



Projected explicit and implicit Taylor series methods for DAEs

Diana Estévez Schwarz¹ · René Lamour²

Received: 3 October 2019 / Accepted: 29 November 2020 / Published online: 29 January 2021
© The Author(s) 2021

Abstract

The 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 to 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 in a general framework. In particular, we address higher-order Padé methods that stand out due to their stability. We further discuss several aspects of our prototype implemented in Python using Automatic Differentiation. The methods have been successfully tested on examples arising from multibody systems simulation and a higher-index DAE benchmark arising from servo-constraint problems.

Keywords Taylor series methods · Integration · DAE · Differential-algebraic equation · Consistent initial value · Index · Derivative array · Projector-based analysis · Nonlinear constrained optimization · Automatic differentiation

1 Introduction

Higher-index DAEs do not only represent integration problems but also differentiation problems, as well (see, e.g., [4, 22, 23, 28]). Therefore, it seems worthwhile to solve an associated ODE with classical integration schemes and the differentiation

✉ Diana Estévez Schwarz
estevez@beuth-hochschule.de

René Lamour
lamour@math.hu-berlin.de

¹ Beuth Hochschule für Technik Berlin, Berlin, Germany

² Humboldt Universität zu Berlin, Berlin, Germany

problems using Automatic Differentiation (AD). However, depending on the structure, both differentiations and integrations may be intertwined in a complex manner such that this plausible idea may 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 [26, 27] and the related work, the structural index was used to determine the degree of freedom.
- In [13], we used the tractability matrix sequence to solve the inherent ODE for DAEs of index up to two. The generalization to higher-index DAEs seemed rather complicated.
- In [17], we described briefly how an approach based on the differentiation index defined in [14, 16] leads to an explicit Taylor series methods for DAEs. An analysis of the corresponding projected explicit ODE can be found in [18]. 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 [17]. Here we focus on the methods from [7, 21].

The main idea in this context is that the computation of Taylor coefficients of a solution of an implicit ODE can be considered as the solution of a nonlinear system of equations. In this sense, we will see that a generalization for DAEs can be obtained by solving a nonlinear optimization problem [16]. The obtained solution corresponds to a projected method. There are several advantages of this approach:

- We assume weak structural properties of the DAEs (1), 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. Indeed, due to our formulation as an optimization problem, implicitly we consider the orthogonally projected explicit ODE introduced in [18].
- 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 5 and integration schemes up to order 8 were successful.

The purpose of our implementation is the improvement of our diagnosis software [17]. Therefore, we do not focus on high-performance, but on information about the reliability of the numerical result, specially with regard to higher-index issues and the diagnosis of singularities like [19]. Due to their stability and order properties, the new higher-order methods presented here are a considerable improvement in comparison to [17], where only explicit Taylor series methods were considered.

The paper is organized as follows. In Section 2, we introduce DAEs and summarize some of our previous results that are crucial for the approach presented here.

The properties of Taylor coefficients in the derivative array of a DAE are discussed in Section 3. Using these properties, in Section 4, we define the general projected explicit/implicit Taylor series method that are, indeed, a generalization of the explicit method from [17].

Within this framework, in Section 5, we present four different types of projected methods: explicit methods, fully implicit method, two-halfstep (TH) schemes, and higher-order Padé (HOP) schemes.

The properties of the considered optimization problems that provide the projection are discussed in Section 6 and some practical considerations for the implementation are addressed in Section 7.

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

For completeness, in the Appendix, we summarize the stability functions and stability regions for the considered Taylor series methods, since they are essential for the development of HOP methods. We also summarize some linear algebra results for decoupling DAEs and provide the DAE formulation of the tested examples resulting from servo-constraint problems for multi-body systems.

2 DAEs: index, consistent values, and decoupling

In this article, we consider general DAEs:

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

for $f : \mathcal{G}_f \rightarrow \mathbb{R}^n$, $\mathcal{G}_f \subset \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}$, where the partial Jacobian $f_{x'}$ is singular and $\ker f_{x'}(x', x, t)$ is constant. For our purposes, we define the constant orthogonal projector Q onto $\ker f_{x'}$ as well as the complementary orthogonal projector $P := I - Q$.

Recall that the singularity of $f_{x'}$ means that (1) contains derivative-free equations, called explicit constraints, and that the differentiation of (1) may lead to further derivative-free equations, called hidden constraints. A consistent initial value x_0 has to fulfill all explicit and implicit constraints. The characterization of all these constraints motivated the following definition for the differentiation index, cf. [4].

Definition 1 [14] The differentiation index is the smallest integer μ such that:

$$f(x', x, t) = 0, \tag{2}$$

$$\frac{d}{dt} f(x', x, t) = 0, \tag{3}$$

⋮

$$\frac{d^{\mu-1}}{dt^{\mu-1}} f(x', x, t) = 0, \tag{4}$$

uniquely determines Qx as a function of (Px, t) .

If μ is the differentiation index according to Definition 1, then the conventional differentiation index (see, e.g., [4]) results to be μ as well. According to this definition, in the following we will never prescribe initial values for Qx_0 , since we may compute Qx_0 evaluating a function at (Px_0, t_0) . Moreover, in the higher-index case, the Eqs. (2)–(4) contain explicit and hidden constraints that restrict the choice for Px_0 .

According to [16], for an initial guess $\alpha \in \mathbb{R}^n$, consistent initial values x_0 can be computed solving the following constrained optimization problem:

$$\min \quad \|P(x_0 - \alpha)\|_2 \quad (5)$$

$$\text{subject to all explicit and hidden constraints.} \quad (6)$$

Equivalently, we can solve the system of equations:

$$\Pi(x_0 - \alpha) = 0 \quad (7)$$

$$\text{all explicit and hidden constraints} \quad (8)$$

for a suitable orthogonal projector Π with $\text{rank } \Pi = d \leq \text{rank } P$, where d is the degree of freedom of (1), cf. [15]. However, in particular for nonlinear DAEs, Π may be difficult to compute and therefore, in practice, (5) may be more convenient than (7), cf. Appendix 2. The approach (5)–(6) was implemented in InitDAE, a Python program to determine the index and consistent initial values for DAEs [12, 17].

Furthermore, for linear DAEs of the form:

$$A(t)x' + B(t)x = q(t), \quad (9)$$

in [18], it was shown that using the derivative array (2)–(4), the decoupling:

$$(\Pi x)' = \varphi_1(\Pi x, q, q', \dots, q^{(\mu)}), \quad (10)$$

$$(I - \Pi)x = \varphi_2(\Pi x, q, q', \dots, q^{(\mu-1)}), \quad (11)$$

can be obtained for suitable functions φ_1, φ_2 , where (10) is an ODE in the invariant subspace $\text{im } \Pi$. Therefore, theoretically, we can set up (10) and solve it with an integration scheme for ODEs. Subsequently, $(I - \Pi)x$ can be computed at each time point using (11). Notice that in doing so, the error in $(I - \Pi)x$ depends only on the error made solving (10) and the properties of φ_2 from (11). Moreover, the values of $(I - \Pi)x$ at previous time points do not influence (10).

For nonlinear DAEs, analogous considerations can be undertaken considering the linearization along a solution. However, of course, the properties of f are decisive in practice. Since a detailed analysis for the nonlinear case goes far beyond the scope of this article, we focus on a general formulation of projected Taylor methods using (5)–(6), having in mind that at present, the theoretical basis has been developed for linear DAEs only. At least the numerical tests from Section 8 suggest the applicability for some classes of nonlinear DAEs.

3 Taylor series and DAEs

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

$$[(c_0)_j, (c_1)_j, (c_2)_j, \dots, (c_{k_c})_j] \approx [x(t_j), x'(t_j), \frac{1}{2}x''(t_j), \dots, \frac{1}{k_c!}x^{(k_c)}(t_j)] \quad (12)$$

is given and that we look at adequate methods to compute:

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

If we suppose that the ODE/DAE is described by (1), we require that:

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

are fulfilled. For our purposes, given $K \geq 1$, we further consider the order $(K - 1)$ derivative array derivative array [4] containing (1) and $K - 1$ derivatives of (1):

$$\begin{pmatrix} f(x', x, t) \\ \frac{d}{dt} f(x', x, t) \\ \frac{d^2}{dt^2} f(x', x, t) \\ \vdots \\ \frac{d^{(K-1)}}{dt^{(K-1)}} f(x', x, t) \end{pmatrix} = \begin{pmatrix} f(x', x, t) \\ f_{x'}(x', x, t)x'' + f_x(x', x, t)x' + f_t(x', x, t) \\ \vdots \\ \vdots \end{pmatrix}. \quad (13)$$

Therefore, we suppose that for the corresponding function:

$$\begin{aligned} r(c_0, c_1, \dots, c_K, t) &:= \begin{pmatrix} r_0(c_0, c_1, t) \\ r_1(c_0, c_1, c_2, t) \\ \vdots \\ r_{K-1}(c_0, c_1, \dots, c_K, t) \end{pmatrix} \\ &:= \begin{pmatrix} f(c_1, c_0, t) \\ 2f_{x'}(c_1, c_0, t)c_2 + f_x(c_1, c_0, t)c_1 + f_t(c_1, c_0, t) \\ \vdots \end{pmatrix} \end{aligned} \quad (14)$$

it holds:

$$r((c_0)_j, (c_1)_j, (c_2)_j, \dots, (c_K)_j, t_j) = 0 \quad (15)$$

and

$$r((c_0)_{j+1}, (c_1)_{j+1}, (c_2)_{j+1}, \dots, (c_K)_{j+1}, t_{j+1}) = 0. \quad (16)$$

In practice, the function r can be provided by automatic differentiation (AD) [17, 31].

Using this notation, the index from Definition 1 is the smallest integer μ such that for $K = \mu$, the derivative array r uniquely determines Qc_0 as a function of (Pc_0, t) . For this purpose, the 1-fullness of the Jacobian of r is verified in [16, 17], cf. Appendix 2. We emphasize that the main difference to the conventional differentiation index [4] is precisely that for 1-fullness the columns corresponding

to c_0 (and not c_1) are considered. With this index definition in mind, we can define consistency for the Taylor coefficients.

