Projected Explicit and Implicit Taylor Series Methods for DAEs

Diana Estévez Schwarz and René Lamour

October 1, 2019

Abstract

The only recently developed new algorithm for computing consistent initial values and Taylor coefficients for DAEs using projector based constrained optimization opens new possibilities to apply Taylor series integration methods. In this paper, we show how corresponding projected explicit and implicit Taylor series methods can be adapted for DAEs of arbitrary index. Owing to our formulation as a projected optimization problem constrained by the derivative array, no explicit description of the inherent dynamics is necessary and various Taylor integration schemes can be defined straightforward. In particular, we address higher-order Padé methods that stand out due to their stability and order properties. We further discuss several aspects of our prototype implemented in Python using Automatic Differentiation. The methods have been successfully tested for examples arising from multibody systems simulation and a higher-index DAE benchmark arising from servo-constraint problems.

Keywords: Taylor series methods, DAE, differential-algebraic equation, consistent initial value, index, derivative array, projector based analysis, nonlinear constrained optimization, automatic differentiation

MSC-Classification: 65L05, 65L80, 34A09, 34A34, 65D25, 90C30, 90C55

1 Introduction

Higher index DAEs do not only represent integration problems, but differentiation problems, too. Therefore, it seems straightforward to solve an associated ODE with classical integration schemes and the differentiation problems using Automatic Differentiation (AD). However, depending on the structure, both differentiations and integrations may be intertwined in a complex manner such that this simple idea results to be difficult to realize in general.

In this context, different approaches have been considered for DAEs in order to combine AD with ODE integrations schemes. By construction, the approaches are based on corresponding index definitions and lead, therefore, to quite different algorithms.

- In [20], [21] and the related work the structural index was used to find a corresponding ODE.
- In [9] we used the tractabiliy matrix sequence to solve the inherent ODE for DAEs of index up to two. The realization for higher-index DAEs seemed to be rather complicated.
- In [13] we briefly described how an approach based on the differentiation index defined in [10], [12] leads to an explicit Taylor series methods for DAEs. An analysis of the corresponding projected explicit ODE can be found in [14]. In this sense, these methods can be considered as projected Taylor series methods.

In this paper, we analyze more general classes of the latter mentioned projected Taylor series methods. In particular, we discuss how projected implicit Taylor series methods can be defined for DAEs, generalizing the approach from [13]. Here we focus on the methods from [16], [3].

The main idea in this context is that the computation of Taylor coefficients of a solution of an implicit ODE can be considered the solution of a nonlinear system of equations. In this sense, we will see that a generalization for DAEs can be obtained solving a nonlinear optimization problem [12]. The obtained solution corresponds to a projected method. The advantages of this approach are obvious:

- We assume weak structural properties of the DAEs such that ODEs and semi-explicit DAEs are simple special cases. Theoretically, we can consider DAEs of any index.
- An explicit description of the inherent dynamics is not required for the algorithmic realization.
- We can use higher-order integration schemes, also for stiff ODEs/DAEs.

The described methods were implemented in a prototype and first numerical tests for DAEs up to index 4 and integration schemes up to order 8 were successful.

The paper is organized as follows. In Section 2 we introduce the notation for Taylor Series and DAEs that we use and summarize the result of [12], which is crucial for our approach.

Explicit and fully implicit Taylor series methods of ODEs are revised in Section 3 such that a generalization for DAEs of the approach from [13] results straightforward.

A general description of projected Taylor series methods is given in Section 4. Within that framework, we present how two-halfstep (TH) schemes and higher order Padé (HOP) schemes can be applied to DAEs in Section 5.

The properties of the resulting minimization problems are discussed in Section 6 and some practical consideration for the implementation are addressed in Section 7.

A prototype implementation of the proposed projected methods for DAEs is tested for several well-known examples and benchmarks from literature in Section 8. An outlook summarizing directions for further investigations concludes the paper.

For completeness, in the Appendix, on the one hand we summarize the stability functions and stability regions for the considered Taylor series methods, since they were essential for the development of HOP methods. On the other hand, we provide the used DAE formulation of the tested examples resulting from servo-constraint problems for multi-body systems.

2 Taylor Series and DAEs

Since we want to analyze one-step methods, we will consider the computation of an approximation of the solution of the ODE/DAE at a time-point t_{j+1} if an approximation of the solution at a time-point t_j is given. Consequently, in order to describe our method in terms of Taylor expansion coefficients, for $K \in \mathbb{N}$ we suppose that a suitable approximation

$$[(c_0)_j, (c_1)_j, (c_2)_j, \ldots, (c_K)_j] \approx [x(t_j), x'(t_j), \frac{1}{2}x'(t_j), \ldots, \frac{1}{K!}x^{(K)}(t_j)]$$

is given and that we look at adequate methods to compute

$$[(c_0)_{j+1}, (c_1)_{j+1}, \ldots, (c_K)_{j+1}] \approx [x(t_{j+1}), x'(t_{j+1}), \frac{1}{2}x'(t_{j+1}), \ldots, \frac{1}{K!}x^{(K)}(t_{j+1})].$$

If we suppose that the ODE/DAE is described by

$$f(x', x, t) = 0, \tag{1}$$

then, first of all, we require that

$$f((c_1)_j, (c_0)_j, t_j) = 0 \quad \text{and} \quad f((c_1)_{j+1}, (c_0)_{j+1}, t_{j+1}) = 0$$

is fulfilled. If, more generally, we consider the derivative array

$$\begin{pmatrix} f(x',x,t) \\ \frac{d}{dt} f(x',x,t) \\ \frac{d^2}{dt^2} f(x',x,t) \\ \vdots \\ \frac{d^k}{dt^k} f(x',x,t) \end{pmatrix}$$

for $k \in \mathbb{N}$, then we suppose that for the corresponding function

$$r(c_0,c_1,\ldots,c_{k+1},t) := \begin{pmatrix} r_0(c_0,c_1,t) \\ r_1(c_0,c_1,c_2,t) \\ \vdots \\ r_k(c_0,c_1,\ldots,c_{k+1},t) \end{pmatrix} := \begin{pmatrix} f(c_1,c_0,t) \\ \vdots \\ \vdots \end{pmatrix}$$

it holds

$$r((c_0)_j,(c_1)_j,(c_2)_j,\ldots,(c_{k+1})_j,t_j) = 0$$

and

$$r((c_0)_{j+1},(c_1)_{j+1},(c_2)_{j+1},\ldots,(c_{k+1})_{j+1},t_{j+1}) = 0.$$

In practice, the function r can be provided by automatic differentiation (AD).

Let us focus on the relation between k, K, the DAE-index μ and consistent initial values:

- If, instead of (1), we have a nonlinear equation $f(x,t)$ and consider a derivative array r with up to the k-th derivative, then, at t_0, we can compute $k+1$ consistent coefficients

  $$(c_0)_0,(c_1)_0,(c_2)_0,\ldots,(c_k)_0$$

  if $k \leq K$. The coefficients $c_{k+1},\ldots,c_K$ will not be consistent in general, since no equations are considered for them.

- For regular ODEs (1), if we consider a derivative array r with up to the k-th derivative and c_0 is given, then we can compute $k+1$ consistent coefficients

  $$(c_1)_0,(c_2)_0,(c_3)_0,\ldots,(c_{k+1})_0,$$

  at t_0, if $k+1 \leq K$. The coefficients $c_{k+2},\ldots,c_K$ will not be consistent in general, since no equations for them are given.

- For DAEs (1), if we consider a derivative array r with up to the k-th derivative and fix the free initial conditions of c_0, then we may compute $k+2-\mu$ consistent coefficients

  $$(c_0)_0, (c_1)_0, (c_2)_0, \ldots, (c_{k+1-\mu})_0$$

  if $k+1 \leq K$, cf. [13]. The coefficients $c_{k+2-\mu}, \ldots, c_K$ will not be consistent in general. In this case, the challenge is the appropriate fixation of the free initial conditions.

