-C). They represent impulses, if necessary, and automatic local order elevation for state variables, if possible, at the same time. Constraints were formulated on acceleration or velocity level in an implicit way using an augmented Lagrangian technique with semi-smooth Newton schemes without penetration; all other evaluations were explicit, which yielded half-explicit methods. The schemes were derived and explained in detail concerning different criteria and were applied to various examples with multi-contact, flexibility and industrial relevance. As a compromise between classic timestepping schemes and event-driven integration, half-explicit timestepping schemes distinguish impulsive and non-impulsive periods for all examples in a geometrically consistent way and reduce the computing time significantly. For flexible examples, we recognized the drawback of explicit schemes, that is, stability issues.

The third part tackled stability issues and presented a framework of impulsive corrections, when these are necessary at the end of each discretization interval. The correction process was automatic and was evaluated on the same kinematic level as the piecewise non-impulsive trajectory, that is, on velocity level. The resulting overall mixed timestepping schemes were consistent for impulses and benefited from higher order in non-impulsive periods and all advantages of the base integration schemes used to calculate the approximation per time-step. We presented a nonsmooth adaptation of the generalized-\( \alpha \) method, the Bathe method and the ED-\( \alpha \) method. Our method was applied to a slider-crank mechanism with a flexible connecting rod, impacts and dry friction. The elastic behavior of the connecting rod was compared using the different base integration schemes and a modal approach. The results were validated
with respect to the half-explicit timestepping methods from the second part and a rigid body simulation in \textit{SIMPACK}. In comparison with the generalized-\(\alpha\) method, the Bathe-method and the ED-\(\alpha\) method behaved more robust in the nonlinear regime. These methods were more expensive per time-step but they required less steps, and the methods remained stable, even if the Newmark-type integrator failed for large deformations and long time duration dynamic response calculations.

In the fourth part, we proposed a timestepping scheme, which enables to achieve not only compliance of the impact law but also of the non-penetration constraint. It was shown that the decoupled application of the Gear-Gupta-Leimkuhler method can be interpreted as a coordinate projection to the non-penetration constraint at the end of each time step, rather than as a derivative projection. As this strategy does not lead to an energy-consistent discretization, our proposed method couples position and velocity level with an implicit evaluation of the constraint matrix. We demonstrated the overcoming of the drift-off effect and the performance of our approach.
17. Epilogue

17.1. Achievements

The contribution of the manuscript is the discussion and significant extension of time integration methods from nonsmooth mechanics and computational mechanics. The achievements are threefold and discussed in three separate parts of the work.

We embed classic timestepping schemes from nonsmooth mechanics in the broad concept of time-discontinuous Galerkin methods. This yields further theoretical foundation to these methods. As we show, time-discontinuous Galerkin schemes somehow connect classic timestepping schemes and event-driven schemes. Techniques from both time-integration strategies are necessary to formulate the proposed timestepping schemes for nonsmooth mechanical systems. We succeed in gluing these strategies and benefit from advantages as well as avoid disadvantages. A family of highly efficient half-explicit timestepping schemes on acceleration and velocity level is proposed as an example. It reduces the computing time significantly.

We abstract the concept and explain what 'gluing' of timestepping and event-driven schemes means: without resolving the points of discontinuity, we only apply an impulsive correction to a nonsmooth propagation if at least one impact occurs in the preceding discretization interval (Fig. 17.1). The variable $q_i$ denotes the position, the variable $v_i^\pm$ denotes the velocity at the time $t_i$. The velocity may jump, indicating that we have to split the time $t_i$ in the left-hand $t_i^-$ and right-hand limit $t_i^+$. Although,
we discuss only base integration schemes with high-frequency damping in this work, one is free to choose every integration scheme tailored for a specific application. The framework takes care of making the base integration scheme consistent to impulsive forces. These, as we have outlined, do not only occur in rigid body systems, but also in semi-discrete systems from computational contact mechanics. This fact supports the relevance of the proposed ideas.

In this manuscript, we offer a detailed comparison between the generalized-\(\alpha\) method, the Bathe method and the ED-\(\alpha\) method for nonlinear flexible applications. Such a comparison is very rare in the literature as one usually considers the generalized-\(\alpha\) method as the standard solution method for practical problems. We highlight the benefits of the Bathe method as a compromise between the efficient generalized-\(\alpha\) method, which has some drawbacks for nonlinear problems, and the more sophisticated, but also more expensive ED-\(\alpha\) method. We present a comparison of typical phenomena and computational cost.

The base integration schemes from Fig. 17.1 and Moreau-Jean timestepping schemes in general are formulated on velocity level. They are reduced concerning the differential index and insofar better conditioned than position level discretizations. Concerning impacting mechanical systems, it is even more important that the schemes are physically consistent due to the impulsive concept. In this work, missing geometric consistency is added by projection schemes. Coordinate projections are efficient, whereas the Gear-Gupta-Leimkuhler method is more expensive but offers derivative projections in the sense of constraint forces: the residuum is orthogonal to the tangent of the constraint manifold. This makes the latter method energy-consistent, if the underlying integration scheme is well-defined. Coordinate projection methods usually change the total energy of a system. We present and discuss both types of projection methods for nonsmooth mechanical systems.

17.2. Perspective

There are many time-integration schemes for computational mechanics. Each one is said to have certain advantages and disadvantages. Especially, we mention Laursen-Love schemes with contact/velocity updates and Laursen-Chawla schemes with active set strategies [119, 24, 52]. We have discussed similar concepts in this work: impulsive corrections and the Gear-Gupta-Leimkuhler method for nonsmooth mechanical problems. Answering questions concerning differences between these concepts is not only a philosophical challenge. A comparison of time-integration schemes for computational mechanics and nonsmooth mechanics with industrial applications would build-up on our presented work and improve the mutual understanding of time-integration schemes from computational mechanics and nonsmooth mechanics. One could include integration schemes based on high-frequency damping, e.g., the explicit generalized-\(\alpha\) method [98], or variational concepts, like Lie group concepts [99, 114]. Such a comparison may also deepen the mathematical analysis to prove characteristics observed numerically, especially for nonlinear problems. Nowadays, automatic step-size control is done only heuristically for nonsmooth mechanical systems. The
concept of time-discontinuous Galerkin methods may offer the opportunity to develop estimators based on variational concepts. Appropriate metrical descriptions have to be found to measure the integration error. Questions concerning the correct metrics for projection methods are related and should be included in the proposed comparison of time-integration schemes for industrial examples.
Appendix
A. Time Discontinuous Galerkin Methods

The "forecasting trapezoidal rule" from Chapter 7 is the easiest example of $D^+$ or $D^-$ timestepping schemes based on Time-Discontinuous Galerkin (TDG) methods. These time-integration schemes consistently improve the behavior of classic timestepping schemes during non-impulsive periods (Chapter B). In this chapter, we give an introduction to $D^+$ and $D^-$ timestepping schemes and illustrate basic properties. The text is based on [161].

Discontinuous Galerkin (DG) methods are special mortar methods, which have been originally established for space discretization. Nowadays DG methods are mainly used for convection-dominated flow problems [21, 43]. When flow problems exhibit a time-dependence, the time discretization is designed using Runge-Kutta (RK) schemes. The overall discretization is fairly not an easy task because of the connection between space and time discretization due to the Courant-Friedrich-Levy (CFL) condition. Additional stabilization strategies have to be applied. The final Runge-Kutta discontinuous Galerkin methods are of Total Variation Diminishing (TVD), Essentially Non-Oscillatory (ENO) or Weighted Essentially Non-Oscillatory (WENO) type [44]. We use DG methods not for the space discretization but directly for the time discretization. The derivation follows [118], which is considered to be the first contribution. Also [103, 18, 172, 8, 190] have motivated our approach.

DG methods use discontinuous trial functions in any situation. In contrast, eXtended (XFEM) or Generalized Finite Element Methods (GFEM) are based on situation dependent enrichment of a pool of continuous trial functions with discontinuous representatives [39]. GFEM and XFEM are identical developments of two different research groups and follow in general the same ideas as DG methods. We do not further consider these methods.

We start from Problem 1.3 aiming to define proper test functions and a finite dimensional basis for the discrete solution. We assume:

- test functions might have jumps across the intervals,
- test functions are continuous inside the intervals.

The first assumption leads to the expression of discontinuous Galerkin methods. The second assumption states that there is no instantaneous influence of the analytic nonsmooth dynamics on the numerical solution in-between an interval. Hence, the exact time of discontinuity is not resolved. We write $\varphi_q, \varphi_v \in C^0(\mathcal{I})$ for piecewise-continuous functions with respect to a time step-size partition $\mathcal{I} = \{I_1, \ldots, I_N\}$. 
A.1. Evaluations with Discontinuous Test Functions

It is not clear how to formulate (1.30) in terms of discontinuous test functions whose discontinuities coincide with those of the functional. Depending on the usage of appropriate mollifiers, that is, smooth cutoff functions, we define the distributional derivative of a piece-wise continuously differentiable functional \( v \in C^1(I) \) applied to discontinuous functions \( \varphi_v \in C^1(I) \) \([8]\). Let \( \epsilon > 0 \) and \( i \) an arbitrary index. An absolutely continuous characteristic mollifier

\[
\varphi^\epsilon_{v_i} : \mathbb{R} \to \mathbb{R}, \quad t \mapsto \varphi^\epsilon_{v_i}(t) := \varphi_v(t) \chi^\epsilon(t)
\]

of \( \varphi_v \), with support in \((t_{i-1}, t_i + \epsilon)\) (Fig. A.1), can be obtained with

\[
\chi^\epsilon(t) := \begin{cases} 
(t - t_{i-1})/\epsilon & \text{for } t_{i-1} \leq t < t_{i-1} + \epsilon, \\
1 & \text{for } t_{i-1} + \epsilon \leq t < t_i, \\
1 + (t_i - t)/\epsilon & \text{for } t_i \leq t < t_i + \epsilon, \\
0 & \text{elsewhere}.
\end{cases}
\]

The integration by parts formula yields

\[
\int D^- v \varphi^\epsilon_{v_i} \, dt = - \int_{t_{i-1}}^{t_i + \epsilon} v \dot{\varphi}^\epsilon_{v_i} \, dt - \int_{t_{i-1}}^{t_i} v \varphi^\epsilon_{v_i} \, dt - \int_{t_i}^{t_i + \epsilon} \dot{v} \varphi^\epsilon_{v_i} \, dt
\]

\[
= [[v]] \varphi^\epsilon_{v_i}(t_i) + \int_{t_{i-1}}^{t_i} \dot{v} \varphi^\epsilon_{v_i} \, dt + \int_{t_i}^{t_i + \epsilon} \dot{v} \varphi^\epsilon_{v_i} \, dt
\]

because of the continuity of \( \varphi^\epsilon_{v_i} \) at \( t_{i-1} + \epsilon \) and \( t_i \). In the limit \( \epsilon \to 0 \), we use \( \chi^\epsilon(t_i) = 1 \) and LEBESGUE’s theorem to achieve

\[
\lim_{\epsilon \to 0} \int D^- v \varphi^\epsilon_{v_i} \, dt = [[v]] \varphi_v(t_i) + \int_{t_{i-1}}^{t_i} \dot{v} \varphi_v \, dt.
\]
Hence, we define with a partition of unity ansatz

\[ < D^{-} v, \varphi_v > := \lim_{\epsilon \to 0} < D v, \sum_i \varphi_{v,i}^{t_i} > = \sum_i [ [v_i] ] \varphi_v \left( t_i \right) + \sum_i \int_{t_i-1}^{t_i} \dot{v} \varphi_v dt. \]  

This expression focuses on discontinuities at the right border \( t_i \) of \( I_i \). Alternatively, incorporating the left border \( t_{i-1} \) of \( I_i \) with a similar mollifier \( \chi_i^{t_{i-1}} \), the expression

\[ < D^{+} v, \varphi_v > := \sum_i [ [v_{i-1}] ] \varphi_v \left( t_i^{+} \right) + \sum_i \int_{t_{i-1}}^{t_i} \dot{v} \varphi_v dt \]  

is also consistent. The discontinuity evaluations could have been totally omitted, which is physically not satisfactory. Further, both the left and the right border of \( I_i \) could be considered. This would result in two term recursions, that is, multi-step methods, because the three intervals \( I_{i-1}, I_i \) and \( I_{i+1} \) are involved. The choice of the mollifier is crucial and might change both physics and numerical behavior. We use the expression \( d_i^{\pm} \) instead of \( d_i \) for the interaction measure.

### A.2. Timestepping Schemes Based on Time Discontinuous Galerkin Methods

We demonstrate the numerical approximation of Problem 1.2 with time discontinuous Galerkin methods and discuss its properties.