Definition 2 For $K \geq \mu$ and $0 \leq k_c \leq K - \mu$, the Taylor coefficients up to k_c are consistent if they are in the set:

$$\mathcal{T}_{k_c}^j := \left\{ [(c_0)_j, (c_1)_j, \dots, (c_{k_c})_j] \in \mathbb{R}^{(k_c+1) \cdot n} \mid \begin{array}{l} \text{There exist } (\tilde{c}_{k_c+1})_j, \dots, (\tilde{c}_K)_j \text{ such that} \\ r((c_0)_j, \dots, (c_{k_c})_j, (\tilde{c}_{k_c+1})_j, \dots, (\tilde{c}_K)_j, t_j) = 0 \end{array} \right\}$$

Note that \mathcal{T}_0^0 corresponds to the set of consistent initial values and that if sufficient smoothness of f is given, we can suppose that for all $c_0 \in \mathcal{T}_0^0$ in regularity regions, there is a unique solution of the initial value problem. For a discussion of regularity regions and singularities within a projector-based analysis, we refer to [14, 23].

For sufficiently smooth regular linear DAEs (9), Theorem 1 from [18] implies that for any $c_0 \in \mathcal{T}_0^0$, there is a unique solution fulfilling $x(t_0) = c_0$. Analogously, for $[(c_0)_j, (c_1)_j, \dots, (c_{k_c})_j] \in \mathcal{T}_{k_c}^j$, there exists a unique solution $x(t)$ such that $\frac{x^{(k)}}{k!}(t_j) = (c_k)_j$, $0 \leq k \leq k_c$. Indeed, Theorem 1 from [18] provides a general description of the inherent dynamics in terms of the associated orthogonally projected explicit ODE (10).

Let us focus on the relation between k_c , K , the DAE-index μ and the computation of consistent initial values and Taylor coefficients at a particular t_0 considering:

$$r((c_0)_0, (c_1)_0, (c_2)_0, \dots, (c_K)_0, t_0) = 0. \tag{17}$$

- For initial value problems $x(t_0) = c_0$ for ODEs $g(x'(t), x(t), t) = 0$ with regular $g_{x'}$, if we consider (17), then we can compute K consistent coefficients:

$$(c_1)_0, (c_2)_0, (c_3)_0, \dots, (c_K)_0,$$

at t_0 , since c_0 is given. In the above notation, the maximal value for k_c is $k_c = K$.

- If we consider an uniquely solvable nonlinear time-dependent equation $g(x(t), t) = 0$ with regular g_x and a corresponding system of equations (17), then, at t_0 , we cannot prescribe c_0 and compute therefore K consistent coefficients:

$$(c_0)_0, (c_1)_0, (c_2)_0, \dots, (c_{K-1})_0.$$

In this case, the maximal value is $k_c = K - 1$. For the coefficients $(c_K)_0$, no equations are given, since in this particular case, they do not appear in (17). Note that in principle, $g(x, t) = 0$ can be considered an index-1 DAE. In this sense, it fits into the case below.

- For DAEs (1), if we consider (17) and fix the free initial conditions of c_0 , then in general, we may compute $K + 1 - \mu$ consistent coefficients:

$$(c_0)_0, (c_1)_0, (c_2)_0, \dots, (c_{K-\mu})_0, \tag{18}$$

cf. [17]. In this case, we have at most $k_c = K - \mu$. In general, the coefficients $c_{K+1-\mu}, \dots, c_K$ cannot be computed considering (17). Another crucial aspect is that not all components of c_0 can be prescribed, since all the constraints have to be satisfied.

Note that according to (5)–(6), for an arbitrary initial guess α that, in general, may be not consistent, the optimization problem:

$$\begin{aligned} \min \quad & \|P((c_0)_0 - \alpha)\|_2 \tag{19} \\ \text{subject to} \quad & r((c_0)_0, (c_1)_0, (c_2)_0, \dots, (c_K)_0, t_0) = 0, \tag{20} \end{aligned}$$

provides consistent initial values (18). Moreover, in terms of (7)–(8), this minimization problem is equivalent to the system of equations:

$$\begin{aligned} \Pi((c_0)_0 - \alpha) &= 0, \tag{21} \\ r((c_0)_0, (c_1)_0, (c_2)_0, \dots, (c_K)_0, t_0) &= 0, \tag{22} \end{aligned}$$

where Π describes an appropriate orthogonal projector, and the rank of Π coincides with the number of degrees of freedom of the DAE [15, 18]. Note further that, in general, the coefficients $[(c_{K+1-\mu})_0, \dots, (c_K)_0]$ are not uniquely determined neither by (19)–(20) nor by (21)–(22). In our implementation from [17], the minimum norm solution $[(\tilde{c}_{K+1-\mu})_0, \dots, (\tilde{c}_K)_0]$ is computed.

Example 1 Consider the index-4 DAE:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} x' + \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} x = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ e^t \end{pmatrix} \tag{23}$$

with the general solution:

$$x(t; C) = \begin{pmatrix} C e^{-t} - \frac{e^t}{2} \\ -e^t \\ e^t \\ -e^t \\ e^t \end{pmatrix}. \tag{24}$$

For this clearly structured example, the projector-based approach will lead to:

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

This means that Qx corresponds to x_2 and according to the notation introduced in [18], the EOPE-ODE (the essential orthogonally projected ODE describing the dynamics) that corresponds to (10) will be formulated in terms of x_1 .

Let us have a closer look to the derivative array with respect to the index determination and the computation of consistent initial values, both related to the computation of Π , see Appendix 2.

According to (12), for $n = 5$ and $K = 4$, we consider:

$$(c_\ell)_{t_j} = (c_{1\ell} \ c_{2\ell} \ c_{3\ell} \ c_{4\ell} \ c_{5\ell})^T, \quad \ell = 0, \dots, 4,$$

such that the equations on the left, that are formulated for functions as described in (13), correspond to the equations on the right, that can be formulated at some $t = t_j$ for the scalar numbers $c_{k\ell} = \frac{x_k^{(\ell)}(t_j; C)}{\ell!}$ (cf. (14)):

$$\begin{aligned} f &= \begin{pmatrix} x'_1 + x_1 + x_2 \\ x'_3 + x_2 \\ x'_4 + x_3 \\ x'_5 + x_4 \\ x_5 - e^t \end{pmatrix}, & r_0(c_0, c_1, t_j) &= \begin{pmatrix} c_{11} + c_{10} + c_{20} \\ c_{31} + c_{20} \\ c_{41} + c_{30} \\ c_{51} + c_{40} \\ c_{50} - e^{t_j} \end{pmatrix}, \\ \frac{d}{dt}f &= \begin{pmatrix} x''_1 + x'_1 + x'_2 \\ x''_3 + x'_2 \\ x''_4 + x'_3 \\ x''_5 + x'_4 \\ x'_5 - e^t \end{pmatrix}, & r_1(c_0, c_1, c_2, t_j) &= \begin{pmatrix} 2c_{12} + c_{11} + c_{21} \\ 2c_{32} + c_{21} \\ 2c_{42} + c_{31} \\ 2c_{52} + c_{41} \\ c_{51} - e^{t_j} \end{pmatrix}, \\ \frac{d^2}{dt^2}f &= \begin{pmatrix} x'''_1 + x''_1 + x''_2 \\ x'''_3 + x''_2 \\ x'''_4 + x''_3 \\ x'''_5 + x''_4 \\ x''_5 - e^t \end{pmatrix}, & r_2(c_0, \dots, c_3, t_j) &= \begin{pmatrix} 6c_{13} + 2c_{12} + 2c_{22} \\ 6c_{33} + 2c_{22} \\ 6c_{43} + 2c_{32} \\ 6c_{53} + 2c_{42} \\ 2c_{52} - e^{t_j} \end{pmatrix}, \\ \frac{d^3}{dt^3}f &= \begin{pmatrix} x''''_1 + x'''_1 + x'''_2 \\ x''''_3 + x'''_2 \\ x''''_4 + x'''_3 \\ x''''_5 + x'''_4 \\ x'''_5 - e^t \end{pmatrix}, & r_3(c_0, \dots, c_4, t_j) &= \begin{pmatrix} 24c_{14} + 6c_{13} + 6c_{23} \\ 24c_{34} + 6c_{23} \\ 24c_{44} + 6c_{33} \\ 24c_{54} + 6c_{43} \\ 6c_{53} - e^{t_j} \end{pmatrix}. \end{aligned}$$

The colors visualize the chains of calculation by with each entry of $(I - \Pi)c_0$ is uniquely determined by the equations $r = 0$, cf. (15). In particular, the red equations permit the computation of Qc_0 , since:

$$c_{20} = -c_{31} = 2c_{42} = -6c_{53} = -e^{t_j}.$$

Since no representation of c_{20} is possible with less differentiations, the index is $\mu = 4$. Moreover, the violet, blue, and green expressions provide values for components of Pc_0 , in particular c_{50} , c_{40} , and c_{30} , respectively. This means that we cannot prescribe initial values for $(P - \Pi)c_0$.

Indeed, the EOPE-ODE reads:

$$x'_1 + x_1 = e^t. \quad (25)$$

Summarizing, we see that to compute $(c_0)_0$, we have to prescribe a value for $x_1(t_0)$ and consider at least derivatives of order up to three (with $K = \mu = 4$) in the derivative array, i.e., $r((c_0)_0, \dots, (c_3)_0, (c_4)_0, t_0) = 0$.

We emphasize that the gray items must vanish in order to satisfy $r = 0$, but for $K = \mu - 1$, they do not uniquely determine all coefficients of $(c_\ell)_0$ for any $\ell > 0$. This means that the value for k_c from Definition 2 is $k_c = 0 = K - \mu$.

