Structure-Preserving Numerical Solution of Multibody Systems in Energy Formulation

Master Thesis

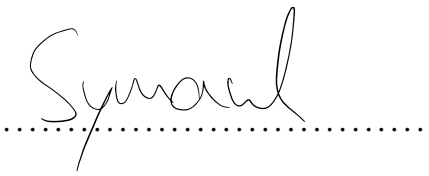
Aron Symank
Technische Universität Berlin
Department of Mathematics
January 2025

Supervisor: Prof. Dr. Volker Mehrmann

Eigenständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit eigenständig ohne Hilfe Dritter und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe. Alle Stellen, die den benutzten Quellen und Hilfsmitteln unverändert oder sinngemäß entnommen sind, habe ich als solche kenntlich gemacht. Es wurden keine generativen KI-Tools verwendet. Ich erkläre weiterhin, dass ich die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt habe.

Berlin, den 14.01.2025



Abstract

Mehrkörpersysteme (MKS) mit holonomischen Zwangsbedingungen bilden Systeme differentiell-algebraischer Gleichungen mit Differentiationsindex 3. Die Überführung in ein System gewöhnlicher Differentialgleichungen durch Indexreduktion oder Variablentransformation ist in der Praxis oft nicht möglich; in diesem Fall stellen die algebraischen Gleichungen eine große Herausforderung für die numerische Lösung dar, insbesondere wenn zeitgleich gute Energieerhaltung gefordert ist. Nach der Betrachtung von MKS in der klassischen Euler-Lagrange-Formulierung wenden wir uns der modernen Modellklasse port-Hamiltonischer Systeme zu und untersuchen deren Eigenschaften. Die Erweiterung des Modells um algebraische Gleichungen in zahlreichen Forschungsarbeiten der letzten Jahre ermöglicht es uns, MKS mit Zwangsbedingungen als port-Hamiltonisches System zu formulieren.

Anschließend betrachten wir numerische Verfahren für die Integration von MKS mit Zwangsbedingungen, mit einem Schwerpunkt auf partitionierten Runge-Kutta Methoden (PRK). Wir untersuchen das Lobatto IIIA-IIIB-Paar sowie eine Methode nach A. Murua (1996). Die Ergebnisse numerischer Simulationen mittels unserer Implementierung der genannten PRK-Methoden sowie zweier gewöhnlicher Runge-Kutta Methoden bilden den Abschluss der Arbeit. Wir untersuchen die Löser im Hinblick auf den Erhalt von Energie und Zwangsbedingungen, sowie für den Fall, dass eine algebraische Gleichung aus der Störung einer schlecht konditionierten Massenmatrix resultiert. Die guten Ergebnisse des weniger bekannten Lösers von Murua in Bezug auf alle drei genannten Aspekte legen nahe, der Methode neue Aufmerksamkeit zu schenken und ihr Potenzial für die Lösung differential-algebraischer Gleichungen weiter zu erforschen. Für unbeschränkte Systeme zeigte die Gauss-Legendre-Kollokation die erwarteten insgesamt sehr guten Ergebnisse, und insbesondere vielversprechende Resultate für den numerisch herausfordernden Fall einer schlecht konditionierten Massenmatrix.

Abstract

In the classic Euler-Lagrange derivation, multibody systems (MBS) with holonomic constraints form differential-algebraic system of equations of index 3. When the transformation to a system of ordinary differential equations through index reduction or variable transformations is not feasible, the presence of algebraic equations poses a great challenge for numerical integration, in particular when one is interested in accurate energy conservation. We look into structure and properties of MBS and then turn our focus to the powerful model class of port-Hamiltonian systems (pH). Within this highly active field of research, algebraic equations have recently been included into the framework, allowing us to formulate constrained MBS as a pH system. We then investigate numerical methods for the solution of MBS, with a focus on partitioned Runge-Kutta methods (PRK). We look into two such methods specifically designed for the numerical integration of constrained MBS, namely the Lobatto IIIA-IIIB pair and a method proposed by A. Murua in 1996, focusing on the conservation of both energy and the algebraic constraints.

We present numerical simulation results of our implementation for the mentioned PRK methods as well as two (non-partitioned) Runge-Kutta methods. Finally, we also investigate solver behaviour for the limiting case that an algebraic equation arises from a perturbation to an ill-conditioned mass matrix. The good performance of the lesser known solver by Murua with regard to all three points suggests to revisit this method and further explore its potential. For the unpartitioned methods, Gauss-Legendre collocation showed the expected overall good solution quality, and in particular promising results for the challenging case of a highly ill-conditioned mass matrix.

Contents

1 Introduction

The modelling and numerical integration of physical systems in our modern understanding have been continuously developing fields of research ever since the fundamental work of Newton in the late 17th century. Novel approaches by mathematicians such as Euler, Lagrange and Hamilton brought about further insights and helped to advance theory and solution techniques along with the overall evolution of mathematics and natural sciences. On these theoretical foundations, the technological progress of the 20th century has enabled us to carry out numerical computations in previously unimagined speed, opening new possibilities and sparking research in the growing fields of numerical mathematics and computer science.

In the realm of mechanics, the multibody system framework is the most common approach to model and simulate systems of interconnected bodies which are constrained in their motion by links, joints, or other conditions such as contact points, see [1, 11]. These connections and constraints introduce algebraic equations into the well-studied systems of ordinary differential equations (ODEs). The resulting systems of differential-algebraic equations (DAEs) pose challenges in theoretical analysis and numerical treatment, see [12]. While a DAE can always be transformed into an ODE in theory, this is not always feasible in practice, in particular for complex problems. Since the early 1970s ([13]), the great practical value of DAE systems has received widespread recognition, and their modelling and the subsequent development of specially designed numerical methods is now a highly active field of research, see [21]. DAEs may also arise as perturbations of ill-conditioned ODEs, e.g., a mechanical system with masses of greatly varying sizes. In that sense, ODEs and DAEs can be numerically close in spite of their structural difference, see [20].

A major difficulty in the numerical solution of constrained multibody systems is to maintain the accuracy of the constraints, especially those that are not explicitly formulated in the model. Another challenge concerns the conservation of energy: While the fundamental physical law of energy conservation is reflected in all theoretical approaches, this does not hold for numerical methods. In the presence of algebraic equations, reconciling constraint and energy conservation is particularly difficult and requires special numerical treatment, cf. [19, 17].

In this thesis, we first introduce constrained multibody systems on the basis of Lagrangian mechanics and analyse some of their properties, based on [3, 21, 11]. With our focus on energy based modelling, we then proceed to Hamiltonian mechanics ([17]) and look into the broad framework of port-Hamiltonian systems. This model class has proven to be very powerful due to intrinsic energy preservation, the option for control

as well as model interconnection across various physical domains with energy as the common language, for an overview see [32, 26]. In recent years, algebraic equations have been included into the framework, see [31, 25, 4, 24], allowing us to incorporate constrained multibody systems with an appropriate formulation.

We then move on to numerical methods. After a recapitulation of Runge-Kutta methods and a related time discretization of port-Hamiltonian systems via collocation methods as presented in [25], we turn our focus to numerical methods for the solution of constrained multibody systems. Since common numerical integrators fail to preserve energy and the algebraic constraints simultaneously, more elaborate methods are needed, designed to exploit the specific problem structure, see [17]. We take a closer look at two such methods based on the concept of partitioned Runge-Kutta methods, namely the Lobatto IIIA-IIIB pair as suggested by Jay in [19] as well as a method proposed by Murua in [28], based on Gauss-Legendre collocation and an adapted Lobatto method. We investigate their properties, again with a focus on energy and constraint preservation.

Finally, we present and discuss numerical simulation results from our implementation of the previously described numerical methods applied to multibody systems. We look into long time energy conservation, the accuracy of algebraic constraints, as well as the solution behaviour in the limiting case of a decreasing mass, turning a differential into an algebraic equation in the limit.

2 Multibody systems

2.1 Introduction

In the second half of the 18th century, Joseph-Louis Lagrange developed a novel approach to classical Newtonian mechanics. Instead of computing all forces acting on a system and then solving the equations derived from Newton's second law, Lagrangian mechanics start by computing the kinetic and potential energy of the system, as well as deriving positional (holonomic) constraints. For an overview on classical mechanics, see for example [15, 7]. For a given configuration space $\mathcal{X} \in \mathbb{R}^d$, the *Lagrangian*

$$\mathcal{L}(q, \dot{q}) = T(q, \dot{q}) - U(q)$$

is defined as the difference between kinetic energy $T(q, \dot{q})$ and potential energy $U(q)$, where $q \in \mathcal{X}$ is a point in configuration space and $\dot{q}$ its derivative with respect to time. Note that in the Lagrangian, however, q and $\dot{q}$ are treated as independent variables. If holonomic constraints $g(q) = 0$ are present, they may be included in the Lagrangian as

$$\mathcal{L}(q, \dot{q}, \lambda) = T(q, \dot{q}) - U(q) - g(q)^T \lambda,$$

or added later to the equations of motion. As has been observed and reformulated by many famous scientists such as Lagrange himself, the development of dynamical physical systems obey what is commonly called the principle of least action: It states that the evolution of a system, given by a path $q(t)$ in configuration space, is an extremum of the action functional $\mathcal{S}$, given as the integral of the Lagrangian over time along the given path, so

$$\mathcal{S}[q(t)] := \int_{t_1}^{t_2} \mathcal{L}(q(t), \dot{q}(t), \lambda(t)) dt,$$

where the time interval $[t_1, t_2]$ as well as the path's endpoints $q_1 = q(t_1),\ q_2 = q(t_2)$ are fixed. Analogous to elementary calculus, for a path to be an extremum, its 'derivative' needs to vanish, meaning that small perturbations δ along the path do not change the action functional. Formalizing this idea led Lagrange, together with his mentor and long-term correspondent Euler, to develop the calculus of variations. The application of the action principle to a mechanical system then yields the famous Euler-Lagrange equations of motion

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}} - \frac{\partial \mathcal{L}}{\partial q} = 0, \tag{2.1}$$

fully describing the dynamics.

In classical mechanics, we have the kinetic energy $T(q, \dot{q}) = \frac{1}{2}\dot{q}^T M(q)\dot{q}$ with $M(q)$ sym-

metric positive definite, the potential energy $U(q)$, and, if present, a set of holonomic constraints $g(q) = 0$. Denoting the gradient $\nabla_x \mathcal{L}$ with respect to a variable x by $\mathcal{L}_x$, we get

$$\begin{aligned}\big(\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}}(q,\dot{q},\lambda)\big)^T &= \frac{d}{dt}\mathcal{L}_{\dot{q}}(q,\dot{q},\lambda)\\ &= \frac{d}{dt}T_{\dot{q}}(q,\dot{q})\\ &= \frac{d}{dt}M(q)\dot{q}\\ &= \frac{\partial}{\partial q}\big(M(q)\dot{q}\big)\dot{q} + M(q)\ddot{q}\end{aligned}$$

and

$$\begin{aligned}\big(\frac{\partial \mathcal{L}}{\partial q}(q,\dot{q},\lambda)\big)^T &= \nabla_q \mathcal{L}(q,\dot{q},\lambda)\\ &= \nabla_q T(q,\dot{q}) - \nabla_q U(q) - Dg(q)^T\lambda,\end{aligned}$$

since

$$\nabla_q\big(g(q)^T\lambda\big) = \big(\frac{\partial}{\partial q}(g(q)^T\lambda)\big)^T = \big(\lambda^T Dg(q)\big)^T = Dg(q)^T\lambda.$$

The second order equations of motion thus read

$$M(q)\ddot{q} = \nabla_q T(q,\dot{q}) - \frac{\partial}{\partial q}\big(M(q)\dot{q}\big)\dot{q} - \nabla_q U(q) - Dg(q)^T\lambda. \tag{2.2}$$

Note that, with the constraints given as $g(q) = 0$, the columns of $Dg(q)^T$ are orthogonal to the constraint manifold $\mathcal{M} := \{q \in \mathbb{R}^d \mid g(q) = 0\}$, and therefore the inaccessible directions of motion at position q. These are scaled by Lagrange multipliers λ such that the overall resulting force is consistent with the constraints, i.e., it confines the resulting motion of q to $\mathcal{M}$.
Reducing the order of (2.2) by introducing generalized velocities $\dot{q} = Z(q)p$ via the pointwise nonsingular rotation matrix Z (see 2.2.3) and including the constraints, we finally get

$$\begin{aligned}\dot{q} &= Z(q)p\\ M(q)\dot{p} &= \underbrace{\nabla_q T(q,Z(q)p) - \frac{\partial}{\partial q}\big(M(q)Z(q)p\big)Z(q)p - \nabla_q U(q) + \varphi(q,p)}_{:=f(q,p)} - Z(q)^T Dg(q)^T\lambda\\ g(q) &= 0,\end{aligned} \tag{2.3}$$

where φ describes forces arising from dissipative elements or external ports which can be added to the equations of motion, see 4.4. An explanation for the appearance of the rotation matrix in front of the term $Dg(q)^T\lambda$ in the second set of equations is given in 2.2.2.

For the subsequent definition of multibody systems and the derivation of its index, we follow [3]. In line with the publication and as is more common in the context of multibody systems, we denote positions by p and (generalized) velocities by v.

Definition 2.1 (Multibody system)**.** A constrained autonomous *multibody system (MBS)* is given by

$$\dot{p} = Z(p)v \tag{2.4}$$

$$M(p)\dot{v} = f(p, v, s) - Z(p)^T G(p, s)^T \lambda \tag{2.5}$$

$$0 = c(p, s) \tag{2.6}$$

$$0 = g(p, s), \tag{2.7}$$

where $p, v \in \mathbb{R}^d$ are the position and velocity vectors, $M(p) \in \mathbb{R}^{d,d}$ is the pointwise positive definite mass matrix including the moments of inertia for rotating bodies (see 2.2.2), $Z(q) \in \mathbb{R}^{d,d}$ is the pointwise nonsingular rotation matrix (see 2.2.3), $f : \mathbb{R}^{2d+r} \to \mathbb{R}^d$ is the vector of applied forces, and G is the total derivative of g. In (2.6), contact point equations are defined by $c \in \mathcal{C}^1(\mathbb{R}^{d+r}, \mathbb{R}^r)$ with $\frac{\partial c}{\partial s}$ nonsingular, such that the contact points s can be uniquely determined as a function of p. In (2.7), the function $g \in \mathcal{C}^2(\mathbb{R}^d, \mathbb{R}^\nu)$ defines ν holonomic constraints, i.e., they depend only on the positional variable p. This is fulfilled since we can set $g(p, s) = \tilde{g}(p) := g(p, s(p))$. We assume that there are no redundant constraints, meaning that G has full row rank.

2.2 Properties

2.2.1 Differentiation index

Since the multibody system 2.1 includes algebraic equations, it is not a system of ordinary differential equations (ODE), but a system of differential-algebraic equations (DAE). On a theoretical level, one can always avoid the difficulties of DAEs by transforming them into an ODE, which used to be the standard way of dealing with DAEs (and often still is), see [21]. There are two ways to achieve this: Either, one uses new coordinates which parametrize the constraint manifold defined by the algebraic equations, resulting in an unconstrained system with fewer variables. As an example, consider a simple two-dimensional pendulum, constrained to move on the unit circle by the algebraic equation $x^2+y^2-1 = 0$. With an appropriate coordinate change, we can easily reduce this to a one-dimensional problem with the angular displacement φ as the only variable. For such a simple problem, this is clearly the preferable approach.

The other way is to apply time derivatives and algebraic transformations to (parts of) the DAE system until an ODE system is obtained, in this case with the same number of variables. The smallest natural number τ such that τ time derivatives, together with algebraic transformations, turn the DAE system into an ODE system, is called the differentiation index. Formally defining this notion is challenging, and there are various other index concepts for DAEs. For under- and overdetermined systems, the differentiation index is not useful at all, since it requires unique solvability, as mentioned in [21]. In this thesis, however, we only consider the differentiation index in the above sense and call it index from here on.

Despite of the theoretical possibility to always transform a DAE into an ODE, it is often preferable to work with the DAE directly. The main reason is that for more complex problems, both coordinate change and index reduction may be unfeasible. In the case of index reduction, we also face two other problems. The first one concerns the constraints: Depending on the chosen formulation, some constraints are given explicitly, while others form the so-called 'hidden' constraints. In the case of an MBS with holonomic constraints, the explicit constraints concern the position of the bodies. The first time derivative of the holonomic constraints thus constitutes constraints on the velocity level, and the second one those on acceleration level. Again considering the pendulum example from above with the explicit holonomic constraint $x^2 + y^2 - 1 = 0$, we have the hidden velocity constraint $x\dot{x} + y\dot{y} = 0$, ensuring that the velocity vector is tangential to the position of the mass, and the hidden acceleration constraint $x\ddot{x}+y\ddot{y}+\dot{x}^2+\dot{y}^2 = 0$, ensuring that the acceleration is such that the velocity vector along the solution remains on the (position-dependent) manifold defined by the velocity constraints.

Clearly, the unique analytical solution satisfies all of these constraints, but a numerical solver can only work with those that are explicitly given. Thus, even if the explicit constraints are exactly satisfied by a numerical solution, the hidden constraints are (generally) not. In case of the pendulum example, if only the positional constraints are explicitly given, the hidden velocity constraints are not satisfied. However, the error with respect to the velocity constraints does not tend to accumulate since the constraint manifold depends on the position, and we (can) thus only observe the error in every individual integration step. The same holds for the acceleration constraints. The positional constraint manifold, on the other hand, is constant in velocity and acceleration. If the holonomic constraints are hidden, we therefore see a 'numerical drift' of the solution, in this case off the unit circle. This effect can also be observed in the numerical experiments, see 7.3.2.

Theorem 2.2. *If the matrix $\Gamma := GZM^{-1}Z^TG^T$ is nonsingular in a neighbourhood of the solution, the index of the MBS 2.1 is 3.*

Proof. First we note that, since we assume $\frac{\partial c}{\partial s}$ to be nonsingular, we get the derivative

of g as

$$\begin{aligned}\frac{dg}{dp}(p,s(p)) &= \Big[\frac{\partial g}{\partial p} \;\; \frac{\partial g}{\partial s}\Big](p,s(p)) \begin{bmatrix} I_d \\ \frac{\partial s}{\partial p} \end{bmatrix}(p) \\ &= \Big[\frac{\partial g}{\partial p} + \frac{\partial g}{\partial s}\frac{\partial s}{\partial p}\Big](p,s(p)) \\ &= \Big[\frac{\partial g}{\partial p} + \frac{\partial g}{\partial s}\big(\frac{\partial s}{\partial c}\big)^{-1}\frac{\partial c}{\partial p}\Big](p,s),\end{aligned}$$

where we used the implicit function theorem to obtain

$$\frac{\partial s}{\partial p} = \big(\frac{\partial s}{\partial c}\big)^{-1}\frac{\partial c}{\partial p}.$$

Thus, we get

$$G(p,s) = \Big[\frac{\partial g}{\partial p} + \frac{\partial g}{\partial s}\big(\frac{\partial s}{\partial c}\big)^{-1}\frac{\partial c}{\partial p}\Big](p,s).$$

Differentiating the constraints (2.7) with respect to time then gives

$$\begin{aligned}0 &= \frac{dg}{dt} \\ &= \frac{\partial g}{\partial p}\,\dot{p} + \frac{\partial g}{\partial s}\,\dot{s} \\ &= \frac{\partial g}{\partial p}\,\dot{p} + \frac{\partial g}{\partial s}\big(\frac{\partial s}{\partial c}\big)^{-1}\frac{\partial c}{\partial p}\,\dot{p} \\ &= \big(\frac{\partial g}{\partial p} + \frac{\partial g}{\partial s}\big(\frac{\partial s}{\partial c}\big)^{-1}\frac{\partial c}{\partial p}\big)\,\dot{p} \\ &= G(p,s)Z(p)v.\end{aligned}$$

Another time derivative gives

$$\begin{aligned}0 &= \frac{d}{dt}(GZ)v + GZ\dot{v} \\ &= \frac{d}{dt}(GZ)v + GZM^{-1}(f - Z^TG^T\lambda),\end{aligned}$$

so

$$GZM^{-1}Z^TG^T\lambda = \frac{d}{dt}(GZ)v + GZM^{-1}f =: \tau(p,v,s).$$

With the nonsingularity of $GZM^{-1}Z^TG^T$, we can therefore multiply the equation with its inverse, and another (third) time derivative of the resulting equation finally gives

$$\dot{\lambda} = \frac{d}{dt}\big[(GZM^{-1}Z^TG^T)^{-1}\,\tau\big].$$

With (2.4), (2.5), we see that $\dot{\lambda} = \frac{d}{dt}[(GZM^{-1}Z^TG^T)^{-1}\ \tau] = \tilde{\tau}(p, v, s)$, showing that the system is of index 3. □

In light of the discussion above, we could directly try to numerically solve the MBS in the index-3 formulation, since this prevents a numerical drift of the solution. However, it has been mentioned in various works, see for example [9, 5, 6] that the solution of high-index problems poses great numerical difficulties, leading to the common approach of index reduction. Intuitively speaking, a DAE system of higher index is 'further away' from an ODE and therefore harder to solve numerically. We would thus like to reduce the index by differentiating the holonomic constraints with respect to time and work with the explicit constraints on velocity level. To avoid the resulting numerical drift, one might consider to simply include the constraints on both levels (or even on acceleration level), in the MBS formulation. This, however, leads to more equations with no additional variables, and the resulting overdetermined system is disadvantageous to solve. A remedy is thus to introduce more variables and find an equivalent reformulation of the system.
The most common approach to this idea has been proposed by Gear, Gupta and Leimkuhler in 1985 ([14]), who found an equivalent formulation with explicit index-2 and index-3 constraints, by introducing additional Lagrange multipliers to avoid an overdetermined system. We now summarize the definition of the new formulation and one central result from [14], assuming the nonsingularity of the matrix Γ from theorem 2.2 as well as full row rank of G.

Definition 2.3 (Gear-Gupta-Leimkuhler formulation)**.** Given the MBS

$$\dot{p} = Z(p)v \tag{2.8}$$
$$M(p)\dot{v} = f(p, v, s) - Z(p)^T G(p, s)^T \lambda \tag{2.9}$$
$$0 = c(p, s) \tag{2.10}$$
$$0 = g(p, s), \tag{2.11}$$

the Gear-Gupta-Leimkuhler formulation (GGL) is given by

$$\dot{p} = Z(p)v + G(p, s)^T \mu \tag{2.12}$$
$$M(p)\dot{v} = f(p, v, s) - Z(p)^T G(p, s)^T \lambda \tag{2.13}$$
$$0 = c(p, s) \tag{2.14}$$
$$0 = g(p, s), \tag{2.15}$$
$$0 = G(p, s)Z(p)v. \tag{2.16}$$

The time derivative (2.16) of the holnomic constraints (2.11) has been added to the

system, and (2.8) has been replaced by (2.12), with another Lagrange multiplier $\mu \in \mathbb{R}^{\nu}$.

Theorem 2.4. *Under the assumption that G has full rank, the GGL formulation of Definition 2.3 has the same solution as the original MBS, in the sense that any solution of the original MBS can be extended to a solution of the GGL formulation by adding the variable μ with $\mu \equiv 0$, and all solutions of the GGL formulation have $\mu \equiv 0$. Further, the GGL formulation is of index 2.*

Proof. Let $x := (p, v, s, \lambda)$ be a solution to the original MBS. Then, $(x, 0)$ is a solution to the GGL formulation, since (2.12) is identical to (2.8) for μ=0, and with (2.11), x also satisfies the time derivative (2.16).
On the other hand, let (x, μ) be a solution to the GGL formulation. Then, differentiating (2.15) once and using (2.16) gives

$$\begin{aligned} 0 &= G(p,s)\dot{p} \\ &= G(p,s)\big(Z(p)v + G(p,s)^T\mu\big) \\ &= G(p,s)Z(p)v + G(p,s)G(p,s)^T\mu \\ &= G(p,s)G(p,s)^T\mu. \end{aligned}$$

Since G has full row rank, GG^T is nonsingular and it follows that $\mu = 0$ along any solution.
For the index, we first note that the analogous derivation of an explicit equation $\dot{\lambda} = \bar{\tau}(p, v, s, \mu)$ from theorem 2.2 still holds, but the first time derivative of the holonomic constraints is already given in the GGL formulation. We can therefore obtain τ with only two time derivatives. For the new variable μ, we just saw above that one time derivative of (2.15) gives

$$0 = G(p,s)Z(p)v + G(p,s)G(p,s)^T\mu \iff G(p,s)G(p,s)^T\mu = -G(p,s)Z(p)v,$$

and the nonsingularity of GG^T gives, together with a second time derivative and equations (2.12),(2.13),

$$\dot{\mu} = \kappa(p, v, s). \square$$

2.2.2 Mass matrix

The mass matrix $M(p)$ is symmetric positive definite, and determines the kinetic energy of the system via $T(p, v) = \frac{1}{2}v^T M(p)v$. If there are rotating bodies, p, v contain angular orientations and velocities and M the corresponding moments of inertia. In a (spatially) $d-$dimensional constrained multibody system with z bodies where no coordinate change

has been applied, we get the constant mass matrix

$$M = \begin{bmatrix} m_1 I_d & 0 & 0 & \dots & 0 \\ 0 & Q_1 & 0 & \dots & 0 \\ 0 & 0 & m_2 I_d & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & Q_z \end{bmatrix} \in \mathbb{R}^{2dz,2dz} \tag{2.17}$$

where $Q_b \in \mathbb{R}^{d,d}$ describes the moments of inertia of body b with respect to the corresponding angles from the positional vector p_b. Depending on the chosen angular coordinates, Q_b is typically diagonal as well. Note that If M is constant, the kinetic energy only depends on the velocity variable v.

A non-constant mass matrix may arise from a coordinate change, see 2.2.3, or when algebraic constraints are analytically resolved. As an example, consider again the simple pendulum with point mass m and a massless rod of length ℓ, where the support x_o is now free to move horizontally on the $x-$ axis, and where the positional constraint $(x-x_0)^2+y^2-\ell^2=0$ has been resolved by only using one angular coordinate φ, describing the displacement from the negative $y-$axis in positive $x-$ direction. The mass then moves in $x-$direction with velocity $\dot{x}_o+\ell\cos(\varphi)\dot{\varphi}$, and in $y-$direction with $\ell\sin(\varphi)\dot{\varphi}$. The kinetic energy is thus given as

$$\begin{aligned} T(x_o,\varphi,\dot{x}_o,\dot{\varphi}) &= \frac{1}{2}m\big[(\dot{x}_o+\ell\cos(\varphi)\dot{\varphi})^2+(\ell\sin(\varphi)\dot{\varphi})^2\big] \\ &= \frac{1}{2}[\dot{x}_o \;\; \dot{\varphi}] \underbrace{\begin{bmatrix} m & m\ell\cos(\varphi) \\ m\ell\cos(\varphi) & m\ell^2 \end{bmatrix}}_{M(\varphi)} \begin{bmatrix} \dot{x}_o \\ \dot{\varphi} \end{bmatrix}, \end{aligned}$$

where $M(\varphi)=M(\varphi)^T>0$ depends on the angle φ.

Looking back at the MBS from definition 2.1 we can see that, though analytically irrelevant, an ill-conditioned mass matrix may pose numerical difficulties. Even though the explicit inversion can usually be avoided in a numerical integration scheme (as an example, see 6.3.3), we are still facing the structural problem that a differential variable is premultiplied by an ill-conditioned matrix in (2.5). This typically happens when the system contains bodies of masses m_k, m_ℓ with $\frac{m_k}{m_\ell}$ large. If M has the constant block diagonal form (2.17), m_k, m_ℓ are eigenvalues of M, therefore a large ratio $\frac{m_k}{m_\ell}$ results in a high condition number. Consider the ODE system

$$m_1\dot{x}=f(x,y), \qquad m_2\dot{y}=g(x,y).$$

For m_2 approaching 0, the system approaches the DAE system

$$m_1\dot{x} = f(x, y), \qquad 0 = g(x, y).$$

Vice versa, the ODE system can result from a slight perturbation of the DAE system as

$$m_1\dot{x} = f(x, y), \qquad \varepsilon\dot{y} = g(x, y)$$

with $0 < \varepsilon \ll 1$. Although the two systems are structurally unalike, we can see the numerical proximity. In practice, many numerical solvers therefore avoid dealing with the ill-conditioned ODE system by instead considering the algebraic equation resulting from a small mass set to 0. This is unproblematic for many cases, but might pose new problems in others: A major issue is the connection of several models with masses of different size. While it may be appropriate to set a small mass m_s to 0 when the other masses in the system are (relatively) much larger, when the system is coupled to another system such that m_s interacts with other masses of similar size, it is heavily inaccurate to have $m_s = 0$. We would thus like the option to use numerical solvers which can work with the ill-conditioned mass matrix to a certain point, rather than resorting to algebraic equations. Ideally, the solutions for a decreasing mass m_s then approach the solution with $m_2 = 0$, bridging the structural gap numerically. The behaviour of numerical solvers with regard to this question is therefore investigated in section 7.

