Technische Universität Berlin
Institut für Mathematik

Signature Method Based Regularization and Numerical Integration of DAEs

Andreas Steinbrecher[†] [‡]

Preprint 2016/01

Preprint-Reihe des Instituts für Mathematik
Technische Universität Berlin
http://www.math.tu-berlin.de/preprints

Report 2016/01 February 2016

Modeling and simulation of dynamical systems often leads to differential-algebraic equations (DAEs) which can be seen as differential equations, where every solution has to satisfy constraints which are contained in the DAE. In general not all these constraints are stated explicitly as equations or can be obtained by algebraic manipulations but are hidden in the DAE and can be obtained from certain derivatives of (parts of) the DAE. Due to those hidden constraints a direct numerical integration of DAEs in general leads to instabilities and possibly non-convergence of numerical methods. Therefore, a regularization or remodeling of the model equations is required. In this article we present three approaches for the regularization of DAEs that are based on the Signature method, which is a structural analysis for DAEs. Furthermore, we present a software package suited for the proposed regularizations and illustrate their efficiency on two examples.

AMS(MOS) subject classification: 34A09, 65L80

Keywords: DAE, hidden constraints, constraint level, structural analysis, signature method, regularization, QUALIDAES

Authors address:

Andreas Steinbrecher
Institut für Mathematik, MA 4–5,
Technische Universität Berlin
Str. des 17. Juni 136
10623 Berlin, Germany
`anst@math.tu-berlin.de`

Signature Method Based Regularization and Numerical Integration of DAEs

Andreas Steinbrecher[†‡]

February 5, 2016

Abstract

Modeling and simulation of dynamical systems often leads to differential-algebraic equations (DAEs) which can be seen as differential equations, where every solution has to satisfy constraints which are contained in the DAE. In general not all these constraints are stated explicitly as equations or can be obtained by algebraic manipulations but are hidden in the DAE and can be obtained from certain derivatives of (parts of) the DAE. Due to those hidden constraints a direct numerical integration of DAEs in general leads to instabilities and possibly non-convergence of numerical methods. Therefore, a regularization or remodeling of the model equations is required. In this article we present three approaches for the regularization of DAEs that are based on the Signature method, which is a structural analysis for DAEs. Furthermore, we present a software package suited for the proposed regularizations and illustrate their efficiency on two examples.

Keywords: DAE, hidden constraints, constraint level, structural analysis, signature method, regularization, QUALIDAES

AMS(MOS) subject classification: 34A09, 65L80

1 Introduction

The complete virtual design of dynamical systems plays a key role in our technological progress. Therefore, the interconnection of automatic modeling tools with efficient and robust simulation tools is of growing interest. The automatic modeling using coupling of modularized subcomponents is frequently used in industrial applications. Here, the coupling of subcomponents is usually described by algebraic constraints yielding large-scale (but often sparse) *differential-algebraic equations* (DAEs) as model equations, e.g., for multibody systems, electrical circuits, or flow problems.
In this article, we consider nonlinear DAEs of the form

$$F(t, z, \dot{z}) = 0, \tag{1}$$

with sufficiently smooth functions $F : \mathbb{I} \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^m$ and $z : \mathbb{I} \to \mathbb{R}^n$, with $t \in \mathbb{I}$ and a compact interval $\mathbb{I} \subset \mathbb{R}$.
Unfortunately, the direct numerical integration of DAEs in general is not feasible due to so called *hidden constraints* which restrict the solution but are not explicitly stated as equations or cannot be obtained only by algebraic manipulations. Hidden constraints usually lead to instabilities and possibly non-convergence of the numerical methods, e.g., see [6, 11, 13, 22]. Hidden constraints can be determined by algebraic manipulations of a certain number of differentiations of (parts of) the DAE. This number of differentiations gives a classification of the difficulties in the treatment of DAEs, and yields in different index concepts, e.g., the

[†] Institut für Mathematik, TU Berlin, Straße des 17. Juni 136, 10623 Berlin, Germany; anst@math.tu-berlin.de.

[‡] The author's work was supported by the ERC Advanced Grant "Modeling, Simulation and Control of Multi-Physics Systems" MODSIMCONMP

differentiation index (d-index) ν_d [6, 11], the strangeness index ν_s [13], the tractability index ν_t [14], the structural index ν_Σ [19], or the maximal constraint level ν_c [22].

To guarantee a stable and robust numerical integration, a regularization or remodeling of the considered DAE is required. The basic idea of an approach to obtain a regularization for high-index DAEs is to consider the original DAE together with a sufficient number of its derivatives (as a *derivative array*). Then a DAE, called *regularized DAE* or *regularization*, which has to be solution equivalent to the original DAE can be derived that now contains all the information on the manifold in which the dynamic of the system take place as algebraic equations. Thus, the regularized DAE does not contain further hidden constraints but contains the originally hidden constraints explicitly as algebraic equations. Often, the regularized DAE will consist of two parts, a purely algebraic part describing the manifold of constraints and a differential part describing the dynamic on this manifold. Since all constraints (including the originally hidden ones) are explicitly stated in the regularized DAE, it can be guaranteed that all these constraints are satisfied within the numerical solution (up to the accuracy that is used to solve these equations) and instabilities or drift from the solution manifold are avoided.

Several general *algebraic approaches* for the regularization of DAEs are developed, e.g., projection methods [14], Kunkel-Mehrmann formulation [13, 20], projected strangeness-free formulations [22], minimal extension [12]. However, the computational complexity for these general algebraic approaches is substantial, since from the derivative array certain nullspaces of the Jacobians and associated projectors onto these nullspaces have to be computed at every integration step. This makes algebraic approaches quite expensive for medium or large scale problems. For structured problems more specific regularization approaches are developed, in particular, the Gear-Gupta-Leimkuhler formulation for the model equations of multibody systems [9] and an element based approach for electrical circuits [4].

In addition to algebraic approaches so called *structural approaches* were developed, e.g., the *Pantelides algorithm* [18] or the *Signature method* (Σ*-method)* [19]. The structural approaches avoid the large computational effort necessary in algebraic approaches. Structural approaches are based on the sparsity pattern of the DAE, to obtain necessary information for a regularization. In these approaches generic structural information, together with certain computer-algebra packages or symbolic differentiation, is used to identify the constraints and interface conditions, as well as the dynamic equations, to determine the index of the system and to compute an index-reduced regularized system model. The advantage of a structural approach in comparison to classical algebraic regularization approaches is that fast algorithms based on graph theory can be applied. Usually, the Pantelides algorithm in combination with the *Dummy Derivative Method* [15] is used in many modeling and simulation tools as the current state-of-the-art to regularize high-index DAEs.

In this article, we present regularization approaches for DAEs that are based on the Σ-method. The article is organized as follows. In Section 2, we review the structural analysis for DAEs based on the Σ-method for DAEs. In Section 3, we discuss regularization techniques for structurally regular problems that are based on the information provided by the Σ-method. Two of the regularization techniques yield overdetermined systems of DAEs. Therefore, in Section 4 we shortly discuss the software package `QUALIDAES` for the numerical integration of (overdetermined) quasi-linear DAEs. In Section 5 we illustrate the applicability of the three proposed regularized formulations numerically integrated with `QUALIDAES` at two examples. We end with some concluding remarks in Section 6.

2 Signature Method - Structural Analysis of DAEs

The Signature method (Σ-method), see [19] and also [16, 17, 21], can be applied to regular nonlinear DAEs of arbitrary high order p of the form $F(t, z, \dot{z}, \ldots, z^{(p)}) = 0$ with $F : \mathbb{I} \times \mathbb{R}^n \times \cdots \times \mathbb{R}^n \to \mathbb{R}^n$ sufficiently smooth and $\mathbb{I} \subset \mathbb{R}$ a compact interval.

Nevertheless, in this article we will restrict our considerations to first order DAEs of the form (1). We denote by F_i the ith component of the vector-valued function F and by z_j the jth component of the vector z. Furthermore, $z^{(k)}$ denotes the kth derivative $\frac{\mathrm{d}^k z}{\mathrm{d}t^k}$ with respect to t and $\dot{z} = z^{(1)}$, $\ddot{z} = z^{(2)}$. Then, the Σ-method consists of the following steps:

Algorithm 2.1 (Signature method (Σ-method) [19])

1. Building the *signature matrix* $\Sigma = [\sigma_{ij}]_{i,j=1,\ldots,n}$ with

$$\sigma_{ij} = \begin{cases} \text{highest order of derivative of } z_j \text{ in } F_i, \\ -\infty \text{ if } z_j \text{ does not occur in } F_i. \end{cases}$$

2. Finding a *highest value transversal (HVT)* of Σ, i.e., a *transversal* $T = \{(1, j_1), (2, j_2), \ldots, (n, j_n)\}$, where $(j_1, \ldots, j_n)$ is a permutation of $(1, \ldots, n)$, with maximal *value* $\mathrm{Val}(T) := \sum_{(i,j)\in T} \sigma_{ij}$.

3. Computing the *offset vectors* $c = [c_i]_{i=1,\ldots,n}$ and $d = [d_j]_{j=1,\ldots,n}$ with $c_i, d_j \in \mathbb{N}_0$ such that

$$d_j - c_i \geq \sigma_{ij} \quad \text{for all} \quad i, j = 1, \ldots, n \quad \text{and} \tag{2a}$$
$$d_j - c_i = \sigma_{ij} \quad \text{for all} \quad (i, j) \in T. \tag{2b}$$

4. Forming the Σ-*Jacobian* $\mathfrak{J} = [\mathfrak{J}_{ij}]_{i,j=1,\ldots,n}$ with

$$\mathfrak{J}_{ij} := \begin{cases} \frac{\partial F_i}{\partial z_j^{(\sigma_{ij})}} & \text{if } d_j - c_i = \sigma_{ij}, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

5. Building the *reduced derivative array*

$$0 = \mathcal{F}(t, \mathcal{Z}) = \begin{bmatrix} \mathcal{F}_1(t, \mathcal{Z}) \\ \vdots \\ \mathcal{F}_n(t, \mathcal{Z}) \end{bmatrix} \tag{4}$$

with

$$\mathcal{F}_i(t, \mathcal{Z}) = \begin{bmatrix} F_i(t, z, \dot{z}) \\ \frac{\mathrm{d}}{\mathrm{d}t} F_i(t, z, \dot{z}) \\ \vdots \\ \frac{\mathrm{d}^{c_i}}{\mathrm{d}t^{c_i}} F_i(t, z, \dot{z}) \end{bmatrix} \quad \text{for } i = 1, \ldots n,$$

and

$$\mathcal{Z}^T = \begin{bmatrix} z_1 & \dot{z}_1 & \ldots & z_1^{(d_1)} & \ldots & z_n & \dot{z}_n & \ldots & z_n^{(d_n)} \end{bmatrix}.$$

6. Success check: if $\mathcal{F}(t, \mathcal{Z}) = 0$, considered locally as an algebraic system, has a solution $(t^*, \mathcal{Z}^*) \in \mathbb{I} \times \mathbb{R}^{n+\sum_{i=1}^{n} d_i}$, and $\mathfrak{J}$ is nonsingular at $(t^*, \mathcal{Z}^*)$, then $(t^*, \mathcal{Z}^*)$ is a consistent point and the method succeeds.

The crucial step in the Σ-method is the success check, i.e., the verification of regularity of the Σ-Jacobian at a consistent point. DAEs for which the Σ-Jacobian is singular for all points $(t^*, \mathcal{Z}^*)$ that solve the reduced derivative array (4) algebraically or DAEs for which there exists no HVT are called *structurally singular*. Accordingly, if the Σ-method succeeds the DAE (1) is called *structurally regular* with the *structural index* ν_Σ defined as

$$\nu_\Sigma := \begin{cases} \max_i c_i & \text{if all } d_j > 0 \\ \max_i c_i + 1 & \text{if some } d_j = 0. \end{cases}$$

and $\mathrm{Val}(\Sigma)$, defined as the value of the highest value transversal T, corresponds to the number of degrees of freedom of the system, see [19]. We call $\mathfrak{J}$ the Σ-Jacobian since it is

in general not equal to the Jacobian $\frac{\partial F}{\partial z}$ or $\frac{\partial F}{\partial \dot{z}}$, but defined by the offset vectors. Note that the success check of the Σ-method is performed locally at a fixed point $(t^*, \mathcal{Z}^*)$, such that the result may hold only locally in a neighborhood of a consistent point.
An HVT as well as the offset vectors can be computed by solving a linear assignment problem (LAP), see [19]. That means, Σ is the matrix of the LAP, where each assignment is specified by a transversal. This LAP (as a special kind of a linear programming problem) also has a dual problem, and the offset vectors c and d are the corresponding solutions of the dual problem. The HVT as well as the offset vectors c and d are not uniquely defined by the conditions (2), since for any feasible solution c and d, also the vectors $[c_i + \theta]_i$ and $[d_j + \theta]_j$ form a solution for any $\theta \in \mathbb{N}_0$. However, there exists a unique element-wise smallest solution, of the dual problem, the so-called *canonical offset vectors.* These are uniquely determined and independent of the chosen HVT, for more details we refer [19].

Remark 2.2 It holds that the set of equations

$$0 = \mathcal{G}(t, \mathcal{Z}) = \begin{bmatrix} \frac{\mathrm{d}^{c_1}}{\mathrm{d}t^{c_1}} F_1(t, z, \dot{z}) \\ \vdots \\ \frac{\mathrm{d}^{c_n}}{\mathrm{d}t^{c_n}} F_n(t, z, \dot{z}) \end{bmatrix}$$

is algebraically uniquely solvable for the highest occurring derivatives of the unknowns z_i, $i = 1, ..., n$ in $\mathcal{G}$, i.e., for $\begin{bmatrix} z_1^{(d_1)} & ... & z_n^{(d_n)} \end{bmatrix}^T$, such that we get from the implicit function theorem (see [3])

$$\begin{bmatrix} z_1^{(d_1)} \\ \vdots \\ z_n^{(d_n)} \end{bmatrix} = \bar{\mathcal{G}}(t, \begin{bmatrix} z_1^{(0)} & \dots & z_1^{(d_1-1)} & \dots & z_n^{(0)} & \dots & z_n^{(d_n-1)} \end{bmatrix}^T).$$

See [19]. ◁

Definition 2.3 *Let the DAE (1) be structurally regular with the canonical offset vectors c and d defined in Algorithm 2.1. For the reduced derivative array $\mathcal{F}$ let*

$$\operatorname{rank}(\begin{bmatrix} \frac{\partial \mathcal{F}}{\partial z} & \frac{\partial \mathcal{F}}{\partial \dot{z}} & \cdots & \frac{\partial \mathcal{F}}{\partial z^{(\max_i d_i)}} \end{bmatrix}) = n + \sum_{i=1}^{n} c_i \tag{5a}$$

$$\textit{and } \operatorname{rank}(\begin{bmatrix} \frac{\partial \mathcal{F}}{\partial \dot{z}} & \cdots & \frac{\partial \mathcal{F}}{\partial z^{(\max_i d_i)}} \end{bmatrix}) = \sum_{i=1}^{n} d_i \tag{5b}$$

for all $(t^*, \mathcal{Z}^*)$ satisfying $0 = \mathcal{F}(t^*, \mathcal{Z}^*)$. Then from $0 = \mathcal{F}$ a set of algebraic equations

$$0 = H(t, z) \quad \textit{with} \quad H(t, z) : \mathbb{I} \times \mathbb{R}^n \to \mathbb{R}^{m_c} \tag{6}$$

with $m_c = n + \sum_{i=1}^{n} c_i - \sum_{i=1}^{n} d_i$ and

$$\operatorname{rank}(\frac{\partial H(t, z)}{\partial z}) = m_c \quad \textit{for all} \quad (t, z) \in \mathbb{M}$$

with

$$\mathbb{M} = \{(t, z) \in \mathbb{I} \times \mathbb{R}^n \ : \ 0 = H(t, z)\}$$

can be extracted.
The set $\mathbb{M}$ is called set of consistency *and the array of equations $0 = H(t, z)$ is called the* array of (hidden) constraints.