In this paper, we assume that

$$\ker f_{c_1}(c_1, c_0, t)$$

does not depend on (c_1, c_0, t) and consider the constant orthogonal projector Q onto $\ker f_{c_1}$ as well as the complementary orthogonal projector $P := I - Q$. Therefore, the Taylor coefficients of $Px(t)$ at t_j correspond to

$$[P(c_0)_j, P(c_1)_j, P(c_2)_j, \ldots, P(c_K)_j].$$

Recall further that according to [12] the optimization problem

$$\begin{array}{rl} \min & \left\| P\left((c_0)_{(0)} - x_0\right) \right\|_2 \\ \text{subject to} & r((c_0)_0, (c_1)_0, (c_2)_0, \ldots, (c_{k+1})_0, t_0) = 0, \end{array}$$

provides consistent initial values. In terms of [11], this minimization problem is equivalent to the system of equations

$$\Pi\left((c_0)_{(0)} - x_0\right) = 0, \tag{2}$$

$$r((c_0)_0, (c_1)_0, (c_2)_0, \ldots, (c_{k+1})_0, t_0) = 0, \tag{3}$$

where Π describes an appropriate orthogonal projector and the rank of Π coincides with the degree of freedom of the DAE.

Example 1. *Consider the index-4 DAE*

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} x' + \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} x = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \mathrm{e}^t \end{pmatrix}, \quad x = \begin{pmatrix} C\mathrm{e}^{-t} - \frac{\mathrm{e}^t}{2} \\ -\mathrm{e}^t \\ \mathrm{e}^t \\ -\mathrm{e}^t \\ \mathrm{e}^t \end{pmatrix} \tag{4}$$

	x_1	x_2	x_3	x_4	x_5
$x_*(0)$	1.00000000e+00	-1.00000000e+00	1.00000000e+00	-1.00000000e+00	1.00000000e+00
$x'_*(0)$	-4.09190611e-15	-1.00000000e+00	1.00000000e+00	-1.00000000e+00	1.00000000e+00
$\frac{1}{2}x''_*(0)$	5.00000000e-01	$\boxed{\text{-4.76883765e-02}}$	5.00000000e-01	-5.00000000e-01	5.00000000e-01
$\frac{1}{3!}x'''_*(0)$	$\boxed{\text{-1.50770541e-01}}$	$\boxed{\text{8.66099724e-03}}$	$\boxed{\text{1.58961255e-02}}$	-1.66666667e-01	1.66666667e-01
$\frac{1}{4!}x^{(iv)}_*(0)$	$\boxed{\text{3.55273860e-02}}$	$\boxed{\text{-1.31582911e-03}}$	$\boxed{\text{-2.16524931e-03}}$	$\boxed{\text{-3.97403138e-03}}$	4.16666667e-02
$\frac{1}{5!}x^{(v)}_*(0)$	$\boxed{\text{-6.84231137e-03}}$	$\boxed{\text{0.00000000e+00}}$	$\boxed{\text{2.63165822e-04}}$	$\boxed{\text{4.33049862e-04}}$	$\boxed{\text{7.94806275e-04}}$

Table 1: Numerical solution of the initialization problem for system (4) from Example 1 for $t_0 = 0$ and $\alpha = [1,0,0,0,0]$ using Taylor coefficients with $k = 4$, $K = 5$. The framed values are not consistent.

with the general solution

$$x = \begin{pmatrix} C\,e^{-t} - \frac{e^t}{2} \\ -e^t \\ e^t \\ -e^t \\ e^t \end{pmatrix}.$$

The corresponding projectors read

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad \Pi = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

According to the notation introduced in [14], the associated essential projected ODE reads

$$x_1' + x_1 = e^t. \tag{5}$$

For the initial value $x_1(0) = 1$, the solution is $x_1(t) = \cosh(t)$. In Table 1 we present the results of the computation of consistent initial values with InitDAE [8].

In [14], it is shown how this orthogonal projector Π can be used to decouple linear DAEs such that a projected explicit ODE for the component Πx is obtained. Therefore, the numerical solution delivered by the methods defined in the following corresponds to

- the numerical solution obtained by Taylor series methods applied to the projected explicit ODE for Πx, and
- corresponding values for the components $(I - \Pi)x$ that result from the values for Πx and the derivative array .

Due to the formulation as an optimization problem, the inherent dynamics of the DAE that can be expressed in terms of Πx is not considered explicitly, but implicitly. Therefore, the stability and order properties of the integration methods defined below can be transferred straightforward from ODEs to DAEs.

3 Projected Explicit and Fully Implicit Taylor Series Methods for DAEs

Since our focus is on the formulation of methods for DAEs, we start rewriting the methods for ODEs in such a way, that the generalization for DAEs results straightforward.

3.1 Explicit Taylor Series Method for ODEs

In terms of the above notation, the explicit Taylor series method corresponds to the steps:

- Initialization: Solve the system of equations

$$\begin{aligned} (c_0)_{(0)} - x_0 &= 0, && (6)\\ r((c_0)_0, (c_1)_0, (c_2)_0, \ldots, (c_{k+1})_0, t_0) &= 0. && (7) \end{aligned}$$

- For time-points t_{j+1}, $j \geq 0$, $h_j = t_{j+1} - t_j$: Solve the systems of equations

$$\begin{aligned} (c_0)_{(j+1)} - \underbrace{\sum_{\ell=0}^{k_e} (c_\ell)_j h_j^\ell}_{\approx x(t_j+h_j)} &= 0, && (8)\\ r((c_0)_{j+1}, (c_1)_{j+1}, (c_2)_{j+1}, \ldots, (c_{k+1})_{j+1}, t_{j+1}) &= 0, && (9) \end{aligned}$$

  successively for $k_e \leq k+1$, where k_e determines the order.

This method is called explicit, since equation (8) is an explicit equation for $(c_0)_{j+1}$ that does not involve any value $(c_\ell)_{j+1}$ for $\ell \geq 1$.

3.2 Explicit Taylor Series Method for DAEs

According to [13], a generalization for DAEs reads:

- Initialization: Solve the optimization problem

$$\begin{aligned} \min \quad & \left\| P\left((c_0)_{(0)} - x_0\right)\right\|_2 && (10)\\ \text{subject to} \quad & r((c_0)_0, (c_1)_0, (c_2)_0, \ldots, (c_{k+1})_0, t_0) = 0. && (11) \end{aligned}$$

- For time-points t_{j+1}, $j \geq 0$, $h_j = t_{j+1} - t_j$: Solve the optimization problems

$$\begin{array}{rl} \min & \left\| P\left((c_0)_{(j+1)} - \sum_{\ell=0}^{k_e} (c_\ell)_j h_j^\ell \right) \right\|_2 , \\ \text{subject to} & r((c_0)_{j+1}, (c_1)_{j+1}, (c_2)_{j+1}, \ldots, (c_{k+1})_{j+1}, t_{j+1}) = 0, \end{array}$$

  successively for $k_e \leq k+1-\mu$, where k_e determines the order.

In contrast to explicit ODEs, with this approach we always have to solve nonlinear systems of equations. Therefore, it seems reasonable to consider also implicit Taylor approximations in the integration scheme.

3.3 Fully Implicit Taylor Series Methods for ODEs

According to [16], among others, the implicit counterpart of the explicit Taylor series method for ODEs consists, at last, in the following steps.

- Initialization: See Section 3.1, equations (6)-(7).
- For time-points t_{j+1}, $j \geq 0$, $h_j = t_{j+1} - t_j$: Solve the systems of equations

$$\begin{aligned} \underbrace{\sum_{\ell=0}^{k_i} (c_\ell)_{j+1} (-h_j)^\ell}_{\approx x(t_{j+1}-h_j)} - (c_0)_{(j)} &= 0, & (12) \\ r((c_0)_{j+1}, (c_1)_{j+1}, (c_2)_{j+1}, \ldots, (c_k)_{j+1}, t_{j+1}) &= 0, & (13) \end{aligned}$$

  successively for $k_i \leq k+1$, where k_i determines the order.

3.4 Fully Implicit Taylor Series Methods for DAEs

For DAEs, the generalization results now straightforward:

- Initialization: See Section 3.2, equations (10)-(11).
- For time-points t_{j+1}, $j \geq 0$, $h_j = t_{j+1} - t_j$: Solve the optimization problems

$$\min \quad \left\| P\left(\sum_{\ell=0}^{k_i} (c_\ell)_{j+1} (-h_j)^\ell - (c_0)_{(j)} \right) \right\|_2 \tag{14}$$

$$\text{subject to} \quad r((c_0)_{j+1}, (c_1)_{j+1}, (c_2)_{j+1}, \ldots, (c_k)_{j+1}, t_{j+1}) = 0, \tag{15}$$

  successively for $k_i \leq k+1-\mu$, where k_i determines the order.