2.2.3 Rotation matrix

In three-dimensional multibody systems, the position vector p_b corresponding to one body *b* contains three positional coordinates, typically denoting the body's centre of mass, as well as three angular coordinates, denoting its orientation in space, so

$$p_b = [x \;\; y \;\; z \;\; \alpha \;\; \beta \;\; \gamma]^T.$$

While the translational movement of the body is simply the time derivative of the positional coordinates, this is not true for its angular movement $[\dot{\alpha} \;\; \dot{\beta} \;\; \dot{\gamma}]^T$. In the modelling process, it is easier and most common to describe rotations with respect to the body's own coordinate system, or reference frame. While the model is based on one constant, global coordinate system, often termed 'world frame', the coordinate systems attached to the moving bodies move with them, so their orientation relative to the global coordinates change. As an example, consider a 3D cuboid with a fixed coordinate frame attached to its centre of mass. When the cuboid rotates around its x-axis, this refers to a rotation around the x-axis that is fixed to the body, independent of the current position and orientation of the cuboid, see figure 1.
This rotation is the body's *angular velocity* $\omega := [\omega_1 \;\; \omega_2 \;\; \omega_2]^T$, whereas the angles α, β, γ

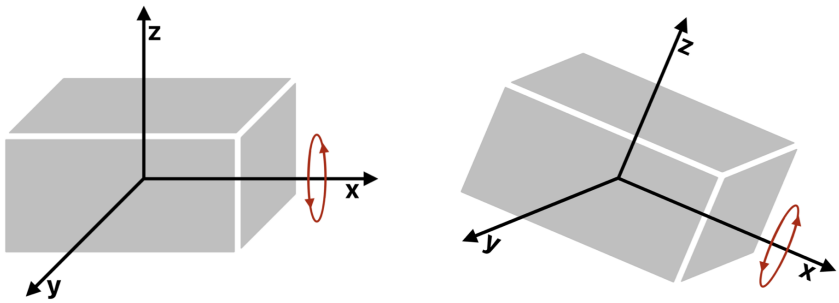


Figure 1: A cuboid with attached coordinate system

describe the body's orientation relative to the global, fixed coordinate system. Therefore, the angular movement $[\dot{\alpha} \;\; \dot{\beta} \;\; \dot{\gamma}]^T$ is not equal to the angular velocity $[\omega_1 \;\; \omega_2 \;\; \omega_3]^T$. However, there is clearly a unique relation between angular movement and angular velocity, depending on the current orientation of the body. To find this relation, we now follow chapter 1.3 in [11] and chapter 2.4 in [1]. First, we note that any rotation of a body in 3D space can be expressed as a consecutive rotation around its x-axis with angle α, then the (resulting) y-axis with angle β, and finally the (resulting) z-axis with angle γ. Denoting these by the elementary rotation matrices $R(\alpha, 0, 0), R(0, \beta, 0), R(0, 0, \gamma)$, we get the total rotation as

$$R(\alpha, \beta, \gamma) = R(0, 0, \gamma)R(0, \beta, 0)R(\alpha, 0, 0).$$

The angular velocities can now be obtained by finding the unique matrix

$$\Omega := \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}$$

which satisfies Poisson's kinematical equation

$$\dot{R}(\alpha, \beta, \gamma) = \Omega R(\alpha, \beta, \gamma).$$

For a given orientation α, β, γ of the body, we can rearrange for $\dot{\alpha}, \dot{\beta}, \dot{\gamma}$ to obtain

$$\begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} = \begin{bmatrix} \frac{\cos(\gamma)}{\cos(\beta)} & \frac{\sin(\gamma)}{\cos(\beta)} & 0 \\ \sin(\gamma) & -\cos(\gamma) & 0 \\ -\cos(\gamma)\tan(\beta) & -\sin(\gamma)\tan(\beta) & 1 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}.$$

Thus, for one body b, we get

$$\underbrace{\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\\ \dot{\alpha}\\ \dot{\beta}\\ \dot{\gamma}\end{bmatrix}}_{\dot{p}_b} = \underbrace{\begin{bmatrix} & I_3 & \\ \frac{\cos(\gamma)}{\cos(\beta)} & \frac{\sin(\gamma)}{\cos(\beta)} & 0\\ \sin(\gamma) & -\cos(\gamma) & 0\\ -\cos(\gamma)\tan(\beta) & -\sin(\gamma)\tan(\beta) & 1\end{bmatrix}}_{Z_b(p_b)} \underbrace{\begin{bmatrix}v_1\\ v_2\\ v_3\\ \omega_1\\ \omega_2\\ \omega_3\end{bmatrix}}_{v_b},$$

which can be extended to all bodies of the system to obtain $Z(p)$.
For $\beta = \frac{\pi}{2}$ or $\beta = \frac{3}{2}\pi$, the matrix $Z(p)$ is undefined, and unbounded for β approaching either value, so other parametrizations are needed. The problem can also be circumvented by using redundant 4-dimensional coordinates, named Euler parameters or quaternions.

The resulting multibody system

$$\dot{p} = Z(p)v \tag{2.18}$$
$$M(p)\dot{v} = f(p, v, s) - Z(p)^T G(p, s)^T \lambda \tag{2.19}$$
$$0 = c(p, s) \tag{2.20}$$
$$0 = g(p, s), \tag{2.21}$$

is not in the classical form of a reduced second order differential equation, since Z modulates the relation between p and v in (2.18), see again [11]. To obtain a system with $Z = I$, we can apply a variable transformation via $\tilde{v} = Z(p)v \iff v = Z(p)^{-1}\tilde{v}$. We then get

$$\dot{v} = \frac{d}{dt}(Z(p)^{-1}\tilde{v}) = Z(p)^{-1}\dot{\tilde{v}} + \dot{Z}(p)^{-1}\tilde{v},$$

and the dynamics (2.19) after transformation read

$$M(p)Z(p)^{-1}\dot{\tilde{v}} = f(p, Z(p)^{-1}\tilde{v}, s) - M(p)\dot{Z}(p)^{-1}\tilde{v} - Z(p)^T G(p, s)^T \lambda. \tag{2.22}$$

We get the kinetic energy

$$\frac{1}{2}v^T M(p)v = \frac{1}{2}\tilde{v}^T Z(p)^{-T} M(p) Z(p)^{-1}\tilde{v} = \frac{1}{2}\tilde{v}^T \tilde{M}(p)\tilde{v}$$

for $\tilde{M}(p) := Z(p)^{-T} M(p) Z(p)^{-1}$, with $\tilde{M} = \tilde{M}^T > 0$. We can now premultiply (2.22) with $Z(p)^{-T}$ to get the transformed system

$$
\begin{aligned}
\dot{p} &= \tilde{v} \\
\tilde{M}(p)\dot{\tilde{v}} &= Z(p)^{-T}\left(f(p, Z(p)^{-1}\tilde{v}, s) - M(p)\dot{Z}(p)^{-1}\tilde{v}\right) - G(p,s)^T\lambda. \\
0 &= c(p,s) \\
0 &= g(p,s).
\end{aligned}
$$

Note that even for M constant, the transformed mass matrix $\tilde{M}(p)$ is not, due to the variable transformation (compare 2.2.2). Also, we can see why the rotation matrix shows up in the term $Z(q)^T G(p,s)^T\lambda$ in the dynamics (2.19) of the original formulation, as already noted for (2.3) in the introduction: After variable transformation, we see that the rotation matrix vanishes and we again get the inaccessible directions of motion G^T as in the original second order equations of motion (2.2), scaled with the Lagrange multipliers λ.

In 2d space, there is only one possible axis of rotation (the axis pointing orthogonally 'out' of the coordinate system). Therefore, independent of a body's current orientation, the angular velocities are equal to the time derivatives of the orientational angles, and we have $Z(p) = I$.

3 Hamiltonian systems

3.1 Introduction

In the 19th century, Sir William Rowan Hamilton further developed our understanding of mechanics and physics, expanding on the work by Lagrange and Euler. One central point is to reduce the second order system of differential equations derived by the Euler-Lagrange formalism to a first order system by introducing (generalized) momenta. Formally, the momenta are defined as

$$p := \frac{\partial \mathcal{L}}{\partial \dot{q}},$$

which, for

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

gives

$$p(q, \dot{q}) = M(q)\dot{q},$$

as well as the *Hamiltonian*, given by

$$\mathcal{H}(p, q) := \dot{q}^T p - \mathcal{L}(q, \dot{q}).$$

Since the Hamiltonian does not explicitly depend on $\dot{q}$, we need for every fixed q a (continuously differentiable) bijection l_q between p and $\dot{q}$ to get from Lagrangian to Hamiltonian and vice versa. This is achieved via the *Legendre transformation*. For a formal definition and more details, see [10, 17].
In order to obtain $\dot{q}$ from p for a given q, it yields $\ell_q(p) := \dot{q}(p)$ as the solution of $p = \frac{\partial \mathcal{L}}{\partial \dot{q}}(q, \dot{q})$. Note that, in the case of classical mechanics, we have $\frac{\partial \mathcal{L}}{\partial \dot{q}}(q, \dot{q}) = M(q)\dot{q}$ with $M(q)$ symmetric positive definite, hence $p = M(q)\dot{q}$ indeed defines a bijection $p \longleftrightarrow \dot{q}$ for a fixed q. With these definitions, the Euler-Lagrange equations of motion are equivalent to Hamilton's equations

$$\dot{p} = -\mathcal{H}_q, \qquad \dot{q} = \mathcal{H}_p$$

where $\mathcal{H}_q := \nabla_q \mathcal{H} = \left(\frac{\partial \mathcal{H}}{\partial p}\right)^T, \mathcal{H}_p := \nabla_p \mathcal{H} = -\left(\frac{\partial \mathcal{H}}{\partial q}\right)^T$.
This is straightforward to verify, here we only show that Hamilton's equations follow from the Euler-Lagrange equations: Assuming that (2.1) holds, we have

$$\begin{aligned}
\frac{\partial \mathcal{H}}{\partial q}(p,q) &= \frac{\partial}{\partial q}\big(\ell_q(p)^T p\big) - \frac{\partial \mathcal{L}}{\partial q}(q, l_q(p)) \\
&= p^T \frac{\partial \ell_q}{\partial q}(p) - \frac{\partial \mathcal{L}}{\partial q}(q,\dot{q}) - \frac{\partial \mathcal{L}}{\partial \dot{q}}(q,\dot{q})\frac{\partial \ell_q}{\partial q}(p) \\
&= p^T \frac{\partial \ell_q}{\partial q}(p) - \frac{\partial \mathcal{L}}{\partial q}(q,\dot{q}) - p^T \frac{\partial \ell_q}{\partial q}(p) \\
&= -\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}}(q,\dot{q}) \\
&= -\frac{d}{dt} p^T \\
&= -\dot{p}^T,
\end{aligned}$$

and

$$\begin{aligned}
\frac{\partial \mathcal{H}}{\partial p}(p,q) &= \frac{\partial}{\partial p}\big(\dot{q}^T p - \mathcal{L}(q,\dot{q})\big) \\
&= \frac{\partial}{\partial p}\big(\ell_q(p)^T p\big) - \frac{\partial \mathcal{L}}{\partial p}(q, \ell_q(p)) \\
&= \ell_q(p)^T + \big(\frac{\partial \ell_q}{\partial p}(p)^T p\big)^T - \frac{\partial \mathcal{L}}{\partial \dot{q}}^T \frac{\partial \ell_q}{\partial p}(p) \\
&= \ell_q(p)^T + p^T \ \frac{\partial \ell_q}{\partial p}(p) - p^T \ \frac{\partial \ell_q}{\partial p}(p) \\
&= \dot{q}^T.
\end{aligned}$$

An analogous computation shows that the Euler-Lagrange equations follow from Hamilton's equations.

Definition 3.1 (Hamiltonian system)**.** Given a state space $\mathcal{X} \subseteq \mathbb{R}^{2d}$ and a function $\mathcal{H} \in C^1(\mathcal{X}, \mathbb{R})$, called the Hamiltonian function or simply Hamiltonian, a *Hamiltonian system* is a set of differential equations

$$\dot{p} = -\mathcal{H}_q(p,q), \qquad \dot{q} = \mathcal{H}_p(p.q)$$

with $q, p \in \mathbb{R}^d$ or, equivalently,

$$\dot{x} = J \ \nabla \mathcal{H}(x)$$

with $J := \begin{bmatrix} 0 & -I_d \\ I_d & 0 \end{bmatrix}$.

Although mathematically equivalent to the Euler- Lagrange equations, Hamilton's equations offer new insights: With the order reduction, the variables (p, q) now completely describe position *and* motion. The resulting configuration space is often called *state space*, as it captures the entire state of the system in that sense. Understand-

ing and analysing a system of (originally) second order differential equations from this point of view is not as intuitive, but it led to important new developments in differential geometry and geometric integration as well as corresponding numerical methods via the analysis of the state space. The notion of momentum instead of velocity can also be advantageous, and the Hamiltonian formulation of classical mechanics is now used in many physical applications, such as particle accelerators [17].
Also note that definition 3.1 does not formally require the equations to actually model a mechanical system, neither the Hamiltonian to express the total energy.

3.2 Properties

Following chapters IV and VI of [17], we investigate some important properties of Hamiltonian systems.

3.2.1 Energy conservation

One key feature of Hamiltonian systems is the conservation of energy along any solution, given by the Hamiltonian function $\mathcal{H}$. Such properties are called an invariant of the system.

Definition 3.2 (Invariants)**.** For a differential equation $\dot{x} = f(x)$, an *invariant* is a (non-constant) function $I(x)$ such that $DI(x)f(x) = 0$ for all x.

Note that the definition also includes functions x which are not solutions to the differential equation. If, however, x is a solution with $x(t_0) = x_0$, the definition implies that

$$\frac{d}{dt}I(x) = DI(x)\dot{x} = DI(x)f(x) = 0,$$

meaning that the quantity $I(x(t)) = I(x_0)$ is constant along any solution. Invariants are also called first integrals, constants of motion or conserved quantities.

Theorem 3.3. *The Hamiltonian function $\mathcal{H}$ is an invariant of the system 3.1.*

Proof. With

$$f(p,q) = \begin{bmatrix} -\mathcal{H}_q \\ \mathcal{H}_p \end{bmatrix} (p,q),$$

we have

$$\begin{aligned} D\mathcal{H}(p,q)f(p,q) &= \begin{bmatrix} \frac{\partial \mathcal{H}}{\partial p} & \frac{\partial \mathcal{H}}{\partial q} \end{bmatrix}(p,q) \begin{bmatrix} -\mathcal{H}_q \\ \mathcal{H}_p \end{bmatrix}(p,q) \\ &= \nabla \mathcal{H}(p,q)^T \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix} \nabla \mathcal{H}(p,q) \\ &= 0. \ \square \end{aligned}$$

Other examples of invariants in Hamiltonian systems are linear and angular momentum, as well as the total mass in chemical reactions [17].

3.2.2 Symplecticity

One characteristic geometric property of Hamiltonian systems is the area preservation of the flow in phase space, called symplecticity. In this part, we follow chapter VI of [17].

Definition 3.4 (Flow)**.** For a Hamiltonian system 3.1, the *flow* $\varphi_t : \mathcal{X} \to \mathcal{X}$ for a fixed time point t is defined as the function which maps an initial value x_0 to the corresponding solution of the system at time t, so $\varphi_t(x_0) = x(t)$ where x is the solution with $x(t_0) = x_0$.

Clearly, φ_0 is the identity map and therefore defined everywhere on $\mathcal{X}$, but for $t > 0$ the flow might not be well-defined for all $x_0 \in \mathcal{X}$. In the following, we always assume that φ_t is well-defined in the sense that the domain is restricted to an open subset $U \subseteq \mathcal{X}$ where necessary.
The flow allows us to investigate the dependency of the solution at a given time t of its initial value. When looking at a connected set A of initial values, one natural question is how A changes over time, meaning what is $\varphi_t(A)$ for a given t. It turns out that a core property of Hamiltonian systems is that the flow is area preserving, which we will now investigate in more detail.
First we note that the area of a parallelogram in two-dimensional space, spanned by vectors $v := [v_1 \ \ v_2]^T, w := [w_1 \ \ w_2]^T \in \mathbb{R}^2$ is given as

$$\det\left(\begin{bmatrix} v_1 & w_1 \\ v_2 & w_2 \end{bmatrix} \right),$$

since the determinant of a linear map is the volume of the unit cube under that map, and the columns of the corresponding matrix are the images of the unit vectors, hence they span the image of the unit cube, which, in two dimensions, is a parallelogram. Also note that a Hamiltonian system is defined on a state space $\mathcal{X}$ comprising the positions $q \in \mathbb{R}^d$ and momenta $p \in \mathbb{R}^d$, so $\mathcal{X} \subseteq \mathbb{R}^{2d}$. In particular, the dimensionality is always

even. Now, consider a 2-dimensional parallelogram P embedded in $\mathbb{R}^{2d}$, spanned by two vectors $\mu, \nu \in \mathbb{R}^{2d}$ with $\mu := [\mu_1^p \ \mu_2^p \ \dots \ \mu_d^p \ \mu_1^q \ \mu_2^q \ \dots \ \mu_d^q]^T, \nu := [\nu_1^p \ \nu_2^p \ \dots \ \nu_d^p \ \nu_1^q \ \nu_2^q \ \dots \ \nu_d^q]^T$. The notion of area in this context is defined by the sum over the areas of the d projections of P onto the coordinate planes (p_i, q_i), so it is given by the bilinear map

$$\tau(\mu, \nu) := \sum_{i=1}^{d} \det\Big(\begin{bmatrix} \mu_i^p & \nu_i^p \\ \mu_i^q & \nu_i^q \end{bmatrix} \Big) = \sum_{i=1}^{d} \mu_i^p \nu_i^q - \nu_i^p \mu_i^q = \mu^T \tilde{J} \nu$$

with $\tilde{J} = \begin{bmatrix} 0 & I_d \\ -I_d & 0 \end{bmatrix}$.

Definition 3.5 (Symplecticity)**.** A differentiable function $f : U \to \mathbb{R}^{2d}$ (with $U \subseteq \mathbb{R}^{2d}$ open) is called *symplectic*, if its Jacobian $Df(x)$ satisfies

$$Df(x)^T \tilde{J} Df(x) = \tilde{J} \qquad \text{for all } x \in U.$$

Looking at the definition of τ, this condition implies that $Df(x)$ satisfies

$$\tau\big(Df(x)\mu, Df(x)\nu\big) = \mu^T Df(x)^T \tilde{J} Df(x) \nu = \mu^T \tilde{J} \nu = \tau(\mu, \nu)$$

for all $\mu, \nu \in \mathbb{R}^{2d}, x \in U$, meaning that the linear approximation to f by its derivative preserves the area of two-dimensional parallelograms in the above sense. From here, the results can be extended to manifolds: Let $M \subseteq \mathbb{R}^{2d}$ be a two-dimensional manifold, given as the image of a compact set $T \subseteq \mathbb{R}^2$ under a continuously differentiable function $\phi : \mathbb{R}^2 \to \mathbb{R}^{2d}$. We can then approximate the surface area of M via parallelograms spanned by $\frac{\partial \phi}{\partial x}(x, y)$ and $\frac{\partial \phi}{\partial y}(x, y)$. Again considering the sum over the projections of these parallelograms onto the (p_i, q_i) planes, and then summing over all such parallelograms, we get the limiting expression

$$\Omega(M) := \iint_T \tau(\frac{\partial \phi}{\partial x}(x, y), \frac{\partial \phi}{\partial y}(x, y)) dx dy$$

which describes the summed area of the d projections of the manifold M onto the (p_i, q_i) planes.

Theorem 3.6. *Let $M \subseteq U \subseteq \mathbb{R}^{2d}$ be a two-dimensional manifold given as above, $f \in \mathcal{C}(U, \mathbb{R}^{2d})$ a symplectic function. Then f preserves the expression $\Omega(M)$, i.e.,*

$$\Omega(f(M)) = \Omega(M).$$

Proof. The manifold $f(M)$ is given as the image of T under the composition $f \circ \phi$.

Because f is symplectic, we get

$$\begin{aligned}\tau\big(\frac{\partial(f\circ\phi)}{\partial x}(x,y),\ \frac{\partial(f\circ\phi)}{\partial y}(x,y)\big) &= \tau\big(Df(\phi(x,y))\frac{\partial\phi}{\partial x}(x,y),\ Df(\phi(x,y))\frac{\partial\phi}{\partial y}(x,y)\big)\\ &= \tau\big(\frac{\partial\phi}{\partial x}(x,y),\ \frac{\partial\phi}{\partial y}(x,y)\big),\end{aligned}$$

and therefore

$$\begin{aligned}\Omega(f(M)) &= \iint_T \tau(\frac{\partial(f\circ\phi)}{\partial x}(x,y), \frac{\partial(f\circ\phi)}{\partial y}(x,y))dxdy\\ &= \iint_T \tau(\frac{\partial\phi}{\partial x}(x,y), \frac{\partial\phi}{\partial y}(x,y))dxdy\\ &= \Omega(M).\ \square\end{aligned}$$

We can now prove that the flow φ from 3.4 in a Hamiltonian system 3.1 is area preserving.

Theorem 3.7. *For a Hamiltonian system with $\mathcal{H}$ twice continuously differentiable and given $t \geq 0$, the flow φ_t is a symplectic map on $\mathcal{X}$.*

Proof. We show that

$$D\varphi_t(x_0)^T \tilde{J} D\varphi_t(x_0) = \tilde{J}$$

for all $x_0 \in X$ and all $t \geq 0$. First, we define $\varphi(t, x_0) := \varphi_t(x_0)$ and note that

$$\begin{aligned}\frac{d}{dt}D\varphi_t(x_0) &= \frac{\partial}{\partial t}\frac{\partial}{\partial x_0}\varphi(t,x_0)\\ &= \frac{\partial}{\partial x_0}\frac{\partial}{\partial t}\varphi(t,x_0)\\ &= \frac{\partial}{\partial x_0}J\nabla\mathcal{H}(\varphi(t,x_0))\\ &= J\nabla^2\mathcal{H}(\varphi(t,x_0))\frac{\partial\varphi}{\partial x_0}(t,x_0)\\ &= J\nabla^2\mathcal{H}(\varphi_t(x_0))D\varphi_t(x_0),\end{aligned}$$

where J is defined as in 3.1 and $\nabla^2\mathcal{H} = \nabla^2\mathcal{H}^T$ is the Hessian matrix of $\mathcal{H}$.

Since $\varphi_0(x_0) = x_0$ and thus $D\varphi_0 \equiv I$, we have

$$\Big(D\varphi_0(x_0)^T \tilde{J} D\varphi_0(x_0)\Big) = \tilde{J}$$

for all x_0. Now, with $J^T = -J, \tilde{J} = J^{-1}$ we have $J^T\tilde{J} = -I, \tilde{J}J = I$ and therefore

$$\begin{aligned}\frac{d}{dt}\Big(D\varphi_t(x_0)^T\tilde{J}D\varphi_t(x_0)\Big) &= \Big(\frac{d}{dt}D\varphi_t(x_0)\Big)^T\tilde{J}D\varphi_t(x_0) + D\varphi_t(x_0)^T\tilde{J}\Big(\frac{d}{dt}D\varphi_t(x_0)\Big)\\ &= D\varphi_t(x_0)^T\nabla^2\mathcal{H}(\varphi_t(x_0))J^T\tilde{J}D\varphi_t(x_0) + D\varphi_t(x_0)^T\tilde{J}J\nabla^2\mathcal{H}(\varphi_t(x_0))D\varphi_t(x_0)\\ &= -D\varphi_t(x_0)^T\nabla^2\mathcal{H}(\varphi_t(x_0))D\varphi_t(x_0) + D\varphi_t(x_0)^T\nabla^2\mathcal{H}(\varphi_t(x_0))D\varphi_t(x_0)\\ &= 0,\end{aligned}$$

finishing the proof. □

The symplecticity of the flow is not just a characteristic of Hamiltonian systems, but is actually uniquely linked to them, as we will now see. First, we prove the famous integrability lemma, stating a condition for which a function $f \in \mathcal{C}^1(\mathbb{R}^n, \mathbb{R}^n)$ is the gradient of another function $H \in \mathcal{C}^2(\mathbb{R}^n, \mathbb{R})$.

Lemma 3.8. *Let $U \in \mathbb{R}^n$ be open, and let $f \in \mathcal{C}(U, \mathbb{R}^n)$ with $Df(x) = Df(x)^T$ for all $x \in U$. Then, for every $x \in U$ there exists a neighbourhood $\tilde{U}$ and a function $H \in \mathcal{C}(\tilde{U}, \mathbb{R})$ such that $f = \nabla H$ on $\tilde{U}$.*

Proof. Let $\tilde{x} \in U$. First, we can assume w.l.o.g. that $\tilde{x} = 0$, since we can otherwise consider the appropriately shifted and defined $\bar{x}, \bar{U}, \bar{f}, \bar{H}$ with no changes to the statement or proof. Next, we consider a sufficiently small open ball $\tilde{U}$ such that $\tilde{x} \in \tilde{U} \subseteq U$. For a function $H : \tilde{U} \to \mathbb{R}$ to satisfy $\nabla H = f$, the direction of maximal ascent of this function at any point $x \in \tilde{U}$ has to be given by $f(x)$. Clearly, this is the case for the path integral along the straight path $\psi : [0, 1] \to \tilde{U}, \psi(t) = tx$ from 0 to x. We verify that this construction indeed gives the desired result: For $H : \tilde{U} \to \mathbb{R}$ with

$$H(x) := \int_\psi f \cdot ds = \int_0^1 \langle f(\psi(t), \psi'(t)\rangle dt = \int_0^1 x^T f(tx) dt$$

we have for $k = 1, \ ..., n$

$$\begin{aligned}
\frac{\partial \mathcal{H}}{\partial x_k}(x) &= \int_0^1 \frac{\partial}{\partial x_k}\big(x^T f(tx)\big)dt \\
&= \int_0^1 f_k(tx) + \sum_{i=1}^n x_i \frac{\partial f_i}{\partial x_k}(tx)t \ dt \\
&= \int_0^1 f_k(tx) + \sum_{i=1}^n x_i \frac{\partial f_k}{\partial x_i}(tx)t \ dt \\
&= \int_0^1 f_k(tx) + t\frac{\partial f_k}{\partial x}(tx)x \ dt \\
&= \int_0^1 \frac{d}{dt}\big(t f_k(tx)\big) \ dt \\
&= \big[t f_k(tx)\big]_0^1 \\
&= f_k(x),
\end{aligned}$$

where we used the fact that $Df = Df^T$ for the third equality. We see that indeed $\nabla \mathcal{H} = f$. $\square$

Theorem 3.9. *Let $f \in \mathcal{C}^1(U, \mathbb{R}^n), U \subseteq \mathbb{R}^n$ open. Then, the following two statements are equivalent:*

(1) For every $x \in U$ there exists a neighbourhood $\tilde{U}$ of x such that $f(x) = J\nabla \mathcal{H}(x)$.

(2) The flow φ_t of the differential equation $\dot{x} = f(x)$ is symplectic for t sufficiently small.

Proof.

$(1) \implies (2)$: For $\tilde{x} \in U$ with neighbourhood $\tilde{U}$ such that $f(x) = J\nabla \mathcal{H}(x)$ on $\tilde{U}$, let $\bar{U} \subseteq \tilde{U}$ be open and t be sufficiently small such that $\varphi_t(\bar{U}) \subseteq \tilde{U}$. Then, theorem 3.7 implies that φ_t is symplectic on $\bar{U}$. Since $\tilde{x} \in U$ is arbitrary, (2) follows.

$(2) \implies (1)$: Let $\tilde{x} \in U$ with a neighbourhood $\tilde{U}$ and $\hat{t}$ sufficiently small such that φ_t is symplectic on $\tilde{U}$ for all $t \in [0, \hat{t}]$. We need to prove the existence of a function $\mathcal{H}$ such that $f = J\nabla \mathcal{H} \iff J^{-1}f = \nabla \mathcal{H}$ on $\tilde{U}$. Lemma 3.8 guarantees the existence of H if $D(J^{-1}f)(x) = J^{-1}Df(x)$ is symmetric for all $x \in \tilde{U}$.