If we increase the number of derivatives with $K = \mu + 1 = 5$ and consider $r((c_0)_0, (c_1)_0, (c_2)_0, \dots, (c_5)_0, t_0) = 0$ together with an initial value for c_{10} , then correct values for $(c_0)_0$ and $(c_1)_0$ can be computed. In general, for $K \geq \mu$, consistent $(c_0)_0, \dots, (c_{K-\mu})_0$ can be obtained.

Since with the approach (19)–(20) the projector Π is not computed explicitly and at least we consider nonlinear under-determined systems of equations, they are solved in a minimum-norm sense. Therefore, the used solver obtains values for all higher derivatives, whereas we cut off $(\tilde{c}_{K-\mu+1})_0, \dots, (\tilde{c}_K)_0$, since only $(c_0)_0, \dots, (c_{K-\mu})_0$ are consistent in the sense of Definition 2.

In Table 1, we present the results of the computation of consistent initial values with InitDAE [12] that solves (19)–(20). For the considered initial value $x_1(0) = 1$, the solution is $x_1(t) = \cosh(t)$. We can appreciate that for $K = 5$, only the Taylor coefficient $(c_0)_0$ and $(c_1)_0$ are consistent, i.e., $k_c = 1 = K - \mu$. Increasing K , correspondingly more consistent Taylor coefficients could be computed.

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 (10) for Πx , and
- corresponding values for the components $(I - \Pi)x$ that result from (11).

Therefore, the stability and order properties of the integration methods defined below can be transferred from ODEs to DAEs. 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.

Table 1 Numerical solution of the initialization problem for system (23) from Example 1 for $t_0 = 0$ and $\alpha = [1, 0, 0, 0, 0]$ using Taylor coefficients with $K = 5$

	x_1	x_2	x_3	x_4	x_5
c_0	1.00000000e+00	-1.00000000e+00	1.00000000e+00	-1.00000000e+00	1.00000000e+00
c_1	-4.09190611e-15	-1.00000000e+00	1.00000000e+00	-1.00000000e+00	1.00000000e+00
\tilde{c}_2	5.00000000e-01	-4.76883765e-02	5.00000000e-01	-5.00000000e-01	5.00000000e-01
\tilde{c}_3	-1.50770541e-01	8.66099724e-03	1.58961255e-02	-1.66666667e-01	1.66666667e-01
\tilde{c}_4	3.55273860e-02	-1.31582911e-03	-2.16524931e-03	-3.97403138e-03	4.16666667e-02
\tilde{c}_5	-6.84231137e-03	0.00000000e+00	2.63165822e-04	4.33049862e-04	7.94806275e-04

Due to the index $\mu = 4$, only c_0 and c_1 are consistent. In fact, the boxed values do not correspond to the solution (24)

4 General definition of explicit/implicit methods

Recall that:

- $P = \Pi = I$ holds for ODEs and that, therefore, for ODEs, the approach (19)–(20) means to compute the Taylor coefficients if c_0 is prescribed.
- We assumed that $\ker f_{x'}(c_1, c_0, t)$ and therefore also P do not depend on (c_1, c_0, t) . Therefore, the Taylor coefficients of $Px(t)$ at t_j correspond to:

$$[P(c_0)_j, P(c_1)_j, P(c_2)_j, \dots, P(c_{k_e})_j].$$

With these two properties in mind, we can present a very general formulation for implicit and explicit Taylor series methods for ODEs and DAEs defining suitable objective functions instead of (19).

In a first step, we focus on consistency.

Lemma 1 Consider:

$$[(c_0)_j, (c_1)_j, (c_2)_j, \dots, (c_K)_j],$$

$k_e, k_i \in \mathbb{N}, 0 \leq k_e, k_i \leq K$ and weights $\omega_{\ell_e}^e, \omega_{\ell_i}^i \in \mathbb{R}$ to define the objective function

$$p((c_0)_{j+1}, \dots, (c_{k_i})_{j+1}) := P \left(\sum_{\ell_i=0}^{k_i} \omega_{\ell_i}^i (c_{\ell_i})_{j+1} (-h_j)^{\ell_i} - \sum_{\ell_e=0}^{k_e} \omega_{\ell_e}^e (c_{\ell_e})_j (h_j)^{\ell_e} \right).$$

Then for any solution:

$$[(c_0)_{j+1}, (c_1)_{j+1}, (c_2)_{j+1}, \dots, (c_K)_{j+1}], \tag{26}$$

of the minimization problem:

$$\min \quad \|p((c_0)_{j+1}, (c_1)_{j+1}, (c_2)_{j+1}, \dots, (c_{k_i})_{j+1})\|_2 \tag{27}$$

$$\text{subject to} \quad r((c_0)_{j+1}, (c_1)_{j+1}, (c_2)_{j+1}, \dots, (c_K)_{j+1}, t_{j+1}) = 0 \tag{28}$$

the values:

$$[(c_0)_{j+1}, (c_1)_{j+1}, (c_2)_{j+1}, \dots, (c_{k_i})_{j+1}], \tag{29}$$

are consistent at t_{j+1} for all $k_i \leq K - \mu$.

Proof According to Definition 2, the coefficients (29) are consistent, since the constraints (28) are fulfilled. Recall further that, under suitable assumptions, the solvability of (27)–(28) follows from the Definition 1 of the index μ and has been discussed in [16]. □

Corollary 1 Consider linear DAEs (9) and suppose that consistent values:

$$[(c_0)_j, (c_1)_j, (c_2)_j, \dots, (c_{k_e})_j],$$

are given. Consider further an integration method defined by (27)–(28) for suitable weights $\omega_{\ell_e}^e, \omega_{\ell_i}^i$. Then the following two approaches provide the same consistent results (29):

- the solution of (27)–(28) for the original DAE (9),
- the solution (27)–(28) for the ODE (10) that is invariant in the subspace in Π , and the subsequent computation of the remaining components according to (11).

Proof On the one hand, according to (21)–(22), the solution of (27)–(28) for the original DAE delivers the same result (29) as:

$$\Pi(t_{j+1}) \left(p \left((c_0)_{j+1}, \dots, (c_{k_i})_{j+1} \right) \right) = 0, \tag{30}$$

$$r \left((c_0)_{j+1}, (c_1)_{j+1}, (c_2)_{j+1}, \dots, (c_K)_{j+1}, t_{j+1} \right) = 0. \tag{31}$$

On the other hand, the derivative array (31) for the DAE contains the derivative array of the ODE (10) and the nonlinear equations (11). □

With the notation of the objective function, different Taylor integration methods can be described with corresponding weights $\omega_{\ell_e}^e, \omega_{\ell_i}^i$. This provides us a very flexible way to implement schemes with different order and stability properties.

5 Projected Taylor integration methods

5.1 Explicit Taylor series method for DAEs

In terms of the above notation, the explicit Taylor series method for ODEs corresponds to $k_e \geq 1, \omega_{\ell_e}^e = 1$ for $0 \leq \ell_e \leq k_e, k_i = 0, \omega_{\ell_i}^i = 1$. Recall that the approach from [17] for DAEs consists of the following steps:

- Initialization: Solve the optimization problem:

$$\min \quad \|P((c_0)_0 - \alpha)\|_2 \tag{32}$$

$$\text{subject to} \quad r((c_0)_0, (c_1)_0, (c_2)_0, \dots, (c_K)_0, t_0) = 0, \tag{33}$$

for an initial guess α .

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

$$\min \quad \left\| P \left((c_0)_{j+1} - \underbrace{\sum_{\ell=0}^{k_e} (c_\ell)_j h_j^\ell}_{\approx x(t_j+h_j)} \right) \right\|_2, \tag{34}$$

$$\text{subject to} \quad r((c_0)_{j+1}, (c_1)_{j+1}, (c_2)_{j+1}, \dots, (c_K)_{j+1}, t_{j+1}) = 0, \tag{35}$$

successively for $k_e \leq K - \mu$, where k_e is the order of the series in t_j .

This method is called explicit, since (32) is an explicit equation for $(c_0)_{j+1}$ that does not involve any value $(c_\ell)_{j+1}$ for $\ell \geq 1$. In contrast to explicit ODEs, where Taylor coefficients may be obtained by function evaluation, with this approach for DAEs in general, we have to solve a nonlinear system of equations. Therefore, it seems reasonable to consider also implicit Taylor approximations in the integration scheme.

5.2 Fully implicit Taylor series methods for DAEs

The implicit counterpart of the explicit Taylor series method for ODEs corresponds to $k_i \geq 1$, $\omega_{\ell_i}^i = 1$ for $0 \leq \ell_i \leq k_i$, $k_e = 0$, $\omega_0^e = 1$. Our generalization for DAEs consists, therefore, of the following steps.

- Initialization as in Section 5.1, Eqs. (32)–(33).
- For time-points t_{j+1} , $j \geq 0$, $h_j = t_{j+1} - t_j$: Solve the optimization problems:

$$\min \quad \underbrace{\|P(\sum_{\ell=0}^{k_i} (c_{\ell})_{j+1} (-h_j)^{\ell} - (c_0)_j)\|_2}_{\approx x(t_{j+1}-h_j)} \quad (36)$$

$$\text{subject to} \quad r((c_0)_{j+1}, (c_1)_{j+1}, (c_2)_{j+1}, \dots, (c_K)_{j+1}, t_{j+1}) = 0, \quad (37)$$

successively for $k_i \leq K - \mu$, where k_i is the order at t_{j+1} .

Obviously, if, instead of (34) and (36), more general conditions of the type (27) are considered, then the dimension of the system that has to be solved remains equal. Therefore, it seems natural to consider more general schemes with better convergence and stability properties than the explicit and the fully implicit Taylor series methods.