The array of constraints $0 = H(t, z)$ contains all (hidden) constraints of the DAE (1) and, therefore, restricts every solution z of the DAE (1) into the set of consistency $\mathbb{M}$. This means that $(t, z(t)) \in \mathbb{M}$ has to be satisfied by every solution z for all $t \in \mathbb{I}$.

Remark 2.4 Replacing all derivatives of all variables by algebraic variables, i.e., replacing $z_i^{(j)}$ by ω_i^j for $i = 1, ..., n$ and $j = 1, ..., d_i$, in the reduced derivative array (4) yields the *algebraic reduced derivative array*

$$\begin{aligned} 0 \;&=\; \mathcal{F}(t, z, \omega) = \mathcal{F}(t, z_1, \omega_1, \ldots, z_n, \omega_n) \qquad (7) \\ &\text{with} \;\; \omega^T = \begin{bmatrix} \omega_1^T & \cdots & \omega_n^T \end{bmatrix} \\ &\text{and} \;\; \omega_i = \begin{cases} [\omega_i^1, ..., \omega_i^{d_i}]^T & \text{if } d_i \geq 1, \\ [\cdot] & \text{if } d_i = 0. \end{cases} \end{aligned}$$

From (5b), the implicit function theorem, and the assumed structural regularity of (1) it follows that (7) can be solved uniquely for ω, such that we obtain

$$\omega = \Omega(t, z) \tag{8}$$

such that $0 = \mathcal{F}(t, z, \Omega(t, z))$. ◁

Definition 2.5 *[23] Two DAEs (1) and*

$$0 = \hat{F}(t, \hat{z}, \dot{\hat{z}}) \tag{9}$$

with their sets of consistency $\mathbb{M} \subset \mathbb{I} \times \mathbb{R}^n$ and $\hat{\mathbb{M}} \subset \mathbb{I} \times \mathbb{R}^{\hat{n}}$, respectively, are called solution equivalent *if there uniquely exists a bijective map $\Phi \; : \; \mathbb{M} \to \hat{\mathbb{M}}$ such that $\Phi(x)$ is a solution of (9) if x is a solution of (1) and $\Phi^{-1}(\hat{x})$ is a solution of (1) if $\hat{x}$ is a solution of (9).*

Definition 2.6 *A DAE (9) with structural index $\hat{\nu}_\Sigma$ is called a* structural regularization *of the DAE (1) with structural index ν_Σ if both DAEs are solution equivalent and $\hat{\nu}_\Sigma < \nu_\Sigma$.*

Example 2.7 Let us consider the simple pendulum (Fig. 1) of mass $m > 0$ and length $\ell > 0$ under gravity $\mathfrak{g}$, see also [19, 21].

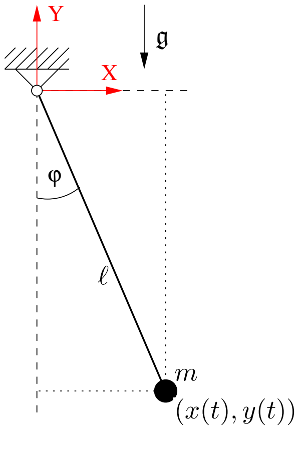

Figure 1: Simple Pendulum

The state of the pendulum is described by the position x and y, the velocities v and w of the mass point, and the Lagrange multiplier λ. The system equations are given by

$$0 = F(z, \dot{z}) = \begin{bmatrix} F_1(z, \dot{z}) \\ F_2(z, \dot{z}) \\ F_3(z, \dot{z}) \\ F_4(z, \dot{z}) \\ F_5(z, \dot{z}) \end{bmatrix} = \begin{bmatrix} \dot{x} - v \\ \dot{y} - w \\ m\dot{v} + 2x\lambda \\ m\dot{w} + 2y\lambda + m\mathfrak{g} \\ x^2 + y^2 - \ell^2 \end{bmatrix}, \qquad \begin{array}{r} (10a) \\ (10b) \\ (10c) \\ (10d) \\ (10e) \end{array}$$

where $z^T = \begin{bmatrix} x & y & v & w & \lambda \end{bmatrix}$.
The model equations (10) contain hidden constraints which constrain the solution in addition to the explicitly stated constraint (10e). In particular, as one hidden constraint we have the *constraint on level 1 (velocity level)*

$$0 = 2xv + 2yw \tag{11a}$$

obtained from the total time derivative of the constraint (10e), where the derivatives $\dot{x}$ and $\dot{y}$ are replaced by (10a) and (10b), respectively. The constraint (11a) is called hidden constraint on level 1 since this constraint is obtained after differentiation of (certain) model equations at most once. A further hidden constraint is the *constraint on level 2 (acceleration level)*

$$0 = 2v^2 + 2w^2 - 2y\mathfrak{g} - \frac{4}{m}(x^2 + y^2)\lambda. \tag{11b}$$

This is obtained from the total time derivative of the constraint on level 1 (11a), where the derivatives $\dot{x}$, $\dot{y}$, $\dot{v}$, and $\dot{w}$ are replaced by (10a)-(10d), respectively. The constraint (11b) is called hidden constraint on level 2 since this constraint is obtained after differentiation of (certain) model equations at most twice.
Following the Σ-method we get the signature matrix Σ for (10) and the canonical offset vectors c and d as

$$\begin{array}{rcl} \Sigma & = & \begin{bmatrix} \colorbox{gray}{1} & - & \colorbox{lightgray}{0} & - & - \\ - & \colorbox{lightgray}{1} & - & \colorbox{gray}{0} & - \\ 0 & - & \colorbox{gray}{1} & - & \colorbox{lightgray}{0} \\ - & 0 & - & \colorbox{lightgray}{1} & \colorbox{gray}{0} \\ \colorbox{lightgray}{0} & \colorbox{gray}{0} & - & - & - \end{bmatrix}, \quad c = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 2 \end{bmatrix}, \\ d^T & = & \begin{bmatrix} 2 & 2 & 1 & 1 & 0 \end{bmatrix}, \end{array} \tag{12}$$

where the two possible HVTs

$$\begin{array}{rcl} T_1 & = & \{(1,3),(2,2),(3,5),(4,4),(5,1)\}, \\ T_2 & = & \{(1,1),(2,4),(3,3),(4,5),(5,2)\} \end{array}$$

are marked by light gray boxes for T_1 and dark gray boxes for T_2. In Σ the entry "$-$" stands for $-\infty$. The corresponding Σ-Jacobian $\mathfrak{J}$ in (3) and the reduced derivative array $\mathcal{F}$ in (4) takes the form

$$\mathfrak{J} = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & m & 0 & 2x \\ 0 & 0 & 0 & m & 2y \\ 2x & 2y & 0 & 0 & 0 \end{bmatrix}, \tag{13}$$

$$0 = \mathcal{F}(t, \mathcal{Z}) = \begin{bmatrix} \dot{x} - v \\ \ddot{x} - \dot{v} \\ \dot{y} - w \\ \ddot{y} - \dot{w} \\ m\dot{v} + 2x\lambda \\ m\dot{w} + 2y\lambda + m\mathfrak{g} \\ x^2 + y^2 - \ell^2 \\ 2x\dot{x} + 2y\dot{y} \\ 2x\ddot{x} + 2\dot{x}^2 + 2y\ddot{y} + 2\dot{y}^2 \end{bmatrix} \qquad \begin{array}{r} (14a) \\ (14b) \\ (14c) \\ (14d) \\ (14e) \\ (14f) \\ (14g) \\ (14h) \\ (14i) \end{array}$$

with $\mathcal{Z}^T = \begin{bmatrix} x & \dot{x} & \ddot{x} & y & \dot{y} & \ddot{y} & v & \dot{v} & w & \dot{w} & \lambda \end{bmatrix}$. Since $\det(\mathfrak{J}) = -4m(x^2 + y^2) = -4m\ell^2 \neq 0$, the Σ-Jacobian $\mathfrak{J}$ is nonsingular at every consistent point, and the Σ-method succeeds with $\nu_\Sigma = \max_i c_i + 1 = 3$.

As mentioned in Definition 2.3 the reduced derivative array (14) allows the determination of all (hidden) constraints yielding the array of (hidden) constraints (6). In particular, the constraint on level 0 (10e) (which is not hidden) is contained explicitly as equation (14g). Furthermore, the hidden constraint on level 1 (velocity level) (11a) can be obtained from (14h), where $\dot{x}$ and $\dot{y}$ are replaced by use of equations (14a) and (14c), respectively. The hidden constraint on level 2 (acceleration level) (11b) can be obtained from (14i), where $\ddot{x}$ and $\ddot{y}$ are replaced by use of equations (14b) and (14d) and $\dot{x}$, $\dot{y}$, $\dot{v}$, and $\dot{w}$ are replaced by use of equations (14a), (14c), (14e), and (14f), respectively. Summarizing we get the array of (hidden) constraints as

$$0 = H(t,z) = \begin{bmatrix} x^2+y^2-\ell^2 \\ 2xv+2yw \\ v^2+w^2-\frac{2}{m}(x^2+y^2)\lambda-y\mathfrak{g} \end{bmatrix}. \tag{15}$$

◁

Example 2.8 In the second example we are interested in a path following problem of a mass-spring-chain as in Figure 2. On the first and the last body the force F is applied while the body in the center has to follow a prescribed path given as constraint (16g). The unknown variables are p_i and v_i as the position and velocity of body $i = 1, 2, 3$ and F as acting force.

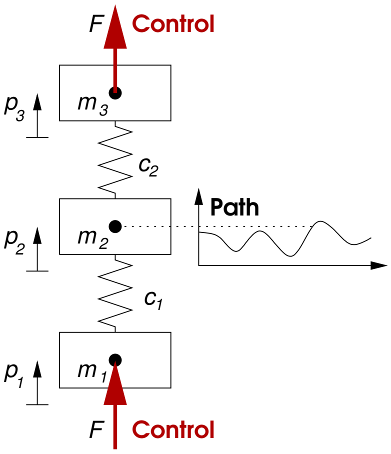

Figure 2: Mass-Spring-Chain

The model equations are given by

$$0 = F(z,\dot{z}) = \begin{bmatrix} \dot{p}_1-v_1 \\ \dot{p}_2-v_2 \\ \dot{p}_3-v_3 \\ m\dot{v}_1+c(p_1-p_2)-F \\ m\dot{v}_2-c(p_1-p_2)+c(p_2-p_3) \\ m\dot{v}_3-c(p_2-p_3)-F \\ p_2-\sin(t) \end{bmatrix}, \qquad \begin{array}{r} (16a) \\ (16b) \\ (16c) \\ (16d) \\ (16e) \\ (16f) \\ (16g) \end{array}$$

where $z^T = \begin{bmatrix} p_1 & p_2 & p_3 & v_1 & v_2 & v_3 & F \end{bmatrix}$. Beside the explicitly stated constraint (16g), which prescribes the path, there exists the hidden constraints

$$0 = v_2-\cos(t), \tag{17a}$$

$$0 = \frac{c}{m}(p_1-p_2)-\frac{c}{m}(p_2-p_3)+\sin(t), \tag{17b}$$

$$0 = \frac{c}{m}(v_1-v_2)-\frac{c}{m}(v_2-v_3)+\cos(t), \tag{17c}$$

$$0 = \frac{c}{m^2}(-3c(p_1-p_2)+3c(p_2-p_3)+2F)-\sin(t). \tag{17d}$$

These hidden constraints (17a)-(17d) are on level 1 up to 4, respectively, and are obtained in a similar way as in Example 2.7 by differentiating (parts of) the DAE once up to four times, respectively, with replacing derivatives of the states by use of equations (16a)-(16f). Following the Σ-method we get the signature matrix Σ for (16) and the canonical offset vectors c and d as

$$\begin{array}{rcl}
\Sigma & = & \begin{bmatrix}
\colorbox{gray}{1} & - & - & \colorbox{lightgray}{0} & - & - & - \\
- & 1 & - & - & \colorbox{lightgray}{$\colorbox{gray}{$0$}$} & - & - \\
- & - & \colorbox{lightgray}{1} & - & - & \colorbox{gray}{0} & - \\
0 & 0 & - & \colorbox{gray}{1} & - & - & \colorbox{lightgray}{0} \\
\colorbox{lightgray}{0} & 0 & \colorbox{gray}{0} & - & 1 & - & - \\
- & 0 & 0 & - & - & \colorbox{lightgray}{1} & \colorbox{gray}{0} \\
- & \colorbox{lightgray}{$\colorbox{gray}{$0$}$} & - & - & - & - & -
\end{bmatrix}, \quad c = \begin{bmatrix} 1 \\ 3 \\ 1 \\ 0 \\ 2 \\ 0 \\ 4 \end{bmatrix}, \\
d^T & = & \begin{bmatrix} 2 & 4 & 2 & 1 & 3 & 1 & 0 \end{bmatrix},
\end{array} \tag{18}$$

where the two possible HVTs

$$\begin{aligned}
T_1 &= \{(1,4),(2,5),(3,3),(4,7),(5,1),(6,6),(7,2)\}, \\
T_2 &= \{(1,1),(2,5),(3,6),(4,4),(5,3),(6,7),(7,2)\}
\end{aligned}$$

are marked by light gray boxes for T_1 and dark gray boxes for T_2. Here again in Σ the entry "$-$" stands for $-\infty$. The corresponding Σ-Jacobian $\mathfrak{J}$ in (3) and the reduced derivative array $0 = \mathcal{F}$ in (4) takes the form

$$\mathfrak{J} = \begin{bmatrix}
1 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & m & 0 & 0 & -1 \\
-c & 0 & -c & 0 & m & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & m & -1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}, \tag{19}$$

$$0 = \mathcal{F}(t,\mathcal{Z}) = \begin{bmatrix}
\dot{p}_1 - v_1 \\
\ddot{p}_1 - \dot{v}_1 \\
\dot{p}_2 - v_2 \\
\ddot{p}_2 - \dot{v}_2 \\
p_2^{(3)} - \ddot{v}_2 \\
p_2^{(4)} - v_2^{(3)} \\
\dot{p}_3 - v_3 \\
\ddot{p}_3 - \dot{v}_3 \\
m\dot{v}_1 + c(p_1 - p_2) - F \\
m\dot{v}_2 - c(p_1 - p_2) + c(p_2 - p_3) \\
m\ddot{v}_2 - c(\dot{p}_1 - \dot{p}_2) + c(\dot{p}_2 - \dot{p}_3) \\
m v_2^{(3)} - c(\ddot{p}_1 - \ddot{p}_2) + c(\ddot{p}_2 - \ddot{p}_3) \\
m\dot{v}_3 - c(p_2 - p_3) - F \\
p_2 - \sin(t) \\
\dot{p}_2 - \cos(t) \\
\ddot{p}_2 + \sin(t) \\
p_2^{(3)} + \cos(t) \\
p_2^{(4)} - \sin(t)
\end{bmatrix} \qquad \begin{matrix}
(20a) \\ (20b) \\ (20c) \\ (20d) \\ (20e) \\ (20f) \\ (20g) \\ (20h) \\ (20i) \\ (20j) \\ (20k) \\ (20l) \\ (20m) \\ (20n) \\ (20o) \\ (20p) \\ (20q) \\ (20r)
\end{matrix}$$

with $\mathcal{Z}^T = [p_1 \;\; \dot{p}_1 \;\; \ddot{p}_1 \;\; p_2 \;\; \dot{p}_2 \;\; \ddot{p}_2 \;\; p_2^{(3)} \; p_2^{(4)} \; p_3 \;\; \dot{p}_3 \;\; \ddot{p}_3 \;\; v_1 \;\; \dot{v}_1 \;\; v_2 \;\; \dot{v}_2 \;\; \ddot{v}_2 \;\; v_2^{(3)} \; v_3 \;\; \dot{v}_3 \;\; F]$. Since $\det(\mathfrak{J}) = 2cm \neq 0$, the Σ-Jacobian $\mathfrak{J}$ is nonsingular independent on $\mathcal{Z}$, and the Σ-method succeeds with $\nu_\Sigma = \max_i c_i + 1 = 5$.