To show this, we do a similar computation as in theorem 3.7. First, we note that for all $t < \hat{t}$ and $x_0 \in \tilde{U}$,

$$D\varphi_t(x_0)^T J D\varphi_t(x_0) = J,$$

since $J = -\tilde{J}$ and φ_t is symplectic. Thus,

$$\begin{aligned}
0 &= \frac{d}{dt} J \\
&= \frac{d}{dt}\big(D\varphi_t(x_0)^T J D\varphi_t(x_0)\big) \\
&= \big(\frac{\partial}{\partial x_0}\frac{\partial}{\partial t}\varphi(t,x_0)\big)^T J D\varphi_t(x_0) + D\varphi_t(x_0)^T J\big(\frac{\partial}{\partial x_0}\frac{\partial}{\partial t}\varphi(t,x_0)\big) \\
&= \big(\frac{\partial}{\partial x_0} f(\varphi(t,x_0))\big)^T J D\varphi_t(x_0) + D\varphi_t(x_0)^T J \frac{\partial}{\partial x_0} f(\varphi(t,x_0)) \\
&= D\varphi_t(x_0)^T Df(\varphi_t(x_0))^T J D\varphi_t(x_0) + D\varphi_t(x_0)^T J Df(\varphi_t(x_0)) D\varphi_t(x_0) \\
&= D\varphi_t(x_0)^T \Big(Df(\varphi_t(x_0))^T J + J Df(\varphi_t(x_0))\Big) D\varphi_t(x_0).
\end{aligned}$$

Since this holds in particular for $t = 0$ with $\varphi_0(x_0) = x_0$ and $D\varphi_0(x_0) = I$, we get

$$0 = Df(x_0)^T J + J Df(x_0)$$

and thus

$$\begin{aligned}
\big(J^{-1} Df(x_0)\big)^T &= Df(x_0)^T J^{-T} \\
&= Df(x_0)^T J \\
&= -J Df(x_0) \\
&= J^{-1} Df(x_0). \ \square
\end{aligned}$$

The symplecticity of the flow is therefore (locally) a unique property of Hamiltonian systems. The differential equation $\dot{x} = f(x)$ fulfils this when f is a gradient field, also called conservative gradient field due to the energy conserving property.

The Hamiltonian system 3.1 does not include algebraic equations, which we have seen in the constraint and contact point equations in the multibody system 2.1. The traditional approach is to eliminate the constraints and again work with an unconstrained system with fewer degrees of freedom. As discussed in 2.2.1, this is not always possible, and we would therefore like the option to include algebraic constraints into the Hamiltonian system. In the next section, we look into the broad and modern framework of port-Hamiltonian systems, which extend classic Hamiltonian systems to incorporate energy dissipating elements and external ports, which can be used as a control input or as a connection to other port-Hamiltonian systems, and may also include algebraic equations.

4 Port-Hamiltonian systems of differential-algebraic equations

4.1 Introduction

As the complexity of the world grows, so does the need for ever more elaborate models and appropriate numerical integration schemes. The port-Hamiltonian (pH) framework has emerged as one of the most promising approaches taking on these challenges. Consequently, pH systems have grown into a highly active research topic in the past decades; their explicit study under this name started in the early 1990s, see for example [23]. For an overview of the research development, see [26] or [32].
They manage to incorporate the advantages of various fields and traditions: The connection of physical systems in different domains (e.g., mechanical, chemical, electrical) from port-based modeling, the geometric approach to classical Hamiltonian systems, as well as control theory ([32]). Bringing these together, pH systems have proven to be a very powerful model type, due to the possibility to connect several systems, their inherent power balance, the option to add control, as well as the existence of appropriate discretization techniques.
In this section, we give an introduction to pH systems that (may) include algebraic equations, following the paper 'Structure-preserving discretization for port-Hamiltonian descriptor systems' by Mehrmann and Morandin ([25]).

Definition 4.1 (pHDAE system)**.** Let $\mathbb{I} \subset \mathbb{R}, \mathcal{X} \in \mathbb{R}^n, \mathcal{S} = \mathbb{I} \times \mathcal{X}$. A *port-Hamiltonian system of differential-algebraic equations (pHDAE)* is a system of differential-algebraic equations of the form

$$E(t,x)\dot{x} + r(t,x) = (J(t,x) - R(t,x))z(t,x) + (B(t,x) - P(t,x))u, \tag{4.1}$$

$$y = (B(t,x) + P(t,x))^T z(t,x) + (S(t,x) - N(t,x))u, \tag{4.2}$$

with Hamiltonian $\mathcal{H} \in \mathcal{C}^1(\mathcal{S}, \mathbb{R})$, where $x(t) \in \mathcal{X}$ is the state, $u(t), y(t) \in \mathbb{R}^m$ are input and output, $r, z \in \mathcal{C}(\mathcal{S}, \mathbb{R}^l)$ are the time-flow and effort functions, $E \in \mathcal{C}(\mathcal{S}, \mathbb{R}^{l,n})$ is the flow matrix, $J, R \in \mathcal{C}(\mathcal{S}, \mathbb{R}^{l,l})$ are the structure and dissipation matrices, $B, P \in \mathcal{C}(\mathcal{S}, \mathbb{R}^{l,m})$ are the port matrices, and $S, N \in \mathcal{C}(\mathcal{S}, \mathbb{R}^{m,m})$ are the feed-through matrices.
Further, the following properties must hold:

- The matrices $\Gamma, W \in \mathbb{R}^{l+m,l+m}$, called the extended structure and dissipation matrices, given as
$$\Gamma := \begin{bmatrix} J & B \\ -B^T & N \end{bmatrix}, \quad W := \begin{bmatrix} R & P \\ P^T & S \end{bmatrix},$$
satisfy $\Gamma = -\Gamma^T$ and $W = W^T \geq 0$ pointwise.

- The gradient of the Hamiltonian $\mathcal{H}$ satisfies

$$\nabla_t \mathcal{H} = z^T r, \quad \nabla_x \mathcal{H} = E^T z \quad \text{pointwise.}$$

This very general definition even allows for E to be a non-square matrix, but here we restrict it to the case $l = n$. Also, we assume that the input u is explicitly known so that y, given by the second set of equations (4.2), is a tracking output. Clearly, the autonomous Hamiltonian system 3.1 is a special case of this definition with no input or output, $E = I, r = 0, R = 0, J = \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix}$ and $z = \nabla \mathcal{H}$. The above properties hold with $\Gamma = J = -\Gamma^T, W = 0 = W^T \geq 0, \nabla_t \mathcal{H} = 0 = z^T r$, and $\nabla_x \mathcal{H} = z = E^T z$.

For E nonsingular (pointwise), we have a system of ordinary differential equations (possibly with an output via (4.2)), since we can simply multiply both sides of (4.1) from the left with E^{-1}. Otherwise, the system also contains algebraic equations. Note that in this regard the structure here is more general than in the case of the multibody system from 2.1: The singular matrix E of rank $k < n$ does not necessarily have the form $\begin{bmatrix} E_{11} & 0 \\ 0 & 0 \end{bmatrix}$ with $E_{11} \in \mathbb{R}^{k,k}$. In that case, a separation of x into 'differential' and 'algebraic' variables is not immediately possible, but the desired structure can often be obtained via a coordinate transformation, see 4.2.2.

4.2 Properties

In the following, we investigate some important properties of pHDAE systems, such as the power balance, variable transformations, and interconnection, still summarizing results from [25].

4.2.1 Power balance

Theorem 4.2. *Along any solution x of 4.1, we have*

$$\frac{d}{dt}\mathcal{H}(t, x(t)) = -\begin{bmatrix} z \\ u \end{bmatrix}^T W \begin{bmatrix} z \\ u \end{bmatrix} + u(t)^T y(t)$$

Proof. First, we note that the pHDAE system can be written in compact form as

$$\begin{bmatrix} E\dot{x} + r \\ 0 \end{bmatrix} = (\Gamma - W) \begin{bmatrix} z \\ u \end{bmatrix} + \begin{bmatrix} 0 \\ y \end{bmatrix}.$$

Along any solution x, we then have

$$\begin{aligned}
\frac{d}{dt}\mathcal{H}(t,x(t)) &= D\mathcal{H}(t,x(t)) \cdot \begin{bmatrix} 1 \\ \dot{x}(t) \end{bmatrix} \\
&= \nabla_t \mathcal{H}^T + \nabla_x \mathcal{H}^T \dot{x} \\
&= r^T z + z^T E\dot{x} \\
&= z^T(E\dot{x} + r) \\
&= \begin{bmatrix} z \\ u \end{bmatrix}^T \begin{bmatrix} E\dot{x} + r \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} z \\ u \end{bmatrix}^T \left((\Gamma - W) \begin{bmatrix} z \\ u \end{bmatrix} + \begin{bmatrix} 0 \\ y \end{bmatrix} \right) \\
&= \begin{bmatrix} z \\ u \end{bmatrix}^T \Gamma \begin{bmatrix} z \\ u \end{bmatrix} - \begin{bmatrix} z \\ u \end{bmatrix}^T W \begin{bmatrix} z \\ u \end{bmatrix} + u^T y \\
&= - \begin{bmatrix} z \\ u \end{bmatrix}^T W \begin{bmatrix} z \\ u \end{bmatrix} + u^T y,
\end{aligned}$$

where

$$\begin{bmatrix} z \\ u \end{bmatrix}^T \Gamma \begin{bmatrix} z \\ u \end{bmatrix} = 0$$

since $\Gamma = -\Gamma^T$. $\square$

We see that the change of energy along a solution corresponds exactly to the energy given into (flowing out of) the system via the ports, $u^T y$, minus the energy lost to dissipation, $\begin{bmatrix} z \\ u \end{bmatrix}^T W \begin{bmatrix} z \\ u \end{bmatrix}$. In particular, in absence of ports and energy dissipation, energy is conserved.

Further, with $W = W^T \geq 0$ we have along any solution that the dissipation inequality

$$\begin{aligned}
\mathcal{H}(t_2,x(t_2)) - \mathcal{H}(t_1,x(t_1)) &= \int_{t_1}^{t_2} \mathcal{H}(\tau,x(\tau))d\tau \\
&= \int_{t_1}^{t_2} - \begin{bmatrix} z \\ u \end{bmatrix}^T W \begin{bmatrix} z \\ u \end{bmatrix} + u(\tau)^T y(\tau) d\tau \\
&\leq \int_{t_1}^{t_2} u(\tau)^T y(\tau) d\tau,
\end{aligned}$$

holds, so the energy increase in the system is bounded by the energy given into it via

ports, reflecting the laws of physics in the model. Without input, the energy can only decrease, so no energy can be generated within the system.

4.2.2 Representations and transformations

The representation of a system in pHDAE form is not unique; for many variable transformations, the pHDAE structure is preserved. In [24], Mehrmann and van der Schaft have shown that the index of dissipative Hamiltonian DAE systems (dHDAE), i.e., with no ports, is at most two. They also show that, under some additional assumptions, any system of index at most two has a dHDAE representation.
In the presence of algebraic equations, we would naturally like a representation which splits the system into a differential and an algebraic part, with corresponding variables. Under which conditions and how such a representation can be achieved is currently under investigation, but for some cases it has already been shown. Here, we want to present a simplified and slightly modified theorem and (shortened) proof from [4] for the case of a linear time-varying pHDAE system of index (at most) one.

Theorem 4.3. *Let*

$$E\dot{x} = [(J - R)Q - EK]x + (B - P)u \tag{4.3}$$
$$y = (B - P)^T Qx + (S + N)u \tag{4.4}$$

be a pHDAE system of index at most one with $E^T Q = Q^T E$ *and Hamiltonian* $\mathcal{H}(x) = \frac{1}{2}x^T E^T Qx$*. Then, there exists an equivalent system*

$$\bar{E}\dot{\bar{x}} = [(\bar{J} - \bar{R})\bar{Q} - \bar{E}\bar{K}]\bar{x} + (\bar{B} - \bar{P})\bar{u}$$
$$\bar{y} = (\bar{B} - \bar{P})^T \bar{Q}\bar{x} + (\bar{S} + \bar{N})\bar{u}$$

with $\bar{E} = \begin{bmatrix} \bar{E}_{11} & 0 \\ 0 & 0 \end{bmatrix}, \bar{Q} = \begin{bmatrix} \bar{Q}_{11} & 0 \\ 0 & \bar{Q}_{22} \end{bmatrix}$ *and equivalent Hamiltonian* $\bar{\mathcal{H}}$ *independent of the second part of the state ,* $\bar{x_2}$*.*

Proof. The authors have shown in the same publication that the index of at most one implies that $E(t) \in \mathbb{R}^{n,n}$ has constant rank, which implies the existence of pointwise orthogonal matrix functions $\tilde{U}, \tilde{V}$ such that

$$\tilde{E} = \tilde{U}^T E \tilde{V} = \begin{bmatrix} \tilde{E}_{11} & 0 \\ 0 & 0 \end{bmatrix}$$

with $\tilde{E}_{11}$ nonsingular ([21]). With $\tilde{Q} := \tilde{U}^T Q \tilde{V} = \begin{bmatrix} \tilde{Q}_{11} & \tilde{Q}_{12} \\ \tilde{Q}_{21} & \tilde{Q}_{22} \end{bmatrix}$ and $E^T Q = Q^T E$ we get

$$\begin{aligned} E^T Q &= (\tilde{U}\tilde{E}^T\tilde{V}^T)^T \tilde{U}\tilde{Q}\tilde{V}^T = \tilde{V}\tilde{E}\tilde{Q}\tilde{V}^T = \tilde{V} \begin{bmatrix} \tilde{E}_{11}^T\tilde{Q}_{11} & \tilde{E}_{11}^T\tilde{Q}_{12} \\ 0 & 0 \end{bmatrix} \tilde{V}^T \\ &= \tilde{V} \begin{bmatrix} \tilde{Q}_{11}^T\tilde{E}_{11} & 0 \\ \tilde{Q}_{12}^T\tilde{E}_{11} & 0 \end{bmatrix} \tilde{V}^T = Q^T E. \end{aligned}$$

Therefore, $\tilde{E}_{11}^T\tilde{Q}_{11} = \tilde{Q}_{11}^T\tilde{E}_{11}$ and, with $\tilde{E}_{11}$ nonsingular, $\tilde{Q}_{12} = 0$. Setting $\tilde{J} := \tilde{U}^T J \tilde{U}, \tilde{R} := \tilde{U}^T R \tilde{U}, \tilde{L} := \tilde{J} - \tilde{R}$, we see that the system

$$\tilde{E}\dot{\tilde{x}} = [(\tilde{J} - \tilde{R})\tilde{Q} - \tilde{E}\tilde{K}]\tilde{x}$$

is equivalent to (4.3) (without input), via the time dependent variable transformation $x = \tilde{V}x$:

$$\begin{aligned} E\dot{x} &= [(J-R)Q - EK]x \\ \iff \tilde{U}^T E \frac{d}{dt}(\tilde{V}x) &= \tilde{U}^T[(J-R)Q - EK]\tilde{V}\tilde{x} \\ \iff \tilde{U}^T E\big(\tilde{V}\dot{\tilde{x}} + \tilde{V}\tilde{V}^T\dot{\tilde{V}}\tilde{x}\big) &= [(\tilde{U}^T J - \tilde{U}^T R)\tilde{U}\tilde{U}^T Q - \tilde{U}^T E\tilde{V}\tilde{V}^T K]\tilde{V}\tilde{x} \\ \iff \tilde{E}\dot{\tilde{x}} + \tilde{E}\tilde{V}^T\dot{\tilde{V}}\tilde{x} &= (\tilde{J} - \tilde{R})\tilde{Q}\tilde{x} - \tilde{E}\tilde{V}^T K\tilde{V}\tilde{x} \\ \iff \tilde{E}\dot{\tilde{x}} &= (\tilde{J} - \tilde{R})\tilde{Q}\tilde{x} - \tilde{E}(\tilde{V}^T K\tilde{V} + \tilde{V}^T\dot{\tilde{V}})\tilde{x} \\ \iff \tilde{E}\dot{\tilde{x}} &= (\tilde{J} - \tilde{R})\tilde{Q}\tilde{x} - \tilde{E}\tilde{K}\tilde{x} \\ \iff \tilde{E}\dot{\tilde{x}} &= \tilde{L}\tilde{Q}\tilde{x} - \tilde{E}\tilde{K}\tilde{x} \\ \iff \begin{cases} \tilde{E}_{11}\dot{\tilde{x}}_1 &= (\tilde{L}_{11}\tilde{Q}_{11} + \tilde{L}_{12}\tilde{Q}_{12})\tilde{x}_1 - \tilde{E}_{11}\tilde{K}_{11}\tilde{x}_1 + \tilde{L}_{21}\tilde{Q}_{22}\tilde{x}_2 \\ 0 &= (\tilde{L}_{21}\tilde{Q}_{11} + \tilde{L}_{22}\tilde{Q}_{21})\tilde{x}_1 + \tilde{L}_{22}\tilde{Q}_{22}\tilde{x}_2 \end{cases} \end{aligned}$$

for $\tilde{K} = \tilde{V}^T K\tilde{V} + \tilde{V}^T\dot{\tilde{V}}$. Note that, even if $K = 0$, we get $\tilde{K} = \tilde{V}^T\dot{\tilde{V}}$, and the time dependent variable transformation introduces a structurally new term. In 4.1, this is reflected in the time-flow variable r.

As shown in [21], the variable transformation does not change the system's index and the matrix function $\tilde{L}_{22}\tilde{Q}_{22}$ is therefore pointwise invertible. This allows further transformations, the computations of which we will leave out here. Importantly, the required transformation matrices are not orthogonal and may be ill-conditioned; the procedure should therefore not be carried out numerically. We ultimately get transformed matri-

ces $\bar{E} = \begin{bmatrix} \bar{E}_{11} & 0 \\ 0 & 0 \end{bmatrix}, \bar{Q} = \begin{bmatrix} \bar{Q}_{11} & 0 \\ 0 & \bar{Q}_{22} \end{bmatrix}$, and the transformed system

$$\bar{E}_{11}\dot{\bar{x}}_1 = \bar{L}_{11}\bar{Q}_{11}\bar{x}_1 - \bar{E}_{11}\bar{K}_{11}\bar{x}_1 \tag{4.5}$$

$$0 = \bar{L}_{21}\bar{Q}_{11}\bar{x}_1 + \bar{L}_{22}\bar{Q}_{22}\bar{x}_2, \tag{4.6}$$

where $x = \bar{V}\bar{x}, \bar{E}_{11} = \tilde{E}_{11}$ nonsingular and $\bar{L}_{22}\bar{Q}_{22} = \tilde{L}_{22}\tilde{Q}_{22}$, so in particular $\bar{L}_{22}\bar{Q}_{22}$ is nonsingular. We thus have obtained a system where (4.5) is an ODE in $\bar{x}_1$ and (4.6) can be solved explicitly for $\bar{x}_2$ as a function of $\bar{x}_1$.

The transformed Hamiltonian $\bar{\mathcal{H}}(\bar{x}) = \frac{1}{2}\bar{x}^T\bar{E}^T\bar{Q}\bar{x}$ is indeed equivalent to $\mathcal{H}$ along the corresponding solution $\bar{x} = \bar{V}^{-1}x$, with

$$\begin{aligned}
\bar{\mathcal{H}}(\bar{x}) &= \frac{1}{2}x^T\bar{V}^{-T}\left(\bar{U}^TE\bar{V}\right)^T\bar{U}^{-1}Q\bar{V}\bar{V}^{-1}x \\
&= \frac{1}{2}x^T\bar{V}^{-T}\bar{V}^TE^T\bar{U}\bar{U}^{-1}Qx \\
&= \frac{1}{2}x^TE^TQx \\
&= \mathcal{H}(x).
\end{aligned}$$

Further, with $\bar{E} = \begin{bmatrix} \tilde{E}_{11} & 0 \\ 0 & 0 \end{bmatrix}$, we have

$$\bar{\mathcal{H}}(\bar{x}) = \frac{1}{2}\bar{x}^T\bar{E}^T\bar{Q}\bar{x} = \frac{1}{2}\bar{x}_1\tilde{E}_{11}^T\bar{Q}_{11}\bar{x}_1,$$

so the transformed Hamiltonian only depends on the first (transformed) variable $\bar{x}_1$. The input and output can easily be added to the transformed system via $\begin{bmatrix} B_1 & P_1 \\ B_2 & P_2 \end{bmatrix} = \bar{U}^T\begin{bmatrix} B & P. \end{bmatrix}$ □

4.2.3 Autonomy

We can make a non-autonomous pHDAE system autonomous by adding time as a state. This can be achieved by increasing the dimension of the sate space $\mathcal{X}$ by one and considering the new extended state space $\tilde{\mathcal{S}} = \mathbb{I} \times \tilde{\mathcal{X}}$, where $\tilde{\mathcal{X}} = \mathcal{X} \times \mathbb{I}$, so the last component of the new variable $\tilde{x}$ corresponds to time in the original system. Adding the equation $\dot{t} = 1$ and extending all matrix functions appropriately, we get the equivalent

autonomous system

$$\underbrace{\begin{bmatrix} E & r \\ 0 & 1 \end{bmatrix}}_{:=\tilde{E}} \underbrace{\begin{bmatrix} \dot{x} \\ \dot{t} \end{bmatrix}}_{:=\tilde{x}} = \Big(\underbrace{\begin{bmatrix} J & 0 \\ 0 & 0 \end{bmatrix}}_{:=\tilde{J}} - \underbrace{\begin{bmatrix} R & 0 \\ 0 & 0 \end{bmatrix}}_{:=\tilde{R}} \Big) \underbrace{\begin{bmatrix} z \\ 0 \end{bmatrix}}_{:=\tilde{z}} + \Big(\underbrace{\begin{bmatrix} B & 0 \\ 0 & 1 \end{bmatrix}}_{:=\tilde{B}} - \underbrace{\begin{bmatrix} P & 0 \\ 0 & 1 \end{bmatrix}}_{:=\tilde{P}} \Big) \underbrace{\begin{bmatrix} u \\ 1 \end{bmatrix}}_{:=\tilde{u}}$$

$$\underbrace{\begin{bmatrix} y \\ 0 \end{bmatrix}}_{:=\tilde{y}} = \Big(\begin{bmatrix} B^T & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} P^T & 0 \\ 0 & 1 \end{bmatrix} \Big) \begin{bmatrix} z \\ 0 \end{bmatrix} + \Big(\underbrace{\begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix}}_{:=\tilde{S}} - \underbrace{\begin{bmatrix} N & 0 \\ 0 & 0 \end{bmatrix}}_{:=\tilde{N}} \Big) \begin{bmatrix} u \\ 1 \end{bmatrix}.$$

Clearly, the new extended structure and dissipation matrices

$$\tilde{\Gamma} = \begin{bmatrix} \tilde{J} & \tilde{B} \\ -\tilde{B}^T & \tilde{N} \end{bmatrix}, \tilde{W} = \begin{bmatrix} \tilde{R} & \tilde{P} \\ \tilde{P}^T & \tilde{S} \end{bmatrix}$$

still fulfil $\tilde{\Gamma} = -\tilde{\Gamma}^T$ and $\tilde{W} = \tilde{W}^T \geq 0$. With no explicit time dependence on the 'new' time variable $\tilde{t}$, we have $\nabla_{\tilde{t}} \tilde{\mathcal{H}} = 0$, and with $\tilde{r} = 0$ the condition $\nabla_{\tilde{t}} \tilde{\mathcal{H}} = \tilde{r}^T \tilde{z}$ is satisfied, as well as

$$\nabla_{\tilde{x}} \tilde{\mathcal{H}} = \begin{bmatrix} \nabla_x \mathcal{H} \\ \nabla_t \mathcal{H} \end{bmatrix} = \begin{bmatrix} E^T z \\ z^T r \end{bmatrix} = \begin{bmatrix} E^T & 1 \\ r^T & 0 \end{bmatrix} \begin{bmatrix} z \\ 0 \end{bmatrix} = \tilde{E}^T \tilde{z}.$$

The autonomous system thus still has the same pHDAE structure.

4.2.4 Interconnection

For $i = 1, 2$ let

$$\begin{aligned} E_i \dot{x}_i &= (J_i - R_i) z_i + (B_i + P_i) u_i, \\ y_i &= (B_i + P_i)^T z_i + (S_i - N_i) u_i \end{aligned}$$

be two autonomous pHDAEs with Hamiltonians $\mathcal{H}_i$ with

$$My + Nu = 0$$

for the aggregated input and output $y := (y_1, y_2), u := (u_1, u_2)$ and $M, N \in \mathbb{R}^{k, m_1 + m_2}$. Then, the aggregated system is again a pHDAE system with Hamiltonian $\mathcal{H} = \mathcal{H}_1 + \mathcal{H}_2$. For a detailed analysis see [27].

4.3 Dirac structure

4.3.1 Definition

We now look into the geometric structure underlying the pHDAE, still following and summarizing [25]. Figure 2 illustrates the overall structure of a port-Hamiltonian

system, where f, e are the flow and effort variables:

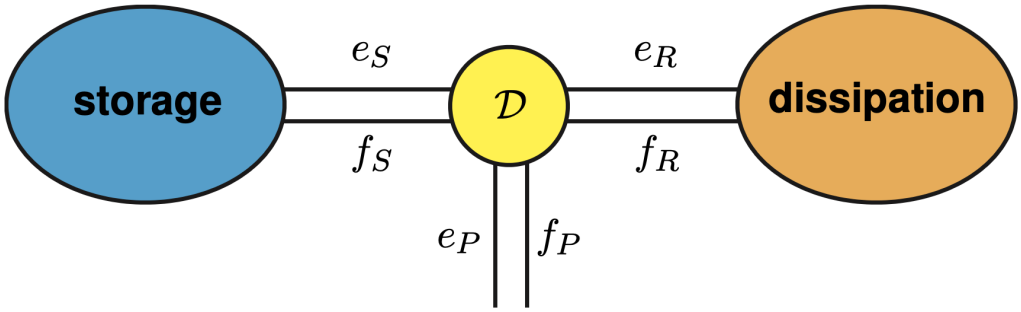


Figure 2: dissipative port-Hamiltonian structure, from [32]

The storage part describes the energy stored in the system and thus corresponds to the Hamiltonian function. We can also see a dissipation element and the external port. The product ${e_\ell}^T f_\ell, \ell \in \{s, p, d\}$ corresponds to the energy moving in direction to $\mathcal{D}$, denoting the Dirac structure. Thus, ${e_s}^T f_s$ describes the loss in stored energy and is therefore equal to $-\dot{\mathcal{H}}$, whereas ${e_p}^T f_p = y^T u$ is the energy put into the system via the external port, and ${e_d}^T f_d$ describes the energy 'gained' through dissipation, therefore ${e_d}^T f_d \leq 0$ always holds, as we will see. (Note that, in the figure we have e_R, f_R for resistance but in the following, they will always be denoted by e_d, f_d in line with the notation in [25]). In particular, the gain in stored energy should be equal to the energy given into the system via the port minus the energy lost to dissipation, meaning that

$$-{e_s}^T f_s = {e_p}^T f_p + {e_d}^T f_d,$$

or, equivalently,

$$(e_s, e_p, e_d)^T (f_s, f_p, f_d) = 0.$$

Definition 4.4. Let $\mathcal{F}$ be a vector space, $\mathcal{E}$ its dual space, and let $\langle\langle \cdot, \cdot \rangle\rangle : (\mathcal{F} \times \mathcal{E})^2 \to \mathbb{R}$ be the bilinear form defined as

$$\langle\langle (f_1, e_1), (f_2, e_2) \rangle\rangle := \langle e_1 | f_2 \rangle + \langle e_2 | f_1 \rangle,$$

where $\langle \cdot, \cdot \rangle : \mathcal{E} \times \mathcal{F} \to \mathbb{R}$ is the duality pairing $\langle e | f \rangle = e(f)$.
A *Dirac structure* on $\mathcal{F} \times \mathcal{E}$ is a subspace $\mathcal{D} \subseteq \mathcal{F} \times \mathcal{E}$ such that $\mathcal{D} = \mathcal{D}^{\perp\!\!\!\perp}$.

In case that $\dim \mathcal{F} < \infty$, the condition in 4.4 is equivalent to the two conditions $\dim(\mathcal{D}) = \dim(\mathcal{F})$ and $\langle e | f \rangle = 0$ for all $(f, e) \in \mathcal{D}$. However, this structure does not suffice for an equivalent definition of the pHDAE system, but rather corresponds to the structure for

one point x in the state space. Consequently, we need to extend the notion of a Dirac structure from vector spaces to vector bundles over $\mathcal{X}$.