5.3 Two-halfstep explicit/implicit 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:

$$x(t_j + \sigma h_j) \approx \sum_{\ell_e=0}^{k_e} (c_{\ell_e})_j (\sigma h_j)^{\ell_e}, \quad (38)$$

$$x(t_{j+1} - (1 - \sigma)h_j) \approx \sum_{\ell_i=0}^{k_i} (c_{\ell_i})_{j+1} (-(1 - \sigma)h_j)^{\ell_i}, \quad (39)$$

and equalize the expressions from both right-hand sides. The properties of the methods (38)–(39) are described in [21]. 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 [2, 9].

Remark 1 Note that another closely related implicit/explicit scheme is described in the literature, see, e.g., [29]. There, the first step is implicit and the second one explicit, in contrast to the approach from above. According to the extensive analysis from [29], $\sigma = \frac{1}{2}$ is convenient also in that case. However, for $0 < \sigma < 1$, 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 and t_{j+1} .

In the notation from Section 4, choosing $\sigma = \frac{1}{2}$ in (38)–(39) means to consider:

$$p := P \left(\sum_{\ell_i=0}^{k_i} (c_{\ell_i})_{j+1} \left(-\frac{h_j}{2}\right)^{\ell_i} - \sum_{\ell_e=0}^{k_e} (c_{\ell_e})_j \left(\frac{h_j}{2}\right)^{\ell_e} \right) \tag{40}$$

for $0 \leq k_e, k_i \leq K - \mu$, i.e.:

$$\omega_{\ell_e}^e = \left(\frac{1}{2}\right)^{\ell_e}, \ell_e = 0, \dots, k_e, \quad \omega_{\ell_i}^i = \left(\frac{1}{2}\right)^{\ell_i}, \ell_i = 0, \dots, k_i.$$

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 1) of a (k_e, k_i) -TH method is not a Padé approximation of the exponential function. Consequently, the maximally achievable order of the integration method for fixed k_e and k_i is not given for these particular weights in general. Therefore, further higher-order schemes for stiff ODEs have been developed, namely the HOP-methods described in [7].

5.4 Higher-order Padé methods

According to [7], HOP may be interpreted as Hermite-Obrechhoff-Padé or higher-order Padé. The corresponding integration schemes may be considered as implicit Taylor series methods based on Hermite quadratures.

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

$$\begin{aligned} \omega_{\ell_e}^e &:= \frac{k_e!(k_e + k_i - \ell_e)!}{(k_e + k_i)!(k_e - \ell_e)!}, & \ell_e = 0, \dots, k_e, \\ \omega_{\ell_i}^i &:= \frac{k_i!(k_e + k_i - \ell_i)!}{(k_e + k_i)!(k_i - \ell_i)!}, & \ell_i = 0, \dots, k_i. \end{aligned}$$

These coefficients correspond to the (k_e, k_i) -Padé approximation of the exponential function such that the stability function $R(z)$ is precisely this approximation, see Appendix 1. Indeed, (k_e, k_i) -HOP methods have the following properties, cf. [7]:

- 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. [7].

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

6 Properties of the minimization problems

In [16], 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. [17]. To appreciate the properties for implicit methods (i.e., $k_i > 0$), we define, for $k \geq \max \{k_e, k_i\}$, the matrices:

$$T_e := \left(P\omega_0^e \ P\omega_1^e h_j \ P\omega_2^e h_j^2 \ \dots \ P\omega_k^e h_j^k \right) \in \mathbb{R}^{n \times n \cdot (k+1)}$$

$$T_i := \left(P\omega_0^i \ P\omega_1^i (-h_j) \ P\omega_2^i h_j^2 \ \dots \ P\omega_k^i (-h_j)^k \right) \in \mathbb{R}^{n \times n \cdot (k+1)},$$

assuming $\omega_{\ell_i} = 0$ for $\ell_i > k_i$ and $\omega_{\ell_e} = 0$ for $\ell_e > k_e$, and the vectors:

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

With this notation, we write:

$$p := P \left(\sum_{\ell_i=0}^{k_i} \omega_{\ell_i}^i (c_{\ell_i})_{j+1} (-h_j)^{\ell_i} - \sum_{\ell_e=0}^{k_e} \omega_{\ell_e}^e (c_{\ell_e})_j (h_j)^{\ell_e} \right)$$

$$= T_i X_{j+1} - T_e X_j.$$

Therefore, as in [16], for $\alpha := T_e X_j$, $X := X_{j+1}$, we consider the objective function:

$$f(X) := \frac{1}{2} \|T_i X - \alpha\|^2$$

$$= \frac{1}{2} \|P(T_i X - \alpha)\|^2$$

$$= \frac{1}{2} (T_i X - \alpha)^T P(T_i X - \alpha)$$

$$= \frac{1}{2} \left(X^T (T_i)^T P T_i X - 2\alpha^T P X + \alpha^T P \alpha \right).$$

Observe that the matrix:

$$\tilde{P}_i := (T_i)^T P T_i = (T_i)^T T_i$$

$$= \begin{pmatrix} P(\omega_0^i)^2 & P\omega_0^i \omega_1^i (-h_j)^2 & \dots & P\omega_0^i \omega_k^i (-h_j)^k \\ P\omega_0^i \omega_1^i (-h_j)^2 & & & \\ \vdots & & & \\ P\omega_0^i \omega_k^i (-h_j)^k & P\omega_1^i \omega_k^i (-h_j)^K & \dots & P(\omega_k^i)^2 (-h_j)^{2k} \end{pmatrix}$$

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

is, by construction, positive semi-definite. However, it is not an orthogonal projector in general. Therefore, Theorem 1 and Corollary 1 of [16] cannot be applied directly. Hence, the solvability of the optimization problem is more difficult 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 [16].

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 (27)–(28) leads to a nonlinear system of equations of dimension $2n \cdot (K + 1)$, cf. [16]. 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 introduced Lagrange-multipliers 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 $0 \leq k_e, k_i \leq K - \mu$ 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. So far, we considered schemes with constant order and stepsize only.

7.3 Jacobian matrices

To solve the optimization problems (27)–(28) numerically, we provide the corresponding Jacobians.

- The Jacobian of the constraints (28) is described in [17], since it is also used for the computation of consistent initial values.
- To describe the Jacobian of the objective function (27), which is a gradient, we define:

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

and realize that:

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

for $q \neq 0, 0 \leq \ell_i \leq k_i$.

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 stepsizes. The results can be found in Fig. 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 (25).

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 remarkable.
- 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 Numerical test for examples from the literature

8.2.1 Validation of known results

We 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 [30], see Appendix, Section C.1,
- Extended mass-on-car from [25], see Appendix, Section C.2,
- Pendulum index 3, which can be found in almost all introductions to DAEs, in the reduced to first-order form, with the positive y axis pointing upwards and the parameters $m = 1.0$, $l = 1.0$, and $g = 1.0$. (x_1, x_2) are the coordinates, (x_3, x_4) the corresponding velocities and x_5 the Lagrange parameter. In our computation, the system starts from rest at 45 degrees to the vertical.
- Car axis index 3 formulation with all parameters as given in [24]. In order to avoid a disadvantageous scaling of the Taylor coefficients, we changed the independent variable t to $\tau = 10 t$. This is advantageous, since the time-dependent input-function is $y_b = 0.1 \sin(10 t)$. For large K , the corresponding higher Taylor coefficient lead to considerable scaling differences that are avoided by the substitution with τ . For the details of our reformulation, we refer to [17].

For all examples (see also Table 2), 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.

All tests confirmed the applicability of the method. The solution graphs look identical with those given in the literature [24, 25, 30]. The graphs of the estimated errors of the (3,3)-HOP methods in Fig. 2 confirm the order expectations.

Since it is obvious that our implementation is not competitive with respect to runtime (see Table 3), we have not made a systematic comparison with other solvers here.

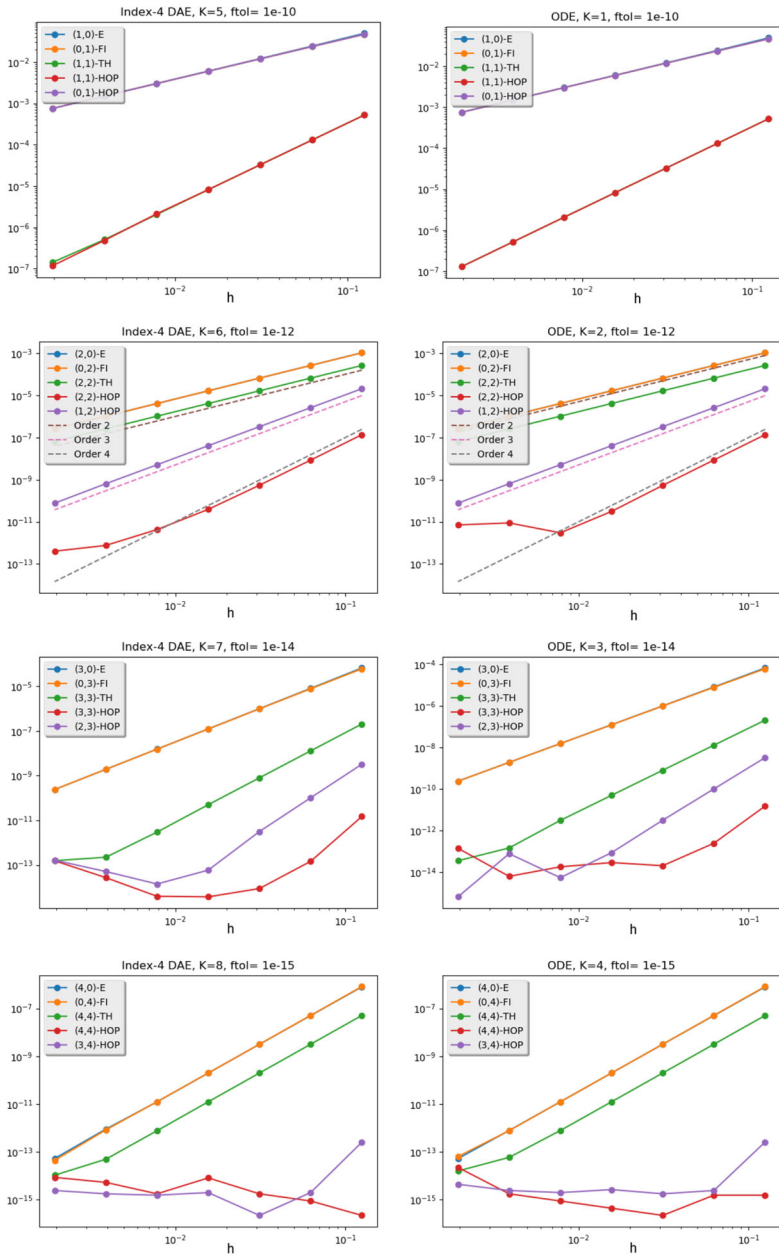


Fig. 1 Stepsize-error diagram for the error $|x_1(1) - \cosh(1)|$ for the DAE (left) and the essential ODE (right) corresponding to Example 1 for different methods and $ftol$ for the module *minimize* from SciPy. For $k_e, k_i=2$, we included graphs of Ch^p for $p = 2, 3, 4$ to appreciate the order of the methods. The crossing of the (4,4)-HOP and (3,4)-HOP methods for the DAE (left) are probably due to rounding errors, since the error is about $1e-14$