Obviously, if, instead of (12) and (14), more general conditions are defined, then the computational costs will be comparable as long as the dimension of the system remains equal. Therefore, it seems natural to search for more general schemes with better convergence and stability properties.

4 General Definition of Explicit/Implicit Methods

Note that $P = I$ holds for ODEs and that in all formulations from above the constraints of the optimization problems are identical. Therefore, in order to characterize different methods, for the time-points t_j, t_{j+1}, we merely consider the corresponding objective function in terms of

$$p\left((c_0)_{j+1},\ldots,(c_{k_i})_{j+1}\right) := P\left(\sum_{\ell_i=0}^{k_i} \omega^i_{\ell_i}(c_{\ell_i})_{(j+1)}\left(-h_j\right)^{\ell_i} - \sum_{\ell_e=0}^{k_e} \omega^e_{\ell_e}(c_{\ell_e})_{(j)}\left(h_j\right)^{\ell_e}\right)$$

for suitable weights $\omega^e_{\ell_e}, \omega^i_{\ell_i}$ and $k_e, k_i \geq 0$. Consequently, it suffices to assert

$$\begin{aligned} \min \quad & \left\| p\left((c_0)_{j+1},(c_1)_{j+1},(c_2)_{j+1},\ldots,(c_{k_i})_{j+1}\right)\right\|_2 && (16) \\ \text{subject to} \quad & r((c_0)_{j+1},(c_1)_{j+1},(c_2)_{j+1},\ldots,(c_k)_{j+1},t_{j+1}) = 0. && (17) \end{aligned}$$

for the corresponding function p.

With this notation, the explicit and the fully implicit Taylor series methods from Section 3 can be characterized as follows:

- Explicit: $k_e \geq 1$, $\omega^e_{\ell_e} = 1$ for $0 \leq \ell_e \leq k_e$, $k_i = 0$, $\omega^i_0 = 1$.
- Fully Implicit: $k_i \geq 1$, $\omega^i_{\ell_e} = 1$ for $0 \leq \ell_i \leq k_i$, $k_e = 0$, $\omega^e_0 = 1$.

In general terms, any method with $k_i \geq 1$ can be considered as implicit method.

5 Two-halfstep and HOP schemes

In this section, we describe two types of known Taylor schemes for ODEs. The first ones are very simple and illustrative, the second ones slightly more sophisticated and optimal.

5.1 Two-halfstep Explicit/Implicit (TH) Schemes

One straightforward combination of the explicit and implicit integration schemes is to approximate $x(t_j + \sigma h_j) = x(t_{j+1} - (1-\sigma)h_j)$ for $0 \leq \sigma \leq 1$ as follows

$$x(t_j + \sigma h_j) \approx \sum_{\ell_e=0}^{k_e} (c_{\ell_e})_{(j)}\left(\sigma h_j\right)^{\ell_e}, \tag{18}$$

$$x(t_{j+1} - (1-\sigma)h_j) \approx \sum_{\ell_i=0}^{k_i} (c_{\ell_i})_{(j+1)}\left(-(1-\sigma)h_j\right)^{\ell_i}, \tag{19}$$

and equalize the expressions from both right-hand sides. The properties of the methods (18)-(19) are described in [16]. The choice $\sigma = \frac{1}{2}$, which can be interpreted as a generalization of the trapezoidal rule, turns out to be convenient. For $k_e = k_i$, $\sigma = \frac{1}{2}$, it coincides with the one tested in [1], [5].

Remark 1. *Note that another closely related implicit/explicit scheme is described in the literature, too. This means that the first step is implicit and the second one explicit, in contrast to the approach from above. According to the extensive analysis from [22],* $\sigma = \frac{1}{2}$ *is convenient also in that case. However, these methods are less suitable for our DAE-scheme since the Taylor coefficients would be considered at* $t_j + \sigma h_j$*, whereas the constraints have to be fulfilled at* t_{j+1}*.*

In the notation from Section 4, choosing $\sigma = \frac{1}{2}$ in (18)-(19) means to consider

$$p := P\left(\sum_{\ell_i=0}^{k_i} (c_{\ell_i})_{(j+1)} \left(-\frac{h_j}{2}\right)^{\ell_i} - \sum_{\ell_e=0}^{k_e} (c_{\ell_e})_{(j)} \left(\frac{h_j}{2}\right)^{\ell_e}\right) \tag{20}$$

for $0 \leq k_e, k_i \leq k+1-\mu$, i.e.,

$$\omega^e_{\ell_e} = \left(\frac{1}{2}\right)^{\ell_e}, \; \ell_e = 0, \ldots, k_e, \qquad \omega^i_{\ell_i} = \left(\frac{1}{2}\right)^{\ell_i}, \ell_i = 0, \ldots, k_i.$$

For shortness, we denote these two-halfstep methods by (k_e, k_i)-TH.

Recall that, in general, the stability function $R(z)$ (cf. Appendix A) of a (k_e, k_i)-TH method is not a Padé approximation of the exponential function such that the maximally achievable order is not given in general. Therefore, further higher-order schemes for stiff ODEs have been developed, namely the HOP-methods described in [3].

5.2 Higher Order Padé (HOP) Methods

According to [3], HOP may be interpreted as Hermite-Obrechkoff-Padé or simple Higher-Order Padé. These schemes may be viewed as implicit Taylor series methods based on Hermite quadratures.

In our notation, a (k_e, k_i)-HOP scheme means choosing

$$\begin{aligned}
\omega^e_{\ell_e} &:= \frac{k_e!(k_e+k_i-\ell_e)!}{(k_e+k_i)!(k_e-\ell_e)!}, \quad \ell_e = 0, \ldots, k_e, \\
\omega^i_{\ell_i} &:= \frac{k_i!(k_e+k_i-\ell_i)!}{(k_e+k_i)!(k_i-\ell_i)!}, \quad \ell_i = 0, \ldots, k_i.
\end{aligned}$$

These coefficients correspond to the (k_e, k_i)-Padé approximation of the exponential function such that $R(z)$ is optimal, see Appendix A.

(k_e, k_i)-HOP methods have the following properties, cf. [3]:

- the order of consistency is $k_e + k_i$,
- the order of the local error is $k_e + k_i + 1$,
- they are A-stable for $k_i - 2 \leq k_e \leq k_i$,
- they are L-stable for $k_i - 2 \leq k_e \leq k_i - 1$.

Note that also in this case the trapezoidal rule corresponds to $k_e = k_i = 1$ and the implicit Euler method to $k_e = 0$, $k_i = 1$. In this sense, the methods with $k_e = k_i$ could be viewed as a generalization of the trapezoidal rule and those with $k_e = k_i - 1$ as a generalization of the implicit Euler method, cf. [3].

In Section 8 we numerically verify the outstanding properties of these methods.

6 Properties of the Minimization Problems

In [12] we analyzed the properties of the minimization problem obtained when computing consistent initial values. That analysis can directly be applied to the explicit Taylor series method, cf. [13]. To appreciate the properties for implicit methods (i.e. $k_i > 0$), we define, for $k \geq \max\{k_e, k_i\}$, the matrices

$$\begin{aligned} T^e &:= \begin{pmatrix} P\omega_0^e & P\omega_1^e h_j & P\omega_2^i h_j^2 & \dots & P\omega_k^i h_j^k \end{pmatrix} \\ T^i &:= \begin{pmatrix} P\omega_0^i & P\omega_1^i(-h_j) & P\omega_2^i h_j^2 & \dots & P\omega_k^i(-h_j)^k \end{pmatrix}, \end{aligned}$$

assuming $\omega_{l_i} = 0$ for $l_i > k_i$ and $\omega_{l_e} = 0$ for $l_e > k_e$ and the vectors

$$X_j = \big((c_0)_j, \dots, (c_k)_j\big), \quad X_{j+1} = \big((c_0)_{j+1}, \dots, (c_k)_{j+1}\big).$$

With this notation, we obtain

$$\begin{aligned} p &:= P\left(\sum_{\ell_i=0}^{k_i} \omega_{\ell_i}^i (c_{\ell_i})_{(j+1)} \big(-h_j\big)^{\ell_i} - \sum_{\ell_e=0}^{k_e} \omega_{\ell_e}^e (c_{\ell_e})_{(j)} \big(h_j\big)^{\ell_e}\right) \\ &= T^i X_{j+1} - T^e X_j. \end{aligned}$$

