



Towards a reliable implementation of least-squares collocation for higher index differential-algebraic equations—Part 1: basics and ansatz function choices

Michael Hanke¹ · Roswitha März²

Received: 25 November 2020 / Accepted: 16 May 2021 / Published online: 15 June 2021
© The Author(s) 2021

Abstract

In the two parts of the present note we discuss several questions concerning the implementation of overdetermined least-squares collocation methods for higher index differential-algebraic equations (DAEs). Since higher index DAEs lead to ill-posed problems in natural settings, the discrete counterparts are expected to be very sensitive, which attaches particular importance to their implementation. In the present Part 1, we provide a robust selection of basis functions and collocation points to design the discrete problem. We substantiate a procedure for its numerical solution later in Part 2. Additionally, in Part 1, a number of new error estimates are proven that support some of the design decisions.

Keywords Least-squares collocation · Higher index differential-algebraic equations · Ill-posed problem

Mathematics Subject Classification (2010) 65L80 · 65L08 · 65F20 · 34A99

1 Introduction

An overdetermined least-squares collocation method for the solution of boundary value problems for higher index differential-algebraic equations (DAEs) has been introduced in [28] and further investigated in [25–27]. A couple of sufficient convergence conditions have been established. Numerical experiments indicate an excellent

✉ Michael Hanke
hanke@nada.kth.se

Roswitha März
maerz@mathematik.hu-berlin.de

¹ Department of Mathematics, KTH Royal Institute of Technology, S-10044 Stockholm, Sweden

² Institute of Mathematics, Humboldt University of Berlin, D-10099 Berlin, Germany

behavior. Moreover, it is particularly noteworthy that the computational effort is not much more expensive than for standard collocation methods applied to boundary value problems for ordinary differential equations. However, the particular procedures are much more sensitive which reflects the ill-posedness of higher index DAEs. The question of a reliable implementation is almost completely open. The method offers a number of parameters and options whose selection has not been backed up by any theoretical justifications. Both parts of the present paper are devoted to a first investigation of this topic. We focus on the choice of collocation nodes, the representation of the ansatz function as well as the shape and structure of the resulting discrete problem. We apply various theoretical arguments, among them also new sufficient convergence conditions in Theorems 1, 2, and 3, and report on corresponding systematic comprehensive numerical experiments.

Considering that so far for the practical simulation of general unstructured DAEs with higher index only such methods are available which are connected with the construction, analysis, and evaluation of high-dimensional prolonged systems (derivative array systems), a reliable direct method, as we aim at, would be a significant progress. This is confirmed by a first comparison with the classical derivative array-based method from [6] in [27, subsection 6.4] and [25, subsection 5.2]. A discussion of some earlier direct approaches can be found in [28, Section 5].

The present Part 1 of the note is organized as follows: Section 2 contains the information concerning the problem to be solved as well as the basics on the overdetermined least-squares approach, and, additionally, the new error estimates. Section 3 deals with the selection and calculation of collocation points and integration weights for the different functionals of interest and Section 4 provides a robust selection of bases of the ansatz space. It should also be mentioned at this point that the resulting discrete least-squares problem is treated in detail in Part 2 [24]. We conclude with Section 5, which contains a summary and further comments.

The algorithms have been implemented in C++11. All computations have been performed on a laptop running OpenSUSE Linux, release Leap 15.1, the GNU g++ compiler (version 7.5.0) [40], the Eigen matrix library (version 3.3.7) [22], SuiteSparse (version 5.6.0) [7], in particular its sparse QR factorization [8], Intel® MKL (version 2019.5-281), all in double precision with a rounding unit of $\varepsilon_{\text{mach}} \approx 2.22 \times 10^{-16}$.¹ The code is optimized using the level -O3.²

2 Fundamentals of the problem and method

Consider a linear boundary value problem for a DAE with properly involved derivative,

$$A(t)(Dx)'(t) + B(t)x(t) = q(t), \quad t \in [a, b], \quad (1)$$

$$G_a x(a) + G_b x(b) = d. \quad (2)$$

¹Intel is a registered trademark of Intel Corporation.