#### A.2.1. Definition of the Galerkin Approximation

Let \( \Phi_q^h, \Phi_v^h \subset C^0(\mathcal{I}) \) be finite dimensional subspaces for test functions with respective bases \( \mathcal{B}_{\Phi_q^h} := \{ \varphi_{q,k}^h \} \) and \( \mathcal{B}_{\Phi_v^h} := \{ \varphi_{v,k}^h \} \). Let further \( \Psi_q^h, \Psi_v^h \subset LBV(\mathcal{I}) \) be conforming subspaces for the choice of \( \dot{q} \)- and \( v \)-trial functions. The corresponding bases are given by \( \mathcal{B}_{\Psi_q^h} := \{ \psi_{q,k}^h \} \) and \( \mathcal{B}_{\Psi_v^h} := \{ \psi_{v,k}^h \} \). Then,

\[ q^h : \mathcal{I} \to \mathbb{R}^{N_d}, \quad t \mapsto q^h(t) := q_0 + \sum_k \int_{t_0}^{t} \psi_{q,k}^h ds \dot{q}_k^h, \]  

\[ v^h : \mathcal{I} \to \mathbb{R}^{N_d}, \quad t \mapsto v^h(t) := \sum_k \psi_{v,k}^h(t) \dot{v}_k^h \]  

is a representation of the numerical solution. The weights \( \{ \dot{q}_k^h \} \) and \( \{ \dot{v}_k^h \} \) are specified later. Inserting (A.3) and (A.4) into Problem 1.3 yields the discrete problem.
Problem A.1 (Petrov-Galerkin distribution differential inclusion) Solve
\[ \sum_k < \psi^h q_k, \varphi^h > q^h = \sum_k < Y \psi^h v_k, \varphi^h > v^h, \quad \forall \varphi^h \in \Phi^h, \]
\[ \sum_k < D^\pm \psi^h v_k, \varphi^h > v^h = \begin{cases} < M^{-1} h, \varphi^h > + < M^{-1} d^\pm, \varphi^h >, & \forall \varphi^h \in \Phi^h \end{cases}, \]

*together with the discrete initial conditions*
\[ q^h(0) := q_0 \in \mathbb{R}^{N_d}, \quad (A.5) \]
\[ v^h(0) := v_0 \in \mathbb{R}^{N_d}. \quad (A.6) \]

It is clear that \( Y, M^{-1}, h, \) and \( d^\pm \) are evaluated using \( q^h \) and \( v^h \). Contact and impact laws are evaluated to compute \( d^\pm \).

### A.2.2. Comparison with Moreau-Jean Timestepping Schemes

Problem A.1 is a general description which does not give appropriate time discretization schemes in all cases. The quality of the schemes is highly depending on the trial and test function subspaces. What are primary drivers for their selection?

- Problem A.2 depends on an initial value and describes a time-evolutionary solution. Also Problem A.1 must state an evolution process not depending on future information at each point in time.
- Experience has shown that for the description of nonsmooth dynamical systems, one-step methods are more appropriate than multi-step methods due to the lack of regularities of the right hand side [3].
- For efficient evaluation of the primal-dual pairings, easy test and trial functions should be used. They have to represent the smoothness of the analytical problem depending on, e.g., external forces but also on constraints.

One possibility to achieve these goals is the selection of piecewise polynomials. Choosing piecewise constant spaces \( \Phi^h = \Psi^h = \Psi^h = \Phi^h = \Psi^h = \Phi^h = \Psi^h = \Phi^h := \{ \chi^i \}_{i} \). Focusing on the time interval \( I_i \in \mathcal{I} \), we gain well-known classical timestepping schemes as a special case of Problem A.1. We distinguish the alternative evaluations, that is, \( D^\pm \) and \( d^\pm \).

**Case \( D^+ \) and \( d^- \)**

Problem A.2 (Implicit Moreau-Jean timestepping scheme) Solve
\[ q_i - q_{i-1} = Y_i v_i \Delta t_i, \quad (A.7) \]
\[ v_i - v_{i-1} = \int_{t_{i-1}}^{t_i} M^{-1} h dt + < M^{-1} d i^-, \chi^i > \] (A.8)

together with the discrete initial conditions (A.5), (A.6).

Again, \( M^{-1}, h \) and \( d i^- \) are evaluated using \( q^h \) and \( v^h \). It is noteworthy that there has been some freedom.

\[ t_i - t_{i-1} M^{-1} h d t \approx (t_i - t_{i-1}) \sum I \beta_f M^{-1} (q_i^h) h (q_i^h, v_i^{h\pm}, t_i) , \] (A.9)
\[ < M^{-1} d i^- , \chi^i > \approx \sum I M^{-1} (q_i^h) \Delta i_i \] (A.10)

with \((q_i^h, v_i^h, \Delta i_i, t_i) \in \mathcal{N}_i \) on velocity level. The classical Moreau-Jean timestepping scheme (\( \theta = 1 \)) can be achieved with \( \beta f_i := 1, t_i := t_i, q_i^h = q_i, v_i^h = v_i \) and \((q_i, v_i, \Delta i_i, t_i) \in \mathcal{N}_i \) on velocity level (Section 1.4.2).

**Case D^- and di^-**

**Problem A.3 (Explicit Moreau-Jean timestepping scheme)** Solve

\[ q_i - q_{i-1} = Y_{i-1} v_{i-1} \Delta t_i , \] (A.11)
\[ \mathbf{v}_i - \mathbf{v}_{i-1} = \int_{t_{i-1}}^{t_i} M^{-1} \mathbf{h} dt + < M^{-1} \mathbf{d}^-, \chi^i > \quad (A.12) \]

together with the discrete initial conditions (A.5), (A.6).

- We have chosen \( \dot{q}_k^h \) and \( \mathbf{v}_k^h \) to coincide with the values of the numerical solutions at the left end of the \( k \)th interval (Fig. A.2 right panel). The constant velocity in \( I_i \) is defined by \( \mathbf{v}_{i-1} = \mathbf{v}^h \left( t_{i-1}^+ \right) \) and the velocity jump, due to \( D^- \), occurs at the right side together with the impact, due to \( \mathbf{d}^- \).

- The classical Moreau-Jean timestepping scheme with \( \theta = 0 \) can be achieved with \( \beta_{f_1} := 1, t_1 := t_{i-1}, q_1^h = q_{i-1}, \mathbf{v}_1^h = \mathbf{v}_{i-1} \) and \( (q_i, \mathbf{v}_i, \Delta \mathbf{i}_i, t_i) \in \mathcal{N}_I \) on velocity level (Section 1.4.2).

**Case \( D^+ / D^- \) and \( \mathbf{d}^+ \)**

With \( \mathbf{d}^+ \), all impact evaluations take place in the semi-open interval \( [t_{i-1}, t_i) \). The evaluation of the impact laws at the right border of \( I_i \) is not maintained by the time discontinuous Galerkin scheme. Repeated tests have shown that this yields poor timestepping schemes [3]. We do not consider this case in the following.

Both Problem A.2 and Problem A.3 do not distinguish between contacts and impacts. The interaction measure \( \mathbf{d}^- \) summarizes both possibilities and is discretized directly. Hence, there is no splitting

\[ \mathbf{d}^-_i = (t_i - t_{i-1}) \beta_i \mathbf{r} \left( q_i^h \right) + \mathbf{p}_i \quad (A.13) \]

in smooth and nonsmooth interactions and the direct application of higher order schemes would not be successful.
B. Higher Order Timestepping Schemes

For the development of higher order timestepping schemes based on time discontinuous Galerkin methods, we start from Problem A.1. The procedure is similar to the embedding of the Moreau-Jean timestepping scheme in Section A.2.2. The interpretation of the 'forecasting trapezoidal rule' (Fig. 7.2) within the context of time-discontinuous Galerkin schemes is given in Section 7.4. The following text is based on [161, 167].

B.1. Selection of Bases Functions

When defining smooth discrete position and velocity solutions inside an interval \( I_i \), several questions arise. How can integrals with respect to arbitrary functions, e.g., \(< M^{-1} h, \varphi^{h,v}_v >\) or \(< M^{-1} d\dot{t}, \varphi^{h,v}_v >\), be calculated efficiently? Is it possible to represent the integrals with respect to polynomials of degree \( 2M_i \), e.g., \(< \psi^h_{q_k}, \varphi^{h,v}_v >\) or \(< \psi^h_{q_k}, \varphi^{h,v}_v >\), exactly by the same formula? This demand occurs when discretizing \( I_i \) with \( M_i + 1 \) points and nodal trial functions. The left and right border of \( I_i \) play a special role according to Section A. How can they be included as integration points? It turns out that the optimal quadrature rules of Gauss, Radau and Lobatto cannot positively respond to all above requirements. The inclusion of borders of \( I_i \) as integration points never allows exactness for polynomials of degree \( 2M_i \). On the other side, Clenshaw-Curtis quadrature formulas have positive weights, can be evaluated fast and stable by Fast Fourier Transformation algorithms and are competitive for general integrands as well [191]. We choose the latter methods and mention that it is no drawback that they are exact only for polynomials up to degree \( M_i \). They evaluate the integrand at the Chebychev points \( t_{i_k} \) for \( M_i \neq 0 \). For \( M_i = 0 \), no rule exists but both \( t_{i_0} = t_{i-1} \) or \( t_{i_0} = t_i \) are popular choices. The weights with respect to \( I_i \) and with respect to its lower subintervals satisfy

\[
\beta_i := \beta_{t_{i_k}} = \beta_{r_{i_k}} = \frac{1}{\Delta t_i} \int_{t_{i_k}}^{t_i} l_{ii} dt , \quad \beta_{i_k}(t^*) := \frac{1}{\Delta t_{i_{k-1}}} \int_{t_{i_k}}^{t^*} l_{ii} dt ,
\]

with the classical pruned Lagrange polynomials

\[
l_{ii} : I \to \mathbb{R} , \quad l_{ii}(t) := \begin{cases} \prod_{j \neq i} \frac{t - t_{ji}}{t_{ji} - t_{ji}} , & \text{for } t \in I_i , \\ 0 , & \text{for } t \notin I_i . \end{cases}
\]
We use these pruned Lagrange polynomials to define piecewise polynomial nodal bases for test functions $\Phi^h = \Phi^{h} := \mathcal{P}^\alpha (I)$ and for trial functions $\Psi^h = \Psi^{h} := \mathcal{P}^\alpha (I)$. This is a consistent approach, which actually yields a classical Galerkin scheme. If needed, multi-index notation $\alpha := (M_1, \ldots, M_N)$ allows for varying polynomial degrees for different elements of $I$. Altogether, this results in respective $(N + \sum M_i)$-dimensional bases $B_{\Phi^h}$, $B_{\Psi^h}$, $B_{\Phi^h}$ and $B_{\Psi^h}$. Their elements satisfy

\[ \varphi^h_{i+k} = \varphi^h_{i+k} = \psi^h_{i+k} = \psi^h_{i+k} = I_l \text{ with } k = \sum_{j=1}^{i-1} (M_j + 1) + l. \]  

(B.1)

In the following, we can study easy evaluable one-step evolution processes just by focusing on one interval $I_i$ and by using the related index notation.

### B.2. Definition of the General Scheme

Stages are the values of position, velocity or acceleration approximations which coincide with the peaks of the nodal bases relative to a sub-interval $I_i$:

\[
\begin{align*}
q^h_{i-1,0} &= q^h_{i-1}, & q^h_{i-1,1} &= q^h (t_i), & \cdots & q^h_{i-1,M_l-1} &= q^h (t_{iM_l-1}), & q^h_{i-1,M_l} &= q^h, \\
q^h_{i-1,0} &= q^h_{i+1}, & q^h_{i-1,1} &= q^h (t_i), & \cdots & q^h_{i-1,M_l-1} &= q^h (t_{iM_l-1}), & q^h_{i-1,M_l} &= q^h, \\
v^h_{i-1,0} &= v^h_{i+1}, & v^h_{i-1,1} &= v^h (t_i), & \cdots & v^h_{i-1,M_l-1} &= v^h (t_{iM_l-1}), & v^h_{i-1,M_l} &= v^h, \\
v^h_{i-1,0} &= v^h_{i+1}, & v^h_{i-1,1} &= v^h (t_i), & \cdots & v^h_{i-1,M_l-1} &= v^h (t_{iM_l-1}), & v^h_{i-1,M_l} &= v^h.
\end{align*}
\]

We insert the subspace specializations for one interval $I_i$, that is, functions like in (B.1), in Problem A.1 and we use $\Delta t^{-}$ because of stability reasons. On the one hand, $\Delta t^{+}$ evaluates impacts in the semi-open interval $[t_{i-1}, t_i)$ according to Section A.2.2. On the other hand, Section B.2.2 shows that $t_{i-1}$ will be the only reasonable candidate in this case. This is an explicit evaluation not ensuring the validity of the constraint after impact time. It is known to have bad properties for classical timestepping schemes [3].

Finally, we get the discrete initial conditions (A.5), (A.6). The position equation of Problem A.1 yields equations

\[
\sum_k \int_{I_i} l_k I_i dt \dot{q}^h_{i-1,k} = \sum_k \int_{I_i} l_k I_i Y dt v^h_{i-1,k}
\]

(B.2)

for $l \in \{0, \ldots, M_i\}$. We have to distinguish if the velocity jump should occur at the left or right interval border (Fig. A.2). With (A.9), (A.10) and (A.13), we obtain the following formulations from the velocity equation of Problem A.1.
B.2.1. Velocity Representation

**\(D^+\) timestepping schemes**

Knowing \(v_{i-1}^h\), we search \(v_{i-1,M_i}^h\) with equations

\[
\left[ v_{i-1,0}^h - v_{i-1}^h \right] l_i \left( t_{i-1}^+ \right) + \sum_k \int_{l_i} l_{i_k} l_{i_k} dt v_{i-1,k}^h = \Delta t_i \sum_k \beta_{i_k} M^{-1}_{i_k} \left[ h_{i_k}^+ + r_{i_k} \right] l_i \left( t_{i_k}^+ \right) + \sum_k M^{-1}_{k} p_k l_i \left( t_{i_k}^- \right), \tag{B.3}
\]

for \(l \in \{0, \ldots, M_i\}\).