Therefore, analogous to [12], for $\alpha := T^e X_j$, $X := X_{j+1}$, we consider the objective function

$$\begin{aligned} f(X) &:= \frac{1}{2}\left\|T^i X-\alpha\right\|^2 \\ &= \frac{1}{2}\left\|P(T^i X-\alpha)\right\|^2 \\ &= \frac{1}{2}(T^i X-\alpha)^T P(T^i X-\alpha) \\ &= \frac{1}{2}\left(X^T (T^i)^T P T^i X - 2\alpha^T P X + \alpha^T P\alpha\right). \end{aligned}$$

Observe that the matrix

$$\begin{aligned} \tilde{P}^i &:= (T^i)^T P T^i = (T^i)^T T^i \\ &= \begin{pmatrix} P(\omega_0^i)^2 & P\omega_0^i\omega_1^i(-h_j)^2 & \dots & P\omega_0^i\omega_k^i(-h_j)^k \\ P\omega_0^i\omega_1^i(-h_j)^2 & & & \\ \vdots & & & \\ P\omega_0^i\omega_k^i(-h_j)^k & P\omega_1^i\omega_k^i(-h_j)^{k+1} & \dots & P(\omega_k^i)^2(-h_j)^{2k} \end{pmatrix} \\ &= \begin{pmatrix} P(\omega_0^i)^2 & \dots & P\omega_0^i\omega_{k_i}^i(-h_j)^{k_i} & 0 \\ \vdots & & & \vdots \\ P\omega_0^i\omega_{k_i}^i(-h_j)^{k_i} & \dots & P(\omega_{k_i}^i)^2(-h_j)^{2k_i} & \\ 0 & \dots & & 0 \end{pmatrix} \in \mathbb{R}^{n\cdot(k+1)\times n\cdot(k+1)} \end{aligned}$$

is, by construction, positive semi-definite. However, it is not an orthogonal projector in general. Therefore, Theorem 1 and Corollary 1 of [12] cannot be applied directly. Hence, the solvability of the optimization problem results to be more difficult to be guaranteed than for explicit Taylor methods. More precisely, we want to emphasize that, for

$$\tilde{P} := \begin{pmatrix} P & 0 \\ 0 & 0 \end{pmatrix} \in \mathbb{R}^{n\cdot(k+1)\times n\cdot(k+1)},$$

the nullspaces

$$\ker \begin{pmatrix} \tilde{P}^i & G^T \\ G & 0 \end{pmatrix} \quad \text{and} \quad \ker \begin{pmatrix} \tilde{P} & G^T \\ G & 0 \end{pmatrix}$$

may be different. However, since $\tilde{P}^i$ depends on h_j, it is reasonable to assume that a suitable stepsize h_j can be found such that the optimization problem becomes solvable in the sense discussed in [12]. Nevertheless, the DAE index and the condition number to monitor singularities should not be computed using $\tilde{P}^i$.

7 Some Practical Considerations

7.1 Dimension of the Nonlinear Systems Solved in Each Step

For a given $K \in \mathbb{N}$, the Lagrange approach for solving (16) - (17) leads to a nonlinear system of equations of the dimension $2n \cdot (K+1)$, cf. [12]. Thereby, consistent coefficients

$$(c_0)_{j+1}, (c_1)_{j+1}, (c_2)_{j+1}, \dots, (c_{K-\mu})_{j+1}$$

are obtained. In contrast, the coefficients $c_{K-\mu+1}, \dots, c_K$ will not be consistent in general and the Lagrange-parameters are not even of interest.

However, increasing K by one means solving a nonlinear system containing $2n$ additional variables and equations.

7.2 Setting k_e and k_i in a Simple Implementation

Dealing with automatic differentiation (AD), the number $K \in \mathbb{N}$ has to be prescribed a priori in order to consider $(K+1)$ Taylor coefficients. Since $0 \leq k_e, k_i \leq k+1-\mu$ and $k+1 \leq K$ must be given in general, for the (k_e, k_i) TH and HOP methods, we set

$$k_i := K - \mu \quad \text{and} \quad k_e := k_i,$$

by default. We further tested $k_i := K - \mu$, $k_e := k_i - 1$ for HOP methods. Therefore we determine the index μ using InitDAE [13]. So far, we considered schemes with constant order and step-size only.

7.3 Jacobian Matrices

To solve the optimization problem (16)-(17) numerically, we provide the corresponding Jacobians.

- The Jacobian of the constraints (17) is described in [13], since it is also used for the computation of consistent initial values.
- To describe the Jacobian of the objective function (16), which is a gradient, we define

  $$q\left((c_0)_{j+1}, \dots, (c_{k_i})_{j+1}\right) := \left\| p\left((c_0)_{j+1}, \dots, (c_{k_i})_{j+1}\right) \right\|_2$$

  and realize that

  $$\frac{\partial q}{\partial (c_{\ell_i})_{j+1}} = \frac{1}{q\left((c_0)_{j+1}, \dots, (c_{k_i})_{j+1}\right)} \left(p\left((c_0)_{j+1}, \dots, (c_{k_i})_{j+1}\right)\right)^T \cdot \omega^i_{\ell_i} \cdot (-h_j)^{\ell_i},$$

  for $q \neq 0$.

8 Numerical Tests

8.1 Order Validation

To visualize the order of the methods, we integrate Example 1 in the interval $[0,1]$ with different step-sizes. The results can be found in Figure 1. On the left-hand side, we show the results for the index-4 DAE. On the right-hand side, we report the results obtained for the corresponding ODE described in equation (5).

Summarizing, we observe that:

- For $k_i, k_e \leq 1$, the methods coincide with the explicit and implicit Euler methods or the trapezoidal rule. Therefore, the graphs coincide up to effects resulting from rounding errors.
- The similarity of the overall behavior for the DAE and the ODE is awesome.
- As expected, the HOP methods are considerably more accurate due to the higher order.
- For small h and large k_e, k_i, scaling and rounding errors impede more accurate results in dependence of the tolerance `ftol` from the module *minimize* from SciPy.

8.2 Examples from the Literature

We further report numerical results obtained by the methods (3,3)-HOP and (4,4)-HOP for the following examples from the literature:

- Mass-on-car from [23], see Appendix, Section B.1,
- Extended mass-on-car from [19], see Appendix, Section B.2,
- Pendulum index 3, which can be found in almost all introductions to DAEs, for $m = 1.0$, $l = 1.0$, and $g = 1.0$,
- Car axis index 3 formulation with all parameters as given in [17]. In order to avoid a disadvantageous scaling of the Taylor coefficients, we changed the independent variable t to $\tau = 10\,t$ as described in [13].

For all examples we used `ftol` for the tolerance of the module *minimize* from SciPy. To estimate the error, we considered the difference between the results obtained by (3,3)-HOP and (4,4)-HOP.