²The interested reader can get access to the code by writing to the corresponding author. However, it should be noted that the code has been written with the aim of a thorough testing of the ingredients of the proposed methods. So it does not yet have production quality.

with $[a, b] \subset \mathbb{R}$ being a compact interval, $D = [I \ 0] \in \mathbb{R}^{k \times m}$, $k < m$, with the identity matrix $I \in \mathbb{R}^{k \times k}$. Furthermore, $A(t) \in \mathbb{R}^{m \times k}$, $B(t) \in \mathbb{R}^{m \times m}$, and $q(t) \in \mathbb{R}^m$ are assumed to be sufficiently smooth with respect to $t \in [a, b]$. Moreover, $G_a, G_b \in \mathbb{R}^{l \times m}$ and $\ker G_a \supseteq \ker D$, $\ker G_b \supseteq \ker D$. Thereby, l is the dynamical degree of freedom of the DAE, that is, the number of free parameters which can be fixed by initial and boundary conditions.

Unlike regular ordinary differential equations (ODEs) where $l = k = m$, for DAEs it holds that $0 \leq l \leq k < m$, in particular, $l = k$ for index-one DAEs, $l < k$ for higher index DAEs, and $l = 0$ can certainly happen.

Supposing accurately stated initial and boundary conditions, index-one DAEs yield well-posed problems in natural settings and can be numerically treated quite well similarly as ODEs [34]. In contrast, in the present paper, we are mainly interested in higher index DAEs which lead to essentially ill-posed problems even if the boundary conditions are stated accurately [27, 33, 34]. The tractability index and projector-based analysis serve as the basis for our investigations. We refer to [33] for a detailed presentation and to [27, 34, 36] for corresponding short sketches.

We assume that the DAE is regular with arbitrarily high index $\mu \in \mathbb{N}$ and the boundary conditions are stated accurately so that solutions of the problem (1)–(2) are unique. We also assume that a solution $x_* : [a, b] \rightarrow \mathbb{R}^m$ actually exists and is sufficiently smooth.

For the construction of a regularization method to treat an essentially ill-posed problem a Hilbert space setting of the problem is most convenient. For this reason, as in [26–28], we apply the spaces

$$H_D^1 := H_D^1((a, b), \mathbb{R}^m) = \{x \in L^2((a, b), \mathbb{R}^m) : Dx \in H^1((a, b), \mathbb{R}^k)\},$$

$$L^2 := L^2((a, b), \mathbb{R}^m),$$

which are suitable for describing the underlying operators. In particular, let $\mathcal{T} : H_D^1 \rightarrow L^2 \times \mathbb{R}^l$ be given by

$$(\mathcal{T}x)(t) = \begin{bmatrix} A(t)(Dx)'(t) + B(t)x(t) \\ G_ax(a) + G_bx(b) \end{bmatrix}. \tag{3}$$

Then the boundary value problem can be described by $\mathcal{T}x = (q, d)^T$.

For $K \geq 0$, let \mathfrak{P}_K denote the set of all polynomials of degree less than or equal to K . Next, we define a finite dimensional subspace $X_\pi \subset H_D^1$ of piecewise polynomial functions which should serve as ansatz space for the least-squares approximation: Let the partition π be given by

$$\pi : \quad a = t_0 < t_1 < \dots < t_n = b, \tag{4}$$

with the stepsizes $h_j = t_j - t_{j-1}$, $h = \max_{1 \leq j \leq n} h_j$, and $h_{min} = \min_{1 \leq j \leq n} h_j$.