**\(D^-\) timestepping schemes**

Knowing \(v_{i-1}^h = v_{i-1,0}^h\), we search \(v_{i}^h\) with equations

\[
\left[ v_{i,0}^h - v_{i}^h \right] l_i \left( t_{i}^- \right) + \sum_k \int_{l_i} l_{i_k} l_{i_k} dt v_{i,k}^h = \Delta t_i \sum_k \beta_{i_k} M^{-1}_{i_k} \left[ h_{i_k}^+ + r_{i_k} \right] l_i \left( t_{i_k}^+ \right) + \sum_k M^{-1}_{k} p_k l_i \left( t_{i_k}^- \right), \tag{B.4}
\]

for \(l \in \{0, \ldots, M_i\}\).

B.2.2. Impact Representation

We assume that the discrete velocity behavior inside an interval is continuously represented by the stage propagation. Hence, impacts are only allowed at the interval borders:

\[
\sum_k M^{-1}_{k} p_i l_i \left( t_{i_k} \right) = M^{-1}_{i} p_i l_i \left( t_{i} \right)
\]

with

\[
(q_i^h, v_i^{h,\pm} p_i, t_i) \in \mathcal{N}_i
\]

on velocity level. For \(D^+\), we use \(v_i^{-}\), and for \(D^-\), we use \(v_i^+\) (Fig. A.2). The discretization \(p_i\) equals the right limit of the interaction impulse at \(t_i\).

B.2.3. Weighting Integral Representation: Reduced Evaluation

The order of the local error is governed by the evaluation of (A.9), (A.10) and (A.13) with quadrature rules. Without changing the order, we approximate the
weighting integrals in (B.2), (B.3) and (B.4) by the same quadrature rule according to Clenshaw-Curtis (C-C) \[118, 191]:

\[
\sum_k \int_{l_i} l_k l_i \ ds \ q^h_{i-1,k} \ \overset{C}{=}_C \ \Delta t_i \ \sum_k \ \beta_{ik} \ q^h_{i-1,k, l_i} \ (t^\pm_{i_k}) = \Delta t_i \ \beta_{i} \ q^h_{i-1,l}, \quad (B.5)
\]

\[
\sum_k \int_{l_i} l_k l_i Y \ ds \ v^h_{i-1,k} \ \overset{C}{=}_C \ \Delta t_i \ \sum_k \ \beta_{ik} \ Y_{ik} \ v^h_{i-1,k, l_i} \ (t^\pm_{i_k}) = \Delta t_i \ \beta_{i} \ Y_{i} \ v^h_{i-1,l}, \quad (B.6)
\]

\[
\sum_k \int_{l_i} l_k l_i \ ds \ v^h_{i-1,k} \ \overset{C}{=}_C \ \Delta t_i \ \sum_k \ \beta_{ik} \ v^h_{i-1,k, l_i} \ (t^\pm_{i_k}) = \Delta t_i \ \beta_{i} \ v^h_{i-1,l}. \quad (B.7)
\]

Thereby, we evaluate \( l_i \) at the interior limit \( t^\pm_{i_k} \) of the sub-interval borders.

**B.2.4. Runge-Kutta Representation**

Substitution of (B.5) and (B.6) into (B.2) yields the collocation of \( M_i + 1 \) velocity stages

\[
q^h_{i-1,l} = Y_{i} \ v^h_{i-1,l}. \quad (B.8)
\]

We will search position stages \( \{ q^h_{i-1,l} \} \) knowing \( q^h_{i-1,0} \) with the fundamental theorem of calculus (cf. (B.13), (B.14), (B.18), (B.19)). The velocity expressions (B.3) and (B.4) are simplified to respective \( M_i + 1 \) equations by evaluating the nodal bases and by inserting (B.7):

\[
D^+ : \ [v^h_{i-1,0} - v^h_{i-1} l_i \ (t^+_i)] = \Delta t_i \ \beta_{i} \ \left\{ M_{i}^{-1} [h^+_i + r_i] - v^h_{i-1,l} \right\} + M_{i}^{-1} p_i l_i \ (t^-_i). \quad (B.9)
\]

For constant trial functions and appropriate definition of the integration point, e.g., either \( t_i \) or \( t_{i-1} \), Equation (B.9) is reduced to the implicit or explicit Moreau-Jean timestepping scheme. For at least linear trial functions, condition (B.9) expresses velocity jumps

\[
D^+ : \ v^h_{i-1,0} = v^h_{i-1} + \Delta t_i \ \beta_{i_0} \ \left\{ M_{i_0}^{-1} [h^+_i + r_i] - v^h_{i-1,0} \right\}, \quad (B.10)
\]

\[
D^- : \ v^h_{i} = v^h_{i-1,M_i} + \Delta t_i \ \beta_{i,M_i} \ \left\{ M_{i,M_i}^{-1} [h^-_i + r_i] - v^h_{i-1,M_i} \right\} + M_{i}^{-1} p_i \quad (B.11)
\]

and respective \( M_i \) stage relationships for accelerations in Table B.1. The values \( \dot{v}^h_{i-1,0} \) or \( \dot{v}^h_{i-1,M_i} \) are needed for the evaluation of (B.10) or (B.11) and are still missing. However, we observe that these are values of the acceleration, which is a polynomial of degree \( M_i - 1 \) in \( t_i \). This polynomial can be uniquely represented by the known \( M_i \) nodal values in Table B.1 as well as by respective and appropriate pruned Lagrangian
we obtain the following Runge-Kutta interpretation of higher order timestepping

Table B.1.: Stage relationship for accelerations [161, Table 1].

<table>
<thead>
<tr>
<th>stage</th>
<th>$D^+$</th>
<th>$D^-$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$l = 0$</td>
<td>$\dot{v}_{i-1,0}^h = ?$</td>
<td>$\dot{v}<em>{i-1,0}^h = M</em>{i_0}^{-1} [h_{i_0}^+ + r_{i_0}]$</td>
</tr>
<tr>
<td>$l \in {1, \ldots, M_i - 1}$</td>
<td>$\dot{v}<em>{i-1,l}^h = M</em>{iM_i}^{-1} [h_{iM_i}^- + r_{iM_i}] + \frac{M_{iM_i}^{-1} p_i}{\Delta t_{iM_i}}$</td>
<td>$\dot{v}_{i-1, M_i}^h = ?$</td>
</tr>
</tbody>
</table>

bases $\{\tilde{t}_{ik}^\pm\}_{k}$ [118]:

$$D^+ : \dot{v}^h (t) = \sum_{k=1}^{M_i} \tilde{t}_{ik}^+(t) \dot{v}_{i-1,k}^h, \quad D^- : \dot{v}^h (t) = \sum_{k=0}^{M_i-1} \tilde{t}_{ik}^-(t) \dot{v}_{i-1,k}^h.$$ (B.12)

Now, we evaluate the acceleration polynomials at $t_{i-1}$ or $t_i$ and get $\dot{v}_{i-1,0}^h$ or $\dot{v}_{i-1,M_i}^h$. Equation (B.12) will be also used to derive stage representations of the velocity with the fundamental theorem of calculus (cf. (B.16), (B.17), (B.20)). With

$$\beta^+_i := \frac{1}{\Delta t_i} \int_{t_{i-1}}^{t_i} \tilde{t}_{ik}^+ dt, \quad \beta^-_i (t^*) := \frac{1}{\Delta t_i} \int_{t_{i-1}}^{t^*} \tilde{t}_{ik}^- dt,$$

we obtain the following Runge-Kutta interpretation of higher order timestepping schemes based on time discontinuous Galerkin methods.

Problem B.1 ($D^+$ timestepping scheme) Let $M_i$ positive and $l \in \{0, \ldots, M_i\}$ for $i \in \mathbb{N}$. Solve simultaneously for the position

$$q_{i-1,l}^h = q_{i-1}^h + \Delta t_i \sum_{k} \beta_{ik} (t_i) Y_{ik} v_{i-1,k}^h,$$ (B.13)

$$q_i^h = q_{i-1}^h + \Delta t_i \sum_{k} \beta_{ik} Y_{ik} v_{i-1,k}^h,$$ (B.14)

and for the velocity

$$v_{i-1,0}^h = v_{i-1,0}^- + \Delta t_i \beta_{i0} \left\{ M_{i0}^{-1} [h_{i0}^+ + r_{i0}] - \sum_{k=1}^{M_i} \tilde{t}_{ik}^- (t_{i-1}) M_{ik}^{-1} [h_{ik}^- + r_{ik}] \right\}$$

$$- \tilde{t}_{iM_i}^+ (t_{i-1}) \frac{\beta_{i0}}{\beta_{iM_i}} M_{iM_i}^{-1} p_i,$$ (B.15)

$$v_{i-1,l}^h = v_{i-1,0}^h + \Delta t_i \sum_{k=1}^{M_i} \beta_{ik} (t_i) M_{ik}^{-1} [h_{ik}^- + r_{ik}] + \tilde{t}_{iM_i}^+ (t_i) \frac{M_{iM_i}^{-1} p_i}{\beta_{iM_i}},$$ (B.16)

$$v_i^h = v_{i-1,0}^h + \Delta t_i \sum_{k=1}^{M_i} \beta_{ik} M_{ik}^{-1} [h_{ik}^- + r_{ik}] + \tilde{t}_{iM_i}^+ \frac{M_{iM_i}^{-1} p_i}{\beta_{iM_i}},$$ (B.17)

together with (A.5), (A.6) and $(q_{i-1,k}^h, v_{i-1,k}^h, r_{ik}, t_{ik}) \in \mathcal{N}_C, (q_i^h, v_i^h, p_i, t_i) \in \mathcal{N}_I$.

For $D^-$, the notation is easier as the jump information is not propagated along $I_i$. 
Problem B.2 \((D^{-}\) timestepping scheme\) Let \(M_i\) positive and \(l \in \{0, \ldots, M_i\}\) for \(i \in \mathbb{N}\). Solve simultaneously for the position

\[
q_{i-1,l}^h = q_{i-1}^h + \Delta t_i \sum_k \beta_{ik} (t_i) Y_{ik} v_{i-1,k}^h , \tag{B.18}
\]

\[
q_i^h = q_i^h + \Delta t_i \sum_k \beta_{ik} Y_{ik} v_{i,k}^h , \tag{B.19}
\]

and for the velocity

\[
v_{i-1,l}^h = v_{i-1}^{h^+} + \Delta t_i \sum_{k=0}^{M_i-1} \beta_{ik} (t_i) M_{ik}^{-1} \left[ h_{ik}^+ + r_{ik} \right] , \tag{B.20}
\]

\[
v_i^{h^+} = v_{i-1, M_i}^{h^+} + \Delta t_i \beta_{i M_i} \left\{ M_{i M_i}^{-1} \left[ h_{i M_i}^- + r_{i M_i} \right] - \sum_{k=0}^{M_i-1} \tilde{\beta}_{ik} (t_i) M_{ik}^{-1} \left[ h_{ik}^+ + r_{ik} \right] \right\} ,
\]

\[+ M_{i}^{-1} p_i \tag{B.21}
\]

\(\) together with (A.5), (A.6) and \((q_{i-1,k}, v_{i-1,k}, r_{ik}, t_i) \in \mathcal{N}_C, (q_i^h, v_i^{h^+}, p_i, t_i) \in \mathcal{N}_I.\)

B.3. Local Error for ODE Methods

Because of (B.8) and Table B.1, \(D^+\) and \(D^-\) timestepping schemes are collocating methods inside each non-impulsive interval \(I_i\) [50]. Hence, the local error for non-impulsive episodes only depends on the adopted quadrature rule.

Theorem B.3 \((\text{Order of local error})\) Using Clenshaw-Curtis quadrature rules, the order of the local error for Problems B.1 and B.2 with contact force calculations on acceleration level satisfies

\[
p = M_i + 1
\]

in sufficiently smooth intervals \(I_i\).