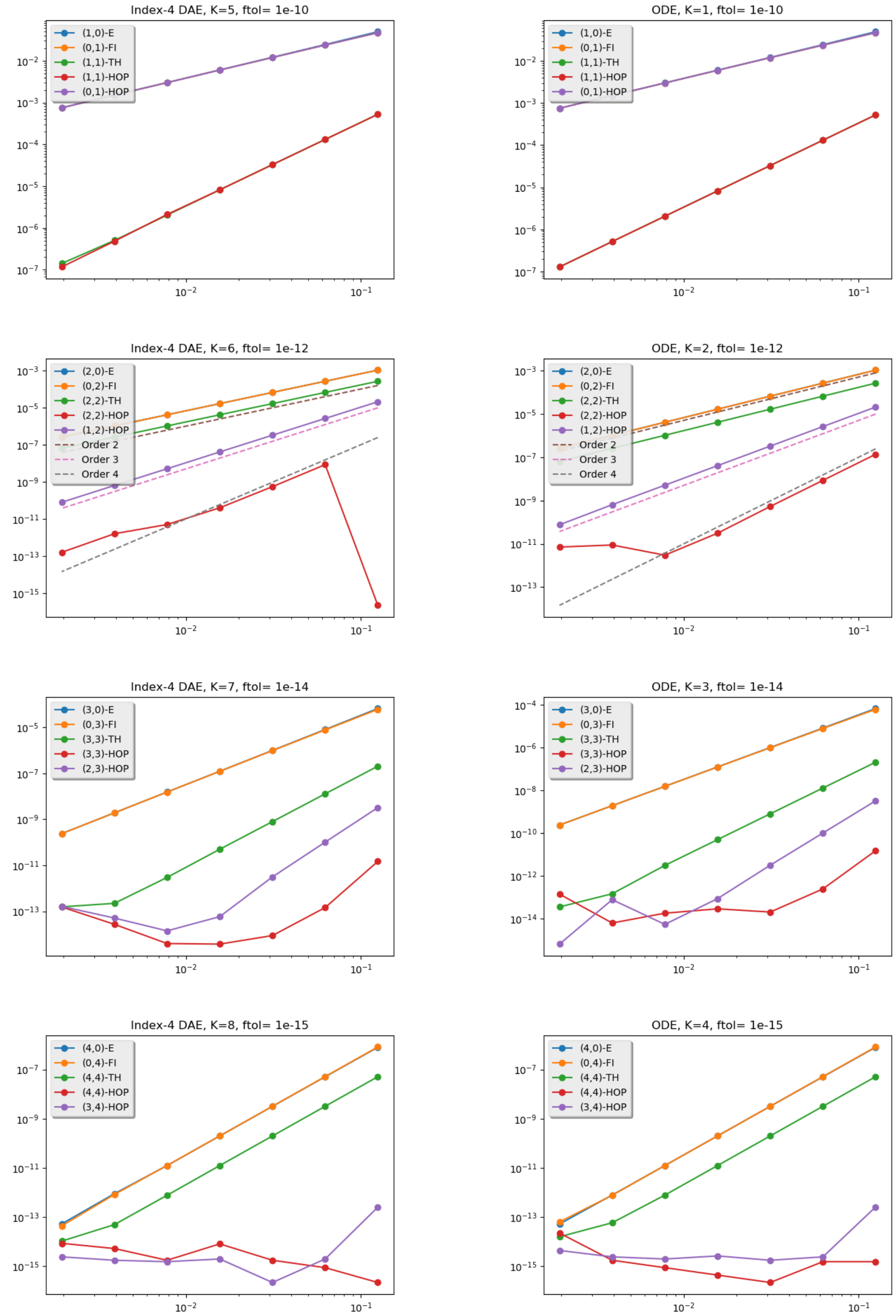

Figure 1: Stepsize-error diagram for the error $|x_1(1) - \cosh(1)|$ for the DAE (left) and the essential ODE (right) corresponding to Example 1 for different methods and *ftol* for the module *minimize* from SciPy. For k_e, k_i=2 we included graphs of Ch^p for $p = 2, 3, 4$ to appreciate the order of the methods. The first value obtained for the DAE-formulation with the $(2,2)$-HOP-method for $K = 6$ and large h seems to be very accurate by chance.

No.	Example	Dimension	Index	rank Π	Linear or not
1.	Mass-on-car	5	3	2	l
2.	Extended mass-on-car	7	4	3	l
3.	Pendulum	5	3	2	nl
4.	Car axis	10	3	4	nl

Table 2: Overview of the examples

No.	Interval	K	$k_i = k_e$	h	Time (s) (3,3)-HOP	Time (s) (4,4)-HOP
1.	0-10	6 / 7	3 / 4	0.025	55.3	80.7
2.	0-20	7 / 8	3 / 4	0.1	154.4	131.7
3.	0-20	6 / 7	3 / 4	0.1	43.7	50.9
4.	0-30	6 / 7	3 / 4	0.025	411.2	201.8

Table 3: Overview of the computations carried out for Figure 2. The CPU Time based on a computation with a 2.3 GHz processor.

All tests confirmed the applicability of the method and the results satisfy the expected accuracy. The solution graphs look identical with those given in the literature. The graphs of the estimated errors confirm the order expectations.

Since it is obvious that our implementation is not competitive with respect to runtime, we have not made a systematic comparison with other solvers here.

9 Summary and Future Work

In this article, we presented a projection approach that permits the extension of explicit/implicit Taylor integrations from ODEs to DAEs. As a result, we obtained higher-order methods that can directly be applied also to higher-index DAEs. The methods are easy to implement and convenient since, thanks to the formulation as an optimization problem, the inherent dynamics of the DAE are considered indirectly. We analysed in detail explicit, fully implicit, two-halfstep (TH) and higher-order-Padé (HOP) methods. Particularly HOP methods present excellent stability and order properties.

The results obtained by a prototype in Python that is based on InitDAE [8] outperform our expectations, in particular for higher-index DAEs. Until now, our focus was on the extension from ODEs to DAEs in order to use higher-order and A-stable methods in InitDAE for our diagnosis purposes during the integration [13]. With this promising first results, we think that more investigations on these projected methods are worthwhile.

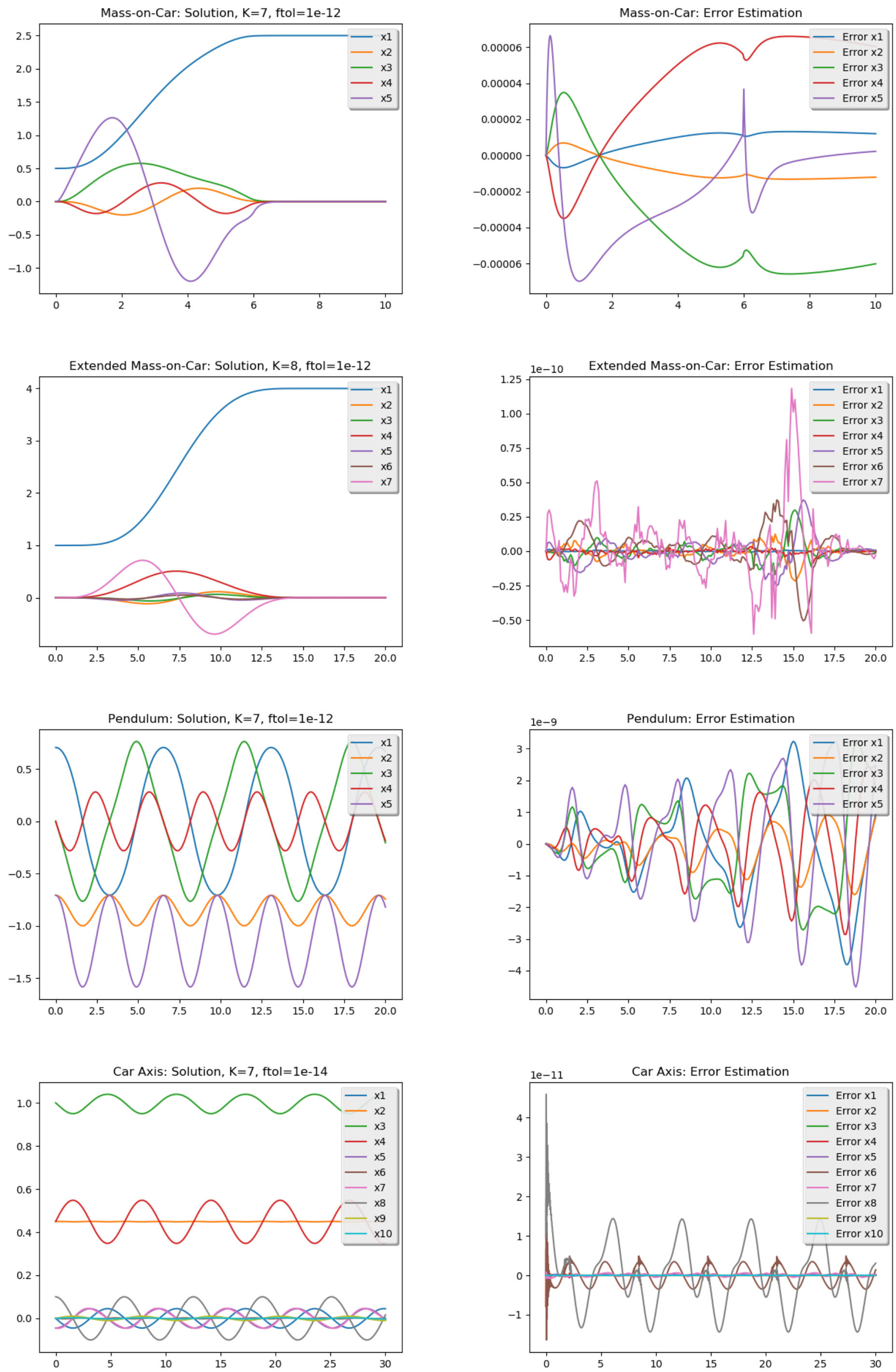

Figure 2: Numerical solutions of the examples from Section 8.2 obtained by (4,4)-HOP (left) and estimation of the error (right) considering the difference between the solution from (3,3)-HOP and (4,4)-HOP.