Definition 4.5. Let $\mathcal{V} \oplus \mathcal{V}^*$ be the Whitney sum of a vector space $\mathcal{V}$ over $\mathcal{X}$ and its dual space $\mathcal{V}^*$. A Dirac structure $\mathcal{D} \subseteq \mathcal{V} \oplus \mathcal{V}^*$ is a subbundle such that $\mathcal{D}_x \subseteq V_x \times V_x^*$ is a Dirac structure (according to definition 4.4) for every $x \in \mathcal{X}$.

Before formulating the pHDAE via the Dirac structure, we first prove the following lemma:

Lemma 4.6. *Let $\mathcal{D} \subseteq \mathcal{V} \oplus \mathcal{V}^*$ be a subbundle with fibres*

$$\mathcal{D}_x := \Big\{ (f, e) \in \mathcal{V}_x \times \mathcal{V}_x^* \Big| f + J(x)e = 0 \Big\},$$

where $J : \mathcal{X} \to \mathcal{L}(\mathcal{V}_x^*, \mathcal{V}_x)$ is a skew-symmetric operator. Then, $\mathcal{D}$ is a Dirac structure.

Proof. We need to show that $\mathcal{D}_x \subseteq \mathcal{V}_x \times \mathcal{V}_x^*$ is a Dirac structure for every $x \in \mathcal{X}$, meaning that $\mathcal{D}_x = \mathcal{D}_x^{\perp\!\!\!\perp}$. Therefore, let $x \in \mathcal{X}$ and let $(f, e) \in \mathcal{D}_x$. We now need to show that

(1) $\langle\langle (f, e), (\tilde{f}, \tilde{e}) \rangle\rangle = 0$ for all $(\tilde{f}, \tilde{e}) \in \mathcal{D}_x$,

(2) for all $(\tilde{f}, \tilde{e}) \in (\mathcal{V}_x \times \mathcal{V}_x^*) \backslash \mathcal{D}_x$, there is at least one $(f, e) \in \mathcal{D}_x$ with $\langle\langle (f, e), (\tilde{f}, \tilde{e}) \rangle\rangle \neq 0$.

For any choice of $(f, e) \in \mathcal{D}_x, (\tilde{f}, \tilde{e}) \in \mathcal{V}_x \times \mathcal{V}_x^*$, we have

$$\begin{aligned} \langle\langle (f, e), (\tilde{f}, \tilde{e}) \rangle\rangle &= \langle e | \tilde{f} \rangle + \langle \tilde{e} | f \rangle \\ &= \langle e | \tilde{f} \rangle + \langle \tilde{e} | - J(x)e \rangle \\ &= \langle e | \tilde{f} \rangle + \langle e | J(x)\tilde{e} \rangle \\ &= \langle e | \tilde{f} + J(x)\tilde{e} \rangle. \end{aligned}$$

Now, (1) follows immediately since $\tilde{f} + J(x)\tilde{e} = 0$ for $(\tilde{f}, \tilde{e}) \in \mathcal{D}_x$.

For (2), let $(\tilde{f}, \tilde{e}) \in (\mathcal{V}_x \times \mathcal{V}_x^*) \setminus \mathcal{D}_x$. We need to show that there exists $(f, e) \in \mathcal{D}_x$ such that $\langle\langle (f, e), (\tilde{f}, \tilde{e}) \rangle\rangle \neq 0$. With $(\tilde{f}, \tilde{e}) \notin \mathcal{D}_x$, we have $\tilde{f} + J(x)\tilde{e} \neq 0$. Therefore, there exists some $e \in \mathcal{V}_x^*$ such that $\langle e | \tilde{f} + J(x)\tilde{e} \rangle = 1$. For $f = -J(x)e$, we then have $(f, e) \in \mathcal{D}_x$ and $\langle\langle (f, e), (\tilde{f}, \tilde{e}) \rangle\rangle = 1$, which finishes the proof. □

4.3.2 pHDAE formulation

We can now formulate the pHDAE in equivalent form using the Dirac structure: Let a pHDAE be given as in 4.1 autonomous form. Let $\mathcal{V} \oplus \mathcal{V}^*$ be a vector bundle with fibres $\mathcal{V}_x = \mathcal{F}_x^s \times \mathcal{F}_x^p \times \mathcal{F}_x^d$, where $\mathcal{F}_x^s := E(x)T_x\mathcal{X} \subseteq \mathbb{R}^l, \mathcal{F}_x^p := \mathbb{R}^m, \mathcal{F}_x^d := \mathbb{R}^{m+l}$ are

the storage, port, and dissipation flows, and let us partition $f = (f_s, f_p, f_d) \in \mathcal{V}$ and $e = (e_s, e_p, e_d) \in \mathcal{V}^*$. Then, $\mathcal{D} \subseteq \mathcal{V} \times \mathcal{V}^*$ with fibres given by

$$\mathcal{D}_x := \left\{ (f, e) \in \mathcal{V}_x \times \mathcal{V}_x^* \middle| f + \begin{bmatrix} \Gamma(x) & I_{\ell+m} \\ -I_{\ell+m} & 0 \end{bmatrix} e = 0 \right\}, \tag{4.7}$$

is a Dirac structure by lemma 4.6.

Theorem 4.7. *The following system of equations, with $\mathcal{D}$ defined as above,*

$$\begin{aligned} f_s &= -E(x)\dot{x}, & e_s &= z(x) \\ f_p &= y, & e_p &= u \\ e_d &= -W(x) f_d, & (f, e) &\in D_x \end{aligned}$$

is equivalent to the original pHDAE, and $\langle e | f \rangle = 0$ is equivalent to the power balance equation.

Proof. We have

$$(f, e) \in \mathcal{D}_x \iff - \begin{bmatrix} f_s \\ f_p \\ \\ f_d \end{bmatrix} = \begin{bmatrix} \Gamma(x) \begin{bmatrix} e_s \\ e_p \end{bmatrix} + e_d \\ \\ - \begin{bmatrix} e_s \\ e_p \end{bmatrix} \end{bmatrix}$$

$$\iff \begin{bmatrix} E(x)\dot{x} \\ -y \end{bmatrix} = \Gamma(x) \begin{bmatrix} z(x) \\ u \end{bmatrix} - W(x) f_d, \qquad f_d = \begin{bmatrix} z(x) \\ u \end{bmatrix}$$

$$\iff \begin{bmatrix} E(x)\dot{x} \\ -y \end{bmatrix} = \Big(\Gamma(x) - W(x) \Big) \begin{bmatrix} z(x) \\ u \end{bmatrix},$$

which is exactly the compact representation of the pHDAE.

For an autonomous system we have $\frac{d}{dt}\mathcal{H}(x(t)) = z(x)^T E(x)\dot{x}$, and therefore

$$\begin{aligned} 0 = \langle e|f\rangle &= \langle (e_s, e_p, e_d)|(f_s, f_p, f_d)\rangle \\ &= {e_s}^T f_s + {e_p}^T f_p + {e_d}^T f_d \\ &= -z(x)^T E(x)\dot{x} + u^T y - \begin{bmatrix} z(x) \\ u \end{bmatrix} W \begin{bmatrix} z(x) \\ u \end{bmatrix} \end{aligned}$$

$$\iff \frac{d}{dt}\mathcal{H}(x(t)) = u^T y - \begin{bmatrix} z(x) \\ u \end{bmatrix} W \begin{bmatrix} z(x) \\ u \end{bmatrix}. \quad \square$$

With figure 2 in mind, we can see that

- the change in stored energy is given by $-{e_s}^T f_s = z^T E\dot{x} = \dot{\mathcal{H}}$,
- the energy put into the system via ports is given by ${e_p}^T f_p = u^T y$,
- the energy lost to dissipation is given by $-{e_d}^T f_d = \begin{bmatrix} z \\ u \end{bmatrix} W \begin{bmatrix} z \\ u \end{bmatrix} \geq 0$.

4.4 port-Hamiltonian formulation of multibody systems

4.4.1 Formulation

We want to show how multibody systems fit into the port-Hamiltonian framework of definition 4.1, from here not following [25] anymore. We consider an autonomous MBS with no contact points and $Z = I$:

$$\dot{p} = v \tag{4.8}$$

$$M(p)\dot{v} = f(p, v) - G(p)^T \lambda \tag{4.9}$$

$$0 = g(p). \tag{4.10}$$

As mentioned earlier, it has been shown in [24] that the index of a dissipative Hamiltonian system is at most two. The above system can therefore not have a pHDAE formulation (at least not without ports), so we first reduce the index from three to two by replacing the holonomic constraints (4.10) with their time derivative. We again assume that $G := Dg$ has full row rank, i.e., that there are no redundant constraints. We get the index-two system

$$\begin{aligned} \dot{p} &= v \\ M(p)\dot{v} &= f(p, v) - G(p)^T \lambda \\ 0 &= G(p)v \end{aligned}$$

with kinetic energy $T(p,v) = \frac{1}{2} v^T M(p) v$, potential energy $U(p)$, and Hamiltonian

$$\mathcal{H}(p,v) = T(p,v) + U(p) = \frac{1}{2} v^T M(p) v + U(p).$$

We can now partition the force vector $f = f^p + f^d + f^u$ into

- forces f^p arising from potential energy,
- dissipative forces f^d, and
- forces f^u coming from an external port,

where $f^p = -\nabla_p \mathcal{H}$ is only linked to a potential in the classical sense for M constant, since we then have $-\nabla_p \mathcal{H} = -\nabla_p U$. Note that the force vector f and its components are not connected to the flows from the Dirac structure formulation, we use superscripts here for distinction.

With the spatial dimension d of the MBS and c constraint equations, we get, for $n := 2d + c$, the state space $\mathcal{X} = \mathbb{R}^n$ with $x = [p \;\; v \;\; \lambda]^T$. Assuming that the dissipation is of the form

$$f^d(p,v) = R(p,v) \begin{bmatrix} p \\ v \\ \lambda \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & R_{22}(p,v) & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ v \\ \lambda \end{bmatrix}$$

for some $R \in \mathcal{X}, \mathbb{R}^{n,n}$ with $R = R^T \geq 0$, we can write the MBS as

$$\begin{aligned} &\underbrace{\begin{bmatrix} I_d & 0 & 0 \\ 0 & M(p) & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{E(x)} \underbrace{\begin{bmatrix} \dot{p} \\ \dot{v} \\ \dot{\lambda} \end{bmatrix}}_{\dot{x}} \\ &= \left(\underbrace{\begin{bmatrix} 0 & I_d & 0 \\ -I_d & 0 & -G(p)^T \\ 0 & G(p) & 0 \end{bmatrix}}_{J(x)} - \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & R_{22}(p,v) & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{R(x)} \right) \underbrace{\begin{bmatrix} \nabla_p \mathcal{H}(p,v) \\ v \\ \lambda \end{bmatrix}}_{z(x)} + \underbrace{\begin{bmatrix} 0 \\ I_d \\ 0 \end{bmatrix}}_{B(x)} \underbrace{f_u}_{u}. \end{aligned}$$

Clearly, the structure and dissipation matrices J, R satisfy $J = -J^T$ and $R = R^T \geq 0$. The matrix functions $B, P \in \mathcal{C}(\mathcal{X}, \mathbb{R}^{n,d})$, as well as $S, N \in \mathcal{C}(\mathcal{X}, \mathbb{R}^{d,d})$ can only be determined with further knowledge / assumptions about the interaction with the external port. Here, we assume that $P = 0$, meaning there is no dissipation of energy along the port connection, and we make no assumptions about S, N except the structural requirements $N = -N^T, S = S^T \geq 0$. We then get, for the extended structure and dissipation

matrices

$$\Gamma = \begin{bmatrix} J & B \\ -B^T & N \end{bmatrix} = -\Gamma^T, \qquad W = \begin{bmatrix} R & 0 \\ 0 & S \end{bmatrix} = W^T \geq 0,$$

where $W \geq 0$ follows immediately from the block matrix structure and $R, S \geq 0$. For the output, we get

$$y = \underbrace{\begin{bmatrix} 0 & I_d & 0 \end{bmatrix}}_{B(x)^T} \underbrace{\begin{bmatrix} \nabla_p \mathcal{H}(p,v) \\ v \\ \lambda \end{bmatrix}}_{z(x)} + (S(x) - N(x)) f_u = v + (S(x) - N(x)) f_u.$$

It remains to verify that the Hamiltonian satisfies $\nabla_t \mathcal{H} = r^T z$ and $\nabla_x \mathcal{H} = E^T z$. First, with the system being autonomous, we have $r = 0$ and $\nabla_t \mathcal{H} = 0$, so $\nabla_t \mathcal{H} = r^T z$ is satisfied. For the second condition, we have

$$\nabla_x \mathcal{H} = \begin{bmatrix} \nabla_p \mathcal{H} \\ \nabla_v \mathcal{H} \\ \nabla_\lambda \mathcal{H} \end{bmatrix} = \begin{bmatrix} \nabla_p \mathcal{H} \\ Mv \\ 0 \end{bmatrix} = \begin{bmatrix} I_d & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & 0 \end{bmatrix}^T \begin{bmatrix} \nabla_p \mathcal{H} \\ v \\ \nabla \end{bmatrix} = E^T z.$$

Strictly speaking, $\nabla_v \mathcal{H} = M(p)v$ only holds for $M(p)$ being symmetric for all p. However, for energy based modeling we can always assume this without loss of generality: If $M(p)$ is not symmetric, consider the decomposition into its symmetric and skew-symmetric parts M_s, M_{ss} as $M(p) = M_s(p) + M_{ss}(p)$. The kinetic energy is then given as

$$T(p,v) = \frac{1}{2} v^T M(p) v = \frac{1}{2} v^T \big(M_s(p) + M_{ss}(p)\big) v = \frac{1}{2} v^T M_s(p) v + \frac{1}{2} v^T M_{ss}(p) v = \frac{1}{2} v^T M_s(p) v,$$

where the term $\frac{1}{2} v^T M_{ss}(p) v$ vanishes due to the skew-symmetry of $M_{ss}(p)$, so only the symmetric part of M contributes to the kinetic energy and therefore the Hamiltonian.

4.4.2 Simple pendulum

As an example, let us consider a simple pendulum in two dimensions, with a point mass m supported at the origin by a massless and stiff rod of length ℓ. First, we use the Euler-Lagrange formalism to derive the second order equations of motion: With $\mathcal{X} = \mathbb{R}^3$, the positional variable $p = (x, y) \in \mathbb{R}^2$ and the Lagrange multiplier $\lambda \in \mathbb{R}$, as well as the constant mass matrix $M = \begin{bmatrix} m & 0 \\ 0 & m \end{bmatrix}$, we get the kinetic energy

$$T(\dot{p}) = \frac{1}{2} \dot{p}^T M \dot{p} = \frac{1}{2} [\dot{x} \;\; \dot{y}] \begin{bmatrix} m & 0 \\ 0 & m \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \frac{1}{2} m (\dot{x}^2 + \dot{y}^2) = \frac{1}{2} m \|\dot{p}\|^2$$

and the potential energy $U(p) = -\bar{g} m (y + \ell)$, where $\bar{g} := -9.81$ is (approximately) the gravitational acceleration of earth. There is one holonomic constraint given by $g(p) =$

$\frac{1}{2}\left(x^2 + y^2 - \ell^2\right) = 0$, where the factor of $\frac{1}{2}$ is arbitrary and only chosen to make the following computations cleaner.
The Lagrangian $\mathcal{L}$ is then given as

$$\mathcal{L}(p, \dot{p}, \lambda) = T(\dot{p}) - U(p) - g(p)^T\lambda = \frac{1}{2}\dot{p}^T M\dot{p} + \bar{g}m(y + \ell) - g(p)^T\lambda.$$

With

$$\frac{d}{dt}\nabla_{\dot{p}}\mathcal{L}(p, \dot{p}, \lambda) = \frac{d}{dt}\nabla_{\dot{p}}T(\dot{p}) = \frac{d}{dt}M\dot{p} = M\ddot{p} = \begin{bmatrix} m\ddot{x} \\ m\ddot{y} \end{bmatrix}$$

and

$$\nabla_p\mathcal{L}(p, \dot{p}, \lambda) = -\nabla_p U(p) - \nabla_p g(p)\lambda = \begin{bmatrix} 0 \\ \bar{g}m \end{bmatrix} - \begin{bmatrix} x \\ y \end{bmatrix}\lambda,$$

the equations of motion are thus

$$\begin{aligned} &\frac{d}{dt}\nabla_{\dot{p}}\mathcal{L}(p, \dot{p}, \lambda) = \nabla_p\mathcal{L}(p, \dot{p}, \lambda) \\ &\quad\iff \begin{bmatrix} m\ddot{x} \\ m\ddot{y} \end{bmatrix} = \begin{bmatrix} 0 \\ \bar{g}m \end{bmatrix} - \begin{bmatrix} \lambda x \\ \lambda y \end{bmatrix}, \end{aligned}$$

together with the constraint $g(p) = \frac{1}{2}\left(x^2 + y^2 - \ell^2\right) = 0$.
Since we only have two spatial dimensions, $Z = I$ and we can reduce the order by introducing velocity variables $\bar{v} = (u, v)$ via $\dot{p} = \bar{v}$, augmenting the configuration space $\mathcal{X}$ to $\mathbb{R}^5$. We reduce the index from three to two by replacing the constraint equation with its time derivative

$$0 = \frac{d}{dt}g(p) = Dg(p)\dot{p} = G(p)\bar{v} = \begin{bmatrix} x & y \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix} = xu + yv,$$

where $G := Dg$. This gives the system

$$\begin{aligned} \dot{x} &= u \\ \dot{y} &= v \\ m\dot{u} &= -\lambda x \\ m\dot{v} &= \bar{g}m - \lambda y \\ 0 &= xu + yv. \end{aligned} \tag{4.11}$$

With the total energy given by the Hamiltonian

$$\mathcal{H}(p, \bar{v}, \lambda) = T(\bar{v}) + U(p) = \frac{1}{2}\bar{v}^T M\bar{v} - \bar{g}m(y + \ell) \tag{4.12}$$

and

$$\nabla_p \mathcal{H}(p, \bar{v}, \lambda) = \begin{bmatrix} 0 \\ -\bar{g}m \end{bmatrix},$$

it can be written in the form

$$\begin{aligned} \dot{p} &= \bar{v} \\ M\dot{\bar{v}} &= -\nabla_p \mathcal{H}(p, \bar{v}, \lambda) - G(p)^T \lambda \\ 0 &= G(p)\bar{v}, \end{aligned}$$

or, equivalently,

$$\underbrace{\begin{bmatrix} I & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{E} \underbrace{\begin{bmatrix} \dot{p} \\ \dot{\bar{v}} \\ \dot{\lambda} \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} 0 & I & 0 \\ -I & 0 & -G(p)^T \\ G(p) & 0 & 0 \end{bmatrix}}_{J(x)} \underbrace{\begin{bmatrix} \nabla_p \mathcal{H}(x) \\ \bar{v} \\ \lambda \end{bmatrix}}_{z(x)}.$$

Clearly, we get $J = -J^T$ and $0 = R = R^T \geq 0$. Also, $\nabla_t \mathcal{H} = 0$ and

$$\nabla_x \mathcal{H} = \begin{bmatrix} \nabla_p \mathcal{H} \\ \nabla_v \mathcal{H} \\ \nabla_\lambda \mathcal{H} \end{bmatrix} = \begin{bmatrix} \nabla_p \mathcal{H} \\ M\bar{v} \\ 0 \end{bmatrix} = E^T z,$$

so the system is indeed in the form of definition 4.1.
To see how dissipation and other external forces can be included into the framework, we add a viscosity term via the force $f_1(x) = -\kappa\bar{v}$, as well as a constant airflow in positive $x-$direction, exerting the force $f_2(x) = [m\alpha \;\; 0]^T$ for constants $\alpha, \kappa \geq 0$. The equations of motion (4.11) then read

$$\begin{aligned} \dot{x} &= u \\ \dot{y} &= v \\ m\dot{u} &= m\alpha - \kappa u - \lambda x \\ m\dot{v} &= \bar{g}m - \kappa v - \lambda y \\ 0 &= xu + yv. \end{aligned}$$

If we consider f_2 an input, we still have the same Hamiltonian (4.12) and the system

can be written in pHDAE form as

$$\underbrace{\begin{bmatrix} I & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{E} \underbrace{\begin{bmatrix} \dot{p} \\ \dot{\bar{v}} \\ \dot{\lambda} \end{bmatrix}}_{\dot{x}} = \left(\underbrace{\begin{bmatrix} 0 & I & 0 \\ -I & 0 & -G(p)^T \\ G(p) & 0 & 0 \end{bmatrix}}_{J(x)} - \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & \kappa I & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{R} \right) \underbrace{\begin{bmatrix} \nabla_p \mathcal{H}(x) \\ \bar{v} \\ \lambda \end{bmatrix}}_{z(x)} + \underbrace{\begin{bmatrix} 0 \\ I \\ 0 \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} \alpha m \\ 0 \end{bmatrix}}_{u},$$

where $R = R^T \geq 0$. Since $E, z, r, \mathcal{H}$ did not change, the conditions on the gradient of $\mathcal{H}$ still hold. With $P = 0$ and assuming that $S = N = 0$, we get the output $y = B^T z = \bar{v}$ and the energy given into the system via the port is $u^T y = \alpha m u$. This is in line with the intuition that the airflow accelerates the mass when it is moving in positive $x-$direction, and with $\alpha m u > 0$ the system's energy increases. When the mass is moving in negative $x-$direction, the airflow slows the mass down, and with $\alpha m u < 0$ energy flows out of the system through the port.
Alternatively, we can also consider the force f_2 as a potential V_2. We then get the Hamiltonian

$$\tilde{\mathcal{H}}(x) = T(v) + V_1(p) + V_2(p) = \frac{1}{2}\bar{v}^T M \bar{v} - \bar{g} m (y + \ell) + \alpha m (\ell - x) \tag{4.13}$$

with

$$\nabla_p \tilde{\mathcal{H}}(x) = \begin{bmatrix} -\alpha m \\ -\bar{g} m \end{bmatrix},$$

and we can write the system without ports as

$$\underbrace{\begin{bmatrix} I & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{E} \underbrace{\begin{bmatrix} \dot{p} \\ \dot{\bar{v}} \\ \dot{\lambda} \end{bmatrix}}_{\dot{x}} = \left(\underbrace{\begin{bmatrix} 0 & I & 0 \\ -I & 0 & -G(p)^T \\ G(p) & 0 & 0 \end{bmatrix}}_{J(x)} - \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & \kappa I & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{R} \right) \underbrace{\begin{bmatrix} \nabla_p \tilde{\mathcal{H}}(x) \\ \bar{v} \\ \lambda \end{bmatrix}}_{\tilde{z}(x)}.$$

In this case, adding a potential force changed the Hamiltonian from (4.12) to (4.13), but we still have $\nabla_x \tilde{\mathcal{H}} = E^T \tilde{z}$, so the system is again in pHDAE form. This highlights the fact that the representation of a physical system is not unique; not only up to coordinate transformations but also due to decisions in the modeling process.

5 Runge-Kutta methods

5.1 Introduction

In this section, we briefly revisit the basic concepts of Runge-Kutta methods and the special case of collocation methods. We then see how a pHDAE system can be discretized using a collocation method. Two Runge-Kutta methods, Gauss-Legendre collocation and the Lobatto IIIC method, are used for numerical simulations in section 7.

Definition 5.1 (Runge-Kutta method)**.** For an ordinary differential equation

$$\dot{x}(t) = f(t, x) \tag{5.1}$$

with $x \in \mathbb{R}^d$, together with initial values x_0, an s-stage *Runge-Kutta method* is given by the equations

$$X_i = x_0 + h \sum_{k=1}^{s} \alpha_{ik} \dot{X}_k, \qquad \dot{X}_i = f(t_i, X_i), \tag{5.2}$$

for $i = 1, ..., s$, and the solution at t_1

$$x_1 = x_0 + h \sum_{i=1}^{s} \beta_i \dot{X}_i, \tag{5.3}$$

with coefficients $\mathcal{A} = [\alpha_{ij}] \in \mathbb{R}^{s,s}, \beta \in \mathbb{R}^s$ and step size $h > 0$.

Many definitions explicitly include the coefficients $c_1 \leq ... \leq c_s$ which give the s stages as $t_0 + c_i h$, but we make the common assumption here that $c_i = \sum_{k=1}^{s} \alpha_{ik}$, so that $c_1, ..., c_s$ are implicitly defined. For consistency, it is also required that $\sum_{i=1}^{s} \beta_i = 1$. From the new solution x_1, the procedure can be repeated to obtain x_2 etc., with the option of changing the step size in each iteration.
Intuitively speaking, we are looking for s estimate function evaluations $\dot{X}_i = f(t_i, X_i)$ at time points $t_i := t_0 + c_i h$ to approximate the exact solution at t_1 as $x_0 + \int_{t_0}^{t_1} f(t, x(t))dt \approx x_0 + h \sum_{i=1}^{s} \beta_i \dot{X}_i$, which explains the necessary consistency condition $\sum_{i=1}^{s} \beta_i = 1$. To compute the values $\dot{X}_i = f(t_i, X_i)$, we also need to estimate solutions $X_i, ..., X_s$ at the points t_i, therefore we analogously approximate X_i as $x_0 + \int_{t_0}^{t_i} f(t, x(t))dt \approx x_0 + h \sum_{k=1}^{s} \alpha_{ik} \dot{X}_k$, explaining the common assumption that $\sum_{k=1}^{s} \alpha_{ik} = c_i$. As originally suggested by Butcher in 1963 ([8]), it is common practice to gather all coefficients

in a so-called Butcher tableau in the form

$$\begin{array}{c|cccc} c_1 & \alpha_{11} & \alpha_{12} & ... & \alpha_{1s} \\ c_2 & \alpha_{21} & \alpha_{22} & ... & \alpha_{2s} \\ \vdots & \vdots & \vdots & \vdots & \\ c_s & \alpha_{s1} & \alpha_{s2} & ... & \alpha_{ss} \\ \hline & \beta_1 & \beta_2 & ... & \beta_s \end{array}$$

If $\alpha_{ij} = 0$ for all $i \leq j$, the equations (5.2) are explicit. Therefore, a Runge-Kutta method is called explicit in this case and implicit otherwise.
The broad framework of Runge-Kutta methods covers a wide range of numerical integration schemes which had historically been used long before, such as the explicit and implicit Euler method, the implicit midpoint rule and many others. It is particularly useful to construct methods of higher order, since explicit order conditions on the coefficients could be derived during the second half of the 20th century by the work of various mathematicians and computer scientists, including Gill, Merson, Butcher, Hairer & Wanner, and many others ([17]).

Definition 5.2. A Runge-Kutta method with coefficients satisifying $c_s = 1, \alpha_{si} = \beta_i$ for $i = 1, ..., s$, is called *stiffly accurate*.

This implies that the last stage $X_s = x_0 + h\sum_{k=1}^{s} \alpha_{sk}\dot{X}_k = x_0 + h\sum_{k=1}^{s} \beta_k\dot{X}_k = x_1$ is equal to the numerical solution at t_1. This type of method is suited for the numerical solution stiff ODEs, hence the name. We will not go into the various characterizations of stiffness here. One typical case of a stiff problem is when components of the solution oscillate at highly different rates, for example in a multibody system where bodies are connected with springs of extremely different stiffness. The limiting case where a small (large) spring constant tends toward $0(\infty)$ results in an algebraic equation. Likewise, turning an algebraic equation into a differential equation via a slight perturbation typically results in a stiff problem. Therefore, stiff ODEs are 'close' to DAEs in that sense. For a DAE system, the algebraic equations are fulfilled for stiffly accurate methods since $x_1 = X_s$ and we require the DAE system to be satisfied at $X_i, ..., X_s$. This does, however, not mean that only stiffly accurate methods are suited to solve DAE systems. For specific problems such as a constrained multibody system, the problem structure can be exploited to construct more advanced solution techniques that do not fall into the category of stiffly accurate Runge-Kutta methods, yet still fulfil the algebraic equations at the numerical solution. We will see two such methods in section 6.