Proof [50, Theorem 6.40] \(\)

This means that whenever we have a non-impulsive propagation of state and (contact) forces, the local error is automatically of higher order, that is, the numerical approximation is consistently improved. However, we do not know anything about errors due to velocity or interaction jumps. As they are propagated, they will affect the global error. The bouncing ball examples (Problems 5.1, 5.2, 5.3) support Theorem B.3 and indicate an order drop due to jumping velocities or interactions.
C. Dense Output and Error Evaluation

Timestepping discretizations define numerical solutions concerning the trial functions. Hence, not considering these trial functions and directly applying error norms to discrete numerical solution sequences may corrupt our error analysis. We have to define dense output formulas for position, velocity and interaction. According to [92], we choose cubic Hermite interpolation for positions and interaction impulses in each interval \( I_i \). We know the position stages and their derivatives, that is, the velocity stages. Concerning the interactions, we are aware of the stages of the contact forces \( \{ r_{i_l} \} \). We do not yet know the stages of the impulses, but we know that the impulse \( p_0 \) vanishes. Moreover our scheme has computed the final interaction \( i_{i_0} := p_{t_{i-1}} \) of the last interval equal to the right limit at \( t_i \). Hence, we use the fundamental theorem of calculus to compute

\[
i_i = p_{t_{i-1}} + \int_{t_{i-1}}^{t_i} r_i (s) \, ds = p_{t_{i-1}} + \Delta t_i \sum_k \beta_{i_k} (t_i) r_{i_k}.
\]

We use the same ansatz as for the velocity because of the mutual dependence in the constraint laws. On the other hand in the classical Moreau-Jean timestepping scheme, the impulse is directly calculated and assumed to be piecewise constant. The cubic Hermite interpolation will be exact up to piecewise quadratic velocity and contact force approximations and is sufficient for most practical applications. For the position, we obtain

\[
\hat{q}_{i_i}^h : [0,1] \to \mathbb{R}^N, \quad \xi \mapsto \hat{q}_{i_i}^h (\xi) := h_0 (\xi) q_{i_{i-1},0}^h + h_1 (\xi) (t_i - t_{i-1}) Y_{i-1,0} v_{i-1,0}^h + h_2 (\xi) q_{i_{i-1},M}^h + h_3 (\xi) (t_i - t_{i-1}) v_{i-1,M}^h ;
\]

and for the interaction, we obtain

\[
\hat{i}_{i_i}^h : [0,1] \to \mathbb{R}^N, \quad \xi \mapsto \hat{i}_{i_i}^h (\xi) := h_0 (\xi) i_{i_0} + h_1 (\xi) (t_i - t_{i-1}) r_{i_0} + h_2 (\xi) i_{i,M} + h_3 (\xi) (t_i - t_{i-1}) r_{i,M} ;
\]

with the standard Hermite polynomials

\[
\begin{align*}
h_0 : [0,1] &\to \mathbb{R}, \quad \xi \mapsto h_0 (\xi) := 2\xi^3 - 3\xi^2 + 1, \\
h_1 : [0,1] &\to \mathbb{R}, \quad \xi \mapsto h_1 (\xi) := \xi^3 - 2\xi^2 + \xi, \\
h_2 : [0,1] &\to \mathbb{R}, \quad \xi \mapsto h_2 (\xi) := -2\xi^3 + 3\xi^2, \\
h_3 : [0,1] &\to \mathbb{R}, \quad \xi \mapsto h_3 (\xi) := \xi^3 - \xi^2 .
\end{align*}
\]
and the parametrization

\[ [0,1] \rightarrow [t_{i-1}, t_i], \quad \xi \mapsto t := t_{i-1} + (t_i - t_{i-1}) \xi. \]

SciPy’s\(^1\) *barycentric interpolation* makes use of the collocation of the velocity stages for defining a dense output formula for the velocity. Finally, SciPy’s *Gauss-Konrod quadrature* based on the dense output formulas provides appropriate error formulas in \(L^1\)-norm for position, velocity and interactions. Hence, we pay attention to evaluate the local and global error as least as exact as the timestepping discretizations of our Python\(^2\) implementation. The Moreau-Jean timestepping with \(\theta = 1\) proposes piecewise linear position, as well as piecewise constant velocity and interaction impulse discretizations. With the forecasting trapezoidal rule, we have piecewise quadratic positions, piecewise linear velocities and piecewise quadratic interaction impulses.

---

1 http://www.scipy.org/
2 http://www.python.org/
D. Generalized-\(\alpha\) Method

For the analysis of the generalized-\(\alpha\) method, it is advantageous to reduce the coupled equations of motion to a series of uncoupled single degree of freedom systems using modal analysis and eigenvector orthogonality. The linear single degree of freedom system with angular frequency \(\omega\) is given by

\[
\ddot{q} + \omega^2 q = 0 ,
\]

where the terms related to external damping and forces are set to zero to study accuracy and stability properties of the algorithm. The generalized-\(\alpha\) method described in (11.1) to (11.6) can be written in the compact form

\[
X_{i+1} = A_{ga} X_i , \quad i \in \{0, 1, \ldots, N - 1\} ,
\]

where \(X_i = (q_i, \Delta tv_i, \Delta t^2 a_i)^T\) and \(A_{ga}\) is the amplification matrix for the generalized-\(\alpha\) method. The accuracy of an algorithm can be determined using the difference equation in terms of the displacement

\[
q_{i+1} - A_1 q_i + A_2 q_{i-1} - A_3 q_{i-2} = 0 ,
\]

where \(A_1\) is the trace of \(A_{ga}\), \(A_2\) is the sum of the principal minors of \(A_{ga}\) and \(A_3\) is the determinant of \(A_{ga}\). It can be shown [41] that the algorithm is second order accurate for unconstrained mechanical systems if

\[
\gamma = \frac{1}{2} - \alpha_m + \alpha_f .
\]

The parameter \(\gamma\) is responsible for numerical dissipation. For \(\gamma = 0.5\), there is no numerical dissipation, whereas for values \(\gamma > 0.5\) the numerical dissipation increases. The stability and numerical behavior of an algorithm depends on the eigenvalues of the amplification matrix. The spectral radius \(\rho\) of an algorithm is defined by

\[
\rho = \max(|\lambda_1|, |\lambda_2|, |\lambda_3|) ,
\]

where \(\lambda_i\) is the \(i\)th eigenvalue of \(A_{ga}\). An algorithm is unconditionally stable for linear problems if \(\rho \leq 1\) for all \(\Omega = \omega \Delta t \in [0, \infty)\). The generalized-\(\alpha\) method is unconditionally stable provided

\[
\alpha_m \leq \alpha_f \leq \frac{1}{2} , \quad \beta \geq \frac{1}{4} + \frac{1}{2} (\alpha_f - \alpha_m) .
\]
The spectral radius is a measure for numerical dissipation. A smaller spectral radius corresponds to greater numerical dissipation. Desirable dissipation properties have a spectral radius close to unity in the low-frequency domain. The value smoothly decreases as $\Omega$ increases. Typically, in the low-frequency domain, $|\lambda_3| \leq |\lambda_{1,2}|$. To preserve the smoothness as $\Omega$ increases, $|\lambda_3| \leq |\lambda_{1,2}|$ for all $\Omega \in [0, \infty)$. Violation of this condition will result in a cusp in the spectral radius plot where $\rho$ increases as $\Omega$ increases (point A in Fig. 11.1). We get the following eigenvalues of the amplification matrix in the high-frequency domain:

$$\lambda_{1,2}^\infty = \frac{1}{4\beta} \left( 4\beta - (2\gamma + 1) \pm j\sqrt{16\beta - (2\gamma + 1)^2} \right),$$

$$\lambda_3^\infty = \frac{\alpha_f}{\alpha_f - 1},$$

where $j = \sqrt{-1}$. High-frequency dissipation is maximized if the principal roots $(\lambda_{1,2}^\infty)$ become real, that is, $\Re(\lambda_{1,2}^\infty) = 0$. It can be shown from (D.6) that for the generalized-$\alpha$ method, this condition is satisfied if

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

which satisfies the second condition in (D.5). The generalized-$\alpha$ method can be described in terms of the two remaining free parameters $\alpha_m$ and $\alpha_f$. Using (D.4) and (D.7), we rewrite $\lambda_{1,2}^\infty$ in (D.6) as

$$\lambda_{1,2}^\infty = \frac{\alpha_f - \alpha_m - 1}{\alpha_f - \alpha_m + 1}.$$

Let $\rho_\infty$ denote the user-specified value of the spectral radius in the high-frequency limit. Since we require that $\lambda_3 \leq \lambda_{1,2}$ for all $\Omega$, it is $\rho = |\lambda_{1,2}^\infty|$. It has been shown that for a given level of high-frequency dissipation, that is, for fixed $\rho_\infty$, low-frequency dissipation is minimized if $\lambda_{1,2}^\infty = \lambda_3^\infty$, that is, if $\alpha_f = (\alpha_m + 1)/3$ (dotted line in Fig. 11.1). It is more convenient to describe this optimal case by defining $\alpha_m$ and $\alpha_f$ in terms of $\rho_\infty$:

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

(D.8)

It is worth to mention that for the exact solution of (D.1), we get:

$$\rho_{ex} = 1, \quad T_{ex} = \frac{1}{\omega \Delta t}. $$
E. Bathe-Method

Applying the Bathe-method (Fig. E.1) to the single degree of freedom system introduced in (D.1), we can write the amplification matrix [17]:

\[
A_{\text{Bathe}} = \frac{1}{D} \begin{pmatrix}
-144\Omega^2 + 19\Omega^4 & -144\Omega^2 + 5\Omega^4 & -28\Omega^2 \\
-96\Omega^2 + \Omega^4 & 144 - 47\Omega^2 & 48 - 4\Omega^2 \\
144 - 19\Omega^2 & 144 - 5\Omega^2 & 28
\end{pmatrix},
\]

where

\[
D = \left(16 + \Omega^2\right) \left(9 + \Omega^2\right).
\]

Analyzing the eigenvalues of \(A_{\text{Bathe}}\) shows that the method is unconditionally stable and \(\rho_\infty = 0\).

**Figure E.1.:** Bathe-method - red: trapezoidal rule, green: Euler backward rule [155, Figure 33].
F. Floating Frame of Reference Description for a Slider-Crank Mechanism

This chapter presents the derivation of the system matrices for the slider-crank mechanism in Fig. F.1.

\[ w_{x_1} = a_0 + a_1 x_1, \]
\[ w_{y_1} = a_2 + a_3 x_1 + a_4 x_1^2 + a_5 x_1^3. \]

Using the above relation for the displacement and considering nodal coordinates (Fig. F.1) for each element, we obtain space-independent shape functions of the beam.
element:

$$S^{ij} = \begin{pmatrix}
1 - \xi & 0 & 3\xi^2 - 2\xi^3 & l(\xi - 2\xi^2 + \xi^3) \\
0 & 1 - 3\xi^2 + 2\xi^3 & 0 & 0 \\
\xi & 0 & 3\xi^2 - 2\xi^3 & l(\xi^3 - \xi^2)
\end{pmatrix};$$  \hspace{1cm} (F.1)

where \( \xi = x/l \).

### F.2. Floating Frame of Reference

**Figure F.2.:** Deformable body coordinate [155, Figure 35].

In the floating frame of reference formulation presented in this section, the configuration of each deformable body in the multibody system is identified by using two sets of coordinates: reference and elastic coordinates [174]. Reference coordinates define the location and orientation of a selected body reference (\( X^i, Y^i \) and \( \theta^i \) in Fig. F.2). Elastic coordinates describe the body deformation with respect to the body reference (\( q^{ij}_1 \) to \( q^{ij}_6 \) in Fig. F.1). The motion of the body is defined as the motion of its reference plus the motion of the material points on the body with respect to its reference (Fig. F.2). We write:

$$u^{ij}_{f} = S^{ij} q^{ij}_{f},$$  \hspace{1cm} (F.2)

where \( u^{ij}_{f} = (u_{f1} \ u_{f2})^T \) is the deformation vector of element \( j \) of deformable body \( i \), \( S^{ij} \) is the shape matrix of element \( j \), \( q^{ij}_{f} \) is the vector of elastic coordinates that contains the time dependent nodal values \( q^{ij}_{f1} \) to \( q^{ij}_{f6} \). For an arbitrary body \( i \) of the system, e.g., the flexible rod in this example, we select a body reference \( \{X^i, Y^i\} \), the location and orientation of which with respect to the global coordinate system are defined by a set of coordinates called reference coordinates and denoted as \( q^i_r \). For the planar motion of deformable bodies, which is a special case of three-dimensional
motion, the vector $q^i_r$ can be written in a partitioned form as

$$q^i_r = \begin{pmatrix} R^i & \theta^i \end{pmatrix}^T,$$

where $R^i$ is a set of Cartesian coordinates that define the location of the origin of the body reference (Fig. F.2) and $\theta^i$ is a set of rotational coordinates that describe the orientation of the selected body reference (in the present planar case, it is a scalar value). There is no rigid body motion between the body and its coordinate system. The floating frame of reference formulation does not lead to a separation between the rigid body motion and the elastic deformation.

In the case of a rigid body, the global position of an arbitrary point $P$ on the rigid body can be written in planar analysis as:

$$r^i_P = R^i + A^i u^i_P,$$

where $u^i_P$ is the local position of point $P$ and $A^i$ is the transformation matrix defined as

$$A^i = \begin{pmatrix} \cos \theta^i & -\sin \theta^i \\ \sin \theta^i & \cos \theta^i \end{pmatrix}.$$

In general, for deformable bodies, the distance between two arbitrary points on the deformable body does not remain constant because of the relative motion between the particles forming the body. The vector $u^i_P$ can be written as:

$$u^{ij}_P = u^{ij}_0 + u^{ij}_f = u^{ij}_0 + S^{ij} q^{ij}_f.$$

According to Fig. F.2 and what we have discussed so far, we describe the new position of an arbitrary point $P^i$ on the flexible body based on reference and elastic coordinates as:

$$r^{ij}_P = R^i + A^i u^{ij}_P = R^i + A^i (u^{ij}_0 + S^{ij} q^{ij}_f).$$

(F.3)

We summarize all the unknowns which are necessary to calculate the new position of the arbitrary point $P$ in the vector $q^{ij}$:

$$q^{ij} = \begin{pmatrix} R^i \\ \theta^i \\ q^{ij}_f \end{pmatrix}.$$

Differentiating (F.3) with respect to time yields

$$\dot{r}^{ij}_P = \dot{R}^i + \dot{A}^i u^{ij}_P + A^i \dot{u}^{ij}_P = \dot{R}^i + \dot{A}^i u^{ij}_P + A^i S^{ij} \dot{q}^{ij}_f,$$

where in case of planar motion, we have $\dot{A} = A_\theta \dot{\theta}$. Then, the velocity vector can be
written as
\[
\dot{r}_{ij}^P = \begin{pmatrix} I & A^i_\theta u^ij_p & A^i S^{ij} \end{pmatrix} \begin{pmatrix} \dot{R}^i \\ \dot{\theta}^i \\ \dot{q}^ij_f \end{pmatrix},
\]
where \( I \) is identity matrix, and \( A^i_\theta \) is the partial derivative of the transformation matrix with respect to the rotational coordinate \( \theta^i \):
\[
A^i_\theta = \begin{pmatrix} -\sin \theta^i & -\cos \theta^i \\ \cos \theta^i & -\sin \theta^i \end{pmatrix}.
\]
Equation (F.4) can also be written as
\[
\dot{r}_{ij}^P = L^{ij} \dot{q}^{ij},
\]
where \( L^{ij} = \begin{pmatrix} I & A^i_\theta u^ij_p & A^i S^{ij} \end{pmatrix} \).