Again the reduced derivative array (14) allows the determination of all (hidden) constraints yielding the array of (hidden) constraints (6). In particular, the constraint on level 0 (16g) (which is not hidden) is contained explicitly as equation (20n). Furthermore, the hidden constraint on level 1 (velocity level) (17a) can be obtained from (20o), where $\dot{p}_2$ is replaced by use of equation (20c). The hidden constraint on level 2 (acceleration level) (17b) can be obtained from (20p), where $\ddot{p}_2$ is replaced by use of equation (20d) and $\dot{v}_2$ is replaced by use of equation (20j). The hidden constraint on level 3 and 4 can be obtained in a similar way. Summarizing we get the array of (hidden) constraints as

$$0 = H(t, z) = \begin{bmatrix} p_2 - \sin(t) \\ v_2 - \cos(t) \\ \frac{c}{m}(p_1 - p_2) - \frac{c}{m}(p_2 - p_3) + \sin(t) \\ \frac{c}{m}(v_1 - v_2) - \frac{c}{m}(v_2 - v_3) + \cos(t) \\ \frac{c}{m^2}(-3c(p_1 - p_2) + 3c(p_2 - p_3) + 2F) - \sin(t) \end{bmatrix}. \tag{21}$$

◁

3 Regularization Approaches for DAEs

Now, we consider structurally regular DAEs (1) with $m = n$ for which the Σ-method succeeds and discuss three different regularization approaches. These regularization approaches mainly use the reduced derivative array $\mathcal{F}$ in (4) provided by the Σ-method as well as the set of (hidden) constraints (6), see Definition 2.3. These proposed regularization approaches differ in its degree of analytical preprocessing and possible degree of automation of the approach.

3.1 Regularization via Structural Extension

To obtain a regularized formulation of a given DAE system (1), the information in the reduced derivative array $\mathcal{F}$ provided by the Σ-method can be used.

The reduced derivative array $\mathcal{F}$ is defined in Step 5 in Algorithm 2.1 by adding the derivatives of F_i up to order c_i to the original DAE resulting in (4) and consists of $M = \sum_i c_i + n$ equations in n unknowns $z_1, \ldots, z_n$. More precisely, it depends on $z_1, \dot{z}_1, \ldots, z_1^{(d_1)}, \ldots, z_n, \dot{z}_n, \ldots, z_n^{(d_n)}$. Thus, to obtain the same number of equations and unknowns, we have to introduce $n_\psi = \sum_i c_i$ new variables.

For that we choose one HVT T^* from the set of all existing HVTs. Then for each $j = 1, \ldots, n$ we have a unique i with $(i, j) \in T^*$, and a regularization can be obtained from the reduced derivative array $\mathcal{F}$ in (4) by replacing the derivatives $z_j^{(\sigma_{ij}+1)}$, ..., $z_j^{(d_i)}$ with the new algebraic variables $\psi_j^{\sigma_{ij}+1}$, ..., $\psi_j^{d_i}$ if $d_i > \sigma_{ij}$.

By replacing derivatives of the unknown variables with algebraic variables in the reduced derivative array $0 = \mathcal{F}$ in (4) as described above, we obtain the *structurally extended (StE) formulation*

$$\mathcal{S}(t, z, \dot{z}, \psi) = 0 \tag{22}$$

with ψ as new algebraic variables, where

$$\psi^T = \begin{bmatrix} \psi_1^T & \cdots & \psi_n^T \end{bmatrix}$$

and

$$\psi_j := \begin{cases} \begin{bmatrix} \psi_j^{\sigma_{ij}+1} & \cdots & \psi_j^{d_i} \end{bmatrix}^T \text{ of size } c_i & \text{if } d_j > \sigma_{ij} \\ [\,\cdot\,] \text{ of size } 0 & \text{if } d_j = \sigma_{ij}. \end{cases}$$

Note that $d_j = \sigma_{ij} + c_i$ by (2b).

The obtained structurally extended formulation (22) is of increased size and not unique, since it depends on the chosen HVT T^*.

The application of the Σ-method to StE-formulation (22) with the vector of unknowns $\left[\begin{array}{cc} z^T & \psi^T \end{array}\right]^T$ yields canonical offset vectors

$$c = \left[\begin{array}{c} 0 \\ \vdots \\ 0 \end{array}\right] \text{ and } d = \left[\begin{array}{c} d^z \\ d^\psi \end{array}\right]$$

with $d^z \in \{0,1\}^n$ corresponding to the existing highest derivatives of z in (22) consisting only zeros and ones and with $d^\psi = 0$ corresponding to ψ in (22), see [21]. Therefore, according to Remark 2.2 and due to $c = 0$, it holds that (22) is algebraically uniquely solvable for $\left[\begin{array}{cccccc} z_1^{(d_1^z)} & \ldots & z_n^{(d_n^z)} & \psi_1 & \ldots & \psi_{n_\psi} \end{array}\right]^T$. Consequently, we get

$$\left[\begin{array}{c} z_1^{(d_1^z)} \\ \vdots \\ z_n^{(d_n^z)} \\ \hline \psi_1 \\ \vdots \\ \psi_{n_\psi} \end{array}\right] = \left[\begin{array}{c} \Psi_z(t,z) \\ \hline \Psi_\psi(t,z) \end{array}\right],$$

in particular, $\psi = \Psi_\psi(t,z)$. Therefore, the StE-formulation (22) is locally solution equivalent to the original DAE (1) as long as the StE-formulation (22) is structurally regular with

$$\Phi : z \mapsto \left[\begin{array}{c} z \\ \Psi_\psi(t,z) \end{array}\right] = \left[\begin{array}{c} z \\ \psi \end{array}\right].$$

Consequently, the StE-formulation (22) has locally the same set of solutions for the original unknowns z as the original DAE (1). Due to the reduced structural index $\nu_\Sigma = 0$ of the StE-formulation (22) it is a local structural regularization of the original DAE (1).

In general the structurally extended formulation (22) is only valid locally in a neighborhood of a consistent point as long as the StE-formulation themselves is structurally regular. Therefore, it may lead to an inappropriate regularization, if an inappropriate HVT T^* is used (see Example 3.1). If an HVT is choosen which is not suitable, the StE-formulation is structurally singular. Therefore, the determination/usage of a new StE-formulation during the (numerical) integration may be necessary (also known as *dynamic state selection*). In this case, the numerical integration has to be interrupted and to be restarted with the newly regularized model equations. Unfortunately, this leads to an increase in integration time and influences the obtained precision negatively. This is a common problem whenever new variables are introduced to obtain a regularization. The same problem occurs in the Method of Dummy Derivatives [15] or in the Index Reduction by Minimal Extension [12].

To choose a suitable HVT, i.e., one that is valid in a preferably large neighborhood of a consistent point, we use a weighting for the different possible HVTs. Based on the Σ-Jacobian (that is regular since the Σ-method succeeds locally at the consistent point) we define the (local) *weighting coefficient* for each possible HVT T as

$$\kappa_T := \prod_{(i,j)\in T} |\mathfrak{J}_{ij}|$$

and choose one with largest value κ_T, i.e., we choose T^* as an HVT of Σ with $\kappa_{T^*} = \max_T(\kappa_T)$. The choice of the HVT based on the weighting factor κ_T is motivated by the observation, that a "good" HVT should be well-conditioned in the sense that the resulting regularized system is structurally regular and well away from a singular point. For more details in the choice of T^* see [21], where it is shown that the structurally extended system (22) is (locally) a regular system of structural index $\nu_\Sigma \leq 1$ if T^* is chosen as described above.

This approach via structural extension completely can be automated using automatic or symbolic differentiation tools. Therefore, an analytical preprocessing is not necessary.
The regularization approach via structural extension based on the Σ-method is similar to the *Method of Dummy Derivatives* [15], where the *Pantelides algorithm* [18] is used to determine the number of times each equation has to be differentiated to obtain a reduced derivative array. It has been shown in [19] that the Pantelides algorithm and the Σ-method are essentially equivalent in the sense that, if they both can be applied and they both succeed (or converge in case of the Pantelides algorithm), they result in the same structural index, and the canonical offset vector $c = [c_i]$ corresponds to the number of differentiations for each equation F_i as determined by the Pantelides algorithm. In the proposed regularization via structural extension, the selection of variables for which derivatives are replaced is directly prescribed by the HVT T^* and the canonical offset vectors. Once the Σ-method is done, this selection of variables is easy to achieve and requires no further numerical computations. In contrast, in the Dummy Derivative Method, the selection of variables is based on an algorithm that aims at selecting certain columns of the Jacobian to obtain a regular submatrix, for details see [15]. As a result, the selected variables are not necessarily the same, and the two approaches might result in different regularizations.

Example 3.1 Let us continue the consideration of the model equations (10) for the simple pendulum. See Example 2.7. The DAE (10) is structurally regular and of structural index $\nu_\Sigma = 3$. The reduced derivative array (14) consists of $M = \sum c_i + n = 9$ equations in $n = 5$ unknowns.
For the HVT $T^* = T_1$ marked by the light gray boxes in the signature matrix (12) (assuming that $\kappa_{T_1} = 4x^2 \geq \kappa_{T_2} = 4y^2$), we have to introduce new algebraic variables as in the following table:

for j	with i s.t. $(i,j) \in T^*$	replace derivatives $z_j^{(\sigma_{ij}+1)}, \ldots, z_j^{(d_j)}$	with new algebraic variables $\psi_j^{\sigma_{ij}+1}, \ldots, \psi_j^{d_j}$
1	5	$z_1^{(1)},\ z_1^{(2)}$	$\psi_1^1,\ \psi_1^2$
2	2	$z_2^{(2)}$	ψ_2^2
3	1	$z_3^{(1)}$	ψ_3^1
4	4	$\emptyset$	$\emptyset$
5	3	$\emptyset$	$\emptyset$

Then we construct the structurally extended formulation as

$$0 = \mathcal{S}_1(t, z, \dot{z}, \psi_1) = \begin{bmatrix} \psi_1^1 - v \\ \psi_1^2 - \psi_3^1 \\ \dot{y} - w \\ \psi_2^2 - \dot{w} \\ m\psi_3^1 + 2x\lambda \\ m\dot{w} + 2y\lambda + m\mathfrak{g} \\ x^2 + y^2 - \ell^2 \\ 2x\psi_1^1 + 2y\dot{y} \\ 2x\psi_1^2 + 2(\psi_1^1)^2 + 2y\psi_2^2 + 2\dot{y}^2 \end{bmatrix} \tag{23}$$

with

$$\begin{aligned} z^T &= \left[\begin{array}{ccccc} x & y & v & w & \lambda \end{array}\right], \\ \psi_1^T &= \left[\begin{array}{cccc} \psi_1^1 & \psi_1^2 & \psi_2^2 & \psi_3^1 \end{array}\right]. \end{aligned}$$

For the HVT $T^* = T_2$ marked by the dark gray boxes in the signature matrix (12) (assuming that $\kappa_{T_2} = 4y^2 \geq \kappa_{T_1} = 4x^2$), we have to introduce new algebraic variables as in the following table:

for j	with i s.t. $(i,j) \in T^*$	replace derivatives $z_j^{(\sigma_{ij}+1)}$, ..., $z_j^{(d_j)}$	with new algebraic variables $\psi_j^{\sigma_{ij}+1}$, ..., $\psi_j^{d_j}$
1	1	$z_1^{(2)}$	ψ_1^2
2	5	$z_2^{(1)},\ z_2^{(2)}$	$\psi_2^1,\ \psi_2^2$
3	3	$\emptyset$	$\emptyset$
4	2	$z_4^{(1)}$	ψ_4^1
5	4	$\emptyset$	$\emptyset$

Then we construct the structurally extended formulation as

$$0=\mathcal{S}_2(t,z,\dot{z},\psi_2)=\begin{bmatrix} \dot{x}-v \\ \psi_1^2-\dot{v} \\ \psi_2^1-w \\ \psi_2^2-\psi_4^1 \\ m\dot{v}+2x\lambda \\ m\psi_4^1+2y\lambda+m\mathfrak{g} \\ x^2+y^2-\ell^2 \\ 2x\dot{x}+2y\psi_2^1 \\ 2x\psi_1^2+2\dot{x}^2+2y\psi_2^2+2(\psi_2^1)^2 \end{bmatrix} \tag{24}$$

with

$$\begin{aligned} z^T &= \begin{bmatrix} x & y & v & w & \lambda \end{bmatrix}, \\ \psi_2^T &= \begin{bmatrix} \psi_1^2 & \psi_2^1 & \psi_2^2 & \psi_4^1 \end{bmatrix}. \end{aligned}$$

Let us check the result by applying the Σ-method to the StE-formulations (23) and (24). For the StE-formulation (23) we obtain the according signature matrix $\widetilde{\Sigma}_1$ and the canonical offset vectors $\widetilde{c}_1$ and $\widetilde{d}_1$ as

$$\widetilde{\Sigma}_1 = \left[\begin{array}{ccccc|cccc} - & - & \colorbox{lightgray}{\colorbox{gray}{0}} & - & - & 0 & - & - & - \\ - & - & - & - & - & - & \colorbox{lightgray}{0} & - & \colorbox{gray}{0} \\ - & \colorbox{lightgray}{\colorbox{gray}{1}} & - & 0 & - & - & - & - & - \\ - & - & - & \colorbox{lightgray}{1} & - & - & - & \colorbox{gray}{0} & - \\ 0 & - & - & - & \colorbox{gray}{0} & - & - & - & \colorbox{lightgray}{0} \\ - & 0 & - & \colorbox{gray}{1} & \colorbox{lightgray}{0} & - & - & - & - \\ \colorbox{lightgray}{\colorbox{gray}{0}} & 0 & - & - & - & - & - & - & - \\ 0 & 1 & - & - & - & \colorbox{lightgray}{\colorbox{gray}{0}} & - & - & - \\ 0 & 1 & - & - & - & 0 & \colorbox{gray}{0} & \colorbox{lightgray}{0} & - \end{array}\right], \quad \widetilde{c}_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$$\widetilde{d}_1^T = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

with the the highest value transversals marked with dark gray and light gray boxes. Furthermore, the value of $\widetilde{\Sigma}_1$ is $\mathrm{Val}(\widetilde{\Sigma}_1)=2$. Since $\widetilde{c}_1=0$, the corresponding reduced derivative array coincides with the system (23), i.e., $\widetilde{\mathcal{F}}_1(t,\widetilde{x}_1,\dot{\widetilde{x}}_1)=\mathcal{S}_1(t,x,\dot{x},\psi_1)$ with $\widetilde{x}_1^T = \begin{bmatrix} x^T & \psi_1^T \end{bmatrix}$

and the Σ-Jacobian is given by

$$\widetilde{\mathfrak{J}}_1 = \left[\begin{array}{ccccc|cccc} 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ 2\lambda & 0 & 0 & 0 & 2x & 0 & 0 & 0 & m \\ 0 & 0 & 0 & m & 2y & 0 & 0 & 0 & 0 \\ 2x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2w_1^1 & 2y & 0 & 0 & 0 & 2x & 0 & 0 & 0 \\ 2w_1^2 & 4\dot{y} & 0 & 0 & 0 & 4w_1^1 & 2x & 2y & 0 \end{array}\right] \tag{25}$$

with $\det(\widetilde{\mathfrak{J}}_1) = 16mx^2(x^2+y^2) = 16mx^2\ell^2$ for all consistent values, in particular, satisfying the seventh equation in (23). Therefore, the Σ-Jacobian $\widetilde{\mathfrak{J}}_1$ can be checked to be nonsingular at a consistent point, since x is assumed to be nonzero ($x \geq y$ and both cannot be zero simultaneously). Thus, the Σ-method for $\mathcal{S}_1$ succeeds with structural index $\nu_\Sigma = 1$. Note that for $x = 0$, the Σ-Jacobian (25) is singular, and the other HVT T_2 would have to be chosen (ensured by the condition that $\kappa_{T_2} > \kappa_{T_1}$ in this case).