In fact, at present, our implementation is not competitive by far. One reason is that setting up the nonlinear equations (16)-(17) and the corresponding Jacobians with AlgoPy, cf. [24], is still very costly. If equations (16)-(17) and the corresponding Jacobians are supplied in a more efficient way, competitive solvers might be achieved. In particular, this seems likely if we take advantage of structural properties, e.g., solving subsystems step-by-step, cf. [6], [7]. Another reason is that the package minimize from SciPy performs more iterations than we expected (often more than 30), although a good initial guess is given in general. This behaviour has to be inspected in more detail. For linear systems, a direct implementation considering the projector Π from equation (2) (or, more precisely, a corresponding basis) should deliver an efficient algorithm. This could be of interest, e.g. for the applications from [19], [23]. Last but not least, competitive solvers require adaptive order and stepsize strategies - a broad field for future work.

Although these algorithms open new possibilities to integrate higher-index DAEs, we want to emphasize that, in practice, a high index is often due to modelling assumptions that should be considered very carefully. The dependencies on higher derivatives should always be well-founded.

A Stability Functions and Stability Regions of Taylor Series Methods

A.1 Stability Functions

The general definition (16)-(17) allows for a straightforward description of the stability function. Applied to ODEs (and therefore $P = I$), the stability function $R : \mathbb{C} \to \mathbb{C}$ results if we consider the test-ODE

$$y' = \lambda y, \quad y(0) = y_0, \quad \lambda \in \mathbb{C}, \tag{21}$$

and describe the numerical method for constant $h = h_j$ in terms of

$$y_{j+1} = R(h\lambda) y_j.$$

For ODEs, the methods described in Section 4 imply

$$\sum_{\ell_i=0}^{k_i} \omega^i_{\ell_i} (c_{\ell_i})_{(j+1)} \left(-h_j\right)^{\ell_i} = \sum_{\ell_e=0}^{k_e} \omega^e_{\ell_e} (c_{\ell_e})_{(j)} \left(h_j\right)^{\ell_e}$$

and, for the test-equation (21), we obtain from

$$(c_{\ell_i})_{j+1} = \lambda^{\ell_i} (c_{i_0})_{j+1} = \frac{\lambda^{\ell_i}}{\ell_i!} y_{j+1} \quad \text{and} \quad (c_{\ell_e})_j = \lambda^{\ell_e} (c_{0_e})_j = \frac{\lambda^{\ell_e}}{\ell_e!} y_j$$

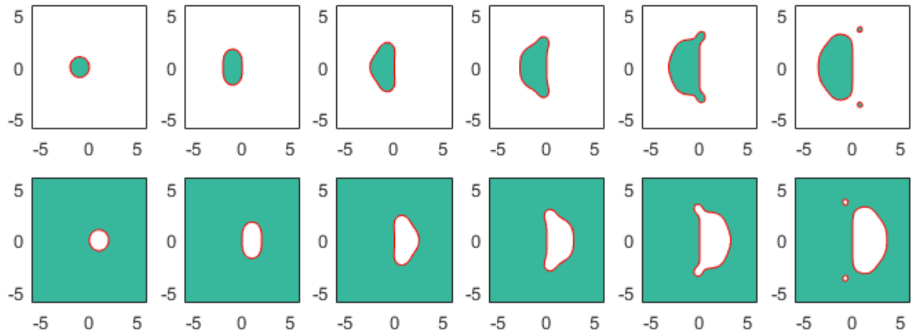

Figure 3: Colored representation of the stability regions S for explicit (top) and fully implicit (bottom) Taylor series methods up to order 6.

the relationship

$$\left(\sum_{\ell_i=0}^{k_i} \omega^i_{\ell_i} \frac{\lambda^{\ell_i}}{\ell_i!} (-h_j)^{\ell_i}\right) y_{j+1} = \left(\sum_{\ell_e=0}^{k_e} \omega^e_{\ell_e} \frac{\lambda^{\ell_e}}{\ell_e!} (h_j)^{\ell_e}\right) y_j,$$

i.e., for $z \in \mathbb{C}$

$$R(z) = \frac{\sum_{\ell_e=0}^{k_e} \frac{1}{\ell_e!} \omega^e_{\ell_e} z^\ell}{\sum_{\ell_i=0}^{k_i} (-1)^{\ell_i} \frac{1}{\ell_i!} \omega^i_{\ell_i} z^\ell}.$$

A.2 Stability Regions

The corresponding stability regions can thus be characterized by

$$S := \{\, z \in \mathbb{C} \;:\; |R(z)| \leq 1 \} = \left\{ z \in \mathbb{C} \;:\; \left| \sum_{\ell_e=0}^{k_e} \frac{1}{\ell_e!} \omega^e_{\ell_e} z^\ell \right| \leq \left| \sum_{\ell_i=0}^{k_i} (-1)^{\ell_i} \frac{1}{\ell_i!} \omega^i_{\ell_i} z^\ell \right| \right\}.$$

For the methods discussed in this article we obtain

- Explicit Taylor:

$$R^E_{k_e,0}(z) = \sum_{\ell_e=0}^{k_e} \frac{z^{\ell_e}}{\ell_e!}.$$

The corresponding stability regions are illustrated in Figure 3 (top), cf. also [2], [15].

- Fully Implicit Taylor:

$$R^{FI}_{0,k_i}(z) = \frac{1}{\sum_{\ell_i=0}^{k_i}(-1)^{\ell_i}\frac{z_i^{\ell}}{\ell_i!}}.$$

  The corresponding stability regions are illustrated in Figure 3 (bottom).

- For the two-halfstep explicit/implicit schemes (20) we obtain:

$$R^{TH}_{k_e,k_i}(z) = \frac{\sum\limits_{\ell_e=0}^{k_e}\frac{1}{\ell_e!}\left(\frac{z}{2}\right)^{\ell_e}}{\sum\limits_{\ell_i=0}^{k_i}\frac{1}{\ell_i!}\left(-\frac{z}{2}\right)^{\ell_i}}.$$

  The corresponding stability regions for $k_e, k_i = 0, \ldots, 6$ are represented in Figure 4. Note that symmetry is due to

$$R^{TH}_{k_e,k_i}(-z) = \frac{1}{R^{TH}_{k_i,k_e}(z)}. \tag{22}$$

  The schemes with $k_e \leq k_i$ seem to be A-stable or $A(\alpha)$-stable for moderate α. Indeed, according to [1], [5], the schemes provide good results for Hamiltonian systems if $k_e = k_i$. Furthermore, A-stable schemes with $k_e < k_i$ are L-stable, since

$$\lim_{z\to-\infty}\left|R^{TH}_{k_e,k_i}(z)\right| = 0 \quad \text{for} \quad k_e < k_i.$$

- For HOP-methods, from

$$\begin{aligned} \omega^e_{\ell_e} &:= \frac{k_e!(k_e+k_i-\ell_e)!}{(k_e+k_i)!(k_e-\ell_e)!}, \quad \ell_e = 0, \ldots, k_e \\ \omega^i_{\ell_i} &:= \frac{k_i!(k_e+k_i-\ell_i)!}{(k_e+k_i)!(k_i-\ell_i)!}, \quad \ell_i = 0, \ldots, k_i \end{aligned}$$

  we obtain

$$R^{HOP}_{k_e,k_i}(z) = \frac{k_e!}{k_i!}\frac{\sum\limits_{\ell_e=0}^{k_e}\frac{1}{\ell_e!}\frac{(k_e+k_i-\ell_e)!}{(k_e-\ell_e)!}z^{\ell}}{\sum\limits_{\ell_i=0}^{k_i}(-1)^{\ell_i}\frac{1}{\ell_i!}\frac{(k_e+k_i-\ell_i)!}{(k_i-\ell_i)!}z^{\ell}}. \tag{23}$$