F.3. Mass Matrix

In this section, we develop the kinetic energy of deformable bodies and point out the differences between the inertia properties of deformable bodies that undergo finite rotations and the inertia properties of both rigid and structural systems. Then, we explain how to assemble the total mass matrix considering two rigid bodies (crank and slider) and joint constraints.

For constructing the mass matrix, we use the following definition of the kinetic energy for element \( j \) of deformable body \( i \):
\[
T^{ij} = \frac{1}{2} \int_{V^{ij}} \rho^{ij} \dot{r}^{ijT} \dot{r}^{ij} dV^{ij},
\]
where \( \rho^{ij} \) and \( V^{ij} \) are the mass density and volume of the element \( j \), \( \dot{r}^{ij} \) is the global velocity vector of an arbitrary point of the element, respectively. Using the expression of the velocity vector (F.4), we write the kinetic energy as
\[
T^{ij} = \frac{1}{2} \dot{q}^{ijT} \int_{V^{ij}} \rho^{ij} L^{ijT} L^{ij} dV^{ij} \dot{q}^{ij},
\]
where \( M^{ij} \) is recognized as the symmetric mass matrix of body \( i \). It is defined as
\[
M^{ij} = \int_{V^{ij}} \rho^{ij} L^{ijT} L^{ij} dV^{ij} = \int_{V^{ij}} \rho^{ij} \begin{pmatrix} I \\ (A^i_\theta u^ij_p)^T \\ (A^i S^{ij})^T \end{pmatrix} \begin{pmatrix} I & A^i_\theta u^ij_p & A^i S^{ij} \end{pmatrix} dV^{ij}.
\]
\[
F.3. \text{ Mass Matrix} \quad 195
\]

\[
\frac{\rho_{ij}}{V_{ij}} \int \left( \begin{array}{c}
I \\
A_i^j u_p^{ij} \\
u_p^{ijT} u_p^{ij} \\
A^i S^{ij} \\
S^{ijT} S^{ij} \\
s_{ij} \end{array} \right) dV^{ij},
\]

(F.6)

where the orthogonality of the transformation matrix \(A_i^j A^i = I\) is used in order to simplify the sub-matrix in the lower right-hand corner. Further, \(A_\theta^i A^i = \tilde{I}\) with \(\tilde{I} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}\).

The mass matrix (F.6) can also be written as

\[
M^{ij} = \left( \begin{array}{c}
m_{RR}^{ij} \\
m_{R\theta}^{ij} \\
m_{Rf}^{ij} \\
m_{\theta\theta}^{ij} \\
m_{\theta f}^{ij} \\
m_{ff}^{ij} \\
s_{ij} \end{array} \right)^{ij},
\]

where

\[
m_{RR}^{ij} = \int_{V_{ij}} \rho^{ij} I dV^{ij}, \quad m_{R\theta}^{ij} = \int_{V_{ij}} \rho^{ij} A_i^j u_p^{ij} dV^{ij},
\]

\[
m_{Rf}^{ij} = \int_{V_{ij}} \rho^{ij} A_i^j S^{ij} dV^{ij}, \quad m_{\theta\theta}^{ij} = \int_{V_{ij}} \rho^{ij} u_p^{ijT} u_p^{ij} dV^{ij},
\]

\[
m_{\theta f}^{ij} = \int_{V_{ij}} \rho^{ij} u_p^{ijT} \tilde{I} S^{ij} dV^{ij}, \quad m_{ff}^{ij} = \int_{V_{ij}} \rho^{ij} S^{ijT} S^{ij} dV^{ij}.
\]

(F.7)

Note that the two sub-matrices \(m_{RR}^{ij}\) and \(m_{ff}^{ij}\), which are associated, respectively, with the translational reference and elastic coordinates, are constant. Other matrices, however, depend on the system generalized coordinates.

The mass matrix in the case of a rigid body motion can be written as

\[
M_i^{rigid} = \left( \begin{array}{c}
m_{RR} \\
\text{symm.} \\
m_{R\theta} \\
m_{\theta\theta} \\
m_{\theta f} \\
m_{ff} \\
s_{ij} \end{array} \right)^i.
\]

In the case of structural systems, the reference coordinates remain constant with respect to time and the mass matrix of the body in this case is the constant matrix \(m_{ff}^{ij}\). When a deformable body undergoes rigid body motion, the mass matrix is defined by (F.7) and the sub-matrices \(m_{Rf}^{ij}\) and \(m_{\theta f}^{ij}\) represent the coupling between the reference motion and the elastic deformation.

In the following, we detail each sub-matrix to find a simplified version for more efficient calculation. The matrix \(m_{RR}^{ij}\) can be defined as

\[
m_{RR}^{ij} = \int_{V_{ij}} \rho^{ij} I dV^{ij} = \begin{pmatrix} m^{ij} & 0 \\ 0 & m^{ij} \end{pmatrix} = I_0^{ij},
\]

where \(I\) is the identity matrix and \(m^{ij}\) is the mass of the element \(j\) of the deformable
body $i$. We write the sub-matrix $m_{R\theta}^{ij}$ as

$$m_{R\theta}^{ij} = \int_{V^{ij}} \rho^{ij} A_0^i u_0^{ij} dV^{ij} = A_0^i \int_{V^{ij}} \rho^{ij} \left[ u_0^{ij} + u_f^{ij} \right] dV^{ij} = A_0^i \left[ I_1^{ij} + \bar{S}^{ij} q_f^{ij} \right],$$

where the matrices $I_1^{ij}$ and $\bar{S}^{ij}$ are defined as

$$I_1^{ij} = \int_{V^{ij}} \rho^{ij} u_0^{ij} dV^{ij}, \quad \bar{S}^{ij} = \int_{V^{ij}} \rho^{ij} S^{ij} dV^{ij}. \quad (F.8)$$

The vector $I_1^{ij}$ is the moment of mass of the body about the axes of the body reference in the undeformed state. It may vanish if the origin of the body reference is initially attached to the body center of mass. The vector $\bar{S}^{ij} q_f^{ij}$ represents the change in the moment of mass due to the deformation. Using (F.7), we verify that

$$m_{R\theta}^{ij} = A_i^{j} \bar{S}^{ij}.$$

The expression for $m_{\theta\theta}^{ij}$ is

$$m_{\theta\theta}^{ij} = \int_{V^{ij}} \rho^{ij} \left[ u_0^{ij} + u_f^{ij} \right]^T \left[ u_0^{ij} + u_f^{ij} \right] dV^{ij} = \int_{V^{ij}} \rho^{ij} \left[ u_0^{ijT} u_0^{ij} + 2 u_0^{ijT} u_f^{ij} + u_f^{ijT} u_f^{ij} \right] dV^{ij} = (m_{\theta \theta}^{ij})_{rr} + (m_{\theta \theta}^{ij})_{rf} + (m_{\theta \theta}^{ij})_{ff},$$

in which the sub-matrix $m_{\theta \theta}^{ij}$ reduces to a scalar that can be written as the sum of three components. The first component, $(m_{\theta \theta}^{ij})_{rr}$, is the mass moment of inertia in the undeformed state:

$$(m_{\theta \theta}^{ij})_{rr} = \int_{V^{ij}} \rho^{ij} u_0^{ijT} u_0^{ij} dV^{ij} = \int_{V^{ij}} \rho^{ij} \left[ (x^{ij})^2 + (y^{ij})^2 \right] dV^{ij} = I_2^{ij}.$$ Clearly, this integral has a constant value and does not depend on the body deformation. The last two scalar components, $(m_{\theta \theta}^{ij})_{rf}$ and $(m_{\theta \theta}^{ij})_{ff}$, represent the change in the mass moment of inertia of the body due to deformation. These two components are evaluated according to

$$(m_{\theta \theta}^{ij})_{rf} = 2 \int_{V^{ij}} \rho^{ij} u_0^{ijT} u_f^{ij} dV^{ij} = 2 \int_{V^{ij}} \rho^{ij} u_0^{ijT} S^{ij} dV^{ij} = q_f^{ijT} \left[ \int_{V^{ij}} \rho^{ij} S^{ijT} S^{ij} dV^{ij} \right] q_f^{ij},$$

$$(m_{\theta \theta}^{ij})_{ff} = \int_{V^{ij}} \rho^{ij} u_f^{ijT} u_f^{ij} dV^{ij} = q_f^{ijT} \left[ \int_{V^{ij}} \rho^{ij} S^{ijT} S^{ij} dV^{ij} \right] q_f^{ij}.$$
If we use definition (F.7), we can write the following:

\[
(m_{\theta\theta})_{ff} = q_f^{ijT} m_{ij}^f q_f^{ij},
\]

where

\[
m_{ij}^f = \int_{\Omega} \rho^{ij} S^{ijT} S^{ij} dV^{ij} = S_{ff}^{ij}.
\] (F.9)

Finally, we introduce

\[
m_{\theta f} = \int_{\Omega} \rho^{ij} \left[ u_0^{ij} + u^{ij}_f \right]^T \tilde{I} S^{ij} dV^{ij} = I_4^{ij} + q_f^{ijT} \tilde{S}^{ij},
\]

where the constant skew symmetric matrix \( \tilde{S}^{ij} \) is defined as

\[
I_4^{ij} = \int_{\Omega} \rho^{ij} u_0^{ijT} \tilde{I} S^{ij} dV^{ij}, \quad \tilde{S}^{ij} = \int_{\Omega} \rho^{ij} S^{ijT} \tilde{I} S^{ij} dV^{ij}.
\] (F.10)

We conclude that, to completely describe the inertia properties of the deformable body in plane motion, a set of inertia shape integrals is required. These integrals, which depend on the assumed displacement field, can be obtained using the GAUSSian quadrature method. As we are dealing with polynomials describing the displacement field, in order to find the optimum number of GAUSSian points, we have to know the highest degree of the polynomials which appear in the calculation of the mass matrix. We expect polynomials with degree 6 at most, which need 4 GAUSSian points to yield an exact integration. Figure F.3 shows GAUSSian points which are necessary to evaluate exact values on a sample element. To evaluate the integrals for \( \xi \in [0,1] \), we have to map GAUSSian points and weights.

![Gaussian point distribution](image-url)

**Figure F.3.:** GAUSSian point distribution for exact integration of polynomial representations in mass matrix evaluations [155, Figure 36].
Once we have calculated the mass matrix on element level, we have to assemble the total mass matrix considering mutual degrees of freedom. One should note that entities in the mass matrix of different elements share the same reference coordinates, but perhaps different elastic coordinates. Figure F.4 shows the regions in which we have the overlap (purple color) between element $j$ (light red) and element $j+1$ (light blue) in the assembly process. As the slider and the rod are attached to each other at one revolute joint, the end point of the flexible rod and the mass center of the slider have the same translational velocity. Therefore, we add the effect of the rigid slider to the mass matrix of the last element of the flexible rod, also by adding one additional degree of freedom for $\theta_3$ (Fig. F.1):

\begin{equation}
T^3 = \frac{1}{2} m_3 \dot{q}_{il} \left [ L^{ilT} L^{il} \right ] \dot{q}_{il} + \frac{1}{2} J_3 \dot{\theta}_3^2 , \tag{F.11}
\end{equation}

where index 3 is related to the slider (third body), and $l$ is related to the last element of the mesh which is connected to the slider by means of a revolute joint. We assemble the first term of (F.11) in the total mass matrix regarding mutual degrees of freedom and we add one additional row and column for the new degree of freedom $\theta_3$ which contain zero everywhere except one diagonal term which is $J_3$.