Table 2 Overview of the examples from Section 8.2.1

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

8.2.2 A challenging index 5 DAE

We consider now the index-5 example from [26] resulting from a model of two pendula, where the Lagrangian multiplier λ_1 of the first one controls the length of the second one:

$$\begin{aligned}
 x_1' &= v_{x_1}, & x_2' &= v_{x_2}, \\
 y_1' &= v_{y_1}, & y_2' &= v_{y_2}, \\
 v_{x_1}' &= -x_1\lambda_1, & v_{x_2}' &= -x_2\lambda_2, \\
 v_{y_1}' &= -y_1\lambda_1 + g, & v_{y_2}' &= -y_2\lambda_2 + g, \\
 0 &= x_1^2 + y_1^2 - L^2, & 0 &= x_2^2 + y_2^2 - (L + c\lambda_1)^2.
 \end{aligned}$$

Note that in this formulation from [26] the positive y axis is pointing downwards. The DAE has index 5 and four degrees of freedom. For the numerical tests, we use the gravity constant $g = 1$, the length of the first pendulum $L = 1$, the parameter $c = 0.1$ and the interval $[0, 80]$. In [26], this example was integrated with constant stepsize $h = 0.05$ and order $k = 7$ as well as $h = 0.025$ and order $k = 8$. For the component x_2 , the two solutions were very close until about $t = 30$ and clearly diverging from there to about $t = 50$ and totally unrelated from $t = 55$ on.

Our implementation leads to a very good results in the sense that two solutions have a small difference up to much larger t . We compare the solutions for $K = 9$ (with HOP method $k_i = k_e = 4$ and order 8) and $K = 8$ (with HOP method $k_i = k_e = 3$ and order 6) for the (numerically) consistent initial value:

$$c_0 = \begin{pmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ v_{x_1} \\ v_{y_1} \\ v_{x_2} \\ v_{y_2} \\ \lambda_1 \\ \lambda_2 \end{pmatrix}_0 = \begin{pmatrix} 1.000000000000000e + 00 \\ -6.346337564282729e - 09 \\ 1.000000000000000e + 00 \\ 3.713317265246974e - 01 \\ 5.183756806486933e - 09 \\ 8.168107595885199e - 01 \\ -9.661740336543358e - 02 \\ 9.641228990309292e - 01 \\ 6.671798106332355e - 01 \\ 8.174254817186853e - 01 \end{pmatrix}.$$

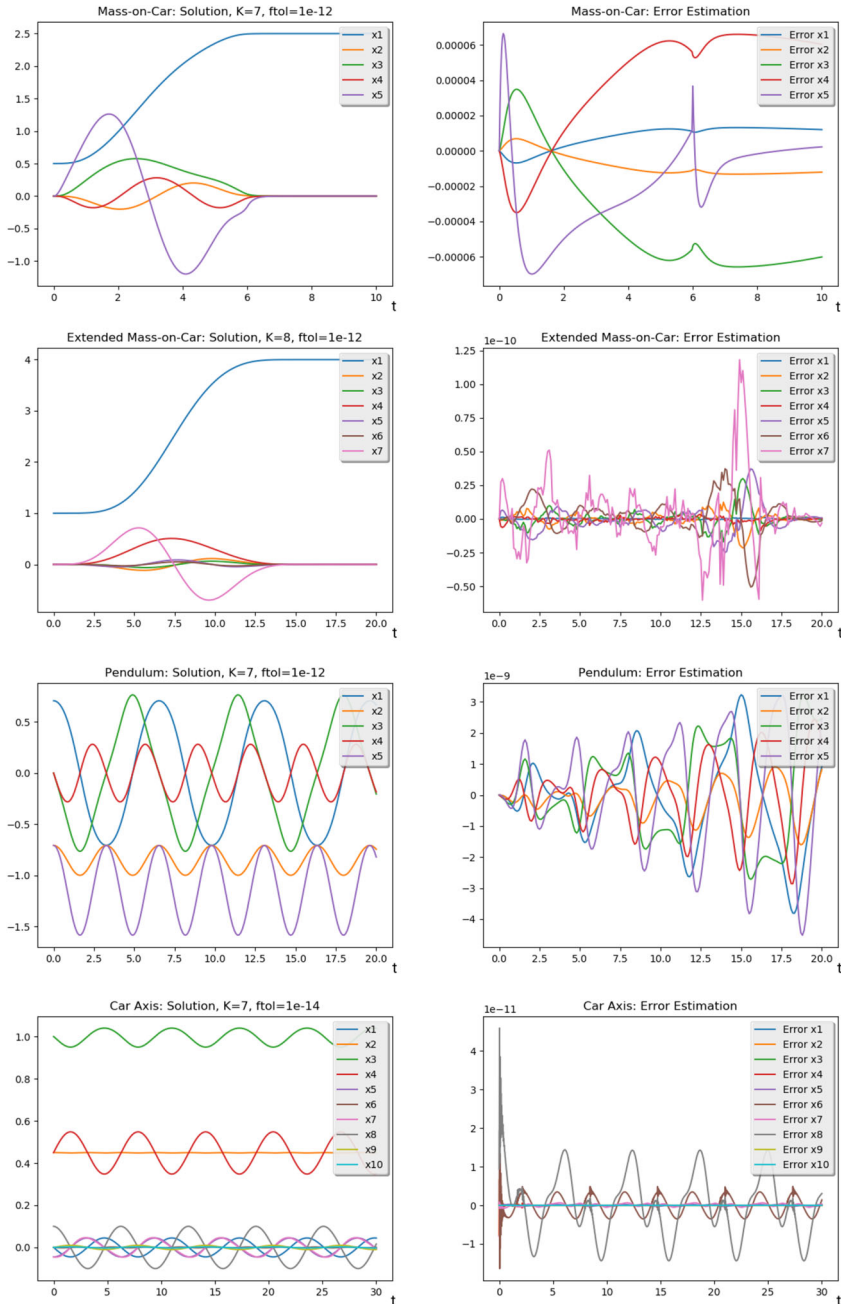


Fig. 2 Numerical solutions of the examples from Section 8.2 obtained by (4,4)-HOP (left) and estimation of the error (right) considering the difference between the solution from (3,3)-HOP and (4,4)-HOP. The specifics of the integration can be found in Table 3

Table 3 Overview of the computations carried out for Fig. 2 with fixed stepsize h

No.	Interval	K	$k_i = k_e$	h	Time (s)	
					(3,3)-HOP	(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

The CPU Time based on a computation with a 2.3-GHz processor

The quality of our results is visualized for x_2 in Fig. 3. For slightly perturbed initial values for x_2 and y_2 , the obtained solutions behave analogously. For arbitrarily perturbed initial values, convergence difficulties during the computation of consistent initial values may appear.

8.2.3 Andrews squeezing mechanism

Finally, we want to report here the behavior we obtained for a well-known index-3 benchmark problem with an extreme scaling. According to [20, 24], the problem is of the form:

$$\begin{pmatrix} q' \\ v' \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} v \\ w \\ M(q)w - f(q, v) + G^T(q)\lambda \\ g(q) \end{pmatrix}$$

for $q \in \mathbb{R}^7, \lambda \in \mathbb{R}^6$. To our surprise, using for α the initial value given in the literature, which leads there to an dynamic behavior, our computation of consistent initial

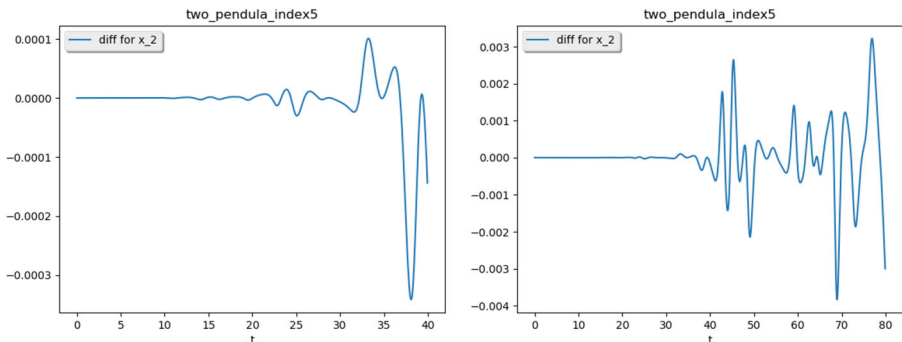


Fig. 3 Difference of the result obtained with (3,3)-HOP with $h = 0.05$ and (4,4)-HOP with $h = 0.025$ for the component x_2 of the index-5 example from Section 8.2.2 in the interval [0,40] (left) and [0,80] (right)

values delivers a stationary solution, such that all our integration methods provide the same constant solution for:

$$q = \begin{pmatrix} \beta \\ \Theta \\ \gamma \\ \Phi \\ \delta \\ \Omega \\ \epsilon \end{pmatrix} = \begin{pmatrix} -1.2456368688861551e - 01 \\ 4.7359420595613766e - 02 \\ 4.5493601541816081e - 01 \\ 2.2197419077471500e - 01 \\ 4.8744266257012064e - 01 \\ -2.2197419077470643e - 01 \\ 1.2302849924800077e + 00 \end{pmatrix}, \tag{41}$$

$$\begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} 9.9283318628552919e + 01 \\ -7.6803614318109092e + 00 \end{pmatrix} \tag{42}$$

and all other components are (numerically) equal to zero. Therefore, we explain here why this happened.