For the StE-formulation (24) we obtain the according signature matrix $\widetilde{\Sigma}_2$ and the canonical offset vectors $\widetilde{c}_2$ and $\widetilde{d}_2$ as

$$\widetilde{\Sigma}_2 = \left[\begin{array}{ccccc|cccc} \colorbox{lightgray}{\colorbox{gray}{1}} & - & 0 & - & - & - & - & - & - \\ - & - & \colorbox{lightgray}{1} & - & - & \colorbox{gray}{0} & - & - & - \\ - & - & - & \colorbox{lightgray}{\colorbox{gray}{0}} & - & - & 0 & - & - \\ - & - & - & - & - & - & - & \colorbox{lightgray}{0} & \colorbox{gray}{0} \\ 0 & - & \colorbox{gray}{1} & - & \colorbox{lightgray}{0} & - & - & - & - \\ - & 0 & - & - & \colorbox{gray}{0} & - & - & - & \colorbox{lightgray}{0} \\ 0 & \colorbox{lightgray}{\colorbox{gray}{0}} & - & - & - & - & - & - & - \\ 1 & 0 & - & - & - & - & \colorbox{lightgray}{\colorbox{gray}{0}} & - & - \\ 1 & 0 & - & - & - & \colorbox{lightgray}{0} & 0 & \colorbox{gray}{0} & - \end{array}\right], \quad \widetilde{c}_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$$\widetilde{d}_2^T = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

again with the the highest value transversals marked with dark gray and light gray boxes. Furthermore, the value of $\widetilde{\Sigma}_2$ is $\mathrm{Val}(\widetilde{\Sigma}_2) = 2$. Again, since $\widetilde{c}_2 = 0$, the corresponding reduced derivative array coincides with the system (24), i.e., $\widetilde{\mathcal{F}}_2(t, \widetilde{x}_2, \dot{\widetilde{x}}_2) = \mathcal{S}_2(t, x, \dot{x}, \psi_2)$ with $\widetilde{x}_2^T = \begin{bmatrix} x^T & \psi_2^T \end{bmatrix}$ and the Σ-Jacobian is given by

$$\widetilde{\mathfrak{J}}_2 = \left[\begin{array}{ccccc|cccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & m & 0 & 2x & 0 & 0 & 0 & 0 \\ 0 & 2\lambda & 0 & 0 & 2y & 0 & 0 & 0 & m \\ 0 & 2y & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2x & 2w_2^1 & 0 & 0 & 0 & 0 & 2y & 0 & 0 \\ 4\dot{x} & 2w_2^2 & 0 & 0 & 0 & 2x & 4w_2^1 & 2y & 0 \end{array}\right] \tag{26}$$

with $\det(\widetilde{\mathfrak{J}}_2) = -16my^2(x^2+y^2) = -16my^2\ell^2$ for all consistent values, in particular, satisfying the seventh equation in (24). Therefore, the Σ-Jacobian $\widetilde{\mathfrak{J}}_2$ can be checked to be nonsingular at a consistent point, since y is assumed to be nonzero ($y \geq x$ and both cannot be zero simultaneously). Thus, the Σ-method for $\mathcal{S}_2$ succeeds with structural index $\nu_\Sigma = 1$. Note also here that for $y = 0$, the Σ-Jacobian (26) is singular, and the other HVT T_1 would have to be chosen (ensured by the condition that $\kappa_{T_1} > \kappa_{T_2}$ in this case).

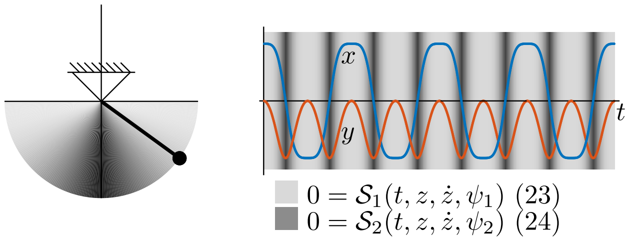

Figure 3: Dynamic state selection

Summarizing, we get two locally valid regularizations in form of the structurally extended formulations. In particular, $\mathcal{S}_1$ (23) is valid as long as $x \neq 0$ while $\mathcal{S}_2$ (24) is valid as long as $y \neq 0$ as illustrated in Figure 3. ◁

Example 3.2 Let us continue the consideration of the model equations (16) for the mass-spring-chain with path control. See Example 2.8. The DAE (16) is structurally regular and of structural index $\nu_\Sigma = 5$. The reduced derivative array (20) consists of $M = \sum c_i + n = 18$ equations in $n = 7$ unknowns. For the two possible HVTs T_1 and T_2 we get $\kappa_{T_1} = \kappa_{T_2} = cm$. Therefore, none of them is preferable.
For the HVT $T^* = T_1$ marked by the light gray boxes in the signature matrix (18) we have to introduce new algebraic variables as in the following table:

for j	with i s.t. $(i,j) \in T^*$	replace derivatives $z_j^{(\sigma_{ij}+1)}, \ldots, z_j^{(d_j)}$	with new algebraic variables $\psi_j^{\sigma_{ij}+1}, \ldots, \psi_j^{d_j}$
1	5	$z_1^{(1)},\ z_1^{(2)}$	$\psi_1^1,\ \psi_1^2$
2	7	$z_2^{(1)}, \ldots,,\ z_2^{(4)}$	$\psi_2^1, \ldots, \psi_2^4$
3	3	$z_3^{(2)}$	ψ_3^2
4	1	$z_4^{(1)}$	ψ_4^1
5	2	$z_5^{(1)},\ z_5^{(2)},\ z_5^{(3)}$	$\psi_5^1,\ \psi_5^2,\ \psi_5^3$
6	6	$\emptyset$	$\emptyset$
7	4	$\emptyset$	$\emptyset$

Then we construct the structurally extended formulation as

$$0 = \mathcal{S}_1(t, z, \dot{z}, \psi_1) = \begin{bmatrix} \psi_1^1 - v_1 \\ \psi_1^2 - \psi_4^1 \\ \psi_2^1 - v_2 \\ \psi_2^2 - \psi_5^1 \\ \psi_2^3 - \psi_5^2 \\ \psi_2^4 - \psi_5^3 \\ \dot{p}_3 - v_3 \\ \psi_3^2 - \dot{v}_3 \\ m\psi_4^1 + c(p_1 - p_2) - F \\ m\psi_5^1 - c(p_1 - p_2) + c(p_2 - p_3) \\ m\psi_5^2 - c(\psi_1^1 - \psi_2^1) + c(\psi_2^1 - \dot{p}_3) \\ m\psi_5^3 - c(\psi_1^2 - \psi_2^2) + c(\psi_2^2 - \psi_3^2) \\ m\dot{v}_3 - c(p_2 - p_3) - F \\ p_2 - \sin(t) \\ \psi_2^1 - \cos(t) \\ \psi_2^2 + \sin(t) \\ \psi_2^3 + \cos(t) \\ \psi_2^4 - \sin(t) \end{bmatrix} \tag{27}$$

with

$$\begin{aligned} z &= \begin{bmatrix} p_1 & p_2 & p_3 & v_1 & v_2 & v_3 & F \end{bmatrix}^T, \\ \psi_1 &= \begin{bmatrix} \psi_1^1 & \psi_1^2 & \psi_2^1 & \psi_2^2 & \psi_2^3 & \psi_2^4 & \psi_3^2 & \psi_4^1 & \psi_5^1 & \psi_5^2 & \psi_5^3 \end{bmatrix}^T. \end{aligned}$$

For the HVT $T^* = T_2$ marked by the dark gray boxes in the signature matrix (18) we have to introduce new algebraic variables as in the following table:

for j	with i s.t. $(i,j) \in T^*$	replace derivatives $z_j^{(\sigma_{ij}+1)}$, ..., $z_j^{(d_j)}$	with new algebraic variables $\psi_j^{\sigma_{ij}+1}$, ..., $\psi_j^{d_j}$
1	1	$z_1^{(2)}$	ψ_1^2
2	7	$z_2^{(1)}, ..., z_2^{(4)}$	$\psi_2^1, ..., \psi_2^4$
3	5	$z_3^{(1)}, z_3^{(2)}$	ψ_3^1, ψ_3^2
4	4	$\emptyset$	$\emptyset$
5	2	$z_5^{(1)}, z_5^{(2)}, z_5^{(3)}$	$\psi_5^1, \psi_5^2, \psi_5^3$
6	3	$z_6^{(1)}$	ψ_6^1
7	6	$\emptyset$	$\emptyset$

Then we construct the structurally extended formulation as

$$0 = \mathcal{S}_2(t, z, \dot{z}, \psi_2) = \begin{bmatrix} \dot{p}_1 - v_1 \\ \psi_1^2 - \dot{v}_1 \\ \psi_2^1 - v_2 \\ \psi_2^2 - \psi_5^1 \\ \psi_2^3 - \psi_5^2 \\ \psi_2^4 - \psi_5^3 \\ \psi_3^1 - v_3 \\ \psi_3^2 - \psi_6^1 \\ m\dot{v}_1 + c(p_1 - p_2) - F \\ m\psi_5^1 - c(p_1 - p_2) + c(p_2 - p_3) \\ m\psi_5^2 - c(\dot{p}_1 - \psi_2^1) + c(\psi_2^1 - \psi_3^1) \\ m\psi_5^3 - c(\psi_1^2 - \psi_2^2) + c(\psi_2^2 - \psi_3^2) \\ m\psi_6^1 - c(p_2 - p_3) - F \\ p_2 - \sin(t) \\ \psi_2^1 - \cos(t) \\ \psi_2^2 + \sin(t) \\ \psi_2^3 + \cos(t) \\ \psi_2^4 - \sin(t) \end{bmatrix} \tag{28}$$

with

$$\begin{aligned} z &= \begin{bmatrix} p_1 & p_2 & p_3 & v_1 & v_2 & v_3 & F \end{bmatrix}^T, \\ \psi_2 &= \begin{bmatrix} \psi_1^2 & \psi_2^1 & \psi_2^2 & \psi_2^3 & \psi_2^4 & \psi_3^1 & \psi_3^2 & \psi_5^1 & \psi_5^2 & \psi_5^3 & \psi_6^1 \end{bmatrix}^T. \end{aligned}$$

Let us again check the result by applying the Σ-method to the StE-formulations (27) and (28).

For the StE-formulation (27) we obtain the according signature matrix $\widetilde{\Sigma}_1$ and the canonical offset vectors $\widetilde{c}_1$ and $\widetilde{d}_1$ as

$$\widetilde{\Sigma}_1 = \left[\begin{array}{ccccccc|ccccccccccc}
- & - & - & \fcolorbox{lightgray}{gray}{0} & - & - & - & 0 & - & - & - & - & - & - & - & - & - & - \\
- & - & - & - & - & - & - & - & \colorbox{lightgray}{0} & - & - & - & - & - & \colorbox{gray}{0} & - & - & - \\
- & - & - & - & \fcolorbox{lightgray}{gray}{0} & - & - & - & - & 0 & - & - & - & - & - & - & - & - \\
- & - & - & - & - & - & - & - & - & - & 0 & - & - & - & - & \fcolorbox{lightgray}{gray}{0} & - & - \\
- & - & - & - & - & - & - & - & - & - & - & 0 & - & - & - & - & \fcolorbox{lightgray}{gray}{0} & - \\
- & - & - & - & - & - & - & - & - & - & - & - & 0 & - & - & - & - & \fcolorbox{lightgray}{gray}{0} \\
- & - & \fcolorbox{lightgray}{gray}{1} & - & - & 0 & - & - & - & - & - & - & - & - & - & - & - & - \\
- & - & - & - & - & \colorbox{lightgray}{1} & - & - & - & - & - & - & - & \colorbox{gray}{0} & - & - & - & - \\
0 & 0 & - & - & - & - & \colorbox{gray}{0} & - & - & - & - & - & - & - & \colorbox{lightgray}{0} & - & - & - \\
\fcolorbox{lightgray}{gray}{0} & 0 & 0 & - & - & - & - & - & - & - & - & - & - & - & - & 0 & - & - \\
- & - & 1 & - & - & - & - & \fcolorbox{lightgray}{gray}{0} & - & 0 & - & - & - & - & - & - & 0 & - \\
- & - & - & - & - & - & - & - & \colorbox{gray}{0} & - & 0 & - & - & \colorbox{lightgray}{0} & - & - & - & 0 \\
- & 0 & 0 & - & - & \colorbox{gray}{1} & \colorbox{lightgray}{0} & - & - & - & - & - & - & - & - & - & - & - \\
- & \fcolorbox{lightgray}{gray}{0} & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - \\
- & - & - & - & - & - & - & - & - & \fcolorbox{lightgray}{gray}{0} & - & - & - & - & - & - & - & - \\
- & - & - & - & - & - & - & - & - & - & \fcolorbox{lightgray}{gray}{0} & - & - & - & - & - & - & - \\
- & - & - & - & - & - & - & - & - & - & - & \fcolorbox{lightgray}{gray}{0} & - & - & - & - & - & - \\
- & - & - & - & - & - & - & - & - & - & - & - & \fcolorbox{lightgray}{gray}{0} & - & - & - & - & -
\end{array}\right], \quad \widetilde{c}_1 = \begin{bmatrix} 0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0 \end{bmatrix},$$

$$\widetilde{d}_1^T = \left[\begin{array}{cccccccccccccccccc} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

with the the highest value transversals marked with dark gray and light gray boxes. Furthermore, the value of $\widetilde{\Sigma}_1$ is $\mathrm{Val}(\widetilde{\Sigma}_1) = 2$. Since $\widetilde{c}_1 = 0$, the corresponding reduced derivative array coincides with the system (27), i.e., $\widetilde{\mathcal{F}}_1(t, \widetilde{x}_1, \dot{\widetilde{x}}_1) = \mathcal{S}_1(t, x, \dot{x}, \psi_1)$ with $\widetilde{x}_1^T = \left[\begin{array}{cc} x^T & \psi_1^T \end{array}\right]$ and the Σ-Jacobian is given by

$$\widetilde{\mathfrak{J}}_1 = \left[\begin{array}{ccccccc|ccccccccccc}
0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
c & -c & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & m & 0 & 0 & 0 \\
-c & 2c & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & m & 0 & 0 \\
0 & 0 & -c & 0 & 0 & 0 & 0 & -c & 0 & 2c & 0 & 0 & 0 & 0 & 0 & 0 & m & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -c & 0 & 2c & 0 & 0 & -c & 0 & 0 & 0 & m \\
0 & -c & 0 & 0 & 0 & m & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0
\end{array}\right] \tag{29}$$

with $\det(\widetilde{\mathfrak{J}}_1) = -2c^3m \neq 0$ independent of z and ψ_1, i.e., for all values. Therefore, the Σ-Jacobian $\widetilde{\mathfrak{J}}_1$ can be checked to be nonsingular at every point. Thus, the Σ-method for $\mathcal{S}_1$