5.2 Collocation methods

Collocation methods are an approach which seems unrelated but is also contained in the Runge-Kutta framework. For the ODE system (5.1) with initial value x_0, we chose

s collocation points $t_i := t_0 + c_i h$ with $0 \leq c_1 < c_2 < ... < c_s \leq 1$. We then construct a solution polynomial $u \in \mathbb{R}^d[t]_{\leq s}$ in the following way:
Considering the s normalized Lagrange polynomials of degree s-1 with respect to the collocation points $c_1, ..., c_s$, given by

$$\ell_k(\tau) := \prod_{\substack{j=1 \\ j \neq k}}^{s} \frac{\tau - c_j}{c_k - c_j},$$

we look for the unique polynomial u of degree s satisfying the differential equation in the collocation points and $u(t_0) = u_0 = x_0$. We thus get

$$u'(t) = \sum_{k=1}^{s} f(t_k, u(t_k)) \ell_k\Big(\frac{t - t_0}{h}\Big)$$

with

$$\begin{aligned} u'(t_i) = u'(t_0 + c_i h) &= \sum_{k=1}^{s} f(t_k, u(t_k)) \ell_k\Big(\frac{t_0 + c_i h - t_0}{h}\Big) \\ &= \sum_{k=1}^{s} f(t_k, u(t_k)) \ell_k(c_i) \\ &= \sum_{k=1}^{s} f(t_k, u(t_k)) \delta_{ik} \\ &= f(t_i, u(t_i)). \end{aligned}$$

Integration, together with the condition $u(t_0) = u_0$, gives

$$\begin{aligned} u(t) &= u_0 + \int_{t_0}^{t} u'(\tau) d\tau \\ &= u_0 + \int_{t_0}^{t} \sum_{k=1}^{s} f(t_k, u(t_k)) \ell_k\Big(\frac{\tau - t_0}{h}\Big) d\tau \\ &= u_0 + h \sum_{k=1}^{s} f(t_k, u(t_k)) \int_0^{\frac{t - t_0}{h}} \ell_k(\tau) d\tau, \end{aligned}$$

which yields the s equations for $t_1, ..., t_s$

$$u(t_i) = u_0 + h \sum_{k=1}^{s} f(t_k, u(t_k)) \int_0^{c_i} \ell_k(\tau) d\tau. \tag{5.4}$$

The solution at t_1 is then given by

$$u(t_1) = u_0 + h \sum_{k=1}^{s} f(t_k, u(t_k)) \int_0^1 \ell_k(\tau) d\tau. \tag{5.5}$$

Setting $X_i := u(t_i), \dot{X}_i := u'(t_i) = f(t_i, X_i), \alpha_{ij} := \int_0^{c_i} \ell_j(\tau) d\tau$ and $\beta_i := \int_0^1 \ell_j(\tau) d\tau$, we see that (5.4),(5.5) are equivalent to (5.2), (5.3). Therefore, the collocation approach leads to an s-stage Runge-Kutta method with the corresponding coefficients.
While collocation methods are a special case of Runge-Kutta methods, they do immediately provide a continuous approximation to the solution and with it further insights, as we will see in the next subsection and 6.3.4. Most importantly though, the conditions on the coefficients of Runge-Kutta methods to construct methods of high order typically lead to collocation methods. The highest possible order $2s$ for an s-stage Runge-Kutta method is achieved by Gauss-Legendre collocation.

5.3 pHDAE time discretization

We will now take a closer look at the application of a collocation method to an autonomous pHDAE system as in definition 4.1 and the associated Dirac structure, in the following again summarizing results from [25]. We again consider a time interval $[t_0, t_1]$, together with s collocation points $0 \leq c_1 \leq c_2 \leq ... \leq c_s \leq 1$ and the corresponding Lagrange polynomials

$$\ell_k(\tau) := \prod_{\substack{j=1 \\ j \neq k}}^{s} \frac{\tau - c_j}{c_k - c_j},$$

and construct the solution polynomial $\tilde{x} \in \mathbb{R}^n[t]_s$ from the condition that $\tilde{x}$ satisfies the system of differential-algebraic equations in all collocation points, so

$$\begin{aligned} E(x_i)\dot{x}_i &= (J(x_i) - R(x_i))z(x_i) + (B(x_i) - P(x_i))u(x_i), \\ y(x_i) &= (B(x_i) + P(x_i))^T z(x_i) + (S(x_i) - N(x_i))u(x_i), \end{aligned}$$

where $x_i := \tilde{x}(t_i)$.

5.3.1 Discrete Dirac structure

Considering the Dirac structure formulation of the pHDAE (4.7), we can define the discrete structure in the collocation points $x_1, ..., x_s$ as

$$\mathcal{D}_{x_i} := \left\{ (f^i, e^i) \in \mathcal{V}_{x_i} \times \mathcal{V}_{x_i}^* \;\middle|\; f^i + \begin{bmatrix} \Gamma(x_i) & I_{\ell+m} \\ -I_{\ell+m} & 0 \end{bmatrix} e^i = 0 \right\}$$

and the discrete flows and efforts accordingly as

$$\begin{aligned} f_s^i &= -E(x_i)\dot{x}_i, & e_s^i &= z(x_i) \\ f_p^i &= y(x_i), & e_p^i &= u(x_i) \\ e_d^i &= -W(x_i)f_d^i, & (f^i, e^i) &\in D_{x_i}. \end{aligned}$$

Now, using the Lagrange polynomials from above, we can define the flow and effort collocation polynomials of degree $s-1$ as

$$\begin{aligned} \tilde{f}_s(t_0+h\tau) &= \sum_{i=1}^{s} f_s^i \ell_i(\tau), & \tilde{e}_s(t_0+h\tau) &= \sum_{i=1}^{s} e_s^i \ell_i(\tau), \\ \tilde{f}_d(t_0+h\tau) &= \sum_{i=1}^{s} f_d^i \ell_i(\tau), & \tilde{e}_d(t_0+h\tau) &= \sum_{i=1}^{s} e_d^i \ell_i(\tau), \\ \tilde{y}(t_0+h\tau) &= \sum_{i=1}^{s} y_i \ell_i(\tau), & \tilde{u}(t_0+h\tau) &= \sum_{i=1}^{s} u_i \ell_i(\tau). \end{aligned}$$

5.3.2 Power balance equation

Since $(f^i, e^i) \in \mathcal{D}_{x_i}$ in all collocation points by construction, the power balance equation ${f^i}^T e^i = 0$ is satisfied in the collocation points. Naturally, we are also interested in the continuous change of energy along the collocation polynomial for the whole interval $[t_0, t_1]$, and in the discrete power balance equation.
Therefore, we investigate $H(t) := \mathcal{H}(\tilde{x}(t))$, the Hamiltonian function evaluated along the collocation polynomial $\tilde{x}$. Since the power balance equation is satisfied in all collocation points, we have

$$\dot{H}(t_i) = \frac{d}{dt}\mathcal{H}(x(t)) = -\langle e_s^i | f_s^i \rangle = \langle e_d^i | f_d^i \rangle + \langle e_p^i | f_p^i \rangle = \langle e_d^i | f_d^i \rangle + \langle y_i | u_i \rangle$$

(see theorem 4.7). Assuming the quadrature rule of the collocation method is of degree $p \in \mathbb{N}$, we get

$$\begin{aligned} H(t_1) - H(t_0) &= h\sum_{j=1}^{s} \beta_j \dot{H}(t_j) + \mathcal{O}(h^{p+1}) \\ &= h\sum_{j=1}^{s} \beta_j \langle e_d^j | f_d^j \rangle + h\sum_{j=1}^{s} \beta_j \langle y_j | u_j \rangle + \mathcal{O}(h^{p+1}). \end{aligned} \tag{5.6}$$

Since we have defined polynomials for the flows and efforts, we can again apply the quadrature rule to obtain

$$\int_{t_0}^{t_1} \langle \tilde{e}_d(s) | \tilde{f}_d(s) \rangle ds = h \sum_{j=1}^{s} \beta_j \langle e_d^j | f_d^j \rangle + \mathcal{O}(h^{p+1}), \tag{5.7}$$

$$\int_{t_0}^{t_1} \langle \tilde{y}(s) | \tilde{u}(s) \rangle ds = h \sum_{j=1}^{s} \beta_j \langle y^j | u^j \rangle + \mathcal{O}(h^{p+1}). \tag{5.8}$$

If $p \geq 2s-2$, equations (5.7) and (5.8) are exact, since $\tilde{e}_d, \tilde{f}_d, \tilde{y}, \tilde{u}$ are polynomials of degree $s-1$ and the integrand in both equations is thus a polynomial of degree $2s-2$. If, additionally, $\beta_j \geq 0$ for all $j = 1, ..., s$, we also have

$$\int_{t_0}^{t_1} \langle \tilde{e}_d(s) | \tilde{f}_d(s) \rangle ds = h \sum_{j=1}^{s} \beta_j \langle e_d^j | f_d^j \rangle \leq 0, \tag{5.9}$$

since the system of equations is fulfilled in the collocation points and $W \geq 0$, so $\langle e_d^j | f_d^j \rangle = {e_d^j}^T f_d^j = -{f_d^j}^T W(x_i) f_d^j \leq 0$. Therefore, also in the discretized version, energy can only be lost to dissipation. Inserting (5.7) and (5.8) into (5.6), we get for the change of $\mathcal{H}$ along $\tilde{x}$ that

$$\begin{aligned} H(t_1) - H(t_0) &= h \sum_{j=1}^{s} \beta_j \langle e_d^j | f_d^j \rangle + h \sum_{j=1}^{s} \beta_j \langle y_j | u_j \rangle + \mathcal{O}(h^{p+1}) \\ &= \int_{t_0}^{t_1} \langle \tilde{e}_d(s) | \tilde{f}_d(s) \rangle + \langle \tilde{y}(s) | \tilde{u}(s) \rangle ds + \mathcal{O}(h^{p+1}). \end{aligned}$$

Clearly, this equation can, in general, not be exact since the Hamiltonian need not be a polynomial, and not of sufficiently small degree. If, however, the Hamiltonian is a polynomial of degree 2, we get $H := \mathcal{H} \circ \tilde{x} \in \mathbb{R}[t]_{2s}$ and thus $\dot{H} \in \mathbb{R}[t]_{2s-1}$. For a quadrature rule of degree $2s-1$, the equations (5.7), (5.8) are also exact, and we get the change of energy in one integration step as

$$\begin{aligned} H(t_1) - H(t_0) &= h \sum_{j=1}^{s} \beta_j \dot{H}(t_j) \\ &= h \sum_{j=1}^{s} \beta_j \langle e_d^j | f_d^j \rangle + h \sum_{j=1}^{s} \beta_j \langle y_j | u_j \rangle \\ &= \int_{t_0}^{t_1} \langle \tilde{e}_d(s) | \tilde{f}_d(s) \rangle + \langle \tilde{y}(s) | \tilde{u}(s) \rangle ds. \end{aligned}$$

The power balance equation therefore maintains its structure in the discretized version, with an exactness depending on the chosen integration scheme and the Hamiltonian function. If the Hamiltonian is quadratic, as is the case in many physical applications, and the chosen quadrature rule is of degree $2s-1$, as is only the case for Gauss-Legendre

quadrature, we get an exact discrete version of the power balance equation along the solution polynomial $\tilde{x}$.
In that case, since $\beta_j \geq 0$ for all $j = 1, ..., s$ in Gauss-Legendre collocation, (5.9) holds and we also get

$$\begin{aligned} H(t_1) - H(t_0) &= h \sum_{j=1}^{s} \beta_j \dot{H}(t_j) \\ &= h \sum_{j=1}^{s} \beta_j \langle e_d^j | f_d^j \rangle + h \sum_{j=1}^{s} \beta_j \langle y_j | u_j \rangle \\ &\leq h \sum_{j=1}^{s} \beta_j \langle y_j | u_j \rangle \\ &= \int_{t_0}^{t_1} \langle \tilde{y}(s) | \tilde{u}(s) \rangle ds, \end{aligned}$$

meaning that both dissipation inequality and power balance equation are qualitatively preserved in discretization.

6 Partitioned Runge-Kutta methods

6.1 Introduction

With the findings from the previous section, it seems natural to use Gauss-Legendre collocation for the numerical solution of pHDAE systems, and, in particular, for multi-body systems with holonomic constraints. The problem is, however, that the system of differential-algebraic equations is only fulfilled in the collocation points $t_i = t_0 + hc_i$, where $c_s < 1$ for Gauss-Legendre collocation. The numerical solution at t_1 does satisfy the constraint if $c_s = 1$, meaning that t_1 is equal to the last stage of the Runge-Kutta method, and $\alpha_{is} = \beta_i$, meaning the numerical solution x_1 is equal to the last stage X_s. This is exactly the definition of stiffly accurate methods (5.2). We could therefore apply a stiffly accurate method to the entire system, but this has two major drawbacks: First, the method cannot achieve the maximal order of Gauss-Legendre collocation. Second, no stiffly accurate Runge-Kutta method is symplectic. This follows directly from the respective conditions for the coefficients, see [19, 17] or [27].
In order to reconcile these rather unfortunate facts, we can apply different Runge-Kutta methods to different variables, leading to partitioned Runge-Kutta methods.

Definition 6.1 (Partitioned Runge-Kutta method)**.** For a partitioned ordinary differential equation of the form

$$\dot{x} = f(x, y), \quad \dot{y} = g(x, y), \tag{6.1}$$

a *partitioned Runge-Kutta method* (PRK) is given by two individual Runge-Kutta methods with coefficients α_{ij}, β_i and $\bar{\alpha}_{ij}, \bar{\beta}_i$, applied to x and y, so

$$\begin{aligned} X_i &= x_0 + h\sum_{k=1}^{s} \alpha_{ik}\dot{X}_k, \qquad \dot{X}_i = f(X_i, Y_i), \\ Y_i &= y_0 + h\sum_{k=1}^{s} \bar{\alpha}_{ik}\dot{Y}_k, \qquad \dot{Y}_i = f(X_i Y_i), \qquad\qquad i = 1, ..., s. \end{aligned}$$

The solution at t_1 is given as

$$x_1 = x_0 + h\sum_{i=1}^{s} \beta_i \dot{X}_i, \qquad y_1 = y_0 + h\sum_{i=1}^{s} \bar{\beta}_i \dot{Y}_i$$

While an MBS with constraints, or any system of differential-algebraic equations, is not of the form (6.1), PRK methods can be adapted to specific problems which include algebraic equations. In this section, we will look into two such methods for constrained MBS. The first is based on the Lobatto IIIA-IIIB pair and described by Laurent Jay in [19, 18], the second based on Gauss-Legendre collocation and an adapted Lobatto

method, described by A. Murua in [28]. For the remainder of this thesis, we refer to the methods by the slightly more compact names J-PRK and M-PRK after their authors.

6.2 Lobatto IIIA-IIIB pair

6.2.1 Definition

In 1977, Ryckaert, Ciccotti & Berendsen ([30]) proposed the SHAKE method, a symmetric, symplectic method of order two which can be applied to separable Hamiltonians of the form

$$H(q,p) = \frac{1}{2}p^T M^{-1} p + U(q),$$

together with the positional constraint $g(q) = 0$.
A reformulation which improves numerical stability and adds a projection step, resulting in the numerical solution satisfying both position and velocity constraints, was introduced by Andersen in 1983 ([2]) and appropriately named RATTLE. In the 1990s, both Jay ([18]) and Reich ([29]) extended the method to general Hamiltonians. In 1996, Jay ([19]) was able to embed the previous findings into the broader framework of PRK methods, in particular the Lobatto IIIA-Lobatto IIIB pair, of which RATTLE is the special case for separable Hamiltonians and 2 stages. Following the original publication by Jay as well as ([17]), we will now take a closer look at this method, with some of the formulations adapted or simplified for our purposes.

Definition 6.2 (J-PRK method)**.** Given the constrained Hamiltonian system

$$\begin{aligned} \dot{p} &= v \\ M(p)\dot{v} &= -\nabla H_p(p,v) - G(p)^T\lambda \\ 0 &= g(p), \end{aligned}$$

as well as consistent initial values (p_0, v_0, λ_0), an $s-$stage ***J-PRK method*** is given by

$$P_i = p_0 + h\sum_{j=1}^{s} \alpha_{ij}\dot{P}_j, \qquad \dot{P}_i = V_i, \tag{6.2}$$

$$V_i = v_0 + h\sum_{j=1}^{s} \bar{\alpha}_{ij}\dot{V}_j, \quad M(P_i)\dot{V}_i = -\nabla H_p(P_i,V_i) - G(P_i)^T\Lambda_i, \tag{6.3}$$

$$g(P_i) = 0, \qquad i = s,...,1, \tag{6.4}$$

with the solution at t_1 defined as

$$p_1 = p_0 + h\sum_{i=1}^{s} \beta_i\dot{P}_i, \qquad v_1 = v_0 + h\sum_{i=1}^{s} \bar{\beta}_i\dot{V}_i,$$

where the coefficients $\alpha_{ij}, \bar{\alpha}_{ij}, \beta_i, \bar{\beta}_i$ as well as the stages $c_1 \leq c_2 ... \leq c_s$, $\bar{c}_1 \leq \bar{c}_2 ... \leq \bar{c}_s$ correspond to the two underlying Runge-Kutta methods.

Clearly, we would now like to chose the underlying RK methods in such a way that the resulting J-PRK method (given consistent initial values) satisfies the following conditions:

- The numerical flow is symplectic
- The numerical solution fulfills the algebraic constraints
- The order of the method should be as high as possible

The analysis of the stages and coefficients offers a meaningful way to derive such a method. In his publication, Jay manages to extend a result from Hairer (1994, [16]) for unconstrained Hamiltonian systems to the constrained case:

Theorem 6.3. *If the numerical method 6.2 has a unique solution (p_1, v_1) for consistent initial values (p_0, v_0, λ_0), then the numerical flow $(p_0, v_0) \rightarrow (p_1, v_1)$ is symplectic if the coefficients of the method satisfy the following conditions:*

$$\begin{aligned} \beta_i &= \bar{\beta}_i, && i = 1, ..., s \\ \beta_i \bar{\alpha}_{ij} + \bar{\beta}_j \alpha_{ji} - \beta_i \bar{\beta}_j &= 0, && i = 1, ..., s, \quad j = 1, ..., s, \end{aligned} \tag{6.5}$$

where the first condition can be omitted for separable Hamiltonians.

In order for the method to satisfy the algebraic constraints at the numerical solution, i.e., $g(p_1) = 0$, we need the 'first' method, treating the variable p, to be stiffly accurate, meaning that

$$\alpha_{sj} = \beta_j, \qquad j = 1, ..., s.$$

Together with the symplecticity conditions (6.5), we get

$$\begin{aligned} \beta_i \bar{\alpha}_{is} + \bar{\beta}_s \alpha_{si} - \beta_i \bar{\beta}_s &= \beta_i \bar{\alpha}_{is} + \bar{\beta}_s \beta_i - \beta_i \bar{\beta}_s \\ &= \beta_i \bar{\alpha}_{is} \\ &= 0, \qquad i = 1, ..., s. \end{aligned}$$

Thus, $\bar{\alpha}_{is} = 0$ for $\beta_i \neq 0$. Since all high order RK methods have $\beta_i \neq 0$ for all i (all collocation methods do, compare the definition in section 5.2), we look for methods which satisfy $\bar{\alpha}_{is} = 0$ for $i = 1, ..., s$.
This affects the J-PRK method in the following way: The unknown $\dot{V}_s$ and therefore

also Λ_s are excluded from the nonlinear system which has to be solved in one step of the method. We can therefore freely chose the value of Λ_s, which then also determines the value of $\dot{V}_s$. This does, however, change the new solution $v_1 = v_0 + h\sum_{i=1}^{s} \bar{\beta}_i \dot{V}_i$ (for $\bar{\beta}_s \neq 0$). This freedom comes with the great benefit that we can set Λ_s such that v_1 satisfies

$$G(p_1)v_1 = 0,$$

meaning that the numerical solution p_1, v_1 satisfies not only the positional constraint, but also the originally hidden constraint on velocity level. Due to this projection step, the velocity constraints are exactly fulfilled in every integration step, as we can also see in the numerical simulations 7.3.1. As mentioned by Jay, this choice of Λ_s does not destroy the symplecticity of the flow.
However, with $\bar{\alpha}_{is} = 0$, the equations (6.3) become

$$V_i = v_0 + h\sum_{j=1}^{s-1} \bar{\alpha}_{ij}\dot{V}_j, \quad M(P_i)\dot{V}_i = -\nabla H_p(P_i, V_i) - G(P_i)^T\Lambda_i.$$

In order for the system (6.2)-(6.4) to have a unique solution, we therefore need to eliminate an unknown. One way to achieve this is by setting P_1 equal to the initial value p_0. For consistent inital values, the constraint $g(P_1) = 0$ is then automatically satisfied. Clearly, $P_1 = p_0$ holds if $\alpha_{1j} = 0, \;\; j = 1, ..., s$.
Again imposing the symplecticity conditions (6.5) and assuming $\beta_i \neq 0$, we get, for $j = 1$,

$$\begin{aligned} &\beta_i\bar{\alpha}_{i1} + \beta_j\alpha_{1i} - \beta_i\beta_1 = \beta_i\bar{\alpha}_{i1} - \beta_i\beta_1 = 0 \\ &\iff \bar{\alpha}_{i1} = \beta_1. \end{aligned}$$

Taking all these conditions together, we are left with the two methods Lobatto IIIA and Lobttao IIIB, which fulfill all requirements and are each of order $2s - 2$. The only Runge-Kutta method of higher order is based on Gauss-Legendre collocation and does not fulfil the requirements for either of the two methods. For the rest of this thesis, we always refer to definition 6.2 in combination with the Lobatto IIIA-IIIB pair when using the term J-PRK method.

6.2.2 Convergence

In his publication [19], Jay manages to prove general convergence conditions for the coefficients of the method in a very elaborate way, using rooted trees for the Taylor expansions. For the special case of the Lobatto IIIA-Lobatto IIIB pair, we get the following result:

Theorem 6.4. *For consistent initial values* (p_0, v_0, λ_0)*, the global error of the method 6.2*

with the underlying RK methods Lobatto IIIA, Lobatto IIIB satisfies

$$p_n - p(t_n) = \mathcal{O}(h^{2s-2}), \qquad v_n - v(t_n) = \mathcal{O}(h^{2s-2}), \qquad \lambda_n - \lambda(t_n) = \mathcal{O}(h^{2s-2}),$$

given that $nh \leq C$ *for a constant* C*. Here,* x_n *denotes the exact and* $x(t_n)$ *the numerical solution at* $t_n = t_0 + nh$ *for the respective variables.*

6.2.3 Energy conservation

Since the method is symplectic, we expect good energy conservation. However, partitioned Runge-Kutta methods do not conserve all quadratic invariants. The Lobatto IIIA-IIIB pair for an unconstrained system $\dot{x} = f(x, y), \dot{y} = g(x, y)$ conserves quadratic invariants of the form

$$I(x, y) = x^T D y, D \in \mathbb{R}^{n,n},$$

see [17] (chapter IV). This does not include the Hamiltonian of an MBS, and the presence of algebraic constraints adds to possible inaccuracies in energy conservation.

6.3 Murua

6.3.1 Definition

In a paper published by Murua in 1997 ([28]), an interesting variation of a PRK method is introduced, suited to solve autonomous, semi-explicit DAE systems of index two. This applies to various Multibody systems, assuming the holonomic constraints have been reduced to index two.

Definition 6.5 (M-PRK method)**.** Given an autonomous, semi-explicit index-two DAE system of the form

$$\dot{x} = f(x, z) \quad g(x) = 0 \tag{6.6}$$

with $x \in \mathbb{R}^d$ as well as two sets of s stages $c_1 < c_2 < ... < c_s, \;\; \bar{c_1} < \bar{c_2} < ... < \bar{c}_s \in [0, 1]$, consistent initial values (x_0, z_0), stepsize h, coefficient matrices $A = [a_{ij}], \bar{A} = [\bar{a}_{ij}] \in \mathbb{R}^{s,s}$ and vectors $\beta, \gamma \in \mathbb{R}^d$, an s-stage *M-PRK method* is given by the equations

$$\begin{aligned} X_i &= x_0 + h \sum_{k=1}^{s} \alpha_{ik} \dot{X}_k, \qquad & \dot{X}_i &= f(X_i, Z_i), \\ \bar{X}_i &= x_0 + h \sum_{k=1}^{s} \bar{\alpha}_{ik} \dot{X}_k, \qquad & g(\bar{X}_i) &= 0, \qquad & i &= s, ..., 1, \end{aligned} \tag{6.7}$$

and the solution at t_1

$$x_1 = x_0 + h\sum_{i=1}^{s} \beta_i \dot{X}_i, \qquad z_1 = \sum_{i=1}^{s} \gamma_i Z_i.$$

This definition can be used to construct PRK methods which exactly fulfil the constraint equations at the solution: If the coefficients are chosen such that there exists (at least) one $l \in \{1, ..., s\}$ with $\bar{\alpha}_{li} = \beta_i$ for all $i \in \{1, ..., s\}$, we have

$$x_1 = x_0 + h\sum_{i=1}^{s} \beta_i \dot{X}_i = x_0 + h\sum_{k=1}^{s} \bar{\alpha}_{lk} \dot{X}_k = \bar{X}_l,$$

with $g(\bar{X}_l) = 0$. Since both $\sum_{i=1}^{s} \beta_i = 1$ and $\sum_{k=1}^{s} \bar{\alpha}_{ik} = \bar{c}_i$ are required for the respective Runge-Kutta method's consistency, we are in particular interested in methods with $\bar{c}_s = 1, \bar{\alpha}_{si} = \beta_i$ and, consequently, $x_1 = \bar{X}_s$.

For certain coefficients, this method is an extension of collocation methods as described in 5.2 to PRK methods, as we will now explore in more detail:
If we choose to approximate the solution $(x, z)(t)$ in the interval $[t_0, t_0+h]$ by polynomials $u, v \in \mathbb{R}^d[t]_{\leq s}$, we get the conditions

$$\begin{aligned} u(t_0) &= x_0, \quad v(t_0) = z_0, \\ u'(t_i) &= f\big(u(t_k), v(t_k)\big), \\ u(\bar{t}_i) &= 0, \qquad\qquad\qquad i = 1, ..., s, \end{aligned} \tag{6.8}$$

where $t_i := t_0 + c_i h, \bar{t}_i := t_0 + \bar{c}_i h$. Again using normed Lagrange polynomials $\ell_1^{(u)}, ..., \ell_s^{(u)}$ with respect to the nodes $c_1, ..., c_s$, integration yields

$$\begin{aligned} u(t_i) &= u_0 + h\sum_{k=1}^{s} \int_0^{c_i} \ell_k(\tau) d\tau \;\; f\big(u(t_k), v(t_k)\big), \\ u(\bar{t}_i) &= u_0 + h\sum_{k=1}^{s} \int_0^{\bar{c}_i} \ell_k(\tau) d\tau \;\; f\big(u(t_k), v(t_k)\big), \\ 0 &= g\big(u(\bar{t}_i)\big). \end{aligned}$$

For the polynomial v, corresponding to the algebraic variables, we use the $s+1$ collocation points $0 = c_0, c_1, ..., c_s$ and the respective Lagrange polynomials $\ell_1^{(v)}, ..., \ell_s^{(v)}$. Note that they are of degree s, since no integration is carried out for v. Setting $Z_0 := z_0$, we can define

$$v(t) = \sum_{k=0}^{s} \ell_k^{(v)}\big(\frac{t - t_0}{h}\big) \;\; Z_k,$$

which is the unique polynomial of degree s satisfying

$$v(t_0) = z_0, \qquad v(t_0 + c_i h) = Z_i, \qquad\qquad , i = 1, ..., s.$$

The solution at t_1 is then obtained as

$$x_1 = u(t_1) = u_0 + h \sum_{k=1}^{s} \int_0^{c_i} \ell_k(\tau) d\tau \;\; f\big(u(t_i), v(t_i)\big),$$
$$z_1 = v(t_1) = \sum_{k=0}^{s} \ell_k^{(v)}\big(\frac{t_1 - t_0}{h}\big) Z_k = \sum_{k=0}^{s} \ell_k^{(v)}(1) Z_k.$$

Setting $X_i := u(t_i), \bar{X}_i := u(\bar{t}_i), \dot{X}_i := u'(t_i) = f(X_i, Z_i), \alpha_{ij} := \int_0^{c_i} \ell_j(\tau) d\tau, \bar{\alpha}_{ij} := \int_0^{\bar{c}_i} \ell_j(\tau) d\tau$, as well as $\beta_i := \int_0^1 \ell_j(\tau) d\tau$ and $\gamma_i := \ell_i^{(v)}(1)$, we can see that this approach is indeed equivalent to the M-PRK method defined in 6.5. We can now also see why the weight vector γ is not called $\bar{\beta}$ as one might expect, as it is not connected to the second Runge-Kutta method. Also note that the weights are either given by

$$\gamma_0 = \gamma_1 = ... = \gamma_{s-1} = 0, \quad \gamma_1 = 1, \text{ for } c_s = 1,$$

which gives $z_1 = Z_s$, or

$$\gamma_i \neq 0 \text{ for all } i = 1, ..., s.$$

In either case, since the Lagrange polynomials are a partition of unity, we have

$$\sum_{i=1}^{s} \gamma_i = \ell_i^{(v)}(1) = 1.$$

6.3.2 Convergence

In order to prove convergence results, Murua first defines the following conditions:

$$C(q) := \sum_{k=1}^{s} \alpha_{ik} c_k^{l-1} = \frac{c_i^l}{l}, \qquad\qquad 1 \leq i \leq s, 1 \leq l \leq q,$$
$$\bar{C}(\bar{q}) := \sum_{k=1}^{s} \bar{\alpha}_{ik} c_k^{l-1} = \frac{\bar{c}_i^l}{l}, \qquad\qquad 1 \leq i \leq s, 1 \leq l \leq \bar{q}.$$

It is mentioned without proof that the collocation method (6.8) fulfils both conditions for $q = \bar{q} = s$, which we will quickly prove here.