First of all, we want to mention that the indicated condition number, introduced in [14], corresponding to the DAE at α is about 10^{11} , such that a clear hint to the scaling difficulties is given. In contrast, at the given stationary solution, the condition number is about 10^6 .

To simplify further considerations, we notice that the last four equations of $g(p) = 0$ are used to compute $\Phi, \delta, \Omega, \epsilon$, such that we can neglect them and consider only $g_{1,2}$ and $f_{1,2,3}$ to determine $\beta, \Theta, \gamma, \lambda_{1,2}$. For a constant solution $v = w = 0$, the relevant equations are therefore:

$$0 = - \begin{pmatrix} mom \\ 0 \\ f_3(\gamma) \end{pmatrix} + \tilde{G}^T(\beta, \Theta, \gamma) \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix},$$

$$0 = \begin{pmatrix} g_1(\beta, \Theta, \gamma) \\ g_2(\beta, \Theta, \gamma) \end{pmatrix} =: \tilde{g}.$$

At the equilibrium point corresponding to (41)–(42), the constant drive torque mom , the spring force f_3 , and the Lagrangian forces are equalized.

Therefore, it only remains to explain why the approach (19)–(20) delivers a stationary solution. Due to the high condition, the Lagrange multipliers λ are numerically difficult to compute. In fact, for other numerical computations, the accuracy for λ is not controlled [20, p. 536ff]. In contrast, if no (numerical) full row rank of the Jacobian \tilde{G} is given, then (19)–(20) computes a minimal norm solution [15], that in this case minimizes the values for λ , leading to the stationary solution. To our knowledge, this stationary solution was not reported before in the DAE literature.

We plan further investigations on this unexpected behavior. Indeed, for some perturbed initial values, we obtained a solution that converge towards the stationary solution (41)–(42). Moreover, with scaled equations and very different initial values, a nonconstant solution that behaves like the one described in [20, 24] has been obtained.

9 Summary and future work

In this article, we presented a projection approach that permits the extension of explicit/implicit Taylor integrations schemes 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 relatively easy to implement using InitDAE and convenient since, thanks to the formulation as an optimization problem, the inherent dynamics of the DAE are considered indirectly. We analyzed 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 [12] outperform our expectations, in particular with regard to the accuracy for higher-index DAEs, cf. Section 8.2.2. Until now, our focus was on the extension from ODEs to DAEs in order to use higher-order and A-stable methods with InitDAE for our diagnosis purposes during the integration to monitor singularities [17]. With this promising first results, we think that further developments of these projected methods are worthwhile.

In fact, at present, our implementation is not competitive by far. One reason is that setting up the nonlinear equations (27)–(28) and the corresponding Jacobians with AlgoPy, cf. [31], is still very costly. If equations (27)–(28) and the corresponding Jacobians are supplied in a more efficient way, competitive solvers might be achieved. At present, we do not even consider the sparsity of matrices. Furthermore, an improvement seems likely if we take advantage of specific structural properties, e.g., solving subsystems step-by-step, cf. [10, 11]. Another reason for our high computational costs is that the package minimize from SciPy often performs more iterations than we expected (often more than 30), although a good initial guess computed with the explicit Taylor series method is given in general. This behavior has to be inspected in more detail. For linear systems, a direct implementation considering the projector Π from (21) (or, more precisely, a corresponding basis) should deliver an efficient algorithm. This could be of interest, e.g., for the applications from [25, 30]. 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.

Funding Open Access funding enabled and organized by Projekt DEAL

Appendix 1. Stability functions and stability regions of Taylor series methods

A.1 Stability functions

The general definition (27)–(28) allows for a straightforward description of the stability function. Applied to ODEs (and therefore $P = I$), the stability function

$R : \mathbb{C} \rightarrow \mathbb{C}$ results if we consider the test-ODE:

$$y' = \lambda y, \quad y(0) = y_0, \quad \lambda \in \mathbb{C}, \tag{43}$$

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:

$$\sum_{\ell_i=0}^{k_i} \omega_{\ell_i}^i (c_{\ell_i})_{j+1} (-h_j)^{\ell_i} = \sum_{\ell_e=0}^{k_e} \omega_{\ell_e}^e (c_{\ell_e})_j (h_j)^{\ell_e}$$

and, for the test-equation (43), we obtain from:

$$(c_{\ell_i})_{j+1} = \lambda^{\ell_i} (c_0)_j = \frac{\lambda^{\ell_i}}{\ell_i!} y_{j+1} \quad \text{and} \quad (c_{\ell_e})_j = \lambda^{\ell_e} (c_0)_j = \frac{\lambda^{\ell_e}}{\ell_e!} y_j$$

the relationship:

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

i.e., for $z = h_j \lambda \in \mathbb{C}$:

$$R(z) = \frac{\sum_{\ell_e=0}^{k_e} \frac{1}{\ell_e!} \omega_{\ell_e}^e z^{\ell_e}}{\sum_{\ell_i=0}^{k_i} (-1)^{\ell_i} \frac{1}{\ell_i!} \omega_{\ell_i}^i z^{\ell_i}}.$$

A.2 Stability regions

The corresponding stability regions can thus be characterized by:

$$S := \{z \in \mathbb{C} : |R(z)| \leq 1\} = \left\{ z \in \mathbb{C} : \left| \sum_{\ell_e=0}^{k_e} \frac{1}{\ell_e!} \omega_{\ell_e}^e z^{\ell_e} \right| \leq \left| \sum_{\ell_i=0}^{k_i} (-1)^{\ell_i} \frac{1}{\ell_i!} \omega_{\ell_i}^i z^{\ell_i} \right| \right\}.$$

For the methods discussed in this article, we obtain:

- Explicit Taylor:

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

The corresponding stability regions are illustrated in Fig. 4 (top), cf. also [3, 20].

- Fully Implicit Taylor:

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

The corresponding stability regions are illustrated in Fig. 4 (bottom).

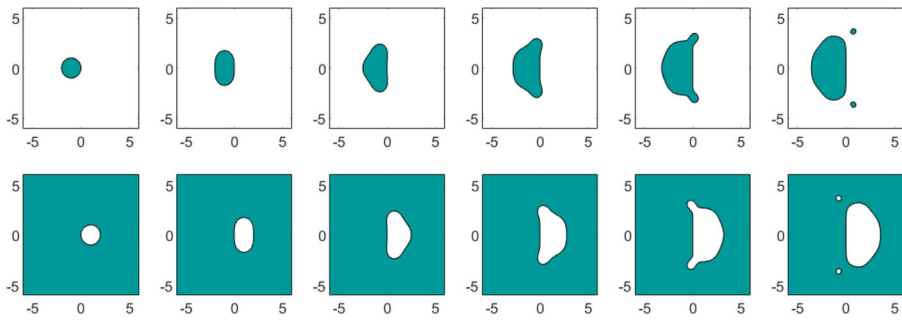


Fig. 4 Colored stability regions S for explicit (top) and fully implicit (bottom) Taylor series methods up to order 6

- For the two-halfstep explicit/implicit schemes (40), we obtain:

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

The corresponding stability regions for $k_e, k_i = 0, \dots, 6$ are represented in Fig. 5.

Note that symmetry is due to:

$$R_{k_e, k_i}^{TH}(-z) = \frac{1}{R_{k_i, k_e}^{TH}(z)}. \tag{44}$$

The schemes with $k_e \leq k_i$ seem to be A -stable or $A(\alpha)$ -stable for moderate α . Indeed, according to [2, 9], 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:

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

- For HOP-methods, from:

$$\begin{aligned} \omega_{\ell_e}^e &:= \frac{k_e!(k_e + k_i - \ell_e)!}{(k_e + k_i)!(k_e - \ell_e)!}, & \ell_e = 0, \dots, k_e \\ \omega_{\ell_i}^i &:= \frac{k_i!(k_e + k_i - \ell_i)!}{(k_e + k_i)!(k_i - \ell_i)!}, & \ell_i = 0, \dots, k_i \end{aligned}$$

we obtain:

$$R_{k_e, k_i}^{HOP}(z) = \frac{k_e! \sum_{\ell_e=0}^{k_e} \frac{1}{\ell_e!} \frac{(k_e+k_i-\ell_e)!}{(k_e-\ell_e)!} z^{\ell_e}}{k_i! \sum_{\ell_i=0}^{k_i} (-1)^{\ell_i} \frac{1}{\ell_i!} \frac{(k_e+k_i-\ell_i)!}{(k_i-\ell_i)!} z^{\ell_i}}. \tag{45}$$