Let $C_\pi([a, b], \mathbb{R}^m)$ denote the space of piecewise continuous functions having breakpoints merely at the meshpoints of the partition π . Let $N \geq 1$ be a fixed integer. Then, we define

$$X_\pi = \{x \in C_\pi([a, b], \mathbb{R}^m) : Dx \in C([a, b], \mathbb{R}^k),$$

$$x_\kappa|_{[t_{j-1}, t_j]} \in \mathfrak{P}_N, \kappa = 1, \dots, k, \quad x_\kappa|_{[t_{j-1}, t_j]} \in \mathfrak{P}_{N-1},$$

$$\kappa = k + 1, \dots, m, \quad j = 1, \dots, n\}. \tag{5}$$

The continuous version of the least-squares method reads: Find an $x_\pi \in X_\pi$ that minimizes the functional

$$\Phi(x) = \|\mathcal{T}x\|^2 = \int_a^b |A(t)(Dx)'(t) + B(t)x(t) - q(t)|^2 dt + |G_a x(a) + G_b x(b) - d|^2. \tag{6}$$

It is ensured by [27, Theorem 4.1] that, for all sufficiently fine partitions π with bounded ratios $1 \leq h/h_{min} \leq \rho$, ρ being a global constant, there exists a unique solution $x_\pi \in X_\pi$ and the inequality

$$\|x_\pi - x_*\|_{H_D^1} \leq Ch^{N-\mu+1} \tag{7}$$

is valid. The constant $C \in \mathbb{R}$ depends on the solution x_* , the degree N , and the index μ , but it is independent of h . If $N > \mu - 1$ then (7) apparently indicates convergence $x_\pi \xrightarrow{h \rightarrow 0} x_*$ in H_D^1 .

At this place it is important to mention that, so far, we are aware of only sufficient conditions of convergence and the error estimates may not be sharp. Not only more practical questions of implementation are open, but also several questions about the theoretical background. We are optimistic that much better estimates are possible since the results of the numerical experiments have performed impressively better than theoretically expected till now.

The following theorem can be understood as a specification of [27, Theorem 4.1] by a more detailed description of the ingredients of the constant C , in particular, now the role of N is better clarified, which could well be important for the practical realization. In particular, it suggests that smooth problems could perhaps be solved better with large N and coarser partitions.

Theorem 1 *Let the DAE (1) be regular with index $\mu \in \mathbb{N}$ and let the boundary condition (2) be accurately stated. Let x_* be a solution of the boundary value problem (1)–(2), and let A, B, q and also x_* be sufficiently smooth.*

Let $N \geq 1$ and all partitions π be such that $h/h_{min} \leq \rho$, with a global constant ρ . Then, for all such partitions with sufficiently small h , the estimate (7) is valid with

$$C = \frac{N!}{(2N)! \sqrt{2N+1}} C_N C_* \rho^{\mu-1} C_{data},$$

where

$$C_* = \max\{\|x_*^{(N)}\|_\infty, \|x_*^{(N+1)}\|_\infty\} (m + 4k(b-a)^3)^{1/2},$$

C_{data} is independent of N and h , it depends only on the data A, D, B, G_a, G_b ,

and C_N is a rather involved function of N . In particular, there is an integer K with $N \leq K \leq 2(\mu - 1) + N$ such that, for $N \rightarrow \infty$, C_N does not grow faster than $K^{2(\mu-1)}$. If A and B are constant, it holds $K = N$.

At this place it should be mentioned that the estimate [38]

$$\sqrt{2\pi N} \left(\frac{N}{e}\right)^N e^{1/(12N+1)} \leq N! \leq \sqrt{2\pi N} \left(\frac{N}{e}\right)^N e^{1/(12N)},$$

or its slightly less sharp version,

$$\sqrt{2\pi N} \left(\frac{N}{e}\right)^N \leq N! \leq \sqrt{2\pi N} \left(\frac{N}{e}\right)^N e^{1/12}$$

allow the growth estimate $\frac{N!}{(2N)!} \leq \sqrt{\pi N} e^{1/6} \frac{1}{N!} \frac{1}{4^N}$, thus

$$\begin{aligned} C &\leq \sqrt{\pi N} e^{1/6} \frac{1}{N!} \frac{1}{4^N \sqrt{2N+1}} C_N C_* \rho^{\mu-1} C_{data} \leq \sqrt{\frac{\pi}{2}} e^{1/6} \frac{1}{N!} \frac{1}{4^N} C_N C_* \rho^{\mu-1} C_{data} \\ &\leq \frac{1}{N!} \frac{1}{4^N} C_N C_* \rho^{\mu-1} \sqrt{\frac{\pi}{2}} e^{1/6} C_{data}. \end{aligned} \tag{8}$$

However, it should be considered that C_N and C_* also depend on N .