To add the effect of the rigid crank mass into the total mass matrix, first we define the constraint for the connecting joint between crank and flexible rod:

\begin{equation}
C^1 = R^i - \begin{pmatrix} l_1 \cos \theta_1 \\ l_1 \sin \theta_1 \end{pmatrix} = 0 , \tag{F.12}
\end{equation}
where \( \mathbf{R}^i = (x^i, y^i)^T \) is the translational coordinate of the rod reference frame. Equation (F.12) shows that the rod reference coordinate, and therefore its derivative with respect to time can be expressed in terms of \( \theta_1 \). For this reason and in order to rewrite the total mass matrix in terms of \( \theta_1 \), we need the time derivative of the constraint \( C^1 \):

\[
C^1 = \dot{\mathbf{R}}^i - \left(-l_1 \sin \theta_1 \right) \dot{\theta}_1 = \dot{\mathbf{R}}^i - C^1_{\theta_1} \dot{\theta}_1 = 0.
\]  

(F.13)

Using (F.13) and keeping in mind the kinetic energy formulation (F.5), we modify the total mass matrix such that it only depends on the coordinate \( \theta_1 \) according to the constraint \( C^1 \):

\[
\bar{m}_{RR} = C^1_{\theta_1} \bar{m}_{RR} C^1_{\theta_1} + I_1,
\]

\[
\bar{m}_{R\theta} = C^1_{\theta_1} \bar{m}_{R\theta}, \quad \bar{m}_{\theta R} = \bar{m}_{R\theta}^T,
\]

\[
\bar{m}_{Rf} = C^1_{\theta_1} \bar{m}_{Rf}, \quad \bar{m}_{fR} = \bar{m}_{Rf}^T,
\]

(F.14)

where \( I_1 = J_1 + \frac{1}{2} m_1 l_1^2 \) is the rigid crank mass moment of inertia with respect to the joint. We consider the contribution of the rigid crank in the total mass matrix according to the kinetic energy of the crank in terms of its only degree of freedom \( \theta_1 \):

\[
T^1 = \frac{1}{2} I_1 \dot{\theta}_1^2,
\]

(F.15)

where index 1 denotes the body number of the crank.

Figure F.5 shows the final mass matrix configuration after considering all the effects in the slider-crank mechanism.

**F.4. Quadratic Velocities**

For the quadratic velocity vector, we need to calculate the derivative of the mass matrix with respect to time and the derivative of the kinetic energy with respect to the degrees of freedom. For the time-derivative of the mass matrix, we write

\[
\dot{\mathbf{M}}^{ij} = \int_{V^{ij}} \rho^{ij} \left( \dot{\mathbf{L}}^{ijT} \dot{\mathbf{L}}^{ij} + \dot{\mathbf{L}}^{ijT} \dot{\mathbf{L}}^{ij} \right) \, dV^{ij} = \begin{pmatrix} \dot{\bar{m}}_{RR} & \dot{\bar{m}}_{R\theta} & \dot{\bar{m}}_{Rf} \\ \dot{\bar{m}}_{\theta R} & \dot{\bar{m}}_{\theta f} & \dot{\bar{m}}_{fR} \end{pmatrix}^{ij} \text{ (symm.)}
\]

(F.16)

where \( \dot{\mathbf{L}}^{ij} \) and \( \dot{\mathbf{L}}^{ij} \) are the modified versions of \( \mathbf{L}^{ij} \) and \( \mathbf{L}^{ij} \) according to the hinge constraint defined in (F.12):

\[
\dot{\mathbf{L}}^{ij} = \begin{pmatrix} C^1_{\theta_1} & A^i_j u^j_0 & A^i_j S^{ij}_j \end{pmatrix},
\]

\[
\dot{\mathbf{L}}^{ij} = \begin{pmatrix} C^1_{\theta_1 \theta_1} & A^i_j S^{ij}_j \dot{q}^j_0 - A^i_j \left( u^j_0 + S^{ij}_j q^j_0 \right) \dot{\theta}_2 \end{pmatrix} A^i_j S^{ij}_j \dot{\theta}_2 \).
Thereby, $C_{\theta_1, \theta_1}^1$ is the second derivative of the joint constraint:

$$C_{\theta_1}^1 = \begin{pmatrix} l_1 \cos \theta_1 \\ l_1 \sin \theta_1 \end{pmatrix} \dot{\theta}_1 = C_{\theta_3, \theta_1}^1 \dot{\theta}_1 .$$

After assembling the element contributions according to the pattern described in Fig. F.4, we add the effect of the rigid slider to the matrix of the last element of the flexible rod, noting that $\dot{J}_3$ is zero:

$$M^3 = m_3 \left( \dot{L}^T \dot{L} + \dot{L}^T \dot{L} \right) .$$

Note that the effect of the rigid crank is already taken into account by modifying the $L$ matrix.

To calculate the second term of the quadratic velocity, first we write the expression for the total kinetic energy as

$$T = \frac{1}{2} \ddot{\theta}_1 \ddot{\theta}_2 + \frac{1}{2} \dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_1 \dot{\theta}_2 \dot{q}_f + \dot{\theta}_2 \dot{q}_f + \frac{1}{2} \dot{q}_f^2 + \frac{1}{2} J_3 \dot{\theta}_3^2 .$$

Keeping in mind definition (F.14), the derivative of the kinetic energy with respect to the nodal coordinates is

$$\frac{\partial T}{\partial q} = \begin{pmatrix} C_{\theta_1}^{1T} \left( m_2 + m_3 \right) C_{\theta_1}^{1T} \dot{\theta}_1 + C_{\theta_1}^{1T} A^i \left( I_1 + S q_f \right) \dot{\theta}_2 + C_{\theta_1}^{1T} A^i S \dot{q}_f \\ -C_{\theta_1}^{1T} A \left( I_1 + S q_f \right) \dot{\theta}_1 \dot{\theta}_2 + C_{\theta_1}^{1T} A^i \dot{\theta}_2 \dot{q}_f \\ 0 \\ C_{\theta_1}^{1T} A^i S \dot{\theta}_1 \dot{\theta}_2 + (I_3 + S_{ff}) q_f \dot{\theta}_2^2 + S \dot{\theta}_2 \dot{q}_f \end{pmatrix}.$$
where $\mathbf{S}$, $\mathbf{S}_{ff}$, $\mathbf{I}_1$, and $\mathbf{I}_3$ are the assembled versions of $\mathbf{S}^{ij}$, $\mathbf{S}_{ff}^{ij}$, $\mathbf{S}^{ij}$, $\mathbf{I}_1^{ij}$ and $\mathbf{I}_3^{ij}$ defined in F.3. Finally, we have

\[
Q_v = - \left( \begin{array}{ccc}
\dot{m}_{RR} & \dot{m}_{R\theta} & 0 \\
\dot{m}_{R\theta} & \dot{m}_{\theta\theta} & 0 \\
0 & 0 & \dot{m}_{ff}
\end{array} \right) \left( \begin{array}{c}
\dot{\theta}_1 \\
\dot{\theta}_2 \\
\dot{\theta}_3 \\
q_f
\end{array} \right) + \frac{\partial T}{\partial \dot{q}}.
\]

**F.5. Stiffness Matrix**

Considering a linear isotropic material, the virtual work due to the elastic forces for element $j$ of body $i$ can be written as

\[
\delta W_{s}^{ij} = - \int_{V^{ij}} \mathbf{\sigma}^{ijT} \delta \mathbf{\varepsilon}^{ij} dV^{ij},
\]

where $\mathbf{\sigma}^{ij}$ and $\mathbf{\varepsilon}^{ij}$ are, respectively, the stress and strain tensors. Since the rigid body motion corresponds to the case of constant strains and since we defined the deformation of flexible bodies with respect to the body reference, the strain displacement relations can be written in the following form:

\[
\mathbf{\varepsilon}^{ij} = D^{ij} \mathbf{u}^{ij}_f,
\]

where $D^{ij}$ is a differential operator. We write

\[
\mathbf{\varepsilon}^{ij} = D^{ij} \mathbf{S}^{ij} \mathbf{q}^{ij}_f.
\]

For a linear isotropic material, the constitutive equations can be written as

\[
\mathbf{\sigma}^{ij} = C^{ij} \mathbf{\varepsilon}^{ij},
\]

where $C^{ij}$ is the symmetric matrix of elastic coefficients. We conclude

\[
\mathbf{\sigma}^{ij} = C^{ij} D^{ij} \mathbf{S}^{ij} \mathbf{q}^{ij}_f,
\]

where the stress tensor is written in terms of the elastic generalized coordinates of body $i$. Finally, we have

\[
\delta W_{s}^{ij} = - \mathbf{q}^{ij}_f^T \left[ \int_{V^{ij}} \left( D^{ij} \mathbf{S}^{ij} \right)^T C^{ij} D^{ij} \mathbf{S}^{ij} dV^{ij} \right] \delta \mathbf{q}^{ij}_f = - \mathbf{q}^{ij}_f^T K_{ff}^{ij} \delta \mathbf{q}^{ij}_f, \quad (F.19)
\]

where

\[
K_{ff}^{ij} = \int_{V^{ij}} \left( D^{ij} \mathbf{S}^{ij} \right)^T C^{ij} D^{ij} \mathbf{S}^{ij} dV^{ij}. \quad (F.20)
\]
Neglecting the shear deformation and using the assumptions of Euler-Bernoulli beam theory, the strain energy for the element \( j \) of the elastic rod can be written as

\[
U_{ij}^{ij} = \frac{1}{2} \int_0^{l_{ij}} \begin{pmatrix} u_{ij}^f & u_{ij}^t \end{pmatrix} \begin{pmatrix} E_{ij} A_{ij} & 0 \\
0 & E_{ij} I_{ij} \end{pmatrix} \begin{pmatrix} c u_{ij}^f \\
u_{ij}^t \end{pmatrix} \, dx ,
\]  

where \( l_{ij} \) is the length of element \( j \) of the beam, \( E_{ij} \) is the modulus of elasticity of the beam, \( A_{ij} \) is the cross-sectional area, \( I_{ij} \) is the second moment of area, \( u_{ij}^f \) and \( u_{ij}^t \) are the axial and transverse displacements of element \( j \) respectively and \((')\) denotes differentiation with respect to the spatial coordinate. Taking the derivative of \((F.21)\) with respect to \( q_f \) and comparing with \((F.19)\), we write

\[
C_{ij}^{ij} = \begin{pmatrix} E_{ij} A_{ij} & 0 \\
0 & E_{ij} I_{ij} \end{pmatrix}.
\]

Using the shape functions introduced in \((F.1)\), we conclude:

\[
D_{ij}^{ij} S_{ij}^{ij} = \begin{pmatrix} 1 & 0 \\
0 & 1/l_{ij} \end{pmatrix} \begin{pmatrix} -1 & 0 & 0 & 1 & 0 & 0 \\
0 & -6 + 12\xi & 0 & 6 - 12\xi & 0 \\
-l_{ij}^2 & 6\xi & -12\xi & 6\xi - 2 \end{pmatrix},
\]

where the first matrix is the inverse of the Jacobian for transforming to local coordinates with \( \xi = x/l \), and the second matrix is the derivative of \( S_{ij}^{ij} \) with respect to \( \xi \). We calculate the stiffness matrix for each element using \((F.20)\). It is worth to mention that as we are dealing with derivatives of the shape functions, the highest degree of polynomials which appears in the integral of \((F.20)\) is 2. Therefore, we only need 2 Gaussian points for the computation of the stiffness matrix on element level.

**F.6. External Forces**

The virtual work of all external forces \( F_i \) acting on element \( j \) of body \( i \) in the multibody system can be written as

\[
\delta W_{eij}^{ij} = \int_{V_{ij}} F_{ij}^T \delta r_{ij} \, dV_{ij},
\]

where

\[
\delta r_i = \begin{pmatrix} I & A_{ij} u_{ij}^f & A_{ij} S_{ij} \end{pmatrix} \begin{pmatrix} \delta R_i^j \\
\delta \theta_i^j \\
\delta q_i^j \end{pmatrix} = L_{ij}^{ij} \delta q_{ij}^{ij}.
\]
Combining the above two equations, we get the following expression for the virtual work of external forces:

$$\delta W_{\epsilon}^{ij} = \left( Q_R^{ijT} \quad Q_{\theta}^{ijT} \quad Q_f^{ijT} \right) \begin{pmatrix} \delta R_i \\ \delta \theta_i \\ \delta q_f^{ij} \end{pmatrix} = Q_e^{ijT} \delta q_f^{ij} ,$$

where

$$Q_R^{ijT} = \int_{V_{ij}} F^{ijT} dV_{ij} ,$$
$$Q_{\theta}^{ijT} = \int_{V_{ij}} F^{ijT} A_{\theta} u_{ij}^{\theta} dV_{ij} ,$$
$$Q_f^{ijT} = \int_{V_{ij}} F^{ijT} A^{f} S^{ij} dV_{ij} .$$

With the same procedure as for the mass matrix, we can assemble the external force vector. To add the virtual work of the rigid slider, we use the revolute joint constraint between slider and rod

$$Q_e^{ijT} = F^{T} L_{ii} ,$$

where index 3 is related to the slider (third body), and index l is related to the last element of the mesh which is connected to the slider by means of a revolute joint. We assemble (F.23) in the total external force vector regarding mutual degrees of freedom and adding one additional row and column for the degree of freedom $\theta_3$ which is zero as there is no moment in this direction. To add the effect of the rigid crank mass into the total external force vector, we use the constraint for the connecting joint between crank and flexible rod (F.12). We modify the sub-vector introduced in (F.22), such that it only depends on coordinate $\theta_1$ according to the constraint $C_1$

$$\bar{Q}_R^{ijT} = Q_R^{ijT} C_{\theta_1} - mg \frac{l}{2} \sin \theta_1 ,$$

where the second term comes from the work of the gravitational force with the rigid crank. It is worth to mention that according to the polynomial order in the external force integral, 2 GAUSSian points are sufficient for each element.