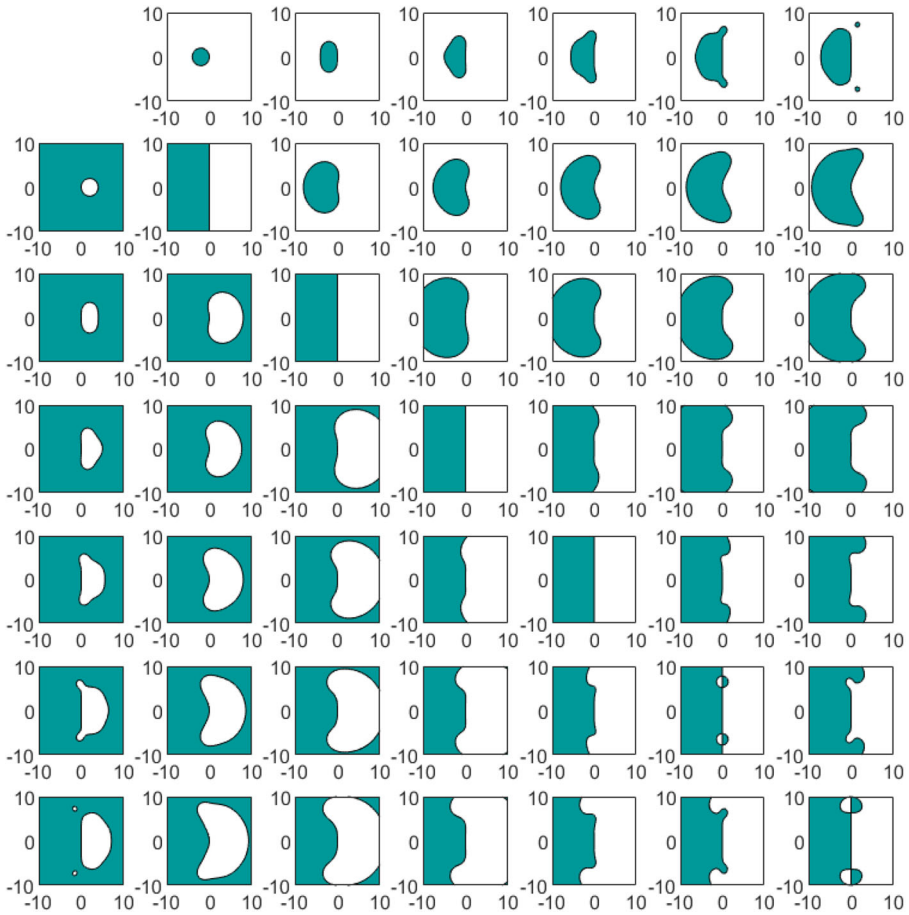


Fig. 5 Colored stability regions S of two-halfstep schemes considering R_{k_e, k_i} for all combinations of $k_e, k_i = 0, \dots, 6$, where k_i corresponds to the rows and k_e to the columns. The symmetry with respect to the main diagonal of the images results from (44). We can realize that, for $k_i = k_e = 5, 6$, they are not A-stable. This contradicts the statement in [21]

Moreover, analogously as before, we have:

$$R_{k_e, k_i}^{\text{HOP}}(-z) = \frac{1}{R_{k_i, k_e}^{\text{HOP}}(z)}. \tag{46}$$

Therefore, in Fig. 6, we obtain again symmetric stability regions that are in accordance with the stability properties reported in Section 5.4.

Since we obtained (44) and (46) for TH and HOP methods with $k = k_e = k_i$, it holds for these methods that:

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

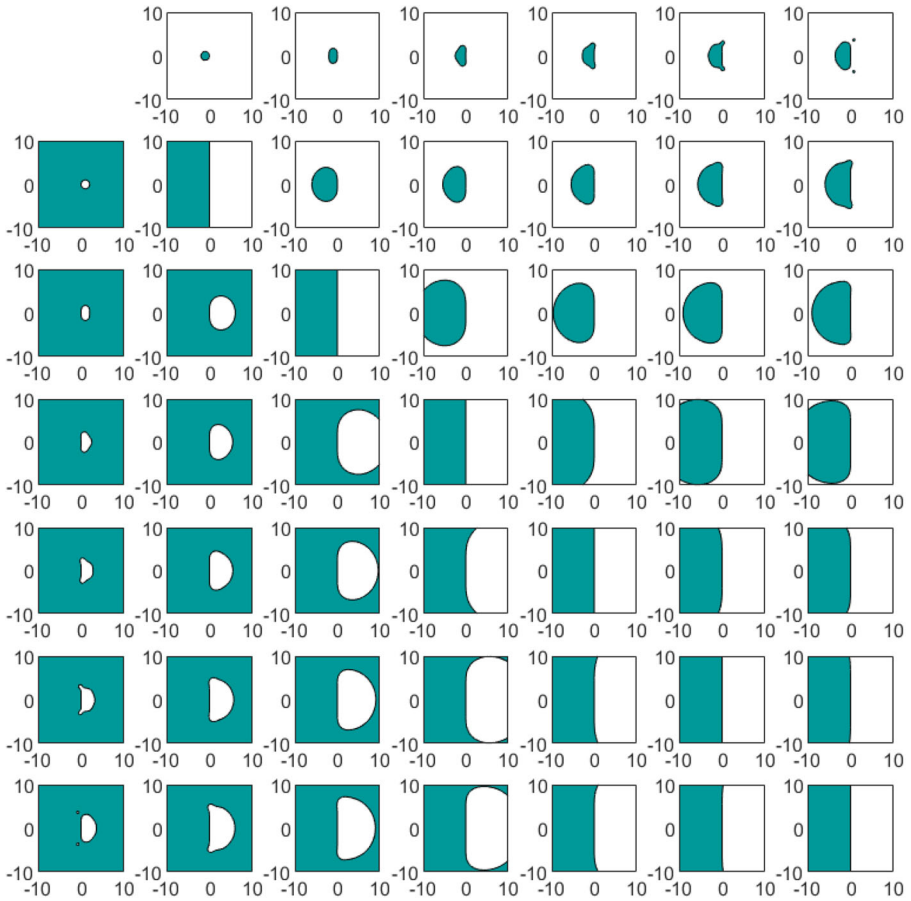


Fig. 6 Colored stability regions S of HOP-methods considering R_{k_e, k_i} for all combinations of $k_e, k_i = 0, \dots, 6$, where k_i corresponds to the rows and k_e to the columns. The symmetry with respect to the main diagonal of the images results from (46). For $k_e = k_i$, we observe $S = \mathbb{C}^-$

According to Lemma 6.20 from [8] and to the representations of the stability regions from Fig. 5, $R_{k,k}^{TH}$ seems to have poles in \mathbb{C}^- in general.

Appendix 2: Linear algebra toolbox for DAEs

According to [6], a linear system of equations:

$$\mathcal{A} \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} = b$$

is 1-full with respect to s_1 , if s_1 is uniquely determined for any consistent b , cf. also [5]. Since we focus on properties of the matrix \mathcal{A} , according to [16], we prefer the following equivalent formulation:

$$\mathcal{A} \text{ is 1-full} \quad :\Leftrightarrow \quad \ker \mathcal{A} \subseteq \left\{ \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} : s_1 = 0 \right\}. \tag{47}$$

For our analysis of DAEs, we focus on 1-full matrices \mathcal{A} of the following particular form:

$$\mathcal{A} = \begin{pmatrix} P & 0 \\ \mathcal{G}_1 & \mathcal{G}_2 \end{pmatrix} \tag{48}$$

for an orthogonal projector P . For DAEs, $\mathcal{G}_1, \mathcal{G}_2$ result from the derivative array and P is the orthogonal projector introduced in Section 2.

Let W_2 denote an arbitrary matrix fulfilling $\ker W_2 = \text{im } \mathcal{G}_2$. Then for $N := W_2 \mathcal{G}_1$, the 1-fullness of (48) implies:

$$\ker \begin{pmatrix} P \\ N \end{pmatrix} = 0.$$

For linear DAEs, N results from the constraints.

Let further W denote an arbitrary matrix fulfilling $\ker W = \text{im } NQ$, such that $WN = WNP$. Then, the orthogonal projector Π onto:

$$\ker \begin{pmatrix} Q \\ WN \end{pmatrix} = \text{im } \Pi$$

fulfills $\Pi := P - (WN)^+(WN)$.

For linear DAEs, this projector Π allows for the formulation of the orthogonally projected explicit ODE (10) in the invariant subspace $\text{im } \Pi$, cf. Theorem 1 in [18]. In practice, we can avoid the explicit computation of the orthogonal projector Π considering the constrained optimization problem (5)–(6) instead of (7)–(8).

Appendix 3: Linear examples from Section 8.2

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

$$Ax' + Bx = q.$$

C.1 Example: mass-on-car

The DAE resulting from the spring-mass system mounted on a car from [30] 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:

$$y_d(t) = \begin{cases} y_0 + p_9\left(\frac{t}{t_{\max}}\right) \cdot (y_f - y_0) & \text{for } 0 \leq t \leq t_{\max} \\ y_f & \text{for } t \geq t_{\max} \end{cases}$$

for $y_0 = 0.5, y_f = 2.5, t_{\max} = 6.0$ and the polynomial p_9 described in Appendix C.3.

For $0 < \alpha < \frac{\pi}{2}$, the DAE-index is 3 and the projector Π from (21) 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.

C.2 Example: extended mass-on-car system

The DAE resulting from the extension of the mass-on car systems described in [25] corresponds to:

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

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

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

$$z_d(t) = \begin{cases} z_0 + p_{15} \left(\frac{t}{t_{\max}} \right) (z_f - z_0) & \text{for } 0 \leq t \leq t_{\max}, \\ z_f & \text{for } t \geq t_{\max}, \end{cases}$$

for $z_0 = 1.0, z_f = 4.0, t_{\max} = 15.0$, and the polynomial p_{15} described in Appendix C.3. 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 .

C.3 About formulas for the prescribed path

In the examples of Sections C.1 and C.2, the prescribed path is described by a polynomial that results to be the incomplete regularized beta function.