succeeds globally with structural index $\nu_\Sigma = 1$.
For the StE-formulation (28) we obtain the according signature matrix $\widetilde{\Sigma}_2$ and the canonical offset vectors $\widetilde{c}_2$ and $\widetilde{d}_2$ as

$$\widetilde{\Sigma}_2 = \left[\begin{array}{ccccccc|ccccccccccc}
\colorbox{lightgray}{$\colorbox{gray}{$1$}$} & - & - & 0 & - & - & - & - & - & - & - & - & - & - & - & - & - & - \\
- & - & - & \colorbox{lightgray}{1} & - & - & - & \colorbox{gray}{0} & - & - & - & - & - & - & - & - & - & - \\
- & - & - & - & \colorbox{lightgray}{$\colorbox{gray}{$0$}$} & - & - & - & 0 & - & - & - & - & - & - & - & - & - \\
- & - & - & - & - & - & - & - & - & 0 & - & - & - & - & \colorbox{lightgray}{$\colorbox{gray}{$0$}$} & - & - & - \\
- & - & - & - & - & - & - & - & - & - & 0 & - & - & - & - & \colorbox{lightgray}{$\colorbox{gray}{$0$}$} & - & - \\
- & - & - & - & - & - & - & - & - & - & - & 0 & - & - & - & - & \colorbox{lightgray}{$\colorbox{gray}{$0$}$} & - \\
- & - & - & - & - & \colorbox{lightgray}{$\colorbox{gray}{$0$}$} & - & - & - & - & - & - & 0 & - & - & - & - & - \\
- & - & - & - & - & - & - & - & - & - & - & - & - & \colorbox{lightgray}{0} & - & - & - & \colorbox{gray}{0} \\
0 & 0 & - & \colorbox{gray}{1} & - & - & \colorbox{lightgray}{0} & - & - & - & - & - & - & - & - & - & - & - \\
0 & 0 & \colorbox{lightgray}{$\colorbox{gray}{$0$}$} & - & - & - & - & - & - & - & - & - & - & - & 0 & - & - & - \\
1 & - & - & - & - & - & - & - & 0 & - & - & - & \colorbox{lightgray}{$\colorbox{gray}{$0$}$} & - & - & 0 & - & - \\
- & - & - & - & - & - & - & \colorbox{lightgray}{0} & - & 0 & - & - & - & \colorbox{gray}{0} & - & - & 0 & - \\
- & 0 & 0 & - & - & - & 0 & - & - & - & - & - & - & - & - & - & - & \colorbox{lightgray}{$\colorbox{gray}{$0$}$} \\
- & \colorbox{lightgray}{$\colorbox{gray}{$0$}$} & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - \\
- & - & - & - & - & - & - & - & \colorbox{lightgray}{$\colorbox{gray}{$0$}$} & - & - & - & - & - & - & - & - & - \\
- & - & - & - & - & - & - & - & - & \colorbox{lightgray}{$\colorbox{gray}{$0$}$} & - & - & - & - & - & - & - & - \\
- & - & - & - & - & - & - & - & - & - & \colorbox{lightgray}{$\colorbox{gray}{$0$}$} & - & - & - & - & - & - & - \\
- & - & - & - & - & - & - & - & - & - & - & \colorbox{lightgray}{$\colorbox{gray}{$0$}$} & - & - & - & - & - & -
\end{array}\right], \quad \widetilde{c}_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$$\widetilde{d}_2^T = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

again with the highest value transversals marked with dark gray and light gray boxes. Furthermore, the value of $\widetilde{\Sigma}_2$ is $\mathrm{Val}(\widetilde{\Sigma}_2) = 2$. Again, since $\widetilde{c}_2 = 0$, the corresponding reduced derivative array coincides with the system (28), i.e., $\widetilde{\mathcal{F}}_2(t, \widetilde{x}_2, \dot{\widetilde{x}}_2) = \mathcal{S}_2(t, x, \dot{x}, \psi_2)$ with $\widetilde{x}_2^T = \begin{bmatrix} x^T & \psi_2^T \end{bmatrix}$ and the Σ-Jacobian is given by

$$\widetilde{\mathfrak{J}}_2 = \left[\begin{array}{ccccccc|ccccccccccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \\
0 & -c & 0 & m & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 2c & -c & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & m & 0 & 0 & 0 \\
-c & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2c & 0 & 0 & 0 & -c & 0 & 0 & m & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -c & 0 & 2c & 0 & 0 & 0 & -c & 0 & 0 & m & 0 \\
0 & -c & c & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & m \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right] \qquad (30)$$

also with $\det(\widetilde{\mathfrak{J}}_2) = -2c^3 m \neq 0$ independent of z and ψ_2, i.e., for all values. Therefore, also

the Σ-Jacobian $\widetilde{\mathfrak{J}}_2$ can be checked to be nonsingular at every point. Thus, the Σ-method for $\mathcal{S}_2$ succeeds globally with structural index $\nu_\Sigma = 1$.
Summarizing, we get two different but globally valid regularizations of the original DAE (16) in form of the structurally extended formulations $\mathcal{S}_1$ (27) and $\mathcal{S}_2$ (28). Due to the global validity a change of the formulation has not to be performed during the numerical integration. ◁

3.2 Regularization via Algebraic Derivative Arrays

As we have seen in the previous section, in particular, in Example 3.1, the structurally extended regularization in general is only locally valid. Therefore, it may be necessary to switch to another structurally extended formulation within a numerical integration. This influences the time integration negatively. As a remedy in this section we propose an extended regularization.
As mentioned in Definition 2.3 the reduced derivative array (4) contains all necessary equations to obtain all hidden constraints. In particular, from the algebraic reduced derivative array (7) it is possible to extract the array of (hidden) constraints (6) in an algebraic way which, in particular, can be done automatically within the numerical integration.
From the algebraic reduced derivative array (7) in combination with the original DAE (1) we obtain the *algebraic derivative array (ADA) formulation*

$$0 = \mathcal{A}(t, z, \dot{z}, \omega) = \begin{bmatrix} F(t, z, \dot{z}) \\ \mathcal{F}(t, z, \omega) \end{bmatrix} \tag{31}$$

with ω as in (7) which forms an overdetermined DAE for z and ω and is uniquely solvable for consistent initial values $z(t_0)$, $w(t_0)$ as long as the original DAE was uniquely solvable for consistent initial values $z(t_0)$.
According to Remark 2.4, the algebraic derivative array formulation (31) is solution equivalent to (1) with

$$\Phi : z \mapsto \begin{bmatrix} z \\ \Omega(t, z) \end{bmatrix}.$$

Therefore, the algebraic derivative array formulation (31) has the same set of solutions for the original unknowns z as the original DAE (1). Since the structural index is not defined for (31) due to its overdetermindness it does not fit into the Definition 2.6 of structural regularization. Nevertheless, (31) can be seen as a regularization of (1) due to the solution equivalence and the fact that all hidden constraints from (1) are contained in the lower part of (31). Therefore, (31) has no further hidden constraints.
The obtained algebraic derivative array formulation (31) is of more increased size as the structurally extended formulation (22). This increases the needed simulation time. On the other hand, the obtained algebraic derivative array formulation (31) is globally valid such that the regularized formulation has to be determined only once before the numerical integration in contrast to the structurally extended formulation (22). This regularization approach via algebraic derivative arrays also can be automated using automatic or symbolic differentiation tools. Therefore, an analytical preprocessing is not necessary.

Example 3.3 Let us use the model equations (10) of the simple pendulum to illustrate the regularization based on the algebraic derivative array formulation. The reduced derivative array is given in (14). Replacing all occurring derivatives with algebraic variables, i.e., $\dot{x} \to \omega_1^1$, $\ddot{x} \to \omega_1^2$, $\dot{y} \to \omega_2^1$, $\ddot{y} \to \omega_2^2$, $\dot{v} \to \omega_3^1$, and $\dot{w} \to \omega_4^1$, leads to the algebraic reduced derivative array

$$0 = \mathcal{F}(t, z, \omega) = \begin{bmatrix} \omega_1^1 - v \\ \omega_1^2 - \omega_3^1 \\ \omega_2^1 - w \\ \omega_2^2 - \omega_4^1 \\ m\omega_3^1 + 2x\lambda \\ m\omega_4^1 + 2y\lambda + m\mathfrak{g} \\ x^2 + y^2 - L^2 \\ 2(x\omega_1^1 + y\omega_2^1) \\ 2(x\omega_1^2 + (\omega_1^1)^2 + y\omega_2^2 + (\omega_2^1)^2) \end{bmatrix} \tag{32}$$

with

$$z = \begin{bmatrix} x & y & v & w & \lambda \end{bmatrix}^T, \tag{33a}$$

$$\omega = \begin{bmatrix} \omega_1^1 & \omega_1^2 & \omega_2^1 & \omega_2^2 & \omega_3^1 & \omega_4^1 \end{bmatrix}^T. \tag{33b}$$

Together with the original DAE (10), we get the algebraic derivative array formulation (31) with $F(t, z, \dot{z})$ given in (10) and $\mathcal{F}(t, z, \omega)$ given in (32) as

$$0 = \mathcal{A}(t, z, \dot{z}, \omega) = \left[\begin{array}{c} \dot{x} - v \\ \dot{y} - w \\ m\dot{v} + 2x\lambda \\ m\dot{w} + m\mathfrak{g} + 2y\lambda \\ x^2 + y^2 - L^2 \\ \hline \omega_1^1 - v \\ \omega_1^2 - \omega_3^1 \\ \omega_2^1 - w \\ \omega_2^2 - \omega_4^1 \\ m\omega_3^1 + 2x\lambda \\ m\omega_4^1 + 2y\lambda + m\mathfrak{g} \\ x^2 + y^2 - L^2 \\ 2(x\omega_1^1 + y\omega_2^1) \\ 2(x\omega_1^2 + (\omega_1^1)^2 + y\omega_2^2 + (\omega_2^1)^2) \end{array}\right] \tag{34}$$

with (33). This corresponds to an overdetermined system of DAEs containing 14 equations for 11 unknowns. ◁

Example 3.4 For the model equations (16) for the mass-spring-chain with path control we obtain the algebraic reduced derivative array from (20) by replacing all derivatives of z by the algebraic representations, i.e., $\dot{p}_1 \to \omega_1^1$, $\ddot{p}_1 \to \omega_1^2$, $\dot{p}_2 \to \omega_2^1$, $\ddot{p}_2 \to \omega_2^2$, $p_2^{(3)} \to \omega_2^3$, $p_2^{(4)} \to \omega_2^4$, $\dot{p}_3 \to \omega_3^1$, $\ddot{p}_3 \to \omega_3^2$, $\dot{v}_1 \to \omega_4^1$, $\dot{v}_2 \to \omega_5^1$, $\ddot{v}_2 \to \omega_5^2$, $v_2^{(3)} \to \omega_5^3$, $\dot{v}_3 \to \omega_6^1$. We get the algebraic reduced derivative array as

$$0 = \mathcal{F}(t, z, \omega) = \begin{bmatrix} \omega_1^1 - v_1 \\ \omega_1^2 - \omega_4^1 \\ \omega_2^1 - v_2 \\ \omega_2^2 - \omega_5^1 \\ \omega_2^3 - \omega_5^2 \\ \omega_2^4 - \omega_5^3 \\ \omega_3^2 - v_3 \\ \omega_3^2 - \omega_6^1 \\ m\omega_4^1 + c(p_1 - p_2) - F \\ m\omega_5^1 - c(p_1 - p_2) + c(p_2 - p_3) \\ m\omega_5^2 - c(\omega_1^1 - \omega_2^1) + c(\omega_2^1 - \omega_3^1) \\ m\omega_5^3 - c(\omega_1^2 - \omega_2^2) + c(\omega_2^2 - \omega_3^2) \\ m\omega_6^1 - c(p_2 - p_3) - F \\ p_2 - \sin(t) \\ \omega_2^1 - \cos(t) \\ \omega_2^2 + \sin(t) \\ \omega_2^3 + \cos(t) \\ \omega_2^4 - \sin(t) \end{bmatrix}$$

with

$$z = \begin{bmatrix} p_1 & p_2 & p_3 & v_1 & v_2 & v_3 & F \end{bmatrix}^T, \tag{35a}$$

$$\omega = \begin{bmatrix} \omega_1^1 & \omega_1^2 & \omega_2^1 & \omega_2^2 & \omega_2^3 & \omega_2^4 & \omega_3^1 & \omega_3^2 & \omega_4^1 & \omega_5^1 & \omega_5^2 & \omega_5^3 & \omega_6^1 \end{bmatrix}^T. \tag{35b}$$

Together with the original model equations (16) we obtain the algebraic derivative array formulation (31) as

$$0 = \mathcal{A}(t, z, \dot{z}, \omega) = \left[\begin{array}{c} \dot{p}_1 - v_1 \\ \dot{p}_2 - v_2 \\ \dot{p}_3 - v_3 \\ m\dot{v}_1 + c(p_1 - p_2) - F \\ m\dot{v}_2 - c(p_1 - p_2) + c(p_2 - p_3) \\ m\dot{v}_3 - c(p_2 - p_3) - F \\ p_2 - \sin(t) \\ \hline \omega_1^1 - v_1 \\ \omega_1^2 - \omega_4^1 \\ \omega_2^1 - v_2 \\ \omega_2^2 - \omega_5^1 \\ \omega_2^3 - \omega_5^2 \\ \omega_2^4 - \omega_5^3 \\ \omega_3^2 - v_3 \\ \omega_3^2 - \omega_6^1 \\ m\omega_4^1 + c(p_1 - p_2) - F \\ m\omega_5^1 - c(p_1 - p_2) + c(p_2 - p_3) \\ m\omega_5^2 - c(\omega_1^1 - \omega_2^1) + c(\omega_2^1 - \omega_3^1) \\ m\omega_5^3 - c(\omega_1^2 - \omega_2^2) + c(\omega_2^2 - \omega_3^2) \\ m\omega_6^1 - c(p_2 - p_3) - F \\ p_2 - \sin(t) \\ \omega_2^1 - \cos(t) \\ \omega_2^2 + \sin(t) \\ \omega_2^3 + \cos(t) \\ \omega_2^4 - \sin(t) \end{array}\right] \tag{36}$$

with (35). This corresponds to an overdetermined system of DAEs containing 25 equations for 20 unknowns. ◁

3.3 Regularization via Hidden Constraints

In the previous sections, we did obtain regularizations that increased in its size more then necessary. Therefore, in the numerical integration more computational time is needed than necessary.
As mentioned in Definition 2.3 the reduced derivative array (4) allows the determination of the array of (hidden) constraints (6) in an analytical preprocessing. From the array of (hidden) constraints (6) in combination with the original DAE (1), we obtain the set

$$0 = \mathcal{O}(t, z, \dot{z}) = \begin{bmatrix} F(t, z, \dot{z}) \\ H(t, z) \end{bmatrix}, \tag{37}$$

which forms the *hidden constraints (HiC) formulation* for z and is uniquely solvable for consistent initial values $z(t_0)$ as long as the original DAE was uniquely solvable for these consistent initial values $z(t_0)$. The obtained overdetermined formulation is of increased number of equations for the same unknown variables as in the original DAE. For more details see [22].
The hidden constraints formulation (37) is solution equivalent to the original DAE (1) with

$$\Phi : z \mapsto z.$$

Therefore, (37) has the same set of solutions as the original DAE (1). Since the structural index is not defined for the hidden constraints formulation (37) due to its overdetermindness, also (37) does not satisfy the Definition 2.6 of a structural regularization. Nevertheless, (37) can be seen as a regularization of (1) due to the solution equivalence and the fact that all hidden constraints from (1) are contained in (37) explicitly as algebraic equations. Therefore, (37) has no further hidden constraints.