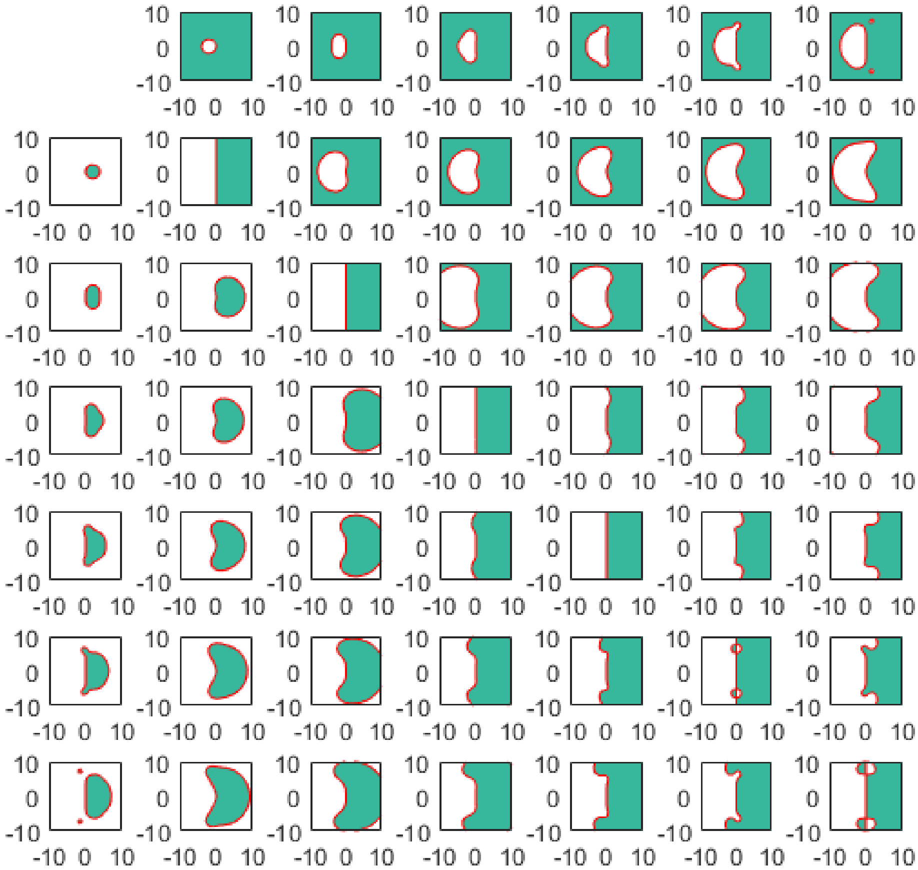

Figure 4: Colored representation of the stability regions S of two-halfstep schemes considering R_{k_e,k_i} for all combinations of $k_e, k_i = 0, \ldots, 6$, where k_e corresponds to the rows and k_i to the columns. The symmetry results form (22). We can realize that, for $k_i = k_e = 5, 6$, they are not A-stable. This contradicts the statement in [16].

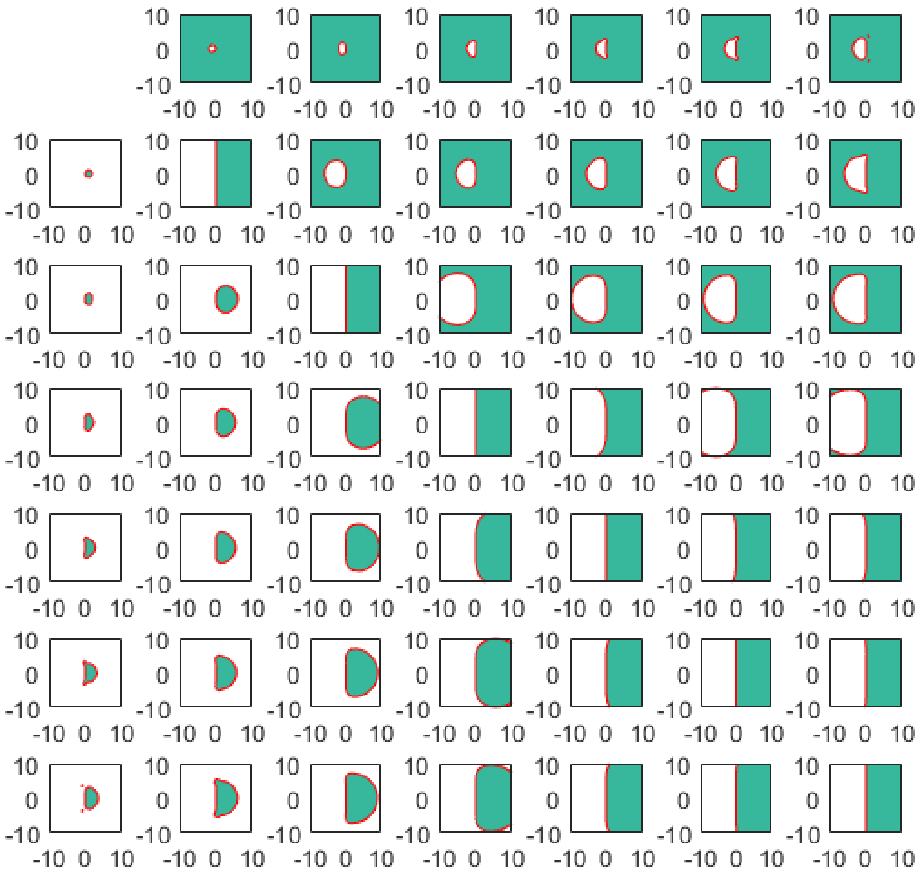

Figure 5: Colored representation of the stability regions S of HOP-methods considering R_{k_e,k_i} for all combinations of $k_e, k_i = 0, \ldots, 6$, where k_e corresponds to the rows and k_i to the columns. The symmetry results form (24). For $k_e = k_i$, we observe $S = \mathbb{C}^-$.

Moreover, analogously as before, we have

$$R^{HOP}_{k_e,k_i}(-z) = \frac{1}{R^{HOP}_{k_i,k_e}(z)}. \tag{24}$$

Therefore, in Figure 5 we obtain again symmetric stability regions that are in accordance with the stability properties reported in Section 5.2.

Since we obtained (22) and (24) for TH and HOP methods with $k = k_e = k_i$, it holds for these methods that

$$1 = R_{k,k}(it)R_{k,k}(-it) = R_{k,k}(it)\overline{R_{k,k}(it)} = \left|R_{k,k}(it)\right| \quad \text{for all } t \in \mathbb{R}.$$

According to Lemma 6.20 from [4] and to the representations of the stability regions from Figure 4, $R^{TH}_{k,k}$ seems to have poles in $\mathbb{C}^-$ in general.

B Linear Examples from Section 8.2

The following two examples, wich result from servo-constraint problems for multibody systems, are linear DAEs of the form

$$Ax' + Bx = q.$$

B.1 Example: Mass-on-Car

The DAE resulting from the spring-mass system mounted on a car from [23] corresponds to

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & m_1+m_2 & m_2\cos(\alpha) & 0 \\ 0 & 0 & m_2\cos(\alpha) & m_2 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$B = \begin{pmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & k & 0 & d & 0 \\ 1 & \cos(\alpha) & 0 & 0 & 0 \end{pmatrix}, \quad q = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ y_d \end{pmatrix},$$

for $x = (x_1, s, v_{x_1}, v_s, F)$. We used the parameters $m_1 = 1.0$, $m_2 = 2.0$, $k = 5.0$, $d = 1.0$, $\alpha = \frac{5}{180}\pi$. y_d is a predefined trajectory for the position of the mass m_2

and reads

$$y_d(t) = \begin{cases} y_0 \quad + \quad \left(126\left(\frac{t}{t_{max}}\right)^5 - 420\left(\frac{t}{t_{max}}\right)^6 + 540\left(\frac{t}{t_{max}}\right)^7 \right. \\ \qquad \left. -315\left(\frac{t}{t_{max}}\right)^8 + 70\left(\frac{t}{t_{max}}\right)^9\right) \cdot (y_f - y_0) \\ \qquad \text{for } 0 \leq t \leq t_{max} \\ y_f \qquad \text{for } t \geq t_{max} \end{cases}$$

for $y_0 = 0.5$, $y_f = 2.5$, $t_{max} = 6.0$.