**F.7. Unilateral Constraints**

Considering the geometry of the slider according to Fig. 4.2 and keeping in mind the revolute joint between slider and end point of the rod, we write the position of the
slider center according to (F.3) and (F.12) as
\[
\mathbf{r}_{cg} = \begin{pmatrix} r^x_{cg} \\ r^y_{cg} \end{pmatrix} = \mathbf{R}^i + \mathbf{A}^i \mathbf{u}_{cg} = \begin{pmatrix} l_1 \cos \theta_1 \\ l_1 \sin \theta_1 \end{pmatrix} + \mathbf{A}^i (\theta_2) \begin{pmatrix} l_2 \\ 0 \end{pmatrix} + \mathbf{S}^i \mathbf{q}_f^i, \tag{F.24}
\]
where index \( l \) is related to the last element of the mesh of the flexible rod with initial length \( l_2 \). Therefore, the nonlinear normal and tangential gap functions as well as the matrices of generalized force directions can be calculated similar to Chapter 4.

### F.8. Boundary Conditions

![Figure F.6: Tangential and pinned reference system [155, Figure 42].](image)

When solving a finite element problem with the floating frame of reference formulation, the system becomes singular. This is because rigid body motion is added to the equations of motion at the same time as the deformation field also contains rigid body motion. A modal analysis would show that the first three eigenvalues are equal to zero which correspond to the degrees of freedom of the rigid body in planar motion. In this model, boundary conditions for the body reference system are introduced to avoid the singularity in the system of equations. There are different ways to define the boundary conditions. Shabana [173] shows that two sets of modes associated with two sets of boundary conditions can be used to obtain the same solution if the coordinate system is properly selected. Therefore, the physical deformation is unique in the inertial frame. For a beam, it is most common to use a clamped-free or a simply supported reference system (Fig. F.6), where three conditions are given in both cases. Clamped-free, that is, tangential, means that the reference system is tangential to the beam deflection at the root of the beam, that is, both displacements and rotations are equal to zero:
\[
q_{f1}^{i1} = q_{f2}^{i1} = q_{f3}^{i1} = 0.
\]

For a simply supported, pinned, reference system, the root of the beam is locked and the end of the beam is moving but only along the local \( x \) direction:
\[
q_{f1}^{i1} = q_{f2}^{i1} = q_{f5}^{i1} = 0,
\]
where index \( l \) stands for the last beam element. In the present simulation, the tangential reference system is chosen. We compare different boundary conditions using modal coordinates in Section 12.5 as introduced in Chapter G.
G. Modal Analysis

Modal analysis is a process where the nodal displacement vector is approximated by a linear combination of dominant eigenvectors (also called mode shapes) as it is shown in Fig. G.1. Elimination of high-frequency mode shapes decreases the number of numerical operations per time step because the size of the matrices in the equations of motion is much less than in the non-reduced case.

\[
\begin{align*}
\text{Figure G.1.: Superposition of the modes in modal analysis [155, Figure 43].}
\end{align*}
\]

For the flexible beam with mass matrix \( m_{ff} \) and stiffness matrix \( K_{ff} \), we obtain the angular frequency \( \omega_k \) and the relative mode shape \( \varphi_k \) using the free vibration equations of motion. The free vibration equations of motion can be derived from (1.1) in case of linearity, no external forces and no damping:

\[
\left( K_{ff} - \omega_k^2 m_{ff} \right) \varphi_k = 0 . \quad (G.1)
\]

The deformations are expressed as

\[
q_f = \sum_{k=1}^{n_m} \Phi_{jk} q_{mk} , \quad (G.2)
\]

where \( q_{fj} \) is the deformation of the degree of freedom number \( j \). Considering that the transformation matrix \( \Phi \) consists of the eigenvectors \( \varphi_k \), the values \( \Phi_{jk} \) and \( q_{mk} \) are components of the eigenvectors and modal coordinates (the new unknowns). The number of reduced coordinates \( n_m \) is chosen depending on the accuracy. We rewrite the above equation:

\[
q_f = \Phi q_m . \quad (G.3)
\]

The deformation field (F.2) is described with

\[
u_{ij}^f = S_{ij}^{ij} \Phi_{ij}^q m = S_{ij}^{ij} q_{m} . \quad (G.4)
\]

Using characteristics of normalized orthogonal eigenvectors (\( \phi_k \)), we decouple the
sub-matrices $m_{ff}$ and $K_{ff}$ in (F.20) and (F.9) respectively:

$$\Phi^T m_{ff} \Phi = I_{n_m \times n_m}, \quad \Phi^T K_{ff} \Phi = \omega_k^2 \delta_{kl} . \quad (G.5)$$

In order to complete the decoupling in the mass matrix, we deal with the sub-matrices $m_{R\theta}$, $m_{Rf}$ and $m_{\theta f}$ on element level using the modified shape functions introduced in (G.4). Moving the body coordinate system to the mass center, we show that the integrals in (F.8) and (F.10) vanish if

$$\int_V \phi_k (r) \, dV = 0 , \quad \int_V r \phi_k (r) \, dV = 0 . \quad (G.6)$$

It can be shown that the free-free modes satisfy the condition above and also the mean-axis conditions which is obtained by minimizing the kinetic energy of the elastic motion with respect to an observer sitting on the flexible body [6]. This ideal coordinate system and the free-free eigenfunctions are shown in Fig. G.2. Removing the first three modal coordinates (corresponding to the rigid body motion) results in decoupled versions of the total mass matrix $M$. It is clear from Fig. G.2 that the deformation at the two ends of the beam do not vanish as defined in this coordinate system. Shabana [173] modifies the free-free shape functions to satisfy the boundary condition which results in a simply supported, pinned, mode. However, it cannot be used to decouple the mass matrix anymore. Using articulated-free modes as shown in Fig. G.3 [56], that is, fixing the position of the left end and leaving the right end free, we can fix the boundary condition at the joint in addition to satisfying the second condition in (G.6), which results in vanishing terms regarding $m_{\theta f}$.
Figure G.3.: Articulated-free boundary condition for partially decoupling the reference and the elastic coordinates [155, Figure 45].
Bibliography


[17] **Bathe, Klaus-Jürgen; Noh, Gunwoo:** Insight into an implicit time integration scheme for structural dynamics. In: *Comput Struct* 98-99 (2012), pp. 1–6. – ISSN 0045–7949


[27] BRASEY, Valerie: HEM 5 user’s guide / Universite de Geneve, Suisse. 1994. – research report


[37] Cebulla, Thomas; Grundl, Kilian; Schindler, Thorsten; Ulbrich, Heinz; Velde, Arie van d.; Pijpers, Han: Spatial Dynamics of Pushbelt CVTs: Model Enhancements. In: *SAE Technical Paper of SAE World Congress, Detroit, 24th-26th April 2012*, 2012


[58] Esefeld, Bastian ; Schindler, Thorsten ; Ulbrich, Heinz: A coupling approach for the numerical integration of non-smooth multibody systems. In: *Proceedings of the 4th Canadian Conference on Nonlinear Solid Mechanics, McGill University Montreal, Canada, 23rd-26th July 2013*, 2013


[70] *fmacvec - fast matrix-vector library*. GNU Lesser General Public License
https://github.com/mbsim-env/


[88] Grundl, Kilian; Cebulla, Thomas; Schindler, Thorsten; Ulbrich, Heinz: The Maxwell-contact. In: PAMM 12 (2012), Nr. 1, pp. 73–74. – ISSN 1617–7061


[93] HDF5Serie - a HDF5 wrapper for time series. GNU Lesser General Public License https://github.com/mbsim-env/


[110] Kleinert, Jan; Simeon, Bernd; Obermayr, Martin: An inexact interior point method for the large-scale simulation of granular material. In: *Comput Meth Appl Mech Eng* 278 (2014), Nr. 0, pp. 567–598. – ISSN 0045–7825


[122] Lieberman, Daniel; Venkadesan, Madhusudhan; Werbel, William; Daoud, Adam; D’Andrea, Susan; Davis, Irene; Mang’Eni, Robert O.; Pitsiladis, Yannis: Foot strike patterns and collision forces in habitually barefoot versus shod runners. In: *Nature* 463 (2010), pp. 531–535. – ISSN 0028–0836


[128] MAZHAR, Hammad; HEYN, Toby; PAZOUKI, Arman; MELANZ, Daniel; SEIDL, Andrew; BARTHOLOMEW, Aaron; TASORA, Alessandro; NEGRUT, Dan: Chrono: a parallel multi-physics library for rigid-body, flexible-body, and fluid dynamics. In: Mechanical Sciences 4 (2013), Nr. 1, pp. 49–64. – ISSN 2191–9151


[138] OpenMBV - open multi body viewer. GNU General Public License
https://github.com/mbsim-env/

In: J Differ Equat 211 (2005), Nr. 2, pp. 247 – 281. – ISSN 0022–0396

[140] PAOLI, Laetitia ; SCHATZMAN, Michelle: A numerical scheme for impact
702–733. – ISSN 0036–1429

[141] PAOLI, Laetitia ; SCHATZMAN, Michelle: A numerical scheme for impact
pp. 734–768. – ISSN 0036–1429

[142] PESKHIN, Michael ; SANDERSON, Arthur: Minimization of energy in quasi-
(1989), pp. 53–60. – ISSN 1552–3098

[143] PFEIFFER, Friedrich: Lecture notes in applied and computational mechanics.
978–3–540–79435–6

[144] PFEIFFER, Friedrich ; BORCHSENIEUS, Fredrik ; FÖRG, Martin: New hy-
draulic system modelling. In: ASME Design Engineering Technical Conference,
Chicago, 2nd-6th September 2003, 2003

[145] PFEIFFER, Friedrich ; GLOCKER, Christoph: Multibody dynamics with uni-
ISBN 0–471–15565–9

[146] PFEIFFER, Friedrich ; SCHINDLER, Thorsten: Einführung in die Dynamik.

[147] PFEIFFER, Friedrich ; SCHINDLER, Thorsten: Introduction to Dynamics. Berlin


[149] PISANO, Alessandro ; USAI, Elio: Sliding mode control: A survey with
979. – ISSN 0378–4754

[150] POPP, Alexander ; WALL, Wolfgang: Dual mortar methods for computa-
tional contact mechanics - overview and recent developments. In: GAMM-
Mitteilungen 37 (2014), Nr. 1, pp. 66–84. – ISSN 1522–2608

[151] PRECLIK, Tobias: Models and algorithms for ultrascale simulations of non-
smooth granular dynamics, Friedrich-Alexander-Universität Erlangen-Nürnberg,
PhD thesis, 2014


[154] Rezaei, Shahed; Schindler, Thorsten: Mixed timestepping schemes for nonsmooth mechanics with high frequency damping. In: Multibody System Dynamics to be submitted. – ISSN 1384–5640


[160] Schindler, Thorsten: Multibody simulation / Technische Universität München. 2014. – lecture notes. 3rd edition, 1st revision; based on lecture notes of Martin Förg


[193] **TRINKLE, Jeffrey** ; **PANG, Jong-Shi** ; **SUDARSKY, Sandra** ; **LO, Grace**: On dynamic multi-rigid-body contact problems with Coulomb friction. In: **J Appl Math Mech** 4 (1997), pp. 267–279. – ISSN 0044–2267


[199] **ZANDER, Roland** ; **SCHINDLER, Thorsten** ; **FRIEDRICH, Markus** ; **HUBER, Robert** ; **FÖRG, Martin** ; **ULBRICH, Heinz**: Non-smooth dynamics in academia and industry: recent work at TU München. In: **Acta Mech** 195 (2008), pp. 167–183. – ISSN 0001–5970