Example 3.5 For the simple pendulum, see Example 2.7, the (hidden) constraints are known to be (10e), (11a), and (11b) summarized in the array of (hidden) constraints (15). Then the regularization via hidden constraints is determined by (10), (15) in the form

$$0 = \mathcal{O}(t, z, \dot{z}) = \left[\begin{array}{c} \dot{x} - v \\ \dot{y} - w \\ m\dot{v} + 2x\lambda \\ m\dot{w} + m\mathfrak{g} + 2y\lambda \\ x^2 + y^2 - \ell^2 \\ \hline x^2 + y^2 - \ell^2 \\ 2xv + 2yw \\ v^2 + w^2 - \frac{2}{m}(x^2 + y^2)\lambda - y\mathfrak{g} \end{array}\right] \tag{38}$$

as a system of DAEs containing 8 equations for 5 unknowns. ◁

Example 3.6 For the mass-spring-chain with path control, see Example 2.8, we get the array of (hidden) constraints in (21). Consequently, the regularization via hidden constraints is determined by (16), (21) in the form

$$0 = \mathcal{O}(t, z, \dot{z}) = \left[\begin{array}{c} \dot{p}_1 - v_1 \\ \dot{p}_2 - v_2 \\ \dot{p}_3 - v_3 \\ m\dot{v}_1 + c(p_1 - p_2) - F \\ m\dot{v}_2 - c(p_1 - p_2) + c(p_2 - p_3) \\ m\dot{v}_3 - c(p_2 - p_3) - F \\ p_2 - \sin(t) \\ \hline p_2 - \sin(t) \\ v_2 - \cos(t) \\ \frac{c}{m}(p_1 - p_2) - \frac{c}{m}(p_2 - p_3) + \sin(t) \\ \frac{c}{m}(v_1 - v_2) - \frac{c}{m}(v_2 - v_3) + \cos(t) \\ \frac{c}{m^2}(-3c(p_1 - p_2) + 3c(p_2 - p_3) + 2F) - \sin(t) \end{array}\right]$$

as a system of DAEs containing 12 equations for 7 unknowns. ◁

There exist further (algebraic) approaches for the determination of the hidden constraints. In particular, let us mention the strangeness-index concept [13] which is based on rank decisions and the evaluation of kernels and cokernels of the Jacobians of the derivative array. Furthermore, the constraint level concept [22] determines the hidden constraints within an iterative procedure.

4 The Software Package QUALIDAES

In Sections 3.2 and 3.3 the proposed regularization approaches lead to overdetermined formulations in form of the algebraic derivative array formulation (31) and the hidden constraints formulation (37). Since in these formulations the number of equations and the number of unknowns are not the same, usually used numerical methods, like `DASSL` [6], `DASPK` [25], `MEBDF` [1], or `RADAU5` [10, 11] cannot be applied. Therefore, adapted methods suited for overdetermined DAEs have to be used, like `ODASSL` [7, 8], `OVDBDF` [5], or `QUALIDAES` [24].
In particular, the software package `QUALIDAES` (`QUA`si `LI`near `DAE` `S`olver) is suited for the numerical integration of (overdetermined) quasi-linear DAEs of the form

$$\left[\begin{array}{c} E(t,z) \\ \hline 0 \end{array}\right] \dot{z} = \left[\begin{array}{c} k(t,z) \\ \hline g(t,z) \end{array}\right] \quad \begin{array}{l} \text{differential part,} \\ \text{algebraic constraints.} \end{array} \tag{39}$$

with $E : \mathbb{I} \times \mathbb{R}^n \to \mathbb{R}^{m_d \times n}$, $k : \mathbb{I} \times \mathbb{R}^n \to \mathbb{R}^{m_d}$, $g : \mathbb{I} \times \mathbb{R}^n \to \mathbb{R}^{m_c}$, $z : \mathbb{I} \to \mathbb{R}^n$, and $m_d + m_c \geq n$, with $t \in \mathbb{I}$, where $\mathbb{I} \subset \mathbb{R}$ is the compact interval. For the usage of `QUALIDAES` it is required that the algebraic equations $0 = g(t,z)$ are nonredundant, i.e., $\operatorname{rank}(\frac{\partial g}{\partial z}) = m_c$ for all $(t,z) \in \mathbb{M} = \{(t,z) \in \mathbb{I} \times \mathbb{R}^n \ : \ 0 = g(t,z)\}$.
`QUALIDAES` is implemented in FORTRAN. The discretization scheme is the 3-stage Runge-Kutta RADAU IIa of order 5 ,e.g., see [10, 11]. The used decomposition methods in `QUALIDAES` is adapted w.r.t. the overdetermined structure (39) such that the (hidden) algebraic constraints $0 = g(t,z)$ are solved (numerically) precisely, while the differential part $E(t,z)\dot{z} = k(t,z)$ is solved in an 'approximative sense'.
Further features of `QUALIDAES` are variable step size control, continuous output, and check and correction of the initial values with respect to its consistency. The specification of the DAE (39) can be done in FORTRAN source code or alternatively in MODELICA using a simple MODELICA-parser. For MODELICA a Matlab interface is provided, see [2]. For more information to `QUALIDAES` we refer to [24].

5 Numerical Results

In the following, we illustrate the efficiency of the proposed regularization techniques for the numerical integration with `QUALIDAES` on two examples. The first one is the simple pendulum (see Example 2.7) and the second one is the mass-spring-chain with path constraints (see Example 2.8).
The numerical integrations are done on an AMD Phenom(tm) II X6 1090T, 3210 MHz, 16GB RAM, openSuSE 13.1 (Linux 3.11.10), GNU Fortran compiler gcc version 4.8.1, no compiler options.

Example 5.1 The model equations of the simple pendulum, see Example 2.7, are stated in (10) and have structural index $\nu_\Sigma = 3$. The mass and length are given by $m = 1$ and $\ell = 1$ while we use the gravitational acceleration $\mathfrak{g} = 13.7503716373294544$. In this case the exact solution has a period of 2s which allows the comparison of the accuracy every period. The initial values are given by

$$\begin{array}{rclrclrcl} x(0) &=& 1, & v(0) &=& 0, & \lambda(0) &=& 0, \\ y(0) &=& 0, & w(0) &=& 0. \end{array}$$

In Figure 4 the solution for $t = 2i$, $i \in \{0, 1, ..., 1000\}$ is illustrated. In this figure only the solution after every period is plotted such that the curves are illustrated to be constant.

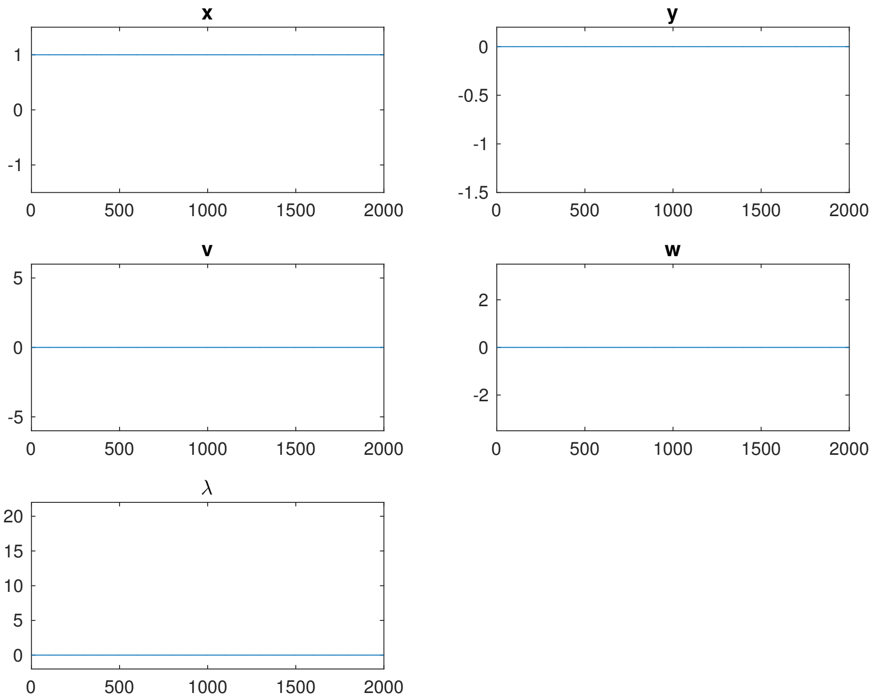

Figure 4: Solution for the simple pendulum (illustrated at every period, i.e., at $t = 2i$, $i \in \{0, 1, ..., 1000\}$)

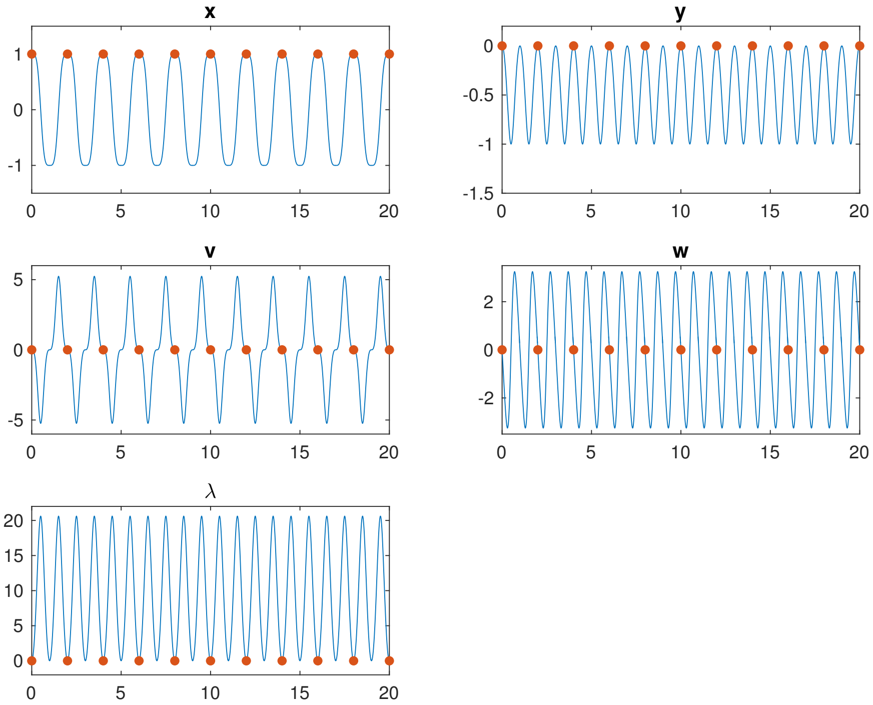

Figure 5: Solution for the simple pendulum (zoom)

Furthermore, in Figure 5 the solution is illustrated for the first 10 periods, i.e., in $\mathbb{I} = [0\text{s}, 20\text{s}]$. The numerical simulation is done for the time domain $\mathbb{I} = [0\text{s}, 2000\text{s}]$. In the following we investigate the suitability of the proposed regularization approaches.

The numerical results for the proposed regularization approaches are obtained with the numerical integration by QUALIDAES. The proposed regularized formulations for the simple pendulum are

- the structurally extended (StE) formulations given in (23) and (24) depending on the current position (x, y) of the pendulum mass, i.e,

  $$\mathcal{S}(t,z,\dot{z},\omega) = \begin{cases} \mathcal{S}_1(t,z,\dot{z},\omega_1) & \text{for } |x| > |y|, \\ \mathcal{S}_2(t,z,\dot{z},\omega_1) & \text{else,} \end{cases}$$

  where the case criterion $(|x| > |y|)$ is evaluated only at the beginning of an integration step within the numerical integration,

- the algebraic derivative array (ADA) formulation given in (34), and

- the hidden constraints (HiC) formulation given in (38).

In comparison we use the widely used solvers RADAU5, with the Gear-Gupta-Leimkuler (GGL) formulation [9] given by

$$0 = \mathcal{G}(t,z,\dot{z},\mu) = \begin{bmatrix} \dot{x} - v + 2x\mu \\ \dot{y} - w + 2y\mu \\ m\dot{v} + 2x\lambda \\ m\dot{w} + 2y\lambda + m\mathfrak{g} \\ x^2 + y^2 - \ell^2 \\ 2xv + 2yw \end{bmatrix} \tag{40}$$

and DASSL with the d-index-1 (DI1) formulation consisting the differential equations (10a)-(10d) and the constraint on acceleration level (11b), i.e.,

$$0 = \mathcal{D}(t,z,\dot{z}) = \begin{bmatrix} \dot{x} - v \\ \dot{y} - w \\ m\dot{v} + 2x\lambda \\ m\dot{w} + 2y\lambda + m\mathfrak{g} \\ 2v^2 + 2w^2 - 2y\mathfrak{g} - \frac{4}{m}(x^2 + y^2)\lambda \end{bmatrix}. \tag{41}$$

Note that the GGL formulation (40) also can be seen as a regularization of the model equations (10) since both are solution equivalent and the number of hidden constraints is lowered in the GGL formulation (40) in comparison to the original DAE (10). On the other hand, the d-index-1 formulation (41) is only a so called *index reduction* since the index is reduced and with that also the number of the hidden constraints but the d-index-1 formulation (41) is not solution equivalent to the the model equations (10) due to the loss of the constraints on level 0 and on level 1. Therefore, the dimension of the set of solutions is increased and we expect that the numerical solution for the the d-index-1 formulation (41) drifts from the set of consistency, i.e., violates the constraints from the model equations (10).

The numerical solution (for every period, i.e., for $t = 2i$, $i \in \{0, 1, ..., 1000\}$) for a prescribed tolerance TOL=10^{-7} and the corresponding absolute errors are illustrated in Figure 6 and Figure 7, respectively. Comparing Figure 4 with Figure 6 it is obvious that the numerical solution DASSL(DI1) deviates strongly from the exact solution.