For $0 < \alpha < \frac{\pi}{2}$, the DAE-index is 3 and the projector Π from equation (2) reads

$$\Pi = \frac{1}{1+\cos^2(\alpha)} \begin{pmatrix} \cos(\alpha)^2 & -\cos(\alpha) & 0 & 0 & 0 \\ -\cos(\alpha) & 1 & 0 & 0 & 0 \\ 0 & 0 & \cos(\alpha)^2 & -\cos(\alpha) & 0 \\ 0 & 0 & -\cos(\alpha) & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

i.e., it depends on α only and is independent of the other parameters.

B.2 Example: Extended Mass-on-Car System

The DAE resulting from the extension of the mass-on car systems described in [19] corresponds to

$$A = \begin{pmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & 1 & & & & \\ & & & m_1+m_2+m_3 & m_2+m_3 & m_3\cos\alpha & \\ & & & m_2+m_3 & m_2+m_3 & m_3\cos\alpha & \\ & & & m_3\cos\alpha & m_3\cos\alpha & m_3 & \\ & & & & & & 0 \end{pmatrix},$$

$$B = \begin{pmatrix} & & & 1 & & & \\ & & & & 1 & & \\ & & & & & 1 & \\ & & & & & & -1 \\ & k_1 & & d_1 & & & \\ & & k_2 & & d_2 & & \\ 1 & 1 & \cos\alpha & & & & \end{pmatrix}, \quad q = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ z_d(t) \end{pmatrix},$$

for $x = (x_1, s_1, s_2, v_{x_1}, v_{s_1}, v_{s_2}, F)$ and with

$$z_d(t) = \begin{cases} z_0 + \Bigg(-3432\left(\frac{t}{t_{max}}\right)^{15} + 25740\left(\frac{t}{t_{max}}\right)^{14} - 83160\left(\frac{t}{t_{max}}\right)^{13} \\ \quad +150150\left(\frac{t}{t_{max}}\right)^{12} - 163800\left(\frac{t}{t_{max}}\right)^{11} + 108108\left(\frac{t}{t_{max}}\right)^{10} \\ \quad -40040\left(\frac{t}{t_{max}}\right)^{9} + 6435\left(\frac{t}{t_{max}}\right)^{8}\Bigg)(z_f - z_0) \\ \quad \text{for } 0 \le t \le t_{max}, \\ z_f \quad \text{for } t \ge t_{max}, \end{cases}$$

for $z_0 = 1.0$, $z_f = 4.0$, $t_{max} = 15.0$, according to [18]. We used the parameters $m_1 = 1.0$, $m_2 = 1.0$, $m_3 = 2.0$, $k_1 = 5.0$, $k_2 = 5.0$, $d_1 = 1.0$, $d_2 = 1.0$, $\alpha = \frac{\pi}{4}$. In this case, the index (and therefore also the rank and shape of Π) depends on the parameters α, d_1 and d_2.

References

[1] P.G. Akishin, I.V. Puzynin, and S.I. Vinitsky. A hybrid numerical method for analysis of dynamics of the classical Hamiltonian systems. *Computers and Mathematics with Applications Volume 34, Issues 2-4, pp. 45-73*, 1997.

[2] R. Barrio. Performance of the Taylor series method for ODEs/DAEs. *Appl. Math. Comput.*, 163(2):525–545, 2005.

[3] G.F. Corliss, A. Griewank, P. Henneberger, G. Kirlinger, F.A. Potra, and H.J. Stetter. High-order stiff ODE solvers via automatic differentiation and rational prediction. *In: Vulkov L., Waśniewski J., Yalamov P. (eds) Numerical Analysis and Its Applications. WNAA 1996. Lecture Notes in Computer Science.vol 1196*, pages 114–124, 1997.

[4] P. Deuflhard and F. Bornemann. *Numerical mathematics 2. Ordinary differential equations. (Numerische Mathematik 2. Gewöhnliche Differentialgleichungen.) 4th revised and augmented ed.* de Gruyter Studium. Berlin, 2013.

[5] S.N. Dimova, I.G. Hristov, R.D. Hristova, I. V. Puzynin, T.P. Puzynina, Z.A. Sharipov, N.G. Shegunov, and Z.K. Tukhliev. Combined explicit-implicit Taylor Series Methods. In *Proceedings of the VIII International Conference "Distributed Computing and Grid-technologies in Science and Education" (GRID 2018), Dubna, Moscow region, Russia, September 10 -14*, 2018.

[6] D. Estévez Schwarz. A step-by-step approach to compute a consistent initialization for the MNA. *Int. J. Circuit Theory Appl.*, 30(1):1–6, 2002.

[7] D. Estévez Schwarz. Consistent initialization for DAEs in Hessenberg form. *Numer. Algorithms*, 52(4):629–648, 2009.

[8] D. Estévez Schwarz and R. Lamour. InitDAE's documentation. URL: `https://www.mathematik.hu-berlin.de/~lamour/software/python/InitDAE/html/`.

[9] D. Estévez Schwarz and R. Lamour. Projector based integration of DAEs with the Taylor series method using automatic differentiation. *J. Comput. Appl. Math.*, 262:62–72, 2014.

[10] D. Estévez Schwarz and R. Lamour. A new projector based decoupling of linear DAEs for monitoring singularities. *Numer. Algorithms*, 73(2):535–565, 2016.

[11] D. Estévez Schwarz and R. Lamour. Consistent initialization for higher-index DAEs using a projector based minimum-norm specification. Technical Report 1, Institut für Mathematik, Humboldt-Universität zu Berlin, 2016.

[12] D. Estévez Schwarz and R. Lamour. A new approach for computing consistent initial values and Taylor coefficients for DAEs using projector-based constrained optimization. *Numer. Algorithms*, 78(2):355–377, 2018.

[13] D. Estévez Schwarz and R. Lamour. InitDAE: Computation of consistent values, index determination and diagnosis of singularities of DAEs using automatic differentiation in Python. *Journal of Computational and Applied Mathematics*, 2019. `doi:DOI:10.1016/j.cam.2019.112486`.

[14] D. Estévez Schwarz and R. Lamour. A projector based decoupling of DAEs obtained from the derivative array. Technical report, Institut für Mathematik, Humboldt-Universität zu Berlin, 2019.

[15] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II*. Springer, 1996.

[16] G. Kirlinger and G. F. Corliss. On implicit Taylor series methods for stiff ODEs. In *Computer arithmetic and enclosure methods. Proceedings of the 3rd International IMACS-GAMM Symposium on Computer Arithmetic and Scientific Computing (SCAN-91), Oldenburg, Germany, 1-4 October 1991*, pages 371–379. Amsterdam: North-Holland, 1992.

[17] F. Mazzia and C. Magherini. Test set for initial value problems, release 2.4. Technical report, Department of Mathematics, University of Bari and INdAM, Research Unit of Bari, February 2008. URL: `http://pitagora.dm.uniba.it/~testset`.

[18] S. Otto. Private communication.

[19] S. Otto and R. Seifried. *Applications of Differential-Algebraic Equations: Examples and Benchmarks*, chapter Open-loop Control of Underactuated Mechanical Systems Using Servo-constraints: Analysis and Some Examples. Differential-Algebraic Equations Forum. Springer, Cham, 2019.

[20] J. D. Pryce. Solving high-index DAEs by Taylor series. *Numer. Algorithms*, 19(1-4):195–211, 1998.

[21] J. D. Pryce, N. S. Nedialkov, G. Tan, and X. Li. How AD can help solve differential-algebraic equations. *Optimization Methods & Software*, 33(4–6):729–749, 2018.

[22] J. R. Scott. Solving ODE Initial Value Problems with Implicit Taylor Series Methods. Technical report, NASA/TM-2000-209400, 2000.

[23] R. Seifried and W. Blajer. Analysis of servo-constraint problems for underactuated multibody systems. *Mech. Sci., 4, 113-129*, 2013.

[24] S. F. Walter and L. Lehmann. Algorithmic differentiation in Python with AlgoPy. *Journal of Computational Science*, 4(5):334 – 344, 2013.