### List of Figures

1.1. Modeling process in technical applications. ........................................ 4  
1.2. A multibody system is set-up from Bodys and Links [UML] [160, Figure 2.2]. 5  
1.3. A multibody system is an abstraction of a real world system [160, Figure 2.3]. 6  
1.4. Bouncing ball example [161, Figure 1, adapted]. .............................. 6  
1.5. Abstract bouncing ball example with finite elements. .......................... 8  
1.6. Pushbelt variator and pushbelt with elements [33, Figure 1.1]. ............... 11  
1.7. Model of a rotor test rig [167]. ....................................................... 13  
1.8. Rigid rotor in elastic lubricated journal bearing (left) [113, Figure 2] and  
squeezing machine with elastic lubricated bearing shell (right) [113, Figure 6]. 13  
1.9. Audi-V6 timing-drive [65, Figure 1]. ............................................... 14  
1.10. Robot for gastrointestinal surgery [148, Figure 5]. .............................. 15  
1.11. Grasping hand [194, Figure 28.1]. .................................................. 16  
1.12. Hair dynamics [23, Figure 2]. ....................................................... 17  
1.13. Shaking bodies [189, Figure 7]. ..................................................... 17  
1.14. Characteristics of an ideal diode. .................................................... 18  
1.15. Gene regulatory network [4, Figure 1]. ............................................ 19  
1.16. Constraints of the sledge [147, Figure 1.3]. .................................... 21  
1.17. Contact laws [154, Figure 3]. .................................................... 23  
1.18. Bilateral constraint. ................................................................. 24  
1.19. Impact effects in rigid multibody systems [160, Figure 6.10]. ............ 26  
1.20. Impact effects in multibody systems with elasticities [160, Figure 6.11]. 26  
1.21. Contact distance problem for possible contact points [160, Figure 6.1]. 27  
1.22. A solved contact distance problem [160, Figure 6.3]. ........................ 28  
1.23. A contact may be open, touching or penetrating [160, Figure 6.4]. ....... 28  
2.1. Constraint formulations and solution methods. .................................... 41  
2.2. Force laws for bilateral and unilateral contacts [165, Figure 1]. ............ 42  
2.3. Planar Coulomb friction [165, Figure 2]. ........................................ 47  
2.4. Configuration of a point mass in contact with a surface [165, Figure 5]. 47  
2.5. proj function for a convex set [72, Figure 3.6]. ................................ 49  
2.6. proj functions for contacts and planar friction with $r = 0.1$. ............... 52  
2.7. proj functions for contacts and planar friction with $r = 0.5$. ............... 52  
2.8. proj functions for contacts and planar friction with $r = 1.0$ [165, Figure 4]. 52  
2.9. Configuration of a point mass in contact with a surface [165, Figure 5]. 54  
2.10. Progression of iterates for finding the normal contact force [165, Figure 6]. 55  
2.11. Progression of iterates for finding the tangential contact force [165, Figure 7]. 55  
2.12. Painleve problem with $g > 0$ [165, Figure 8, adapted]. ...................... 56  
2.13. Globally convergent solution: $\Gamma, G > 0$. Contact maintained [165, Figure 9]. 57  
2.14. Globally convergent solution: $\Gamma < 0, G > 0$. Contact lost [165, Figure 10]. 57  
2.15. Two solutions: $\Gamma, G < 0$. Fixed-point iterations can diverge [165, Figure 12]. 58  
2.16. No solution: $\Gamma > 0, G < 0$. Acceleration-based model fails [165, Figure 11]. 59
3.1. Simulation platform *Salome*. ........................................... 61
3.2. Dynamical system classes in *SICONOS* (UML). .................... 63
3.3. Nonsmooth law classes in *SICONOS* (UML). ....................... 64
3.4. Relation classes in *SICONOS* (UML). ............................... 65
3.5. Simulation classes in *SICONOS* (UML). ............................... 65
3.6. Surrounding of *MBSim* [162, Figure 2, adapted]. ................. 66
3.7. Dynamic system classes in *MBSim* (UML) [162, Figure 4, adapted]. ..... 67
3.8. Object classes in *MBSim* (UML). [162, Figure 5, adapted] .......... 68
3.9. Link classes in *MBSim* (UML) [162, Figure 7, adapted]. .......... 68
4.1. Slider-crank mechanism with unilateral constraints and friction [167, Figure 1]. .................................................. 73
4.2. Definition of the gap functions for the slider-crank mechanism with unilateral constraints and friction. ................................. 74
5.1. Interpretation of velocity jumps [171, Figure 4, adapted]. ......... 78
5.2. Experimental convergence analysis: free flight [161, Figure 5, adapted]. ................................. 80
5.3. Experimental convergence analysis: rest phase [161, Figure 6, adapted]. .................................................. 81
5.4. Experimental convergence analysis: combined analysis [161, Figure 7, adapted]. .................................................. 82
5.5. Half-explicit timestepping scheme on acceleration level [167, Figure 3]. .... 87
5.6. Normal gap functions of the slider for half-explicit timestepping on acceleration level [167, Figure 4]. ................................. 88
5.7. Total energy of the slider-crank mechanism for half-explicit timestepping on acceleration level [167, Figure 5]. ................................. 88
6.1. Explicit timestepping scheme on velocity level [166, Figure 7]. ....... 93
6.2. Normal gap functions of the slider for explicit timestepping on velocity level [166, Figure 8]. .................................................. 94
7.1. Normal gap functions of the slider for half-explicit timestepping on acceleration (dashed) and velocity level (solid) [167, Figure 7]. ........ 96
7.2. Half-explicit timestepping scheme on velocity level [167, Figure 6]. .. 99
8.1. Movement of the center of gravity of the slider for different coefficients of restitution and half-explicit timestepping on velocity level [166, Figure 11]. ........ 101
8.2. Angles and angular velocities of the slider-crank mechanism for half-explicit timestepping on velocity level [166, Figure 12]. ........ 102
8.3. Lagrange multipliers of the slider-crank mechanism for half-explicit timestepping on velocity level [167, Figure 8]. .... 103
8.4. Bouncing ball in a box [167, Figure 9]. .................................. 104
8.5. Trajectory of the bouncing ball in a box for half-explicit timestepping on velocity level [166, Figure 15]. ................................. 105
8.6. Position and velocity of the bouncing ball in a box for half-explicit timestepping on velocity level [166, Figure 16]. ........ 106
8.7. Total energy of the bouncing ball in a box for half-explicit timestepping on velocity level [167, Figure 10]. ................................. 107
8.8. Normal gap functions of the bouncing ball in a box for half-explicit timestepping on acceleration (dashed) and velocity level (solid) [167, Figure 11]. ........ 107
9.1. Linear elastic impacting bar on a rigid obstacle [167, Figure 12]. .... 109
9.2. Elastic bar simulation with the half-explicit timestepping scheme on velocity level $\Delta t = 1 \cdot 10^{-7}$ s, $N = 50$ [167, Figure 13]. 110

9.3. Elastic bar simulation with the half-explicit timestepping scheme on velocity level $\Delta t = 1 \cdot 10^{-7}$ s, $N = 1000$ [167, Figure 14]. 111

9.4. Damped elastic bar simulation with the half-explicit timestepping scheme on velocity level $\Delta t = 1 \cdot 10^{-7}$ s, $N = 1000$ [167, Figure 15]. 112

9.5. Model of a rotor test rig [167, Figure 16]. 113

9.6. Curve of the journal frustum in the journal bearing [166, Figure 25]. 114

9.7. Half-explicit timestepping scheme on velocity level, variant [166, Figure 24]. 115

10.1. Flexible slider-crank mechanism [154, Figure 19]. 119

10.2. Flowchart of computational algorithm [154, Figure 4]. 121

10.3. Transition from impulsive to non-impulsive reactions [154, Figure 5]. 122

11.1. Generalized-$\alpha$ stability region in $\alpha_m$-$\alpha_f$ space, different test case $A(-1,1/2)$, $B(-1,0)$, $C(-1,-1/2)$, $D(1/2,1/2)$, $E(0,1/3)$ [154, Figure 6]. 124

11.2. Properties of the HHT-method for different $\alpha = \alpha_f$, $\alpha_m = 0$ [154, Figure 7]. 125

11.3. Properties of the generalized-$\alpha$ method for different $\rho_\infty$ [154, Figure 8]. 126

11.4. Smooth transition of cases $A$ and $C$ by modifying $\alpha_f$ and $\alpha_m$ [154, Figure 9]. 126

11.5. Generalized-$\alpha$ time integration scheme [154, Equation (44)]. 127

11.6. BATHE time integration scheme: first half step [154, Equation (64)]. 128

11.7. BATHE time integration scheme: second half step [154, Equation (65)]. 129

11.8. Comparison: Newmark method, generalized-$\alpha$ method and BATHE-method [154, Figure 10]. 130

11.9. Comparison: ED-$\alpha$ method with $\rho_\infty = 0$ and optimal $\alpha$ (11.24) and BATHE-method [154, Figure 11]. 130

11.10 Comparison: ED-$\alpha$ method with $\alpha_{AR} = 1/6$ and optimal $\alpha$ (11.24) and generalized-$\alpha$ method [154, Figure 12]. 131

11.11 ED-$\alpha$ method in time [154, Figure 13]. 132

11.12 ED-$\alpha$ time integration scheme [154, Equation (74)]. 132

11.13 Model problem of a three degree of freedom mass-spring system, (a): model with stiff spring, (b): model with bilateral constraint [154, Figure 14]. 133

11.14 Displacement, velocity and acceleration of masses 1 and 2 for various integration methods (case (a)) [154, Figure 15]. 133

11.15 Displacement of mass 3 for various integration methods (case (a)) [154, Figure 16]. 134

11.16 Displacement, velocity and acceleration for mass 1, 2 and 3 (case (b)) [154, Figure 17]. 135

11.17 Comparison of acceleration and velocity level for the calculation of contact forces (case (b)) [154, Figure 18]. 135

12.1. Mass center motion for different numbers of elements [154, Figure 20]. 137

12.2. Mass center motion - comparison to a rigid body simulation [154, Figure 21]. 138

12.3. Relative errors of position, velocity and contact force [154, Figure 22]. 138

12.4. Modified configuration of the slider-crank mechanism [154, Figure 23]. 138

12.5. Comparison between generalized-$\alpha$ method and BATHE-method for the angles $\theta_1$ and $\theta_2$ [154, Figure 24]. 139

12.6. Comparison between generalized-$\alpha$ method and BATHE-method for the angular velocities $\omega_1$ and $\omega_2$ [154, Figure 25]. 140
12.7. Comparison between generalized-α method and Bathe-method for the torque at the beam root [154, Figure 26]. .................................................. 140
12.8. Comparison between generalized-α method and Bathe-method for the active component of the normal contact force [154, Figure 27]. ......................... 141
12.9. Comparison between ED-α method with \( \rho_\infty = 0 \) and Bathe-method for the angular velocities [154, Figure 28]. .................................................. 141
12.10. Comparison between ED-α method with \( \rho_\infty = 0 \) and generalized-α method for the normal contact force [154, Figure 29]. .................................................. 142
12.11. Normal contact force convergence for modal approach [154, Figure 31]. .............. 143
12.12. Comparison between modal solutions for different boundary conditions [154, Figure 32]. .................................................. 144

13.1. Movement of the center of gravity of the slider for different coefficients of restitution and Moreau’s midpoint rule [171, Figure 3]. .................................................. 147
13.2. Gap functions and their time derivatives for \( \epsilon = 0.1 \) and Moreau’s midpoint rule [171, Figure 4]. .................................................. 148
14.1. Slider-crank mechanism with bilateral constraints [171, Figure 5]. ......................... 151
14.2. Constraint satisfaction of the slider-crank mechanism with bilateral constraints for different kind of kinematic levels [160, Figure 5.2]. ................................. 153

15.1. Energy of the full projection method for the slider-crank mechanism. ..................... 157
15.2. Flowchart of the unified timestepping scheme [171, Figure 7]. ............................. 159
15.3. Movement of the center of gravity of the slider for different coefficients of restitution and the unified timestepping method [171, Figure 8]. ..................... 160
15.4. Gap functions and their time derivatives for \( \epsilon = 0.1 \) and the unified timestepping method [171, Figure 9]. .................................................. 161
15.5. Total energy of the slider-crank mechanism for different simulation methods [171, Figure 10, adapted]. .................................................. 161

17.1. Framework for impulsive corrections. .................................................. 167

A.1. Characteristic mollifier [161, Figure 3]. .................................................. 174
A.2. Velocity jump interpretation for \( D^+ \) and \( D^- \) [161, Figure 4]. ......................... 177

E.1. Bathe-method - red: trapezoidal rule, green: Euler backward rule [155, Figure 33]. .......................................................................................... 189
F.1. Slider-crank mechanism with flexible rod and unilateral constraints, element \( j \) of body \( i \) [155, Figure 34]. .................................................. 191
F.2. Deformable body coordinate [155, Figure 35]. .................................................. 192
F.3. Gaussian point distribution for exact integration of polynomial representations in mass matrix evaluations [155, Figure 36]. .................................................. 197
F.4. Mass matrix results from assembling of element mass matrices \( j \) and \( j+1 \) - purple: summation of mutual DOFs, red: element mass matrix \( j \), blue: element mass matrix \( j+1 \) [155, Figure 37]. .................................................. 198
F.5. Total mass matrix for the slider-crank mechanism considering joint constraints [155, Figure 38]. .................................................. 200
F.6. Tangential and pinned reference system [155, Figure 42]. .................................................. 204

G.1. Superposition of the modes in modal analysis [155, Figure 43]. ................................. 205
G.2. Free-free boundary condition and new reference frame at the mass center for decoupling the reference and the elastic coordinates [155, Figure 44] . . 206
G.3. Articulated-free boundary condition for partially decoupling the reference and the elastic coordinates [155, Figure 45] . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ..
List of Tables

4.1. Characteristics of the slider-crank mechanism with unilateral constraints and friction [69]. 75

8.1. Bilateral slider-crank example: comparison of the error of the explicit Moreau-Jean and the half-explicit timestepping scheme on velocity level [167, Table 2]. 102
8.2. Unilateral slider-crank example: comparison of the error of the explicit Moreau-Jean and the half-explicit timestepping scheme on velocity level [167, Table 3]. 104
8.3. Characteristics of the bouncing ball example [167, Table 4]. 105

9.1. Characteristics of the elastic bar example [167, Table 5]. 109
9.2. Characteristics of the rotor example [167, Table 6]. 113

12.1. Modified characteristics of the slider-crank mechanism [154, Table 2]. 139
12.2. Relative CPU time for $\Delta t = 10^{-5}$ s (bilateral) [154, Table 3]. 140
12.3. Comparison of the error $\epsilon$ ($T = 0.05$ s, bilateral) [154, Table 3]. 142
12.4. Relative CPU time for $\Delta t = 10^{-5}$ s (unilateral) [154, Table 5]. 142
12.5. Comparison of error $\epsilon$ ($T = 0.05$ s, unilateral) [154, Table 6]. 143

B.1. Stage relationship for accelerations [161, Table 1]. 183