Proof We apply the estimate [27]

$$\|x_\pi - x_*\|_{H_D^1} \leq \frac{\|\mathcal{T}\alpha_\pi\|}{\gamma_\pi} + \alpha_\pi,$$

in which the approximation error α_π and the instability threshold γ_π are given by

$$\alpha_\pi = \inf_{x \in X_\pi} \|x - x_*\|_{H_D^1}, \quad \gamma_\pi = \inf_{x \in X_\pi, x \neq 0} \frac{\|\mathcal{T}x\|}{\|x\|_{H_D^1}}.$$

Owing to [27, Theorem 4.1], there is a constant $c_\gamma > 0$ independent of π so that the instability threshold γ_π satisfies the inequality

$$c_\gamma h_{min}^{\mu-1} \leq \gamma_\pi \leq \|\mathcal{T}\|,$$

for all partitions with sufficiently small h . This leads to

$$\|x_\pi - x_*\|_{H_D^1} \leq \frac{\alpha_\pi}{\gamma_\pi} (\|\mathcal{T}\| + \gamma_\pi) \leq 2 \frac{\alpha_\pi}{\gamma_\pi} \|\mathcal{T}\|.$$

Choosing N interpolation points ρ_i with

$$\begin{aligned} 0 &< \rho_1 < \dots < \rho_N < 1, \\ \tilde{\omega}(\rho) &= (\rho - \rho_1) \dots (\rho - \rho_N), \end{aligned} \tag{9}$$

the approximation error can be estimated by straightforward but elaborate computations by constructing $p_* \in X_\pi$ such that $p'_{*,s}(t_{j-1} + \rho_i h_j) = x'_{*,s}(t_{j-1} + \rho_i h_j)$, $p_{*,s}(a) = x_{*,s}(a)$, $s = 1, \dots, k$, $p_{*,s}(t_{j-1} + \rho_i h_j) = x_{*,s}(t_{j-1} + \rho_i h_j)$, $s = k + 1, \dots, m$, $i = 1, \dots, N$, $j = 1, \dots, n$, and regarding $\alpha_\pi \leq \|p_* - x_*\|_{H_D^1}$. One obtains

$$\begin{aligned} \alpha_\pi &\leq \frac{h^N}{N!} \|\tilde{\omega}\|_{L^2(0,1)} C_*, \\ C_* &= \max \left\{ \|x_*^{(N)}\|_\infty, \|x_*^{(N+1)}\|_\infty \right\} \left(m + 4k(b-a)^3 \right)^{1/2}. \end{aligned} \tag{10}$$

Turning to shifted Gauss-Legendre nodes that minimize $\|\tilde{\omega}\|_{L^2(0,1)}$ we obtain

$$\|\tilde{\omega}\|_{L^2(0,1)} = \frac{(N!)^2}{(2N)! \sqrt{2N+1}}.$$

To verify this, we consider the polynomial

$$\omega(t) = 2^N \tilde{\omega} \left(\frac{t+1}{2} \right) = (t - t_1) \cdots (t - t_N)$$

with zeros $t_j = 2\rho_j - 1, j = 1, \dots, N$, which is nothing else but the standard Legendre polynomial with leading coefficient one. Using the Rodrigues formula and other arguments from [23, Section 5.4], one obtains

$$\|\omega\|_{L^2(-1,1)} = 2^{N+\frac{1}{2}} \frac{(N!)^2}{(2N)!\sqrt{2N+1}}.$$

Finally, shifting back to the interval $(0, 1)$ leads to $\|\tilde{\omega}\|_{L^2(0,1)} = 2^{-(N+\frac{1}{2})} \|\omega\|_{L^2(-1,1)}$.