An important issue in the numerical treatment of DAEs are the compliance with the constraints as well as the energy balance. In particular, the simple pendulum is a conservative dynamical system such that the energy is invariant for every solutions which should be reflected by the numerical results. In Figure 8 the residual of the constraints as well as the energy conservation, i.e., the deviation from the initial energy, is illustrated. The numerical results DASSL(DI1) are drifting from the constraints on level 0 (position level) as well as from the constraints on level 1 (velocity level). The drift from the constraints are provoked

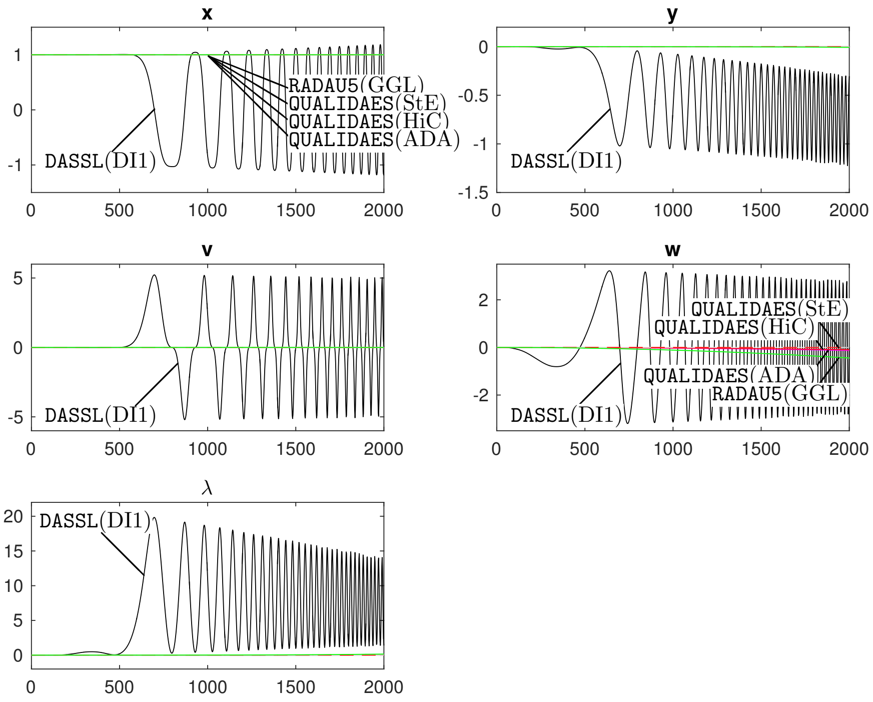

Figure 6: Numerical solution for the simple pendulum (for TOL=10^{-7})

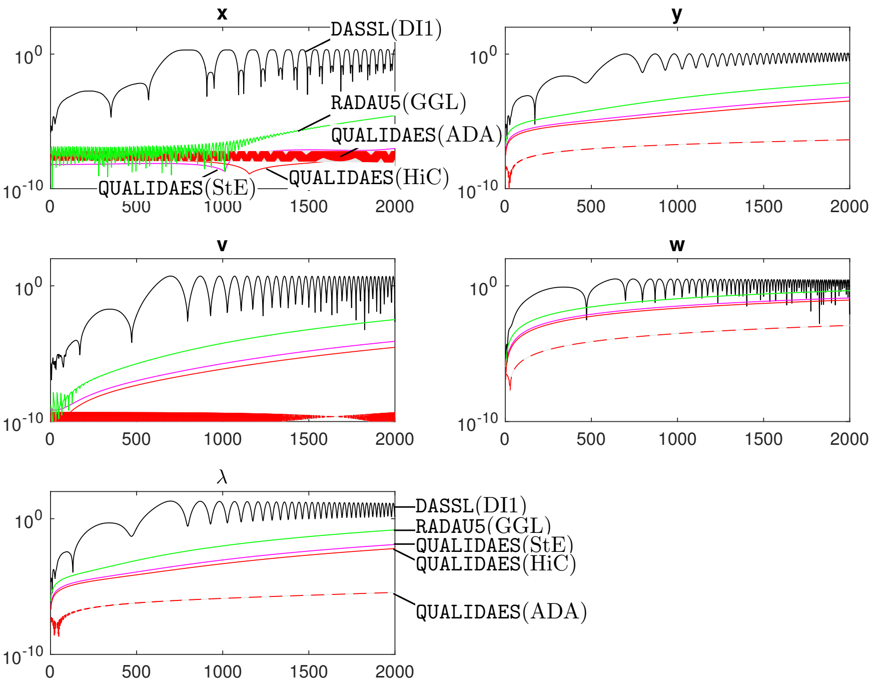

Figure 7: Absolute error for the simple pendulum (for TOL=10^{-7})

from the used formulation (DI1) (41). In this formulation the constraints on level 0 and level 1 are no longer contained (even not hidden). Therefore, the solver cannot satisfy these constraints. The other numerical solutions satisfy the constraints within the prescribed tol-

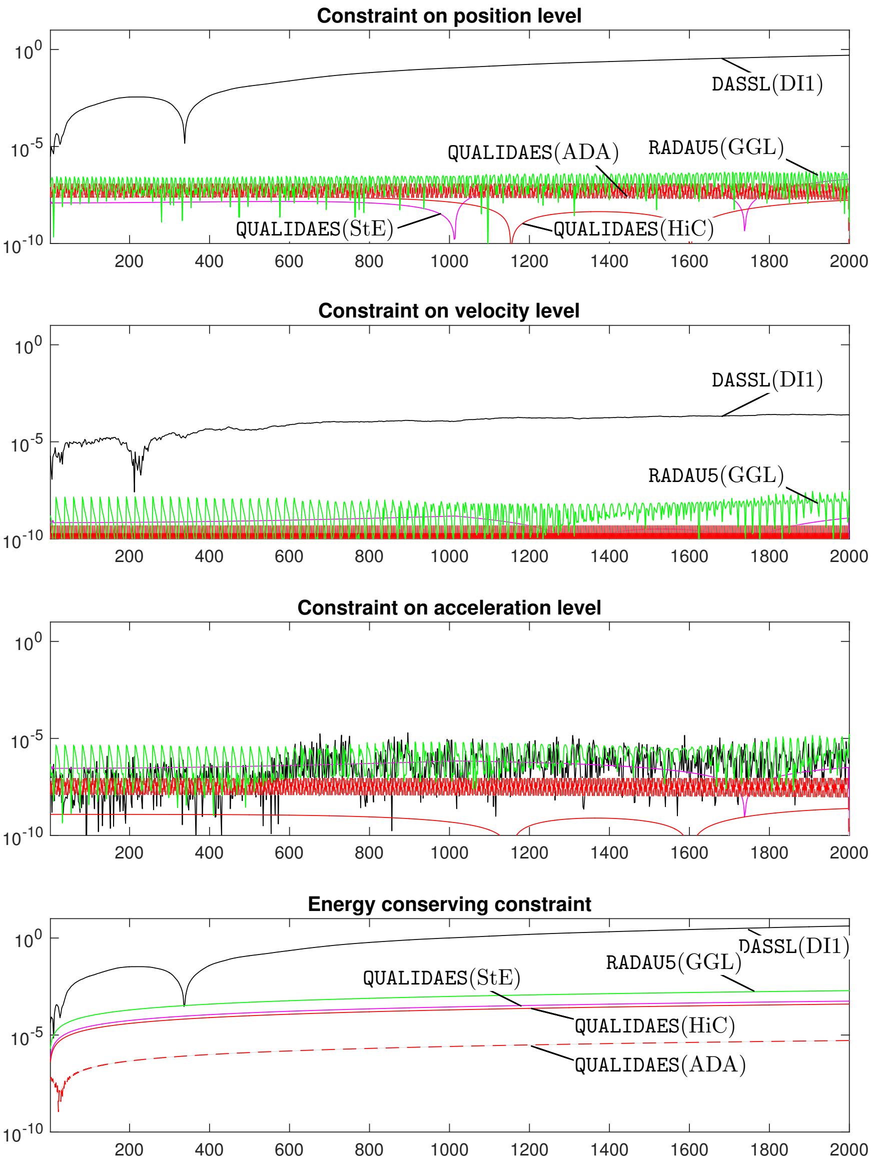

Figure 8: Residual of the constraints for the simple pendulum (for TOL=10^{-7})

erance. The best energy conservations are reached with QUALIDAES, while we get the best energy conservation for the structurally extended formulation (StE).

In Figure 9 the efficiency of the proposed regularization approaches with the numerical integration by QUALIDAES in comparison to the numerical results obtained by DASSL and RADAU5 is shown.

The numerical solution QUALIDAES(HiC) offers the best efficiency, i.e., less computational effort and small absolute error of the numerical solution. This numerical solution is slightly more efficient than the numerical result obtained with RADAU5(GGL) followed by the numerical solutions of QUALIDAES(ADA) and QUALIDAES(StE). The large time consumption for the integration of the StE-formulation lies in its local validity as discussed in Example 3.1. Within one period of 2 seconds it has to be switched between (23) and (24) four times, see Figure 3. Furthermore, due to the increased size of the ADA-formulation and the StE-formulation both are less efficient than the HiC-formulation. The numerical integration using DASSL is not successful at all due to the stability properties of BDF methods, the

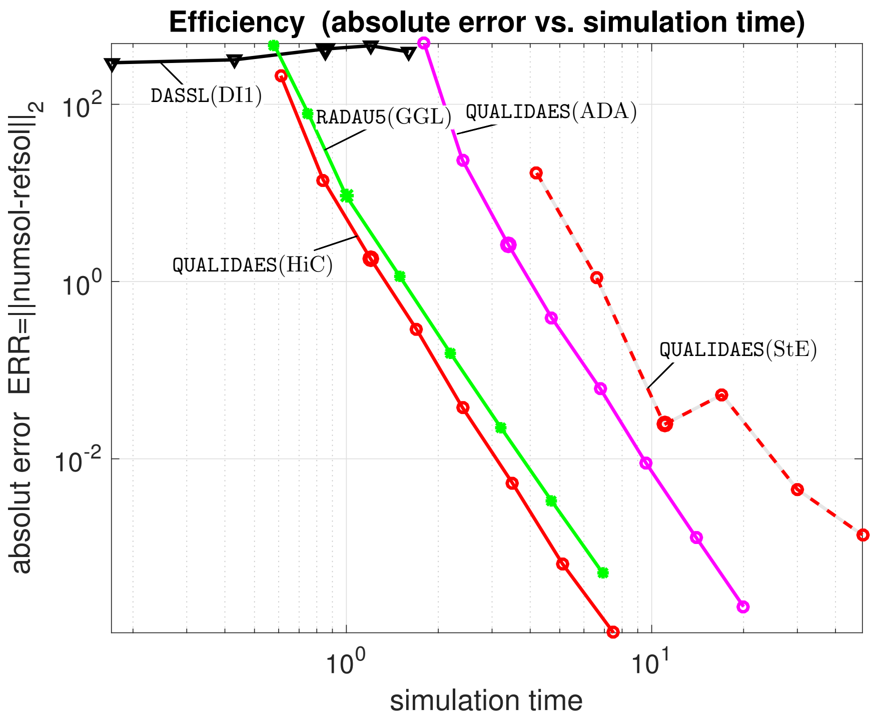

Figure 9: Efficiency (simulation time vs. absolute error for different prescribed tolerances) for the simple pendulum

occurring drift off effect by use of the d-index-1 formulation (DI1), where the constraints on position level and on velocity level are lost, and the large time domain $\mathbb{I}$. Nevertheless, the maximally obtained precision is excellent for `QUALIDAES`(HiC) and `QUALIDAES`(ADA). ◁

Example 5.2 For the example of the mass-spring-chain we use the mass $m = 1$ for all masses and the stiffness $c = 1/6$ for both springs. With initial values

$$\begin{array}{rclrclrcl} p_1(0) &=& 0, & v_1(0) &=& -2, & F(0) &=& 0, \\ p_2(0) &=& 0, & v_2(0) &=& 1, & & & \\ p_3(0) &=& 0, & v_3(0) &=& -2 & & & \end{array}$$

we get the exact solution

$$\begin{array}{rclrclrcl} p_1(t) &=& -2\sin(t), & v_1(t) &=& -2\cos(t), & F(t) &=& \frac{3}{2}\sin(t), \\ p_2(t) &=& \sin(t), & v_2(t) &=& \cos(t), & & & \\ p_3(t) &=& -2\sin(t), & v_3(t) &=& -2\cos(t). & & & \end{array}$$

The simulation is done for the time domain $\mathbb{I} = [0\text{s}, 400\text{s}]$.
For the numerical integrations with `QUALIDAES` we use the proposed regularizations for the mass-spring-chain, i.e.,

- the structurally extended (StE) formulation given in (27) which is globally valid for the mass-spring-chain,
- the algebraic derivative array (ADA) formulation given in (36), and
- the hidden constraints (HiC) formulation given in (39).

Furthermore, for the numerical integrations with `DASSL` and `RADAU5` we use the d-index-1 (DI1) formulation consisting the differential equations (16a)-(16f) and the constraint on

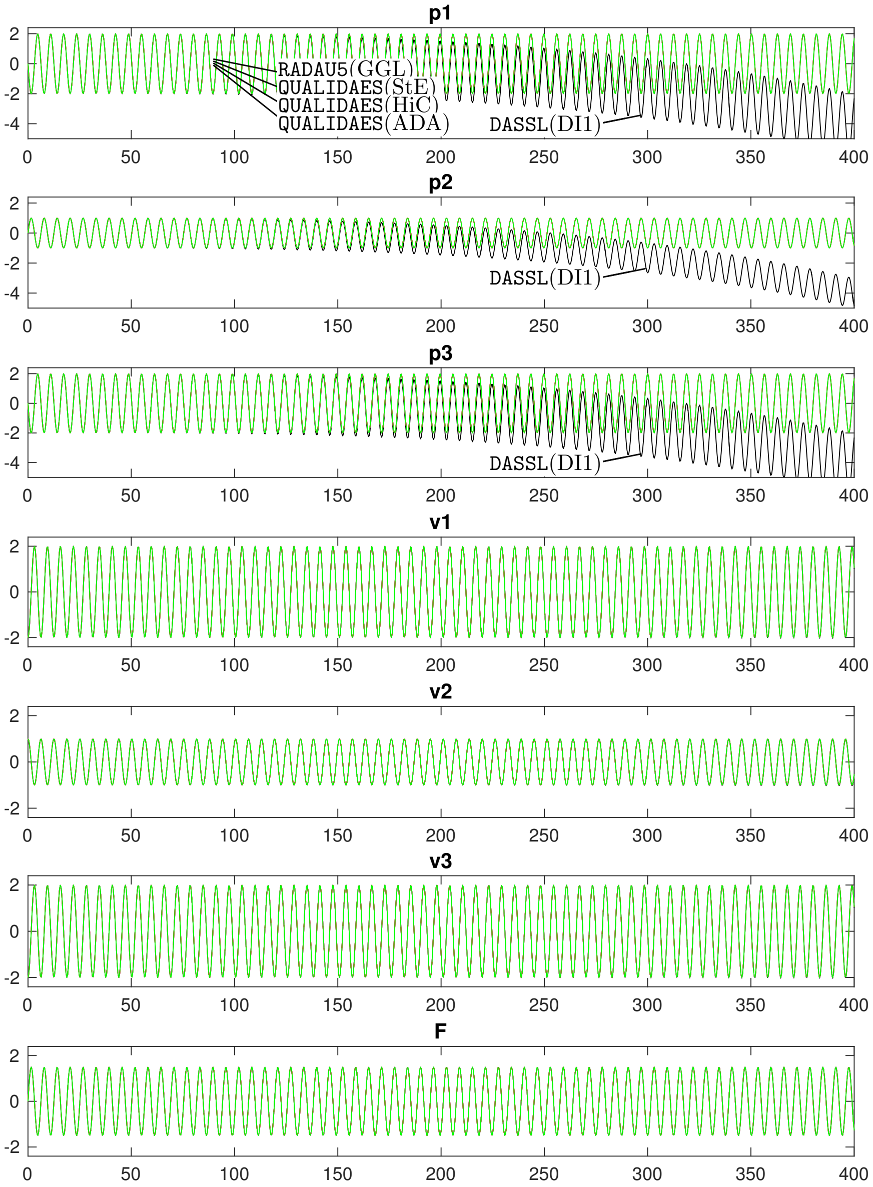

Figure 10: Numerical solution for the mass-spring-chain (for TOL=10^{-7})

level 4 (17d), i.e.,

$$0 = \mathcal{D}(t, z, \dot{z}) = \begin{bmatrix} \dot{p}_1 - v_1 \\ \dot{p}_2 - v_2 \\ \dot{p}_3 - v_3 \\ m\dot{v}_1 + c(p_1 - p_2) - F \\ m\dot{v}_2 - c(p_1 - p_2) + c(p_2 - p_3) \\ m\dot{v}_3 - c(p_2 - p_3) - F \\ \frac{c}{m^2}(-3c(p_1 - p_2) + 3c(p_2 - p_3) + 2F) - \sin(t) \end{bmatrix}. \tag{42}$$

Note again, that the d-index-1 formulation (42) is not a regularization but only a so called index reduction and we expect drift for the numerical results obtained from the d-index-1 formulation (42).