Lemma 6.6. *The collocation method* (6.8) *fulfils* $C(s)$ *and* $\bar{C}(s)$.

Proof. For $k \in \{1, ..., s\}$, consider the polynomial $p^k(\sigma) := \sigma^{k-1}$. Clearly, $deg(p^{(k)}) \leq s - 1$ for all k and it is therefore equal to the Lagrange interpolation polynomial on the s

collocation points $c_1, ..., c_s$ from the collocation method. Thus,

$$p^{(k)}(\sigma) = \sigma^{k-1} = \sum_{j=1}^{s} p^{(k)}(c_j)\ell_j(\sigma) = \sum_{j=1}^{s} c_j^{k-1}\ell_j(\sigma).$$

Integration on both sides yields

$$\begin{aligned}\int_0^{c_i} \sigma^{k-1} d\sigma &= \Big[\frac{1}{k}\sigma^k\Big]_0^{c_i} = \frac{c_i^k}{k} \\ &= \int_0^{c_i} \sum_{j=1}^{s} c_j^{k-1}\ell_j(\sigma)d\sigma = \sum_{j=1}^{s} c_j^{k-1}\int_0^{c_i}\ell_j(\sigma)d\sigma = \sum_{j=1}^{s} c_j^{k-1}\alpha_{ij}.\end{aligned}$$

Similarly, for the condition $\bar{C}$, we get

$$\begin{aligned}\int_0^{\bar{c}_i} \sigma^{k-1} d\sigma &= \frac{\bar{c}_i^k}{k} \\ &= \sum_{j=1}^{s} \bar{c}_j^{k-1}\int_0^{\bar{c}_i}\ell_j(\sigma)d\sigma = \sum_{j=1}^{s} \bar{c}_j^{k-1}\bar{\alpha}_{ij}. \quad \square\end{aligned}$$

Murua then states two order results, which we summarize in an adapted version as follows:

Theorem 6.7. *For an s-stage PRK method as in (6.5) satisfying the conditions $C(s), \bar{C}(s)$ and $b_i = \bar{\alpha}_{si}$ for $i = 1, ..., s$,*

$$\begin{aligned}X_i - x(t + c_i h) &= \mathcal{O}(h^{s+1}) \\ \bar{X}_i - x(t + \bar{c}_i h) &= \mathcal{O}(h^{s+1}) \\ Z_i - z(t + c_i h) &= \mathcal{O}(h^{s}).\end{aligned}$$

If $\bar{c}_s = 1$ and $\bar{c}_i \neq 0$ for $i = 1, ..., s$, the local error of the $x-$ component satisfies

$$x_1 - x(t_0 + h) = \mathcal{O}(h^{min(p,\bar{p})+1}),$$

where $p, (\bar{p})$ is the order of the quadrature formula for the s nodes $c_1, ..., c_s$ (the $s+1$ nodes $\bar{c}_0 = 0, \bar{c}_1, ..., \bar{c}_s = 1$).

This result suggests to use high order quadrature rules such that the collocation points satisfy the stated requirements. The natural choice for the first method is Gauss-Legendre collocation due to the maximal order of $2s - 1$. For the second method, we know that the collocation points need to satisfy $\bar{c}_0 = 0, \bar{c}_s = 1$, suggesting to use the Lobatto quadrature nodes, which are given as the $s + 1$ zeros of the polynomial

$$P(x) := \frac{d^{s-1}}{dx^{s-1}}(x^s(x-1)^s).$$

Clearly, $P(0) = P(1) = 0$. Further, $\bar{c}_0 = 0$ is a simple root, and thus $\bar{c}_i \neq 0$ for all $i = 1, ..., s$, and since both methods are based on collocation, they satisfy $C(s), \bar{C}(s)$ as stated in 6.6. Note that, crucially, the condition $\beta_i = \bar{\alpha}_{si}$, for all $i = 1, ..., s$, holds, ensuring that the constraints are satisfied at the numerical solution x_1. This condition is fulfilled since we have $\bar{c}_1 = 1$, which immediately gives

$$\beta_i = \int_0^1 \ell_i(\tau) d\tau = \int_0^{\bar{c}_1} \ell_i(\tau) d\tau = \bar{\alpha}_{si}.$$

The combination of Gauss-Legendre collocation and the adapted Lobatto method is also suggested by Murua. For the remainder of this thesis, we always refer to this combination when we use the term M-PRK method. The only thing left to do is to compute the appropriate coefficients α_{ij} and the weights $\gamma_k,\ k = 0, ..., s$ for the Lobatto method. The procedure is clear, but since there are a few miscalculations or typos in Murua's paper and the weights γ_k are not explicitly given, we present the Butcher tableaus for $s \leq 3$ in the form

c	$\mathcal{A}$
	β

$\bar{c}$	$\bar{\mathcal{A}}$
$*$	$*$
γ_0	$\gamma_1, ..., \gamma_s.$

Murua PRK method coefficients for $s = 1$

$\frac{1}{2}$	$\frac{1}{2}$
	1

1	1
$*$	$*$
-1	2

Murua PRK method coefficients for $s = 2$

$\frac{1}{2} - \frac{\sqrt{3}}{6}$	$\frac{1}{4}$	$\frac{1}{4} - \frac{\sqrt{3}}{6}$
$\frac{1}{2} + \frac{\sqrt{3}}{6}$	$\frac{1}{4} + \frac{\sqrt{3}}{6}$	$\frac{1}{4}$
	$\frac{1}{2}$	$\frac{1}{2}$

$\frac{1}{2}$	$\frac{1}{4} + \frac{\sqrt{3}}{8}$	$\frac{1}{4} - \frac{\sqrt{3}}{8}$
1	$\frac{1}{2}$	$\frac{1}{2}$
$*$	$*$	$*$
1	$-\frac{3}{\sqrt{3}}$	$\frac{3}{\sqrt{3}}$

Murua PRK method coefficients for $s = 3$

$\frac{1}{2} - \frac{\sqrt{15}}{10}$	$\frac{5}{36}$	$\frac{2}{9} - \frac{\sqrt{15}}{15}$	$\frac{5}{36} - \frac{\sqrt{15}}{30}$
$\frac{1}{2}$	$\frac{5}{36} + \frac{\sqrt{15}}{24}$	$\frac{2}{9}$	$\frac{5}{36} - \frac{\sqrt{15}}{24}$
$\frac{1}{2} + \frac{\sqrt{15}}{10}$	$\frac{5}{36} + \frac{\sqrt{15}}{30}$	$\frac{2}{9} + \frac{\sqrt{15}}{15}$	$\frac{5}{36}$
	$\frac{5}{18}$	$\frac{4}{9}$	$\frac{5}{18}$

$\frac{1}{2} - \frac{\sqrt{5}}{10}$	$\frac{25-\sqrt{5}+6\sqrt{15}}{180}$	$\frac{10-4\sqrt{5}}{45}$	$\frac{25-\sqrt{5}-6\sqrt{15}}{180}$
$\frac{1}{2} + \frac{\sqrt{5}}{10}$	$\frac{25+\sqrt{5}+6\sqrt{15}}{180}$	$\frac{10+4\sqrt{5}}{45}$	$\frac{25+\sqrt{5}-6\sqrt{15}}{180}$
1	$\frac{5}{18}$	$\frac{4}{9}$	$\frac{5}{18}$
$*$	$*$	$*$	$*$
-1	$\frac{5}{3}$	$-\frac{4}{3}$	$\frac{5}{3}$

6.3.3 Application to multibody systems

Let us consider the MBS

$$\begin{aligned} \dot{p} &= v \\ M(p)\dot{v} &= f(p, v, \lambda) - G(p)^T \lambda \\ 0 &= G(p)v \end{aligned}$$

from definition 2.1 without contact points and $Z = I$.
In order for this system to have the appropriate index 2 DAE form 6.6, we multiply the second equation by $(M(p))^{-1}$. Applying the method (6.7) yields

$$P_i = p_0 + h \sum_{k=1}^{s} \alpha_{ik} V_k, \tag{6.9}$$

$$V_i = v_0 + h \sum_{k=1}^{s} \alpha_{ik} \dot{V}_k \tag{6.10}$$

$$\dot{V}_i = M(P_i)^{-1}[f(P_i, V_i, \Lambda_i) - G(P_i)^T \Lambda_i], \tag{6.11}$$

$$\bar{P}_i = p_0 + h \sum_{k=1}^{s} \bar{\alpha}_{ik} \dot{P}_k, \tag{6.12}$$

$$\bar{V}_i = v_0 + h \sum_{k=1}^{s} \bar{\alpha}_{ik} \dot{V}_k, \tag{6.13}$$

$$G(\bar{P}_i)\bar{V}_i = 0, \qquad i = s, ..., 1. \tag{6.14}$$

where $\dot{P}_k$ has already been substituted by V_k in (6.9). Before numerically solving the system, we multiply each of the s equations in (6.11) by $M(P_i)$ again, which gives

$$M(P_i)\dot{V}_i = f(P_i, V_i, \Lambda_i) - G(P_i)^T \Lambda_i.$$

By computing the values for $\dot{V}_i$ explicitly instead of substituting in (6.10), we avoid the explicit inversion of $M(P_i)$. If M is constant, we can do the substitution and multiply all s resulting equations by M, which gives the same system with (6.10), (6.11) combined to

$$MV_i = Mv_0 + h\sum_{k=1}^{s} \alpha_{ik}\dot{V}_k.$$

The M-PRK method can also be applied to the Gear-Gupta-Leimkuhler formulation of the MBS in a straightforward manner, as it has the same semi-explicit index 2 DAE structure 6.6.

6.3.4 Energy conservation

We now want to investigate energy conservation of the numerical solution when the method is applied to the conservative MBS

$$\begin{aligned} \dot{p} &= v \\ M(p)\dot{v} &= -\nabla \mathcal{H}_p(p, v) - G(p)^T \lambda \\ 0 &= G(p)v \end{aligned}$$

in index-two formulation, which is a Hamiltonian system with associated Hamiltonian $\mathcal{H}(p, v)$ describing the systems total energy, see section 4.4. We again consider the collocation polynomials $\tilde{x} = (\tilde{p}, \tilde{v}), \tilde{z} = \tilde{\lambda}$. In the collocation points t_i of the first method, i.e.,

Gauss-Legendre collocation, we get

$$\begin{aligned}
\frac{d}{dt}\mathcal{H}\big((\tilde{x}(t),\tilde{z}(t)\big)\Big|_{t_i} &= \psi(\tilde{p}_i,\tilde{v}_i,\tilde{\lambda}_i)^T E(\tilde{p}_i,\tilde{v}_i,\tilde{\lambda}_i)\begin{pmatrix}\dot{\tilde{p}}_i\\ \dot{\tilde{v}}_i\\ \dot{\tilde{\lambda}}_i\end{pmatrix}\\
&= \begin{bmatrix}\nabla\mathcal{H}_p(\tilde{x}_i,\tilde{\lambda}_i)^T & \tilde{v}_i^T & \tilde{\lambda}_i^T\end{bmatrix}\begin{bmatrix}I & 0 & 0\\ 0 & M(\tilde{p}_i) & 0\\ 0 & 0 & 0\end{bmatrix}\begin{pmatrix}\dot{\tilde{p}}_i\\ \dot{\tilde{v}}_i\\ \dot{\tilde{\lambda}}_i\end{pmatrix}\\
&= \begin{bmatrix}\nabla\mathcal{H}_p(\tilde{x}_i,\tilde{\lambda}_i)^T & \tilde{v}_i^T & \tilde{\lambda}_i^T\end{bmatrix}\begin{bmatrix}\tilde{v}_i\\ -\nabla\mathcal{H}_p(\tilde{x}_i,\tilde{\lambda}_i) - G(\tilde{p}_i)^T\tilde{\lambda}_i\\ 0\end{bmatrix}\\
&= \nabla\mathcal{H}_p(\tilde{x}_i,\tilde{\lambda}_i)^T\tilde{v}_i - \tilde{v}_i^T\nabla\mathcal{H}_p(\tilde{x}_i,\tilde{\lambda}_i)^T - \tilde{v}_i^T G(\tilde{p}_i)^T\tilde{\lambda}_i\\
&= \tilde{\lambda}_i^T G(\tilde{p}_i)\tilde{v}_i.
\end{aligned}$$

However, the time points $t_i = t_0 + c_i h$ are the collocation points of the first method, and the constraint is only satisfied in the collocation points $\bar{t}_i = t_0 + \bar{c}_i h$ of the second method, i.e.,

$$0 = G(\tilde{p}(\bar{t}_i)\tilde{v}(\bar{t}_i).$$

Thus, the power balance equation is not fulfilled in the collocation points (of either method). Instead, using Gauss-Legendre quadrature, we get the approximation

$$\begin{aligned}
\mathcal{H}(\tilde{x}(t_1),\tilde{\lambda}(t_1)) - \mathcal{H}(\tilde{x}(t_0),\tilde{\lambda}(t_0)) &= h\int_0^h \frac{d}{dt}\Big(\mathcal{H}(\tilde{x}(\tau),\tilde{\lambda}(\tau))\Big)d\tau\\
&= h\sum_{i=1}^{s}\omega_i\frac{d}{dt}\Big(\mathcal{H}(\tilde{x}(\tau),\tilde{\lambda}(\tau))\Big)\Big|_{t_i} + \mathcal{O}(h^{2s})\\
&= h\sum_{i=1}^{s}\omega_i\tilde{\lambda}_i^T G(\tilde{p}_i)\tilde{v}_i + \mathcal{O}(h^{2s}),
\end{aligned}$$

or equivalently, in terms of the PRK definition 6.5,

$$\mathcal{H}(x_1,z_1) - \mathcal{H}(x_0,z_0) = h\sum_{i=1}^{s}\omega_i Z_i^T G(P_i)V_i + \mathcal{O}(h^{2s}), \tag{6.15}$$

with the Gauss-Legendre quadrature weights ω_i. As in 5.3.2, the equation is exact for $\mathcal{H} \in \mathbb{R}[t]_{\leq 2}$. This error term and the resulting energy drift is exactly reflected in the numerical results, see 7.3.1. Although the M-PRK method does therefore not precisely

conserve energy, theorem 6.7 states that

$$X_i - x(t + c_i h) = \mathcal{O}(h^{s+1}).$$

With the exact solution satisfying $G(p(t_i))v(t_i) = 0$, we then also have

$$G(\tilde{p}_i)\tilde{v}_i = \mathcal{O}(h^{s+1}),$$

which encourages us to expect near-conservation of energy, in particular for small step sizes. This is supported by the findings in section 7. For a more precise error bound, one would have to locally bound the differential function of the specific problem.

7 Numerical simulations

7.1 Introduction

In this section, we present various results from numerical simulations of multibody systems. All implementations are written in Python (version 3.9.19) with the IDE Spyder (version 6.0.2). The implemented solvers are Gauss-Legendre collocation, the Lobatto IIIC method, and the two more elaborate partitioned Runge-Kutta methods by Jay and Murua from the previous section. All solvers are implemented similarly in the following way:
(1) First, the coefficient matrices required for the respective methods are defined for stages less or equal to three.
(2) To set up a multibody system for a specific problem, one needs to define the dimensionality, the number of constraints, the mass matrix as well as the right hand side function of the differential equations and, if present, the algebraic constraints, as well as problem-specific parameters and initial values. Note that, if algebraic equations are present, the initial values need to be consistent with the explicit and hidden constraints.
(3) To solve the system numerically, a wrapper function is called, which integrates the system until a specified time point is reached. To do so, it repeatedly calls a solver function which carries out the individual integration steps. When calling the wrapper function, we can pass arguments to specify

- initial values
- the end point for the integration time interval
- the number of stages the Runge-Kutta method uses
- the initial step size
- the minimum and maximum step sizes
- the factor by which the step size is decreased (increased) when the tolerance is exceeded (kept)
- the relative error tolerance for which each integration step is tested (two tolerances for the J-PRK method)
- the number of integration steps without exceeding the tolerance until the step size is increased
- a maximum number of integration steps before the program stops and returns results (independent of current integration time)

The function returns arrays which contain, for each integration step,

- the solution of the numerical integration
- the solution of the system of equations given by the Runge-Kutta method(s)

- the values which were tested against the tolerance(s). These are the relative 2-norms of the system of equations evaluated at its numerical solution
- the step sizes, required to scale plots accordingly.

For more details see the documentation and comments in the python files.

7.2 Runge-Kutta methods

Before looking at numerical results for partitioned Runge-Kutta methods from section 6, we first go back to non-partitioned Runge-Kutta methods. In particular, we want to examine the performance of Gauss-Legendre collocation (GLC) and the Lobatto IIIC method (LIIIC) with respect to energy conservation and for the case that the mass of one body tends towards zero.
For this purpose, we use a two-body mass-spring-damper model (see figure 3) with point masses m_1, m_2, three springs which are at rest when their extension is equal to r_i and with spring constants $k_i > 0$. The first and third spring are supported at fixed points with position coordinates 0 and $l := r_1 + r_2 + r_3$, respectively, but there are neither actual boundaries nor contact points in this model, so the masses can move freely along the x-axis. Note that by setting the distance ℓ to be the sum of the three spring lengths at rest, we know that there is an equilibrium point with a potential energy of zero for $p_1 = r_1, p_2 = r_1 + r_2 = l - r_3$, where all springs are at rest. Further, there are two dampers with damping coefficients $d_i \geq 0$. For $d_1 = d_2 = 0$, we can use either the Euler-Lagrange or the Hamiltonian formalism to derive the equations of motion:

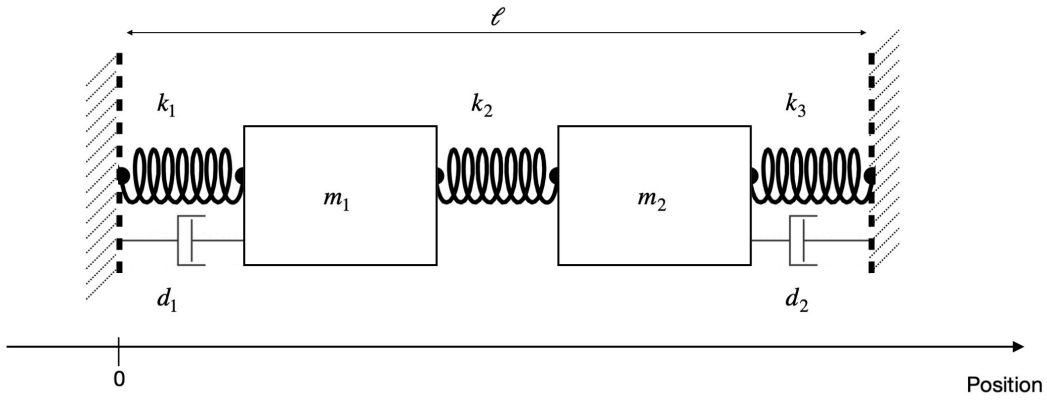


Figure 3: Two-body mass-spring-damper system

With the total energy given by the Hamiltonian

$$\mathcal{H}(p,v) = \frac{1}{2}(m_1 v_1^2 + m_2 v_2^2) + \frac{1}{2}\left[k_1(p_1 - r_1)^2 + k_2(p_2 - p_1 - r_2)^2 + k_2(p_2 - p_1 - r_2)^2 + k_3(l - p_2 - r_3)^2\right]$$

and

$$\nabla_p \mathcal{H}(p,v) = \begin{bmatrix} k_1(p_1 - r_1) - 2k_2(p_2 - p_1 - r_2) \\ 2k_2(p_2 - p_1 - r_2) - k_3(l - p_2 - r_3) \end{bmatrix},$$

we get the equations of motion as the Hamiltonian system

$$\dot{p}_1 = v_1 \tag{7.1}$$

$$\dot{p}_2 = v_2 \tag{7.2}$$

$$m_1\dot{v}_1 = -k_1(p_1 - r_1) + 2k_2(p_2 - p_1 - r_2) \tag{7.3}$$

$$m_2\dot{v}_2 = -2k_2(p_2 - p_1 - r_2) + k_3(l - p_2 - r_3). \tag{7.4}$$

Assuming that the dissipative dampers are linear in the respective body's velocity, we can add them to (7.3), (7.4) and replace the two equations by

$$m_1\dot{v}_1 = -k_1(p_1 - r_1) + 2k_2(p_2 - p_1 - r_2) - d_1v_1$$
$$m_2\dot{v}_2 = -2k_2(p_2 - p_1 - r_2) + k_3(l - p_2 - r_3) - d_2v_2.$$

7.2.1 Energy conservation

First, we want to verify our findings from section 5 concerning energy conservation. With the quadratic Hamiltonian, the symplectic Gauss-Legendre collocation should conserve energy exactly for the non-dissipative case, while the stiffly accurate Lobatto IIIC method is not symplectic and should therefore show changes in energy.
We set up the system with the parameters

$$m_1 = m_2 = 1, r_1 = r_2 = r_3 = 10, k_1 = k_2 = k_3 = 1, d_1 = d_2 = 0$$

and with initial values $p_1 = 6, p_2 = 24, v_1 = v_2 = 0$. Note that $d_1 = d_2 = 0$, so the system is non-dissipative (conservative).
We now solve the system numerically until $t_{end} = 2000$ with both solvers, with inital step size 10^{-3}, tolerance 10^{-5}, and maximum step size 0.1. We can see the expected behaviour of the system's total energy: The Lobatto IIIC solver shows a constant decrease of energy, albeit very slightly, while Gauss-Legendre collocation is energy conserving (figure 4).
We can also see that LIIIC performance with regard to energy conservation deteriorates drastically with increasing step size while it barely affects GLC (figure 5): With a maximum step size of 5, initial step size of 10^{-3} and tolerance of 10^{-12}, the total energy quickly declines to zero for LIIIC, while GLC conserves the total energy. Figure 5b shows a zoom into the LIIIC results from figure 5a, where we can see clearly how the energy drop sets in when the step size reaches a certain threshold, here for $h = 1.25$ (the step size in the graph has been multiplied by 10 to make it visible). This again

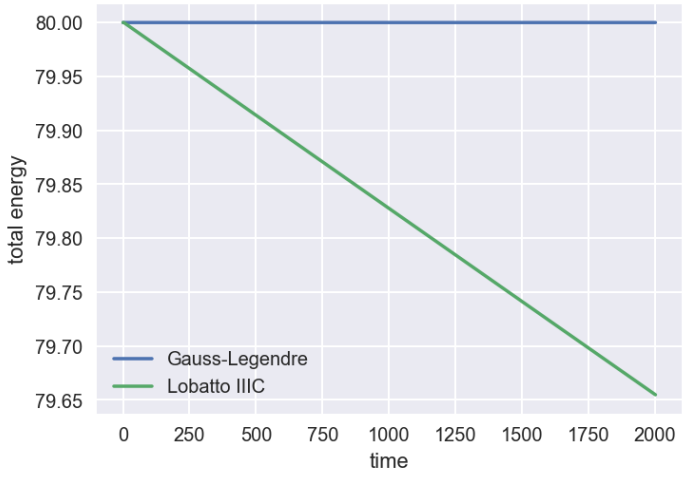
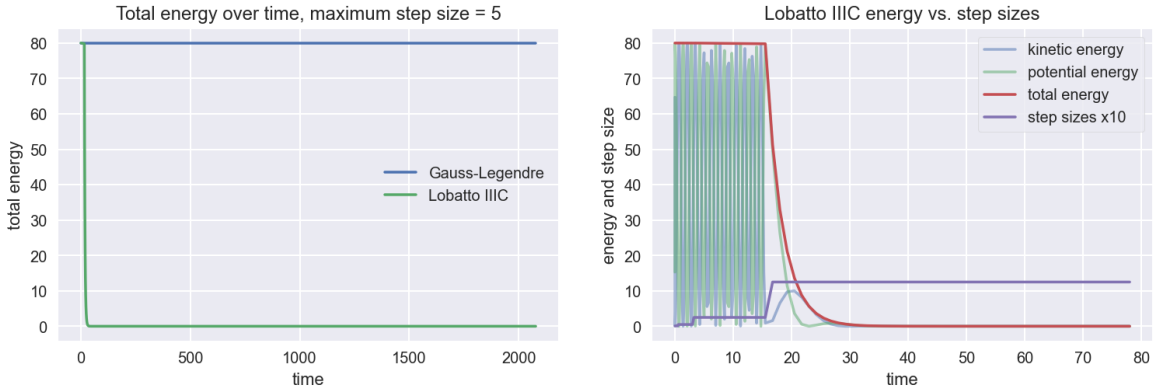


Figure 4: Energy conservation



(a) (b)

Figure 5: Energy conservation with large step size

highlights the importance of symplecticity for long-term integration.

7.2.2 Power balance

For the mass-spring-damper system, we can also test the theoretical results for the discretized power balance equation (PBE) from section 5.3.2. To include energy dissipation and input, we set the damping coefficients $d_1, d_2 > 0$ and add a constant force pushing the first mass to the right, which is given by $m\dot{v}_1 = \alpha, \alpha > 0$ and modelled as an input. These alterations do not affect the Hamiltonian, which is therefore still a polynomial of degree 2, and a polynomial of degree $2s$ along the numerical solution $\tilde{x}$, which we again define as $H(t) := \mathcal{H}(\tilde{x}(t))$. We thus expect, for each integration step from t_i to t_{i+1}, that

$$H(t_{i+1}) - H(t_i) = h \sum_{j=1}^{s} \beta_j \dot{H}(t_j^{(i)}) = h \sum_{j=1}^{s} \beta_j \langle e_d^{j^{(i)}} | f_d^{j^{(i)}} \rangle + h \sum_{j=1}^{s} \beta_j \langle y_j^{(i)} | u_j^{(i)} \rangle, \tag{7.5}$$

is exactly fulfilled up to numerical errors, where β_j are the Gauss-Legendre quadrature weights. We can write the system in port-Hamiltonian form as

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & m_1 & 0 \\ 0 & 0 & 0 & m_2 \end{bmatrix}}_{E} \underbrace{\begin{bmatrix} \dot{p}_1 \\ \dot{p}_2 \\ \dot{v}_1 \\ \dot{v}_2 \end{bmatrix}}_{\dot{x}} = \left(\underbrace{\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix}}_{J} - \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & d_1 & 0 \\ 0 & 0 & 0 & d_2 \end{bmatrix}}_{R} \right) \underbrace{\begin{bmatrix} \nabla_p \mathcal{H} \\ \\ v_1 \\ v_2 \end{bmatrix}}_{z(x)} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}}_{B} \underbrace{\alpha}_{u}.$$

With $P = S = N = 0$, we get the energy dissipation along any solution x as ${e_d}^T f_d = z^T R z = d_1 v_1^2 + d_2 v_2^2 \geq 0$ and the energy-flow into (out of) the system via the port as $u^T y = \alpha B^T z = \alpha v_1$. For the numerical solution, the discrete version of the PBE is still exact, and replacing the terms in (7.5) with the results from above, we expect to have

$$\begin{aligned} H(t_{i+1}) - H(t_i) &= h \sum_{j=1}^{s} \beta_j \langle e_d^{j^{(i)}} | f_d^{j^{(i)}} \rangle + h \sum_{j=1}^{s} \beta_j \langle y_j^{(i)} | u_j^{(i)} \rangle \\ &= h \sum_{j=1}^{s} \beta_j \left(d_1 V_{1j}^{(i)2} + d_2 V_{2j}^{(i)2} \right) + h \sum_{j=1}^{s} \beta_j \alpha V_{1j}^{(i)2}, \end{aligned}$$

where $V_{kj}^{(i)}$ is the variable v_k at stage j from the Runge-Kutta method at the integration step from t_i to t_{i+1}. Numerical simulations yield the expected results, which are shown for three stages and for the parameters $d_1 = 0.2, d_2 = 0.5, \alpha = 3$ in figure 6.