Thus we have

$$\alpha_\pi \leq \frac{h^N}{N!} \frac{(N!)^2}{(2N)!\sqrt{2N+1}} C_* = h^N \frac{N!}{(2N)!\sqrt{2N+1}} C_*. \tag{11}$$

Next, a careful review of the proof of [27, Theorem 4.1 (a)] results in the representation (in terms of [27])

$$\begin{aligned} \frac{1}{c_Y} &= 12c_Y \sqrt{g_{\mu-1}} = 12c_Y \sqrt{d_{1,\mu-1} c_{\mu-1}^* \|D\mathcal{L}_{\mu-1}\|_\infty^2} \\ &= 12c_Y \sqrt{2} \|D\Pi_0 Q_1 \cdots Q_{\mu-1} D^+\|_\infty \|D\mathcal{L}_{\mu-1}\|_\infty \sqrt{c_{\mu-1}^*}. \end{aligned}$$

The factors $\|D\Pi_0 Q_1 \cdots Q_{\mu-1} D^+\|_\infty$ and $\|D\mathcal{L}_{\mu-1}\|_\infty$ depend only on the data A, D, B , likewise the bound c_Y introduced in [27, Proposition 4.3].

In contrast, the term $c_{\mu-1}^*$ depends additionally on N besides the problem data. Let K denote the degree of the auxiliary polynomial $q_{\mu-1} = \mathfrak{A}_{\mu-1}(Dp)' + \mathfrak{B}_{\mu-1}p, p \in X_\pi$ in the proof of [27, Theorem 4.1]. Then we have $N \leq K \leq N + 2(\mu - 1)$ and, by [27, Lemma 4.2], $c_{\mu-1}^* = 4^{\mu-1} \lambda_K \cdots \lambda_{K-\mu+2}$, where each $\lambda_S > 0$ is the maximal eigenvalue of a certain symmetric, positive semidefinite matrix of size $(S + 1) \times (S + 1)$ [28, Lemma 3.3].

Owing to [28, Corollary A.3] it holds that $\lambda_S \leq \frac{4}{\pi^2} S^4 + O(S^2)$ for large S , and therefore

$$\begin{aligned} c_{\mu-1}^* &= 4^{\mu-1} \lambda_K \cdots \lambda_{K-\mu+2} \\ &\leq 4^{\mu-1} \left(\frac{4}{\pi^2} \right)^{\mu-1} K^4 (K - 1)^4 \cdots (K - \mu + 2)^4 + O\left(K^{4(\mu-1)-1}\right) \\ &= 4^{\mu-1} \left(\frac{4}{\pi^2} \right)^{\mu-1} K^{4(\mu-1)} + O\left(K^{4(\mu-1)-1}\right) \\ &\leq 4^{\mu-1} \left(\frac{4}{\pi^2} \right)^{\mu-1} (N + 2(\mu - 1))^{4(\mu-1)} + O\left((N + 2(\mu - 1))^{4(\mu-1)-1}\right). \end{aligned}$$

Finally, letting

$$C_{data} = 2\|\mathcal{T}\| 12c_Y \sqrt{2} \|D\Pi_0 Q_1 \cdots Q_{\mu-1} D^+\|_\infty \|D\mathcal{L}_{\mu-1}\|_\infty, \quad C_N = \sqrt{c_{\mu-1}^*},$$

we are done. □

Observe that, for smooth problems, any fixed sufficiently fine partition π , and $N \rightarrow \infty$, the growth rate of the error $\|x_\pi - x_*\|_{H^1_D}$ is not greater than that of

$$C_* h^N \frac{(N + 2(\mu - 1))^{2(\mu-1)}}{4^N N!} = C_* \left(\frac{h}{4}\right)^N \frac{(N + 2(\mu - 1))^{2(\mu-1)}}{N!} \tag{12}$$

and, for constant matrix function A and B ,

$$C_* h^N \frac{N^{2(\mu-1)}}{4^N N!} = C_* \left(\frac{h}{4}\right)^N \frac{N^{2(\mu-1)}}{N!}. \tag{13}$$

Remember that C_* is a function of N .