For $a > 0, b > 0$, the incomplete beta function is defined by:

$$B_x(a, b) = \int_0^x s^a (1-s)^b dx$$

for $0 < x < 1$ and

$$B_0(a, b) = 0, \quad B_1(a, b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}.$$

Moreover, the incomplete regularized beta function reads:

$$I_x(a, b) = \frac{B_x(a, b)}{B_1(a, b)} = B_x(a, b) \frac{\Gamma(a + b)}{\Gamma(a)\Gamma(b)}$$

and has the obvious properties:

$$I_0(a, b) = 0, \quad I_1(a, b) = 1, \quad I_x(a, b) = 1 - I_{1-x}(a, b).$$

For $n \in \mathbb{N}$, $a = b = n + 1$, the polynomial $p_{2n+1}(x) := I_x(n + 1, n + 1)$ has order $2n + 1$ and can be represented with binomial expansion:

$$\begin{aligned} p_{2n+1}^{(A)}(x) &:= \frac{(2n + 1)!}{n! n!} \int_0^x \sum_{k=0}^n \binom{n}{k} (-1)^k x^{n+k} dx \\ &= \sum_{k=0}^n \left(\frac{(2n + 1)!}{n! n!} \binom{n}{k} \frac{(-1)^k}{n + k + 1} \right) x^{n+k+1}. \end{aligned} \tag{49}$$

Although this formulation has often been used in literature, from a numerical point of view, it is inconvenient due to the alternating sign.

Since the polynomial $p_{2n+1}(x) - \frac{1}{2}$ has a zero at $\frac{1}{2}$, it can be shown that for $h_0 = 1$, the polynomials:

$$h_{2\ell}(x) = \frac{2\ell - 1}{2\ell} 4 \left(x - x^2\right) h_{2\ell-2}(x) \quad \text{and} \quad q_{2n}(x) = \sum_{\ell=0}^n h_{2\ell}(x)$$

allow the representation:

$$p_{2n+1}^{(B)}(x) = \frac{1}{2} + \left(x - \frac{1}{2}\right) q_{2n}(x). \tag{50}$$

Moreover, using the formula for the relation between the incomplete regularized beta function and the binomial expansion for integer a from [1, p. 263], the same polynomial can also be described by:

$$p_{2n+1}^{(C)}(x) = \sum_{k=n+1}^{2n+1} \binom{2n + 1}{k} x^k (1 - x)^{2n+1-k}. \tag{51}$$

Of course, the polynomials described by (49), (50), and (51) are identical, but the formulas from (50) and (51) have the clear advantage that they are completely numerically stable since for $0 < x < 1$ all the terms are positive. The error that is caused by formulation (49) is visualized in Fig. 7.

In particular, for $n = 4$, we obtain:

$$\begin{aligned} p_9^{(A)}(x) &= 126x^5 - 420x^6 + 540x^7 - 315x^8 + 70x^9, \\ p_9^{(B)}(x) &= \frac{1}{2} + \left(x - \frac{1}{2}\right) \cdot \left(1 + 2(x - x^2)^1 + 6(x - x^2)^2 + 20(x - x^2)^3 \right. \\ &\quad \left. + 70(x - x^2)^4 + 252(x - x^2)^5 + 924(x - x^2)^6 + 3432(x - x^2)^7\right), \\ p_9^{(C)}(x) &= 126x^5(1 - x)^4 + 84x^6(1 - x)^3 + 36x^7(1 - x)^2 + 9x^8(1 - x) + x^9. \end{aligned}$$

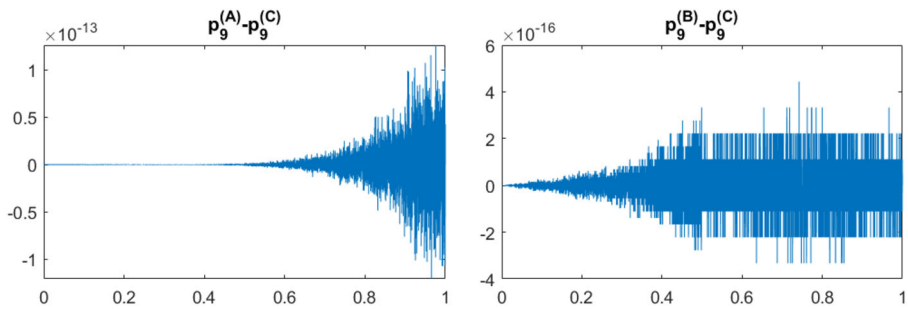


Fig. 7 Difference between the values obtained with the different formulations of p_9 . Obviously, $p_9^{(A)}$ introduces an error that can be avoided using $p_9^{(B)}$ or $p_9^{(C)}$. An analogous comparison of the formulations for $n = 7$ leads to an error of up to $5 \cdot 10^{-11}$ for $p_{15}^{(A)}$

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Abramowitz, M., Stegun, I.A.: Handbook of Mathematical Functions with Formulas Graphs and Mathematical Tables. Dover, New York (1972)
2. Akishin, P.G., Puzynin, I.V., Vinitsky, S.I.: A hybrid numerical method for analysis of dynamics of the classical Hamiltonian systems. *Comput. Math. Appl.* **34**(2-4), 45–73 (1997)
3. Barrio, R.: Performance of the Taylor series method for ODEs/DAEs. *Appl. Math. Comput.* **163**(2), 525–545 (2005)
4. Brenan, K.E., Campbell, S.L., Petzold, L.R.: Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations. Unabridged, corr. republ. Classics in Applied Mathematics, p. 14. SIAM Society for Industrial and Applied Mathematics, Philadelphia (1996)
5. Campbell, S.L.: The numerical solution of higher index linear time varying singular systems of differential equations. *SIAM J. Sci. Stat. Comput.* **6**, 334–348 (1985)
6. Campbell, S.L., Gear, C.W.: The index of general nonlinear DAEs. *Numer. Math.* **72**(2), 173–196 (1995)
7. Corliss, G.F., Griewank, A., Henneberger, P., Kirlinger, G., Potra, F.A., Stetter, H.J.: High-order stiff ODE solvers via automatic differentiation and rational prediction. In: Vulkov, L., Wasniewski, J., Yalamov, P. (eds.) Numerical Analysis and Its Applications. WNAA 1996. Lecture Notes in Computer Science, vol. 1196, pp. 114–124 (1997)
8. Deuffhard, P., Bornemann, F.: Numerical mathematics 2. Ordinary differential equations. (Numerische Mathematik 2 Gewöhnliche Differentialgleichungen.) 4th revised and augmented ed. de Gruyter Studium, Berlin (2013)
9. Dimova, S.N., Hristov, I.G., Hristova, R.D., Puzynin, I.V., Puzynina, T.P., Sharipov, Z.A., Shegunov, N.G., Tukhliev, Z.K.: 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)

10. Estévez Schwarz, D.: A step-by-step approach to compute a consistent initialization for the MNA. *Int. J. Circuit Theory Appl.* **30**(1), 1–6 (2002)
11. Estévez Schwarz, D.: Consistent initialization for DAEs in Hessenberg form. *Numer. Algorithms* **52**(4), 629–648 (2009)
12. Estévez Schwarz, D., Lamour, R.: InitDAE's documentation. Available from: <https://www.mathematik.hu-berlin.de/~lamour/software/python/InitDAE/html/>
13. Estévez Schwarz, D., Lamour, R.: Projector based integration of DAEs with the Taylor series method using automatic differentiation. *J. Comput. Appl. Math.* **262**, 62–72 (2014)
14. Estévez Schwarz, D., Lamour, R.: A new projector based decoupling of linear DAEs for monitoring singularities. *Numer. Algorithms* **73**(2), 535–565 (2016)
15. Estévez Schwarz, D., Lamour, R.: Consistent initialization for higher-index DAEs using a projector based minimum-norm specification. Technical Report 1. Institut für Mathematik, Humboldt-Universität zu Berlin (2016)
16. Estévez Schwarz, D., Lamour, R.: 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)
17. Estévez Schwarz, D., Lamour, R.: InitDAE: Computation of consistent values, index determination and diagnosis of singularities of DAEs using automatic differentiation in Python. *Journal of Computational and Applied Mathematics*. <https://doi.org/10.1016/j.cam.2019.112486> (2019)
18. Estévez Schwarz, D., Lamour, R.: A projector based decoupling of DAEs obtained from the derivative array. In: *Progress in Differential-Algebraic Equations II, Differential-Algebraic Equations Forum (DAE-F)* (2020)
19. Estévez Schwarz, D., Lamour, R., März, R.: Singularities of the Robotic Arm DAE. *Progress in Differential-Algebraic Equations II, Differential-Algebraic Equations Forum (DAE-F)* (2020)
20. Hairer, E., Wanner, G.: *Solving Ordinary Differential Equations II*. Springer (1996)
21. Kirlinger, G., Corliss, G.F.: 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, pp. 371–379, Amsterdam (1992)
22. Kunkel, P., Mehrmann, V.: *Differential-Algebraic Equations - Analysis and Numerical Solution*. EMS Publishing House, Zürich (2006)
23. Lamour, R., März, R., Tischendorf, C.: *Differential-Algebraic Equations: A Projector Based Analysis*. *Differential-Algebraic Equations Forum*, vol. 1. Springer, Berlin (2013)
24. Mazzia, F., Magherini, C.: 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. Available from: <http://pitagora.dm.uniba.it/~testset>
25. Otto, S., Seifried, R.: 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)
26. Pryce, J.D.: Solving high-index DAEs by Taylor series. *Numer. Algorithms* **19**(1–4), 195–211 (1998)
27. Pryce, J.D., Nedialkov, N.S., Tan, G., Li, X.: How AD can help solve differential-algebraic equations. *Optim. Methods Softw.* **33**(4–6), 729–749 (2018)
28. Riaza, R.: *Differential-Algebraic Systems. Analytical Aspects and Circuit Applications*. World Scientific, Hackensack (2008)
29. Scott, J.R., Solving, O.DE.: Initial Value Problems with Implicit Taylor Series Methods. Technical report NASA/TM-2000-209400 (2000)
30. Seifried, R., Blajer, W.: Analysis of servo-constraint problems for underactuated multibody systems. *Mech. Sci.* **4**, 113–129 (2013)
31. Walter, S.F., Lehmann, L.: Algorithmic differentiation in Python with AlgoPy. *J. Comput. Sci.* **4**(5), 334–344 (2013)

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.