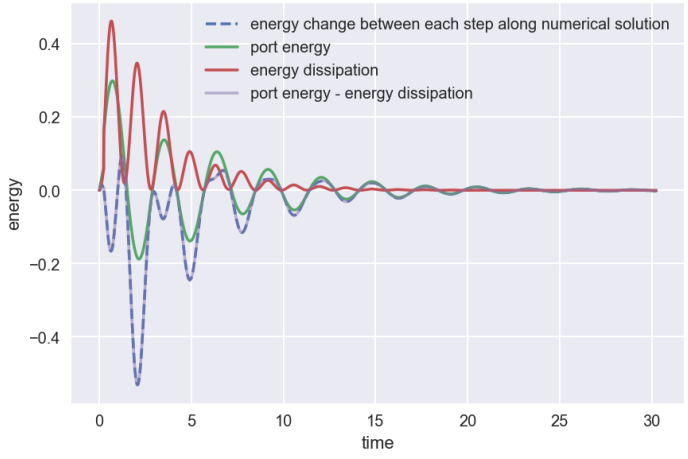


Figure 6: Power balance for Gauss-Legendre Collocation

We can use the exact same procedure for the Lobatto IIIC method, using the respective quadrature weights. At the scale of figure 6, we cannot see a difference to Gauss-Legendre collocation, but we know from section 5.3.2 that the discrete PBE along a quadratic Hamiltonian can only be exact for a quadrature rule of order $2s - 1$, i.e.,

Gauss-Legendre quadrature. Taking a closer look at the results, we indeed see that the PBE for the Lobatto IIIC method is less accurate. Figure 7 shows the difference

$$H(t_{i+1}) - H(t_i) - \left(h \sum_{j=1}^{3} \beta_j \left(d_1 V_{1j}^{(i)2} + d_2 V_{2j}^{(i)2}\right) + h \sum_{j=1}^{3} \beta_j \alpha V_{1j}^{(i)2}\right)$$

for both methods, where β_j are the respective quadrature weights and $V_{kj}^{(i)}$ the respective Runge-Kutta solutions as defined above.

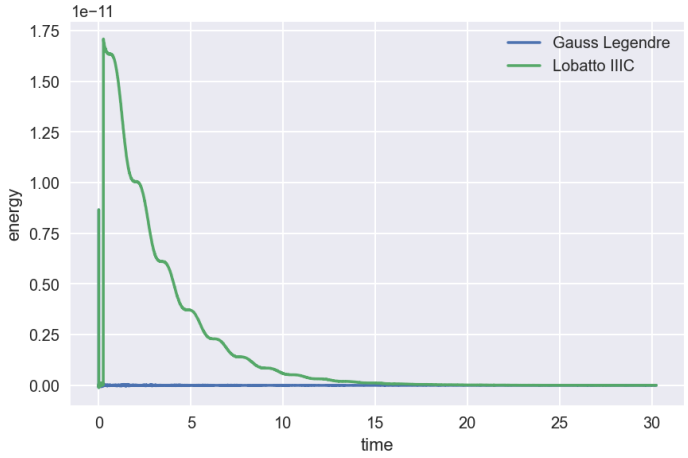


Figure 7: Difference in power balance between solvers

7.2.3 ODE approaches DAE

Now, we want to investigate the case that one mass tends towards zero, and if the ODE solutions tend towards the corresponding DAE solution. For this purpose, we remove the input, set $m_1 = 1$ and choose decreasing values for m_2. In the limit, the ODE system tends towards the DAE system

$$\dot{p}_1 = v_1 \tag{7.6}$$

$$\dot{p}_2 = v_2 \tag{7.7}$$

$$m_1 \dot{v}_1 = -k_1(p_1 - r_1) + 2k_2(p_2 - p_1 - r_2) - d_1 v_1 \tag{7.8}$$

$$0 = -2k_2(p_2 - p_1 - r_2) + k_3(l - p_2 - r_3) - d_2 v_2. \tag{7.9}$$

For $d_2 = 0$, this system is of index two, since we need to rearrange (7.9) for p_2, differentiate twice with respect to time and then substitute into the time derivative of (7.7). This gives an equation of the form $\dot{v}_2 = \tilde{\varphi}(\dot{v}_1)$, where we can substitute $\dot{v}_1$ by (7.8) (after rearranging for $\dot{v}_1$), and we finally obtain an equation of the form $\dot{v}_2 = \varphi(p, v)$. For $d_2 \neq 0$ however, the system is clearly of index one, since we only need to substitute (7.6) and (7.7) into the (rearranged) time derivative of (7.9) to obtain $\dot{v}_2 = \kappa(p, v)$. To improve

solver performance, we therefore set the damping coefficients to $d_1 = 0, d_2 = 0.0001$. With the new algebraic equation of index one, we need to make sure that the chosen initial values are consistent. For the parameters from above, we get a set of consistent initial values by $p_1 \approx 4.275, p_2 \approx 15, v_1 = 0, v_2 \approx 3.55 \cdot 10^4$, where v_1 is arbitrary with respect to consistency. A simple way to find different initial values, close to some which are given but inconsistent, is implemented in the code. Although this consistency is not required for arbitrary small $m_2 > 0$, we use the same (consistent) initial values for all values of m_2 for comparison.

We cannot directly compare the different ODE solutions to the respective DAE solutions at the computed time steps due to step size control and the resulting irregular time points, therefore we explicitly implemented the respective collocation functions $\tilde{g}, \tilde{l}$ for both solvers with three stages. For a GLC integration step from t_i to t_{i+1}, we thus have $\tilde{g}|_{[t_i,t_{i+1}]} \in \mathbb{R}^4[t]_{\leq 3}$, and analogously $\tilde{l}|_{[\tilde{t}_i,\tilde{t}_{i+1}]} \in \mathbb{R}^4[t]_{\leq 3}$ for an LIIIC integration step $\tilde{t}_i$ to $\tilde{t}_{i+1}$.

To measure the difference between an ODE solution $\tilde{x}$ and the respective DAE solution $\tilde{x}_0$, we evaluate both collocation functions at r equidistant time points $t_k \in [0, t_{\text{end}}]$ and average over the relative differences. For one variable y with respective component functions $\tilde{x}_y, \tilde{x}_{0y}$, this yields

$$\delta_y^{\tilde{x},\tilde{x}_0} := \frac{1}{r} \sum_{k=1}^{r} \left| \frac{\tilde{x}_y(t_k) - \tilde{x}_{0y}(t_k)}{\tilde{x}_{0y}(t_k)} \right|,$$

and the averaged relative difference for all four variables is given by

$$\delta(\tilde{x}, \tilde{x}_0) := \frac{1}{4} \Big(\delta_{p_1}^{\tilde{x},\tilde{x}_0} + \delta_{p_2}^{\tilde{x},\tilde{x}_0} + \delta_{v_1}^{\tilde{x},\tilde{x}_0} + \delta_{v_2}^{\tilde{x},\tilde{x}_0} \Big).$$

Using this way of measurement, we compute $\delta(\tilde{g}_m, \tilde{g}_0), \delta(\tilde{l}_m, \tilde{l}_0)$ as well as $\delta_y^{\tilde{g}_m,\tilde{g}_0}, \delta_y^{\tilde{l}_m,\tilde{l}_0}$ for $y \in \{p_1, p_2, v_1, v_2\}$, where $\tilde{g}_m, \tilde{l}_m$ are the collocation functions for $m = m_2 > 0$.

Since we expect better performance for the DAE system from the stiffly accurate LIIIC solver, we assume that for decreasing mass m_2 the GLC solution $\tilde{g}_m$ tends towards the solution $\tilde{l}_0$ rather than $\tilde{g}_0$, which is why we also compute $\delta(\tilde{g}_m, \tilde{l}_0)$. Results are shown in figure 8, for an integration time $t_{end} = 30$, an error tolerance of 10^{-8} and a maximum number of 5000 iterations.

We can see in figures 8a, 8b, that the approaching ODE solution gets closer to the respective DAE solution for the Lobatto IIIC solver. For $m_2 > 0$, (7.9) is of the form

$$m_2 v_2 = -2k_2(p_2 - p_1 - r_2) + k_3(l - p_2 - r_3) - d_2 v_2.$$

For $0 < m_2 \ll 1$, the variable v_2 is therefore highly unstable and the main cause

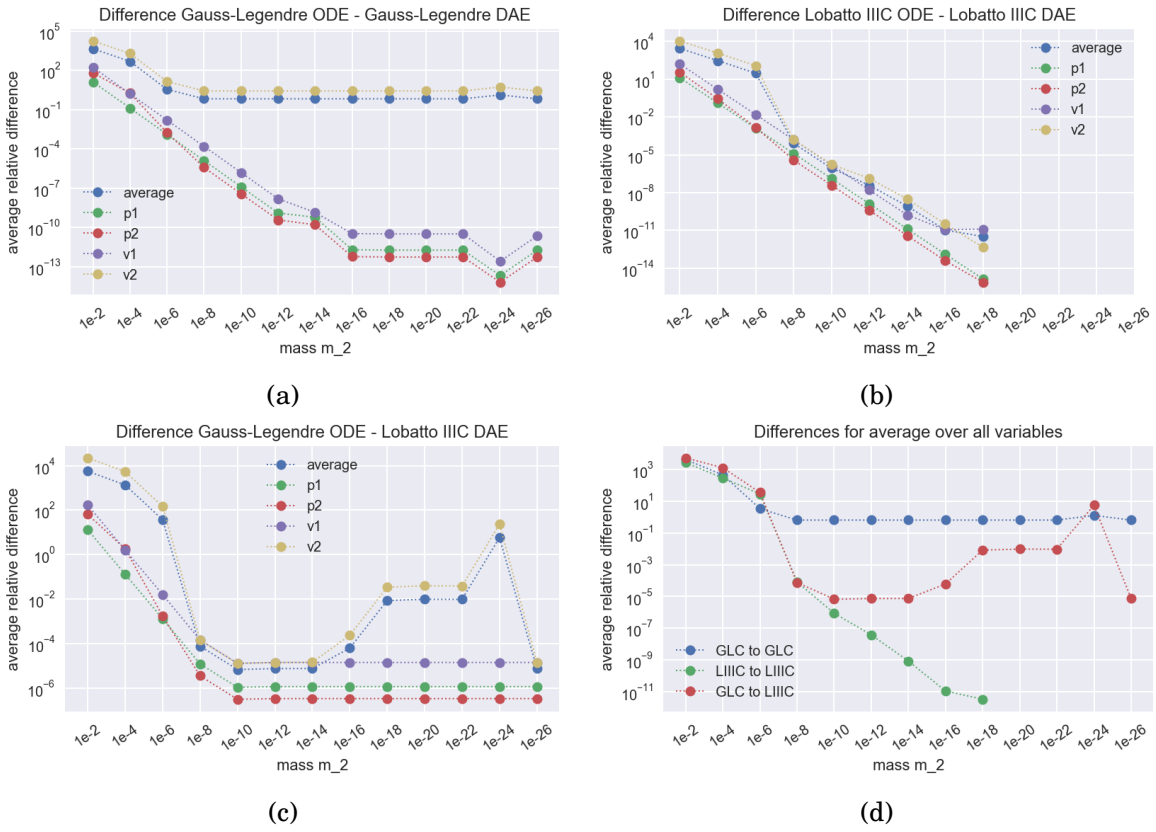


Figure 8: ODE solutions for decreasing m_2 compared to DAE solution for $m_2 = 0$

for the difference between ODE and DAE solutions of Gauss-Legendre collocation. For the Lobatto IIIC solver, this effect is also present but only slightly. We can also see, however, that the Lobatto IIIC solver was not able to terminate with the given parameters for $m_2 \leq 10^{-20}$ due to the limited number of iterations and exceedingly small step sizes (this is indicated in figure 8) by missing data points. This again points to the fact that the stiffly accurate LIIIC might be better fit for the DAE system, but the ODE system, even for very small m_2, is here better solved with GLC.
The most interesting result from figures 8c and 8d is that the GLC ODE solution indeed tends towards the LIIIC DAE solution rather than the GLC DAE solution, except for one outlier for $m_2 = 10^{-24}$. The difference is particularly small for $10^{-8} \geq m_2 \geq 10^{-16}$, where the greater distance for smaller values of m_2 can possibly be explained by the increasingly ill-conditioned system and resulting numerical inaccuracy. Similar results are obtained for varying initial values and tolerances, as well as damping and spring coefficients.

In summary, both solvers perform well for all given test problems. As expected, Gauss-Legendre collocation correctly conserves energy and the power balance equation, while the Lobatto IIIC solver shows slight inaccuracies. Moreover, our findings

suggest that Gauss-Legendre collocation is preferable when the mass matrix is ill-conditioned but invertible, while Lobatto IIIC should be used when one mass is equal to zero.

7.3 Partitioned Runge-Kutta methods

In this section, we carry out numerical simulations with the partitioned Runge-Kutta methods from section 6: The Lobatto IIIA-IIIB pair as described by Jay (J-PRK) in [19] and the method by Murua (M-PRK) from [28], also including the Gear-Gupta-Leimkuhler formulation from [14] for the latter method (M-PRK-GGL). All simulations use three stages unless stated otherwise. We want to compare the methods with respect to energy conservation as well as the constraints on positional and on velocity level. Also, we are interested in the case that one mass tends towards zero. To investigate these questions, we model a simple and a double pendulum with point masses and stiff, massless rods, without energy dissipation or input, as constrained multibody systems.

For a simple pendulum with point mass m and a stiff, massless rod of length ℓ, we have already derived the equations of motion and the resulting multibody system in section 4.4. For the double pendulum, we can proceed analogously to obtain the equations of motion in reduced order as

$$\dot{x}_1 = u_1 \tag{7.10}$$
$$\dot{y}_1 = v_1 \tag{7.11}$$
$$\dot{x}_2 = u_2 \tag{7.12}$$
$$\dot{y}_2 = v_2 \tag{7.13}$$
$$m_1 \dot{u}_1 = \lambda_1 x_1 + \lambda_2 (x_1 - x_2) \tag{7.14}$$
$$m_1 \dot{v}_1 = \bar{g} m_1 + \lambda_1 y_1 + \lambda_2 (y_1 - y_2) \tag{7.15}$$
$$m_2 \dot{u}_2 = \lambda_2 (x_2 - x_1) \tag{7.16}$$
$$m_2 \dot{v}_2 = \bar{g} m_2 + \lambda_2 (y_2 - y_1) \tag{7.17}$$

with holonomic constraints $0 = g(p)$ given by

$$0 = x_1^2 + y_1^2 - \ell_1^2 \tag{7.18}$$
$$0 = (x_2 - x_1)^2 + (y_2 - y_1)^2 - \ell_2^2 \tag{7.19}$$

and the total energy by the Hamiltonian

$$\begin{aligned}\mathcal{H}(p,\tilde{v},\lambda) &= T(\tilde{v}) + U(p)\\ &= \frac{1}{2}m_1(u_1^2+v_1^2)) + \frac{1}{2}m_2(u_2^2+v_2^2)) - \bar{g}m_1(y_1+l_1) - \bar{g}m_2(y_2+l_1+l_2)),\end{aligned}$$

where $p = (x_1, x_2, y_1, y_2), \tilde{v} = u_1, v_1, u_2, v_2$.
To apply the M-PRK method, the holonomic constraints $g(p)$ are replaced by their time derivative $G(p)v$, so (7.18), (7.19) are replaced by the index-2 velocity constraints

$$0 = x_1u_1 + y_1v_1 \tag{7.20}$$

$$0 = (x_1-x_2)u_1 + (y_1-y_2)v_1 + (x_2-x_1)u_2 + (y_2-y_1)v_2. \tag{7.21}$$

For the Gear-Gupta-Leimkuhler formulation, both the index-3 and index-2 constraints are in place, so we use equations (7.10) to (7.21). Further, $\dot{p} = \tilde{v}$ is replaced by $\dot{p} = \tilde{v} + G(p)^T\mu$, so (7.10) to (7.13) become

$$\begin{aligned}\dot{x}_1 &= u_1 + \mu_1x_1 + \mu_2(x_1-x_2)\\ \dot{y}_1 &= v_1 + \mu_1y_1 + \mu_2(y_1-y_2)\\ \dot{x}_2 &= u_2 + \mu_2(x_2-x_1)\\ \dot{y}_2 &= v_2 + \mu_2(y_2-y_1),\end{aligned}$$

with two additional Lagrange multipliers μ_1, μ_2.
The J-PRK method solves the system with the holonomic index 3 constraints, but then adapts the last stage Λ_s of the Lagrange multiplier λ such that the numerical solution satisfies the velocity constraints. Therefore, this solver also uses equations (7.10) to (7.21).
Before we present numerical results, we want to note that all three solvers are more sensitive to changes of system parameters, such as masses, rod lengths, initial values, and of solver parameters, such as maximum step size, tolerance, number of stages, than the (non-partitioned) Runge-Kutta methods from the previous section. This concerns the solution quality, but in particular the required accuracy to keep tolerances, such that a long integration time is not always feasible due to exceedingly small step sizes and resulting computation times. Naturally, the simple pendulum is much less affected than the chaotic double pendulum.

7.3.1 Energy conservation

First, we take a look at the M-PRK method. In section 6.3.4 we have derived an exact formula (6.15) for the energy change between two numerical integration steps. We want to verify this result for the case of the simple pendulum and one stage. We get the

predicted energy change from t_i to t_{i+1} as

$$\delta_E^{(p)}(t_i) := h_i \Lambda_i G(P_i) \tilde{V}_i = h_i \Lambda_i (X_i U_i + Y_i V_i)$$

and the actual energy change of the numerical solution as

$$\delta_E^{(n)} := \mathcal{H}(\tilde{x}_{i+1}) - \mathcal{H}(\tilde{x}_i),$$

where $\tilde{x}_i$ is the numerical solution at t_i. Figure 9 shows $\delta_E^{(p)}, \delta_E^{(n)}$ and $\delta_E^{(p)} - \delta_E^{(n)}$ for an integration time of $t_{end} = 30$ and initial values $[\ell,\ 0,\ \ 0,\ 0,\ 0]$. Since the predicted change $\delta_E^{(p)}(t_i)$ at t_i depends on the current step size h_i, it is included in plots 9a and 9b. As expected, $\delta_E^{(p)}$ grows with h. To make the effect more visible, we have chosen a small initial step size of 10^{-5}, a maximum step size of 0.1 and an adaptation factor of 1.5, enforcing a slow growth of step size. We can see in figure 9c that the difference $\delta_E^{(p)} - \delta_E^{(n)}$ indeed vanishes to the order of 10^{-9}, which we attribute to numerical inaccuracies.
We also implemented two different approaches for the initial guess used to solve the non-linear system of equations in each step. The 'naive' guess simply uses 0 for the variables $\dot{P}, \dot{V}$ and the current solution for all other variables, for all stages. The 'educated' guess uses the current solution (p, v, λ) to estimate $[\dot{P}_1\ \ \dot{V}_1]^T := f(p, v, \lambda)$ and then estimates $[P_1\ \ V_1]^T := [p\ \ v]^T + c_1 h [\dot{P}_1\ \ \dot{V}_1]^T$, where c_1 is the first Gauss quadrature node. In the same way, we successively compute the remaining stages, and analogously the variables $\bar{P}, \bar{V}, \dot{\bar{P}}, \dot{\bar{V}}$. For $\Lambda_1, ..., \Lambda_s$ we again use the current solution λ. Figure 9d shows the system of equations evaluated at the initial guesses, again for the simple pendulum, but with three stages and a larger step size growth rate. While the educated guess is clearly an improvement, we unfortunately had to acknowledge that there is little to no improvement in solver performance for all tested cases, while computation times rise notably in some. We have therefore decided to not implement a similar option in the other solvers and only use the 'naive' guess for the M-PRK solver for all simulations.

For the Gear-Gupta-Leimkuhler formulation, we expect a similar quality of energy conservation, but less predictabilty. The additional equations and Lagrange multipliers introduce further possible inaccuracies, while the holonomic constraint is explicitly given and therefore better overall performance can be expected.
For the J-PRK method, we recall that the Lobatto IIIA-IIIB pair is symplectic (for a proof see [17], chapter VII). We thus expect good energy conservation up to numerical errors. However, there is no exact energy conservation, since PRK methods only preserve general quadratic invariants for unconstrained systems ([17], chapter IV). For holonomic constraints, a more careful analysis is necessary which would go beyond the scope of this thesis, see for example [22].

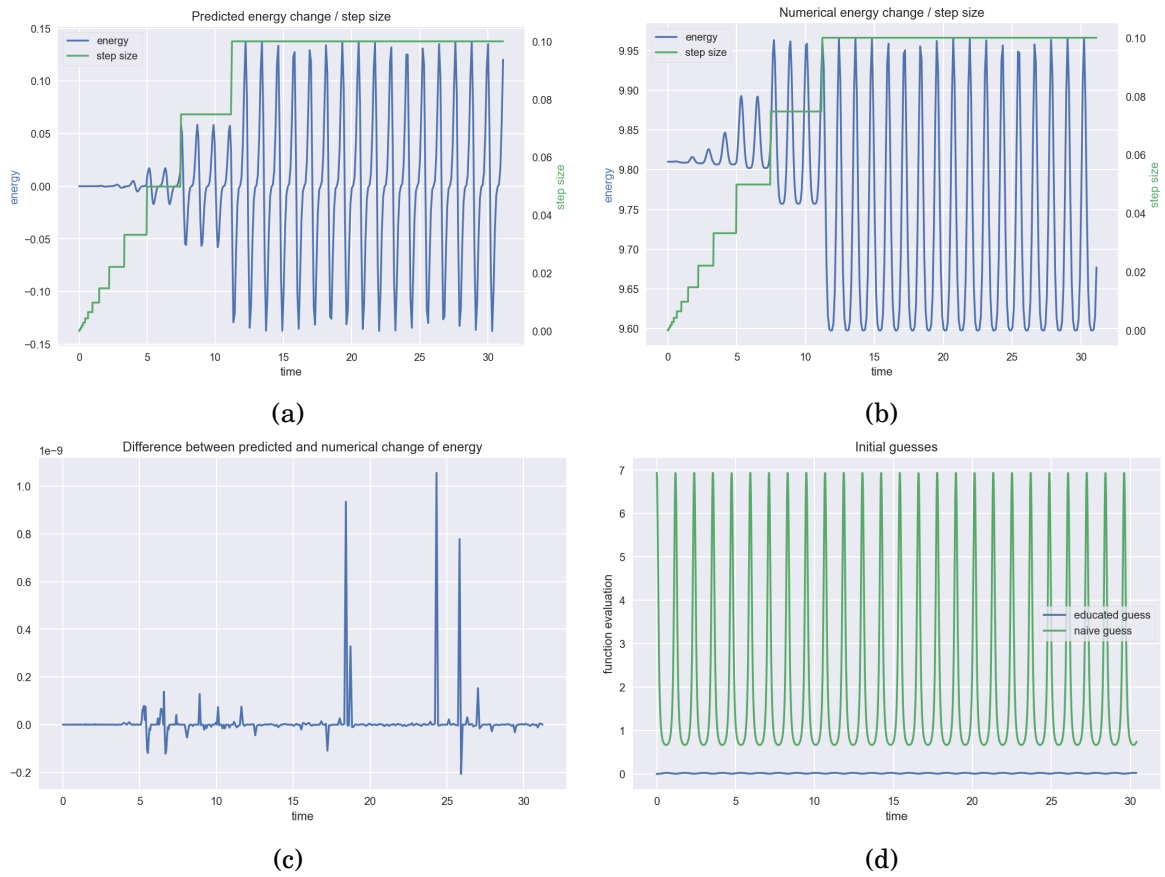
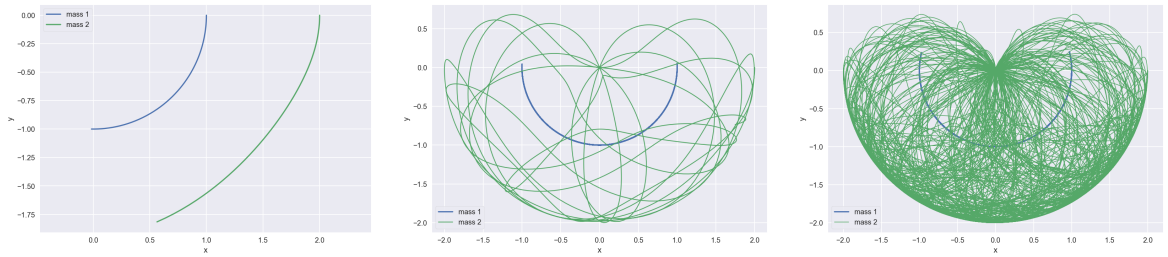


Figure 9: M-PRK method, (a)-(c): predicted and numerical changes of energy for the simple pendulum, (d): function evaluation for different initial guesses

To compare long time energy behaviour, we have integrated both the simple pendulum and the double pendulum with $t_{end} = 240$. For both simulations, we set maximum step sizes to 0.01, tolerances to 10^{-8}, and all rod lengths to 1. Both pendulums set out on the positive x-axis with zero velocities, so initial values for the simple pendulum are $[\ell,\ 0,\ 0,\ 0,\ 0]$ as above and $[\ell_1,\ 0,\ \ell_1 + \ell_2,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0]$ for the double pendulum. For the GGL formulation, the necessary zeros are added for the extra Lagrange multiplier(s). The mass of the simple pendulum is $m = 1$; to make the double pendulum motion more interesting, we set $m_1 = 3, m_2 = 1$. To show that our solvers actually carry out the desired task, figure 10 shows the positional plot for the double pendulum, at three different integration times. The positional plots look almost identical for all solvers, the solution here is from the M-PRK solver and shows the expected chaotic yet symmetrical behaviour.

Figure 11 shows the energy development. We can see that all solvers perform well for both test problems, with changes in energy on the order of 10^{-8} (simple pendulum) and 10^{-5} (double pendulum). For the simple pendulum (11a), the J-PRK method produces strong oscillations in the energy, but around a certain 'average' value - in the figure, line width and opacity have been adapted to make it visible against the other solutions.



(a) 200 iterations (b) 2000 iterations (c) 24151 iterations

Figure 10: Double pendulum position development

However, this average value also shows a slight decrease over time. This is unexpected and can so far only be explained by numerical errors. For the M-PRK solver, we see a very small but constant linear increase in energy, while the M-PRK-GGL solver portrays rather random changes in energy.

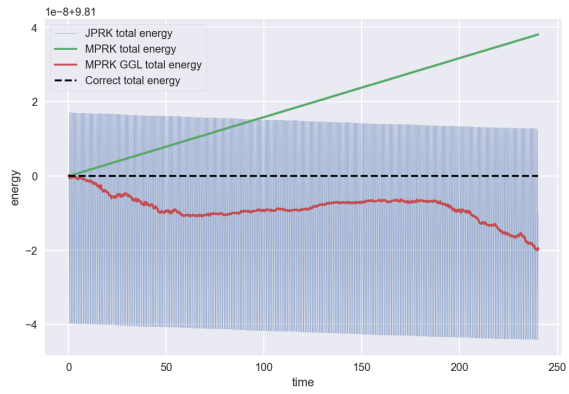
Figure 11b shows the results for the double pendulum. We can see a similar behaviour of the J-PRK solver, but this time without an energy drift. The oscillations are of greater magnitude though, and in this respect both other methods outperform the J-PRK solver. To make the solutions visible, figure 11c shows the results for the double pendulum, but only for the M-PRK and M-PRK-GGL solver, which show changes only on the order 10^{-7}. We can again see a steady linear change in energy for M-PRK, this time a decrease, and again unsystematic changes for M-PRK-GGL. Figure 11d shows how the energy is composed of kinetic and potential energy, limited to the first 1500 integration steps. The plot uses the data from the M-PRK solver, but at this scale there is no visible difference between all three solvers.

These findings are similar for different parameters and initial values, but as already mentioned, less stable settings can easily lead to poorer performance, also with respect to energy conservation. In particular, energy might only be well-preserved when very small step sizes are used.