Remark 1 The specific error estimation provided in [28] for the case of DAEs in Jordan chain form on equidistant grids may provide some further insight into the behavior of the instability threshold γ_π . It is shown that

$$\gamma_\pi \geq \bar{C}_\mu \left(\frac{h}{\sqrt{\lambda_N}}\right)^{\mu-1}$$

holds true for sufficiently small h where \bar{C}_μ is a moderate constant depending only on μ [28, Theorem 3.6]. This leads to the dominant error term

$$\frac{\alpha_\pi}{\gamma_\pi} \leq \frac{C_*}{\bar{C}_\mu} \sqrt{\frac{\pi}{2}} e^{1/6} \frac{1}{2^{2N}} \frac{\lambda_N^{\frac{\mu-1}{2}}}{N!} h^{N-\mu+1} = \frac{1}{\bar{C}_\mu} \sqrt{\frac{\pi}{2}} e^{1/6} \frac{1}{h^{\mu-1}} C_* \left(\frac{h}{4}\right)^N \frac{\lambda_N^{\frac{\mu-1}{2}}}{N!},$$

indicating again that, for smooth problems, it seems reasonable to calculate with larger N and coarse partitions. Moreover, for sufficiently small $\frac{h}{\sqrt{\lambda_N}}$, the estimation $\lambda_N \leq \frac{4}{\pi^2} N^4 + O(N^2)$ becomes valid [28, Remark 3.4], and hence the growth characteristic (13) for large N is confirmed once more.

The functional values $\Phi(x)$, which are needed when minimizing for $x \in X_\pi$, cannot be evaluated exactly and the integral must be discretized accordingly. Taking into account that the boundary value problem is ill-posed in the higher index case $\mu > 1$, perturbations of the functional may have a serious influence on the error of the approximate least-squares solution or even prevent convergence towards the solution x_* . Therefore, careful approximations of the integral in Φ are required. We discuss the following three options:

$$\begin{aligned} \Phi_{\pi,M}^C(x) &= \sum_{j=1}^n \frac{h_j}{M} \sum_{i=1}^M |A(t_{ji})(Dx)'(t_{ji}) + B(t_{ji})x(t_{ji}) - q(t_{ji})|^2 \\ &\quad + |G_a x(a) + G_b x(b) - d|^2, \end{aligned} \tag{14}$$

$$\begin{aligned} \Phi_{\pi,M}^I(x) &= \sum_{j=1}^n h_j \sum_{i=1}^M \gamma_i |A(t_{ji})(Dx)'(t_{ji}) + B(t_{ji})x(t_{ji}) - q(t_{ji})|^2 \\ &\quad + |G_a x(a) + G_b x(b) - d|^2, \end{aligned} \tag{15}$$

and

$$\Phi_{\pi,M}^R(x) = \sum_{j=1}^n \int_{t_{j-1}}^{t_j} \left| \sum_{i=1}^M l_{ji}(t)(A(t_{ji})(Dx)'(t_{ji}) + B(t_{ji})x(t_{ji}) - q(t_{ji})) \right|^2 dt + |G_a x(a) + G_b x(b) - d|^2, \tag{16}$$

in which from the DAE (1) and $x \in X_\pi$ only data at the points

$$t_{ji} = t_{j-1} + \tau_i h_j, \quad i = 1, \dots, M, \quad j = 1, \dots, n,$$

are included, with

$$0 \leq \tau_1 < \dots < \tau_M \leq 1. \tag{17}$$

In the last functional $\Phi_{\pi,M}^R$ Lagrange basis polynomials appear, i.e.,

$$l_{ji}(t) = \frac{\prod_{\substack{\kappa=1 \\ \kappa \neq i}}^M (t - t_{j\kappa})}{\prod_{\substack{\kappa=1 \\ \kappa \neq i}}^M (t_{ji} - t_{j\kappa})} = \frac{\prod_{\substack{\kappa=1 \\ \kappa \neq i}}^M (\tau - \tau_\kappa)}{\prod_{\substack{\kappa=1 \\ \kappa \neq i}}^M (\tau_i - \tau_\kappa)} =: l_i(\tau), \quad \tau = (t - t_{j-1})/h_j. \tag{18}$$