The numerical solution for a prescribed tolerance TOL=10^{-7} and the corresponding absolute errors are illustrated in Figure 10 and Figure 11, respectively. Also for this example the

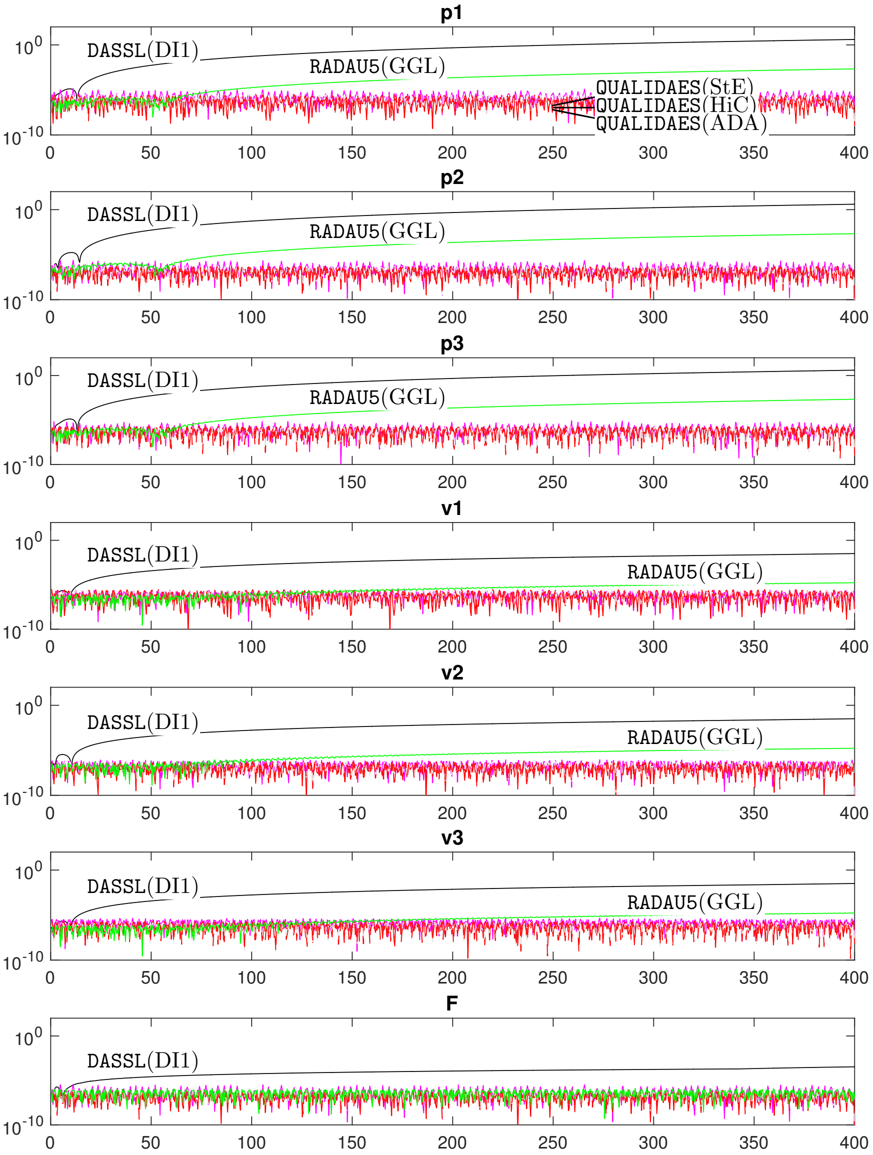

Figure 11: Absolute error for the mass-spring-chain (for TOL=10^{-7})

numerical solution DASSL(DI1) deviates strongly from the exact solution.
The compliance with the constraints is illustrated in Figure 12. While the constraints are satisfied according to the prescribed tolerance by the usage of the proposed regularizations, the numerical solutions for the d-index-1 formulation (DI1) drifts away from all except the constraint on level 4. Here the drift is less intensive for RADAU5 with (DI1). Note, all constraints except the constraint on level 4 are removed within the d-index-1 formulation (DI1).
In the same style as in the last example, the efficiency of the numerical solutions is illustrated in Figure 13. Again the numerical solution QUALIDAES(HiC) offers the best efficiency followed by the numerical solution QUALIDAES(StE). In contrast to the previous example of the simple pendulum, the efficiency using the StE-formulation is better since the StE-formulation (27) for the mass-spring-chain is valid for the whole time domain and has not to be switched during the numerical integration. Furthermore, the numerical solution QUALIDAES(ADA) offers a good efficiency. The efficiency of RADAU5(DI1) is similar, but the obtained precision is

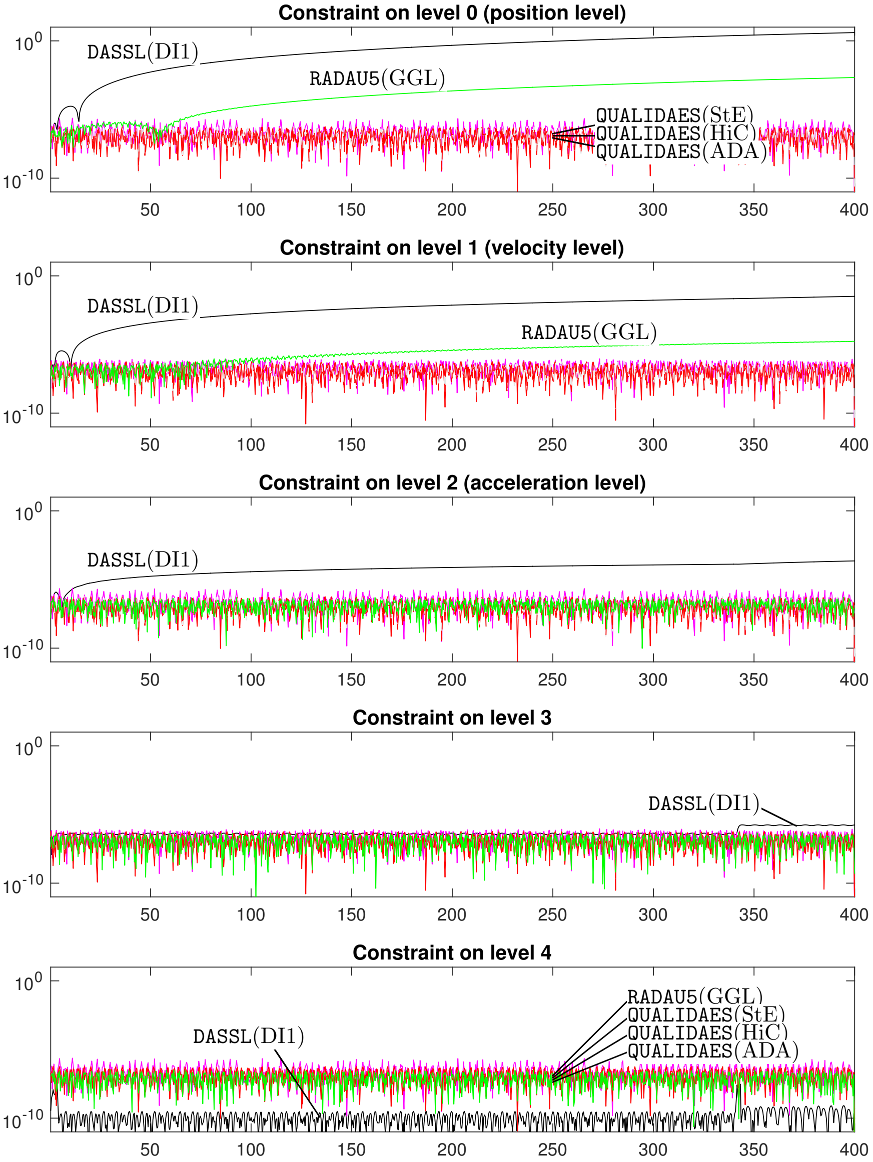

Figure 12: Residual of the constraints for the mass-spring-chain (for TOL=10^{-7})

not as good as for the previous results. The reason is that we used the d-index-1 formulation (DI1), where the hidden constraints on level 4 are included instead of (16g). Therefore, the (hidden) constraints up to level 3 are lost, and the solution drifts away from these lost constraints. Nevertheless, the maximally obtained precision is excellent for QUALIDAES with all regularizations proposed in Section 3, i.e., for the StE-formulation, the ADA-formulation, and the HiC-formulation. ◁

6 Summary

The aim of this article was to discuss several approaches for the regularization of differential-algebraic equations that benefit numerical integration. For this purpose, we have proposed three different regularization approaches which end in the structurally extended formulation (22), the algebraic derivative array formulation (31), and the hidden constraints formulation

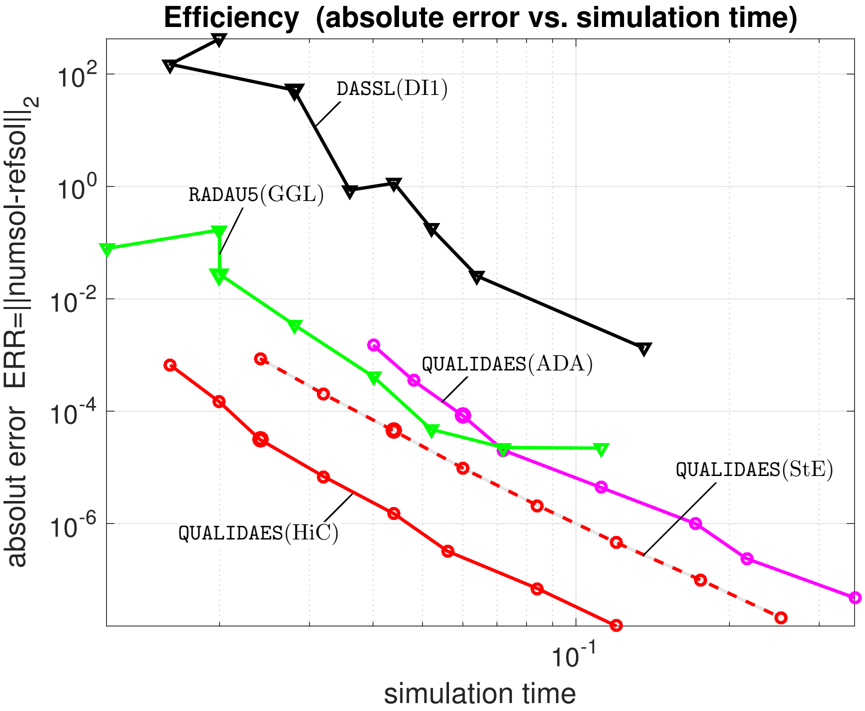

Figure 13: Efficiency (simulation time vs. absolute error for different prescribed tolerances) for the mass-spring-chain

(37). All these regularization approaches are based on the signature method (Σ-method), which was reviewed in Section 2, and the last two of them require numerical integration methods suited for overdetermined differential-algebraic equations. We also briefly introduced the software package `QUALIDAES` suited for quasi-linear differential-algebraic equations which may be overdetermined and illustrated the efficiency of the numerical integration for the numerical simulation of the simple pendulum and the numerical solving of a path following problem for a mass-spring-chain.

References

[1] T.J. Abdulla, J.R. Cash, and M.T. Diamantakis. An MEBDF package for the numerical solution of large sparse systems of stiff initial value problems. *Computers and Mathematics with Applications*, 42:121–129, 2001.

[2] R. Altmeyer and A. Steinbrecher. Regularization and numerical simulation of dynamical systems modeled with Modelica. Preprint 29-2013, Institut für Mathematik, TU Berlin, 2013.

[3] L. Auslander and R.E. MacKenzie. *Introduction to differentiable manifolds.* Dover Publications Inc., New York, 1977. Corrected reprinting.

[4] S. Bächle and F. Ebert. Element-based topological index reduction for differential-algebraic equations in circuit simulation. Technical Report 246, Matheon - DFG Research Center "Mathematics for key techonolgies", 2005.

[5] D. Bankmann and A. Steinbrecher. OVDBDF: A software package for the numerical integration of differential-algebraic equations. Technical report, Institut für Mathematik, Technische Universität Berlin, Berlin, Germany, in preparation.

[6] K.E. Brenan, S.L. Campbell, and L.R. Petzold. *Numerical Solution of Initial-Value Problems in Differential Algebraic Equations*, volume 14 of *Classics in Applied Mathematics.* SIAM, Philadelphia, PA, 1996.

[7] E. Eich-Soellner and C. Führer. *Numerical Methods in Multibody Dynamics.* B.G.Teubner, Stuttgart, 1998.

[8] C. Führer. *Differential-algebraische Gleichungssysteme in mechanischen Mehrkörpersystemen - Theorie, numerische Ansätze und Anwendungen.* PhD thesis, Technische Universität München, 1988.

[9] C.W. Gear, B. Leimkuhler, and G.K. Gupta. Automatic integration of Euler-Lagrange equations with constraints. *Journal of Computional and Applied Mathematics*, 12/13:77–90, 1985.

[10] E. Hairer, C. Lubich, and M. Roche. *The Numerical Solution of Differential-Algebraic Systems by Runge-Kutta Methods.* Springer-Verlag, Berlin, Germany, 1989.

[11] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II - Stiff and Differential-Algebraic Problems.* Springer-Verlag, Berlin, Germany, 2nd edition, 1996.

[12] P. Kunkel and V. Mehrmann. Index reduction for differential-algebraic equations by minimal extension. *Zeitschrift für Angewandte Mathematik und Mechanik*, 84(9):579–597, 2004.

[13] P. Kunkel and V. Mehrmann. *Differential-Algebraic Equations. Analysis and Numerical Solution.* EMS Publishing House, Zürich, Switzerland, 2006.

[14] R. Lamour, R. März, and C. Tischendorf. *Differential-algebraic equations: A projector based analysis.* Differential-Algebraic Equations Forum 1. Berlin: Springer, 2013.

[15] S.E. Mattsson and Söderlind. Index reduction in differential-algebraic equations using dummy variables. *SIAM Journal Scientific Computing*, 14:677–692, 1993.

[16] N. Nedialkov, J. Pryce, and G. Tan. DAESA - a Matlab tool for structural analysis of DAEs: Software. Technical report CAS-12-01-NN, Department of Computing and Software, McMaster University, Hamilton, ON, Canada, 2012.

[17] N.S. Nedialkov and J.D. Pryce. Solving differential-algebraic equations by Taylor series (i): Computing Taylor coefficients. *BIT Numerical Mathematics*, 45(3):561–591, 2005.

[18] C.C. Pantelides. The consistent initialization of differential-algebraic systems. *SIAM Journal on Scientific and Statistic Computing*, 9(2):213–231, 1988.

[19] J.D. Pryce. A simple structural analysis method for DAEs. *BIT*, 41(2):364–394, 2001.

[20] L. Scholz and A. Steinbrecher. DAEs in Applications. In P. Benner, M. Bollhöfer, D. Kressner, C. Mehl, and T. Stykel, editors, *Numerical Algebra, Matrix Theory, Differential-Algebraic Equations and Control Theory*, pages 463–501. Springer International Publishing, 2015.

[21] L. Scholz and A. Steinbrecher. Regularization of DAEs based on the signature method. *BIT Numerical Mathematics*, pages 1–22, 2015.

[22] A. Steinbrecher. *Numerical Solution of Quasi-Linear Differential-Algebraic Equations and Industrial Simulation of Multibody Systems.* PhD thesis, Technische Universität Berlin, 2006.

[23] A. Steinbrecher. Regularization of quasi-linear differential-algebraic equations. *IFAC-PapersOnLine*, 48(1):300–305, 2015. 8th Vienna International Conferenceon Mathematical Modelling MATHMOD 2015.

[24] A. Steinbrecher. QUALIDAES: A software package for the numerical integration of quasi-linear differential-algebraic equations. Technical report, Institut für Mathematik, Technische Universität Berlin, Berlin, Germany, in preparation.

[25] P.E. Van Keken, D.A. Yuen, and L.R. Petzold. DASPK: a new high order and adaptive time-integration technique with applications to mantle convection with strongly temperature- and pressure-dependent rheology. *Geophysical and Astrophysical Fluid Dynamics*, 80(1-2):57–74, 1995.