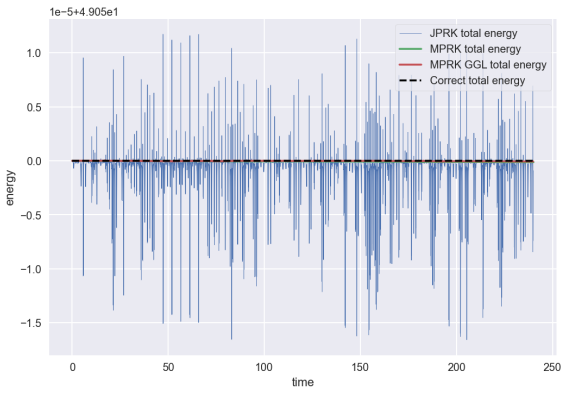
7.3.2 Constraints

For the positional and velocity constraints, we expect different behaviour from all three solvers:

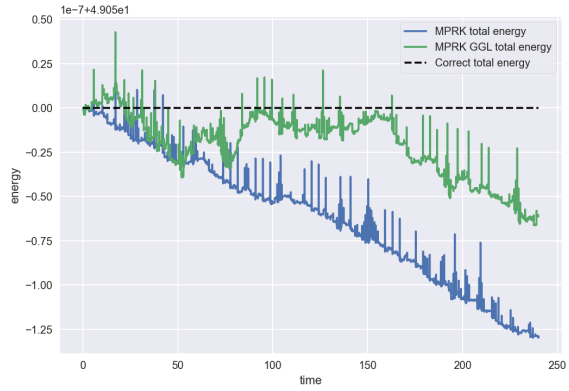
- M-PRK: Maintains velocity constraints, but is prone to numerical drift for the position variables, since the holonomic constraints are not explicitly formulated
- M-PRK GGL: Also maintains the velocity constraints, supposedly on a similar or better level than M-PRK since we expect better overall accuracy. Also maintains the holonomic constraints so there is no numerical drift
- J-PRK: Maintains holonomic constraints. In every step, the solution is projected



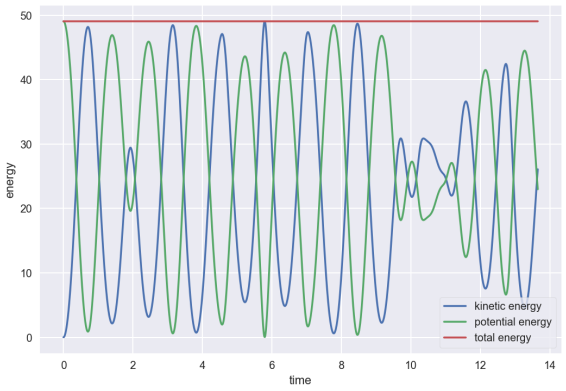
(a) Simple pendulum



(b) Double pendulum



(c) Double Pendulum M-PRK and M-PRK-GGL

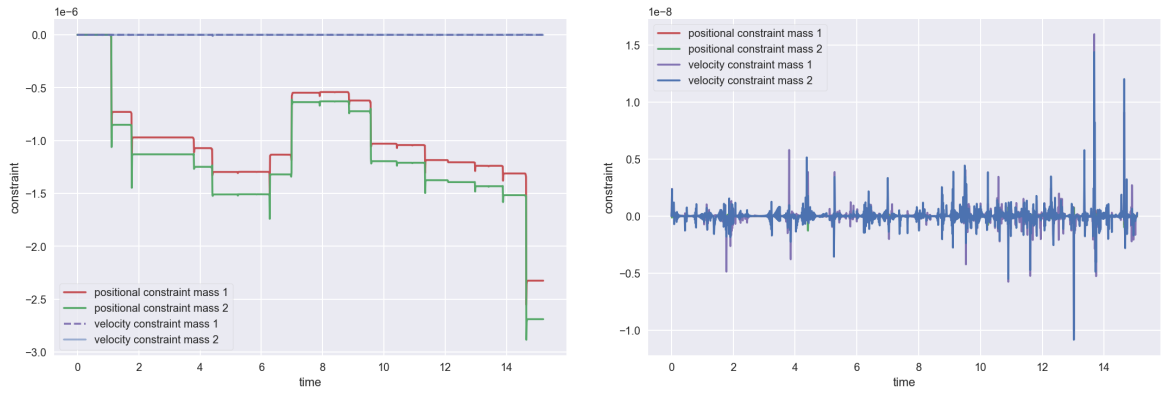


(d) Energy composition

Figure 11: Total energy over time

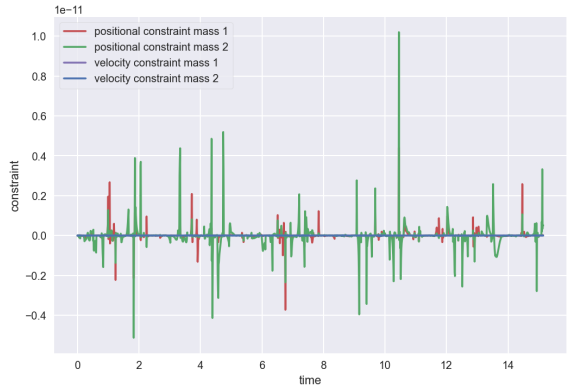
onto the velocity constraint manifold, so these should be satisfied exactly

To test these assumptions and to run our double pendulum model with different settings, we use the somewhat arbitrary parameters $m_1 = 1, m_2 = 8, l_1 = 1, l_2 = 5$ and add an initial velocity of $u_1 = 10$ to the initial values previously used. Of course, we also tested the constraint accuracies for the simple pendulum using the previous setup, but the overall results are similar and we decided to only show the double pendulum results here. We can see the expected results in figure 12, in particular the numerical drift in the holonomic constraints for the M-PRK method in 12a. The inclusion of the Gear-Gupta-Leimkuhler formulation both prevents the numerical drift and improves overall accuracy, see figure 12b. The holonomic constraints are of order 10^{-9} and therefore not visible in the figure. However, the velocity constraints are near identical for both methods (12d). The J-PRK method shows the best performance, with an accuracy of order 10^{-11} for the holonomic constraints and the index-2 constraints exactly satisfied due to the projection step, which are of order 10^{-14} (not visible in the figure).

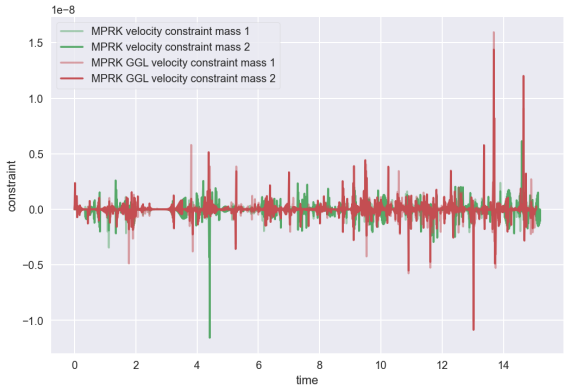


(a) M-PRK

(b) M-PRK-GGL



(c) J-PRK



(d) M-PRK (GGL) velocity constraint

Figure 12: Double pendulum constraints

7.3.3 One mass approaches zero

Finally, we again want to test our solvers for the case that one of the masses tends towards zero, such that the corresponding differential equations become algebraic equations in the limit. Naturally, the solutions for an ever decreasing mass should approach the solution for the mass being exactly zero, but since the mass matrix becomes increasingly ill-conditioned, this is not necessarily reflected in the numerical results. For $m_1 = 0$, equations (7.14), (7.15) become

$$0 = \lambda_1 x_1 + \lambda_2 (x_1 - x_2)$$
$$0 = \lambda_1 y_1 + \lambda_2 (y_1 - y_2).$$

Completely analogous to the mass-spring-damper system in 7.2.3, we can see that the resulting algebraic equations are of index two. Again, we reduce the index by introducing a slight energy dissipation. We achieve this by adding viscosity $\kappa > 0$ via

$$0 = \lambda_1 x_1 + \lambda_2 (x_1 - x_2) - \kappa u_1$$
$$0 = \lambda_1 y_1 + \lambda_2 (y_1 - y_2) - \kappa v_1.$$

The reasoning for the second mass is equivalent, and for simplicity we always use one global viscosity. For the remaining computations, we thus have (7.14) to (7.17) in the form

$$\begin{aligned} m_1\dot{u}_1 &= \lambda_1 x_1 + \lambda_2(x_1 - x_2) - \kappa u_1 \\ m_1\dot{v}_1 &= \bar{g}m_1 + \lambda_1 y_1 + \lambda_2(y_1 - y_2) - \kappa v_1 \\ m_2\dot{u}_2 &= \lambda_2(x_2 - x_1) - \kappa u_2 \\ m_2\dot{v}_2 &= \bar{g}m_2 + \lambda_2(y_2 - y_1) - \kappa v_2. \end{aligned}$$

We use the parameters $\ell_1 = \ell_2 = 0, \kappa = 0.01$ and the same initial values as before, so $[\ell_1,\ 0,\ \ell_1 + \ell_2,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0]$ (again appending another two zeros for the GGL formulation). It is easy to see that these are consistent with the algebraic equations arising from either mass being zero.

Similar to section 7.2.3, we need a way of measuring the difference between discrete solutions with different time points due to varying step sizes. To avoid the rather tedious implementation of the piecewise collocation polynomials for the more elaborate PRK methods, we use the scipy interpolation package to obtain piecewise linear interpolation functions for the numerical solution of each variable over the integration interval $[0, t_{end}]$. From here, the procedure is analogous to before: We choose r equidistant time points $t_k \in [0, t_{end}]$ and average the relative differences for the component functions of each variable, where relative is with respect to the reference solution $\tilde{x}_0$ with $m_1 = 0$ $(m_2 = 0)$. We thus get

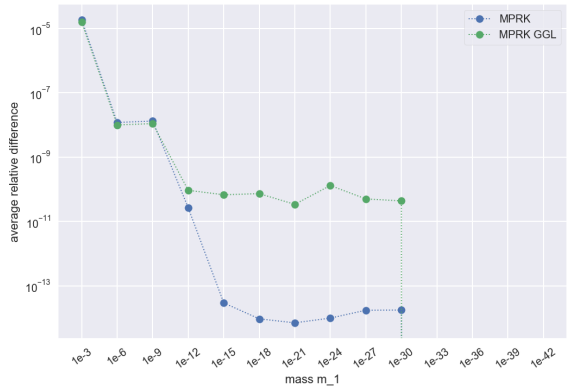
$$\delta^{(1)}(\tilde{x}, \tilde{x}_0) := \frac{1}{10}\left(\delta_{x_1}^{(1)\tilde{x},\tilde{x}_0} + \delta_{y_1}^{(1)\tilde{x},\tilde{x}_0} + ... + \delta_{\lambda_2}^{(1)\tilde{x},\tilde{x}_0}\right)$$

for the first mass and likewise $\delta^{(2)}(\tilde{x}, \tilde{x}_0)$ for the second mass. The GGL formulation includes the additional two Lagrange multipliers μ_1, μ_2. Now, we again compute $\delta^{(1)}(\tilde{x}_{m_1}, \tilde{x}_0)$ for decreasing masses m_1 and $\delta^{(2)}(\tilde{x}_{m_2}, \tilde{x}_0)$ for decreasing masses m_2. In both cases, the other mass is always set equal to 1. Since this leads to many time consuming computations, we only compute solutions for $m = 10^-$ Unfortunately, the J-PRK solver cannot handle this task well: It needs to use exceedingly small step sizes already for one mass on the order of 10^{-6} and barely makes any progress for even smaller masses. For the case where one mass is exactly 0, the solution also barely advances. This might be due to poor implementation or undiscovered errors from our side, but it leads us to believe that the Lobatto IIIA-IIIB pair might be less suited for problems with ill-conditioned mass matrix, at least not without further precautions in the implementation. We therefore only look at the M-PRK solver for this section, both with and without the Gear-Gupta-Leimkuhler formulation.

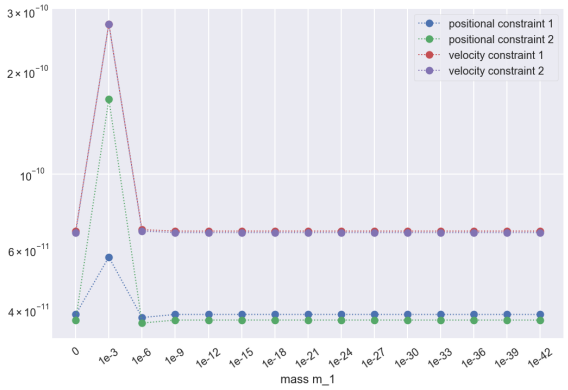
We thus compute, for decreasing mass m_1,

$$\delta^{(1)}(\tilde{\nu}_m, \tilde{\nu}_0) \quad \text{and} \quad \delta^{(1)}(\tilde{\omega}_m, \tilde{\omega}_0),$$

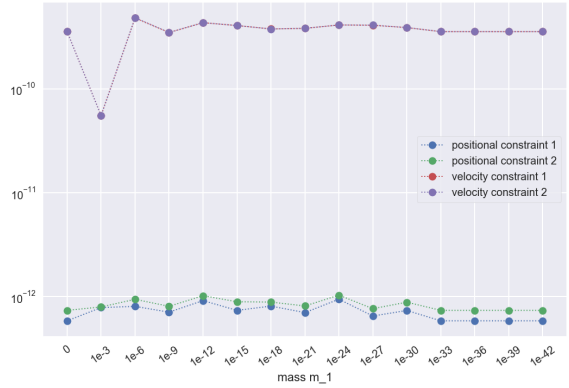
where $\tilde{\nu}_m, \tilde{\omega}_m$ are the numerical solutions for the M-PRK and M-PRK GGL method with $m_1 = m$ and $\tilde{\nu}_0, \tilde{\omega}_0$ are the numerical solutions for $m_1 = 0$. We proceed analogously for decreasing mass m_2. Further, we would like to know how well the solvers maintain the four (original) algebraic constraint equations for these rather challenging cases. We therefore also compute the average absolute value of the individual constraint equations for each solution. For the case that m_1 (m_2) is exactly zero, a new algebraic constraint is introduced; of course, this constraint is not present for any value $m_1 > 0$ $(m_2 > 0)$ but for ever smaller masses we still expect the respective equation to have small values. We therefore also compute their average absolute value for each solution, which are the right hand side of equations (7.14), (7.15) for mass 1 (equations (7.16), (7.17) for mass 2). Figures 13 and 14 show the results for mass 1 and mass 2.



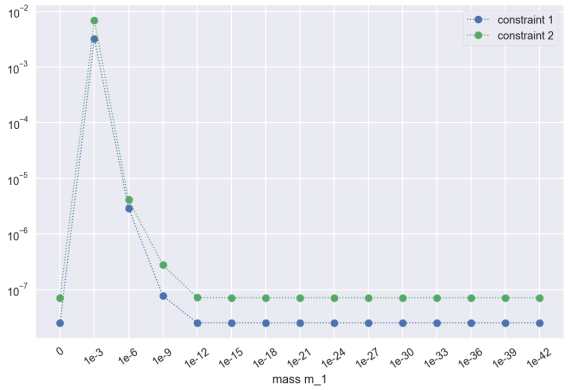
(a) Difference to solution with $m_1 = 0$



(b) Constraints M-PRK



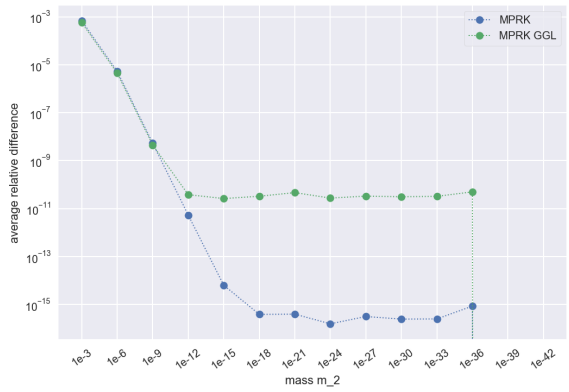
(c) Constraints M-PRK-GGL



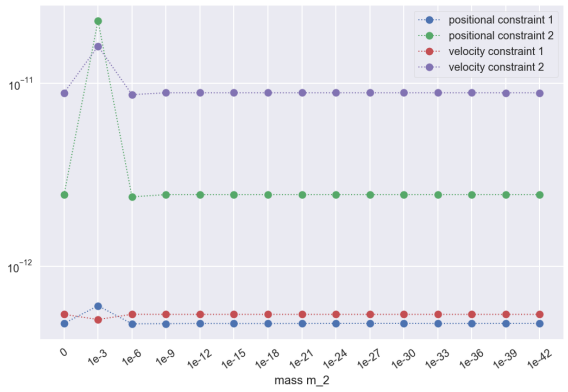
(d) Mass constraint M-PRK

Figure 13: Solutions for decreasing values of m_1. (a) shows how they approach the solution with $m_1 = 0$, (b)-(d) show the constraints

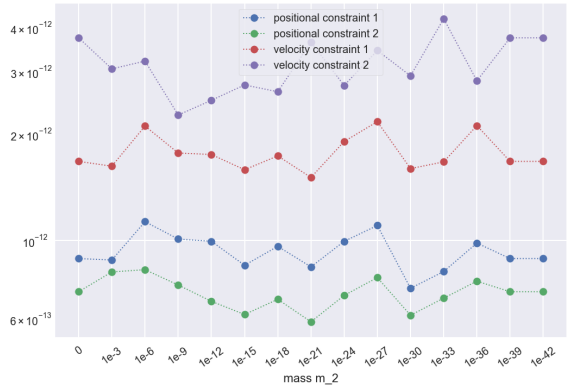
As expected, both M-PRK and M-PRK-GGL solutions approach their respective reference solution for $m_1 = 0$ $(m_2 = 0)$ at a similar rate, see 13a and 14a. Most notably, the



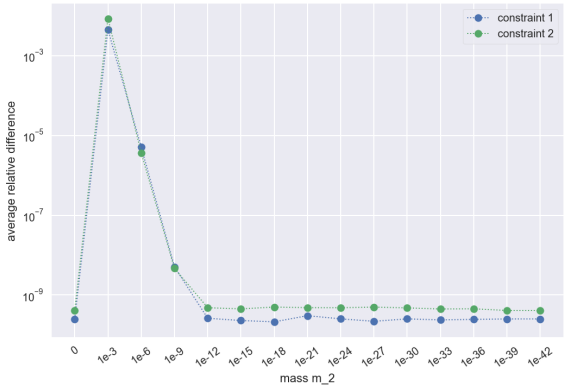
(a) Difference to solution with $m_2 = 0$



(b) Constraints M-PRK



(c) Constraints M-PRK-GGL



(d) Mass constraint M-PRK

Figure 14: Solutions for decreasing values of m_2. (a) shows how they approach the solution with $m_1 = 0$, (b)-(d) show the constraints

solutions are equal to the reference solution with mass zero for masses smaller than 10^{-30} (mass 1) and 10^{-36} (mass 2). This is a satisfying result, as it shows that the solution does not show qualitatively bad behaviour for highly ill-conditioned mass matrices, but rather approaches the respective reference function, until differences vanish completely due to machine precision. This is indicated in the plots by a vertical line, since there is no way to represent 0 in logarithmic scale.

Surprisingly, the solutions' approach stagnates for the GGL formulation for masses smaller than 10^{-12}, for both m_1 and m_2. Of course, this does not say anything about the solution quality since we do not know the correct analytical solutions, but the numerical behaviour of the M-PRK solver is preferable in this regard.

For the constraints, note that the first entry is always produced by the solution for $m_1 = 0$ ($m_2 = 0$). We can see in figures 13b, 13c (mass 1) and 14b, 14c (mass 2) that they are well satisfied, between 10^{-10} and 10^{-13}. The GGL formulation shows slighlty better results, especially with regard to the positional constraints as is usually the case. Figures 13d and 14d show the 'pseudo' constraint equations described above (mass constraints in the figure), which result from one mass being equal to zero. We can see in

the first entries that both solvers managed to satisfy the respective constraint well for $m_1 = 0$ $(m_2) = 0$. For $m_1 > 0$ $(m_2 > 0)$, we are evaluating the right hand side of a differential equation, which, as expected, decreases in absolute value with the respective mass as it approaches an algebraic equation. We have used the data from the M-PRK solver in figure 13d and from the M-PRK-GGL solver in figure 14d, both solvers produce almost the same result for both masses.
For the previous computations concerning energy conservation and constraint accuracy, we have compared the computation times on our private laptop, which show that the J-PRK method was faster than the other two methods. The M-PRK method took slightly longer when the GGL formulation is included, see the following table:

	J-PRK	M-PRK	M-PRK-GGL
simple pendulum energy conservation	60.0	134.7	155.5
simple pendulum constraint accuracy	34.3	34.5	40.9
double pendulum energy conservation	103.6	258.6	300.1
double pendulum constraint accuracy	10.5	18.1	21.0

Overall, the findings from this section are satisfactory since both implemented PRK solvers perform as expected and manage to simulate the behaviour of both the simple and the double pendulum well. Both energy conservation and the accurateness of algebraic constraints are at acceptable levels. In particular, the results for the M-PRK and M-PRK-GGL method for one mass approaching zero are promising in the sense that this solver might generally be well suited to solve problems with ill conditioned mass matrices. However, as already mentioned, some results are prone to change with different model and solver parameters. Also, it remains open why the Lobatto IIIA-IIIB pair in our J-PRK implementation did not show better and more stable behaviour. Presumably, there is an undiscovered error or weakness in our implementation, while on the other hand the solver performs qualitatively correct for all test problems that we applied it to.

8 Conclusions

In this thesis, we investigated multibody systems in the classic Euler-Lagrange formalism and their structural properties. We then studied the model class of port-Hamiltonian systems and saw how multibody systems can be incorporated into their framework, connecting the different traditions.
In the second part, we explored numerical methods and their application to multibody systems, and we presented simulation results from our implementations. We looked into classic Runge-Kutta methods, of which we implemented Gauss-Legendre collocation and the Lobatto IIIC method. We simulated an unconstrained two-body mass-spring-damper system in order to test energy conservation and the limiting case from ODE to DAE for one mass approaching zero. As expected, Gauss-Legendre collocation preserves energy, while the stiffly accurate Lobatto IIIC solver is prone to errors and not suited for the integration over long time periods. The decreasing mass was correctly reflected in the solutions of both solvers, which approach the DAE solution despite the structural gap. This encourages further research on the reliability of Gauss-Legendre collocation for unconstrained multibody systems with ill-conditioned mass matrix, to avoid the artificial creation of a DAE on the one hand, and, on the other, to benefit from the solver's desirable properties, such as the high order and symplecticity.
For the structure-preserving numerical integration of constrained multi-body systems, we looked at partitioned Runge-Kutta methods designed for the multibody system structure. We analysed and implemented the Lobatto IIIA-IIIB pair as well as the method suggested by Murua in [28], with the option to extend the latter by the Gear-Gupta-Leimkuhler formulation. We ran numerical simulations for the simple and double pendulum and found good energy conservation constraint accuracy from both methods. For the algebraic constraints, the highest accuracy was achieved by the Lobatto pair, while we observed the expected slight numerical drift in Murua's method. However, this could easily be prevented by using the GGL formulation. For the case of one mass approaching zero, the Lobatto pair soon failed to converge, while Murua's method had the solutions for a decreasing mass approach the solution of the mass being equal to zero. Our findings suggest to also consider the lesser known method by Murua for the numerical solution of multibody systems, in particular for the case of an ill-conditioned mass matrix.
In the interpretation of all numerical results, we have to take the simplicity of the simulated models into account, as well as the quality of our implementation. Both factors might impair the generalisability of our findings to more complex problems and the highly advanced implementations of industrial solvers. Nevertheless, we hope that this work can be a small contribution to energy-based modelling, further connecting

the fields of multibody systems and port-Hamiltonian systems, and to inspire further research and the design of adapted numerical methods.

References

[1] F. Amirouche. *Fundamentals of Multibody Dynamics. Theory and Applications*. Birkhäuser, 2006.

[2] H. C. Andersen. "Rattle: A 'velocity' version of the shake algorithm for molecular dynamics calculations". In: *Journal of Computational Physics* 52.1 (1983), pp. 24–34.

[3] M. Arnold, V. Mehrmann, and A. Steinbrecher. "Index Reduction in Industrial Multibody System Simulation". Preprint Matheon. 2004.

[4] C. Beattie et al. *Port-Hamiltonian descriptor systems*. 2017. arXiv: 1705.09081.

[5] B. Benhammouda. "A New Numerical Technique for Index-3 DAEs Arising from Constrained Multibody Mechanical Systems". In: *Discrete Dynamics in Nature and Society* 2022.1 (2022).

[6] K. Brenan, S. Campbell, and L. Petzold. *Numerical Solution of Initial-value Problems in Differential-algebraic Equations*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, 1996.

[7] A. Brizard. *An Introduction to Lagrangian Mechanics*. Reference, Information and Interdisciplinary Subjects Series. World Scientific, 2008.

[8] J. C. Butcher. "On Runge-Kutta processes of high order". In: *Journal of the Australian Mathematical Society* 4.2 (1964), pp. 179–194.

[9] S. Campbell and P. Kunkel. "Solving higher index DAE optimal control problems". In: *Numerical Algebra, Control and Optimization* 6.4 (2016), pp. 447–472.

[10] V. Duindam et al. *Modeling and Control of Complex Physical Systems. The Port-Hamiltonian Approach*. Springer, 2009.

[11] E. Eich-Soellner and C. Führer. *Numerical Methods in Multibody Dynamics*. Springer, 1998.

[12] C. L. Ernst Hairer Michel Roche. *The Numerical Solution of Differential-Algebraic Systems by Runge-Kutta Methods*. Lecture Notes in Mathematics. Springer, 1989.

[13] C. Gear. "The Simultaneous Numerical Solution of Differential-Algebraic Equations". In: *Circuit Theory, IEEE Transactions on* CT18 (1971), pp. 89–95.

[14] C. Gear, B. Leimkuhler, and G. Gupta. "Automatic integration of Euler-Lagrange equations with constraints". In: *Journal of Computational and Applied Mathematics* 12-13 (1985), pp. 77–90.

[15] H. Goldstein. *Classical Mechanics*. Addison-Wesley series in advanced physics. Addison-Wesley, 1950.

[16] E. Hairer. “Backward Analysis of Numerical Integrators and Symplectic Methods”. In: *Annals of Numerical Mathematics* 1 (1994), pp. 107–132.

[17] E. Hairer, C. Lubich, and G. Wanner. *Geometric numerical integration*. 2nd ed. Vol. 31. Springer Series in Computational Mathematics. Structure-preserving algorithms for ordinary differential equations. Springer, 2006.

[18] L. Jay. “Runge–Kutta type methods for index three differential-algebraic equations with applications to Hamiltonian systems”. PhD thesis. University of Geneva [VII.1], 1994.

[19] L. Jay. “Symplectic Partitioned Runge-Kutta Methods for Constrained Hamiltonian Systems”. In: *SIAM Journal on Numerical Analysis* 33.1 (1996), pp. 368–387.

[20] M. Knorrenschild. “Differential/Algebraic Equations As Stiff Ordinary Differential Equations”. In: *SIAM Journal on Numerical Analysis* 29.6 (1992), pp. 1694–1715. eprint: https://doi.org/10.1137/0729096.

[21] P. Kunkel and V. L. Mehrmann. *Differential-algebraic equations*. EMS Textbooks in Mathematics. European Mathematical Society, 2006.

[22] L. Li and D. Wang. *Energy and quadratic invariants preserving methods for Hamiltonian systems with holonomic constraints*. 2020. arXiv: 2007.06338.

[23] B. Maschke and A. van der Schaft. “Port-Controlled Hamiltonian Systems: Modelling Origins and Systemtheoretic Properties”. In: *IFAC Proceedings Volumes* 25.13 (1992). 2nd IFAC Symposium on Nonlinear Control Systems Design 1992, Bordeaux, France, 24-26 June, pp. 359–365.

[24] V. Mehrmann and A. J. van der Schaft. *Differential-algebraic systems with dissipative Hamiltonian structure*. 2023. arXiv: 2208.02737.

[25] V. Mehrmann and R. Morandin. *Structure-preserving discretization for port-Hamiltonian descriptor systems*. 2019. arXiv: 1903.10451.

[26] V. Mehrmann and B. Unger. *Control of port-Hamiltonian differential-algebraic systems and applications*. 2022. arXiv: 2201.06590.

[27] R. Morandin. “Modeling and numerical treatment of port-Hamiltonian descriptor systems”. PhD thesis. Technical University Berlin, 2023.

[28] A. Murua. “Partitioned half-explicit Runge-Kutta methods for differential-algebraic systems of index 2”. In: *Computing* 59.1 (1997), pp. 43–61.

[29] S. Reich. “Symplectic integration of constrained Hamiltonian systems by Runge–Kutta methods”. In: *Techn. Report 93-13 (1993), Dept. Comput. Sci., Univ. of British Columbia. [VII.1]* (1993).

[30] J.-P. Ryckaert, G. Ciccotti, and H. Berendsen. “Numerical-Integration of Cartesian Equations of Motion of a System with Constraints – Molecular-Dynamics of N-Alkanes”. In: *Journal of Computational Physics* 23 (1977), pp. 327–341.

[31] A. van der Schaft. “Port-Hamiltonian systems: an introductory survey”. In: *Proceedings of the International Congress of Mathematicians Madrid, August 22–30, 2006*. EMS Press, 2007, pp. 1339–1365.

[32] A. van der Schaft and D. Jeltsema. “Port-Hamiltonian Systems Theory: An Introductory Overview”. In: *Foundations and Trends in Systems and Control* 1 (2014), pp. 173–378.