Remark 2 The direct numerical implementation of $\Phi_{\pi,M}^R(x)$ with the Lagrangian basis functions includes the use of the mass matrix belonging to such functions. It is well known that this matrix may be very bad conditioned thus leading to an amplification of rounding errors. In connection with the ill-posedness of higher index DAEs, this may render the numerical solutions useless. The solution of the least-squares problem with $\Phi_{\pi,M}^I$ is much less expensive than that with $\Phi_{\pi,M}^R$, and in turn, solving system (21)–(22) below for $x \in X_\pi$ in a least-squares sense using the (diagonally weighted) Euclidean norm in \mathbb{R}^{nMm+1} according to $\Phi_{\pi,M}^C$ is even less computationally expensive than using $\Phi_{\pi,M}^I(x)$.

Introducing, for each $x \in X_\pi$ and $w(t) = A(t)(Dx)'(t) + B(t)x(t) - q(t)$, the corresponding vector $W \in \mathbb{R}^{mMn}$ by

$$W = \begin{bmatrix} W_1 \\ \vdots \\ W_n \end{bmatrix} \in \mathbb{R}^{mMn}, \quad W_j = h_j^{1/2} \begin{bmatrix} w(t_{j1}) \\ \vdots \\ w(t_{jM}) \end{bmatrix} \in \mathbb{R}^{mM}, \tag{19}$$

we obtain new representations of these functionals, namely

$$\Phi_{\pi,M}^C(x) = W^T \mathcal{L}^C W + |G_a x(a) + G_b x(b) - d|^2,$$

$$\Phi_{\pi,M}^I(x) = W^T \mathcal{L}^I W + |G_a x(a) + G_b x(b) - d|^2,$$

and

$$\Phi_{\pi,M}^R(x) = W^T \mathcal{L}^R W + |G_a x(a) + G_b x(b) - d|^2,$$

whereby the first two formulae are evident, with $\mathcal{L}^C = \text{diag}(L^C \otimes I_m, \dots, L^C \otimes I_m)$, \otimes denoting the Kronecker product, and $L^C = M^{-1}I_M$ such that finally $\mathcal{L}^C = M^{-1}I_{nMm}$, and further, $\mathcal{L}^I = \text{diag}(L^I \otimes I_m, \dots, L^I \otimes I_m)$ and $L^I = \text{diag}(\gamma_1, \dots, \gamma_M)$. L^C and thus \mathcal{L}^C are positive definite. The matrices L^I and \mathcal{L}^I are positive definite if and only if all quadrature weights are positive.

The formula for $\Phi_{\pi,M}^R(x)$ can be established by straightforward evaluations following the lines³ of [28, Section 2.3], in which $\mathcal{L}^R = \text{diag}(L^R \otimes I_m, \dots, L^R \otimes I_m)$, L^R is the mass matrix associated with the Lagrange basis functions $l_i, i = 1, \dots, M$, (18) for the node sequence (17), more precisely,

$$L^R = \left(L_{ik}^R \right)_{i,k=1,\dots,M}, \quad L_{ik}^R = \int_0^1 l_i(\tau)l_k(\tau)d\tau. \tag{20}$$

L^R is symmetric and positive definite and, consequently, \mathcal{L}^R is so.

We emphasize that the matrices L^C, L^I, L^R depend only on M , the node sequence (17), and the quadrature weights, but do not depend on the partition π and h at all.

We set always $M \geq N + 1$. Although the nodes (17) serve as interpolation points in the functional $\Phi_{\pi,M}^R$, we still call them *collocation nodes* after [28]. It should be underlined here that minimizing each of the above functionals on X_π can be viewed as a special least-squares method to solve the *overdetermined collocation system* $W = 0, G_ax(a) + G_bx(b) = d$, with respect to $x \in X_\pi$, that is in detail, the collocation system

$$A(t_{ji})(Dx)'(t_{ji}) + B(t_{ji})x(t_{ji}) = q(t_{ji}), \quad i = 1, \dots, M, \quad j = 1, \dots, n, \tag{21}$$

$$G_ax(a) + G_bx(b) = d. \tag{22}$$

The system (21)–(22) for $x \in X_\pi$ becomes overdetermined since X_π has dimension $mnN + k$, whereas the system consists of $mnM + l \geq mnN + mn + l \geq nmN + m + l > nmN + k + l \geq mnN + k$ scalar equations.⁴

Remark 3 Based on collocation methods for index-1 DAEs, the first thought in [27, 28] was to turn to the functional $\Phi_{\pi,M}^C$ with nodes $0 < \tau_1 < \dots < \tau_M < 1$. However, the use of the special discretized norm in these papers for providing convergence results is in essence already the use of the functional $\Phi_{\pi,M}^R$.

For a general set of nodes (17), $\Phi_{\pi,M}^C$ represents a simple Riemann approximation of the corresponding integral which has first order of accuracy, only. If, however, the nodes are chosen as those of the Chebyshev integration, the orders 1, . . . , 7 and 9 can be obtained for the corresponding number M of nodes [29, p 349]. The marking with the upper index C indicates now that Chebyshev integration formulas are conceivable. As developed in [23, Section 7.5.2], integration formulas with uniform weights, i.e., Chebyshev formulas, are those where random errors in the function values have the least effect on the quadrature result. This makes these formulas very interesting in our context. However, although a lot of test calculations run well, we are not aware of convergence statements going along with $\Phi_{\pi,M}^C$ so far.

³Hanke and März [28, Section 2.3] is restricted to equidistant partitions π and collocation points $0 < \tau_1 < \dots < \tau_M < 1$. The generalization works without further ado.

⁴If the DAE (1) has index $\mu = 1$, then $l = k$, and hence also the choice $M = N$ makes sense. Then the system (21)–(22) for $x \in X_\pi$ is nothing else but the classical collocation approach, and the least-squares solution becomes the exact solution of the collocation system. We refer to [34] for a detailed survey of classical collocation methods, however, here we mainly focus on higher index cases yielding overdetermined systems.

Remark 4 The functional $\Phi_{\pi,M}^R$ gets its upper index R from the restriction operator $R_{\pi,M}$ introduced in [26] with nodes $0 < \tau_1 < \dots < \tau_M < 1$. Note that [26, Theorem 2.3] generalizes convergence results from [27, 28] to a large extend. Theorem 2 below allows even any nodes with (17).

Remark 5 The functional $\Phi_{\pi,M}^I$ has its upper index I simply from the word *integration formula*. We will see first convergence results going along with $\Phi_{\pi,M}^I$ in Theorem 2 below.

Intuitively, it seems reasonable to use a Gaussian quadrature rule for these purposes. However, it is not known if such a rule is most robust against rounding errors and/or other choices of the overall process.

Remark 6 Our approximations are according to the basic ansatz space X_π discontinuous, with possible jumps at the grid points in certain components. In this respect it does not matter which of our functionals is selected. Since we always have over-determined systems (21)–(22), it can no longer be expected that all components of the approximation are continuous even for the case $\tau_1 = 0, \tau_M = 1$. This is an important difference to the classical collocation methods for index-1 DAEs, which base on classical uniquely solvable linear systems, e.g., [34].

Theorem 2 *Let the DAE (1) be regular with index $\mu \in \mathbb{N}$ and let the boundary condition (2) be accurately stated. Let x_* be a solution of the boundary value problem (1)–(2), and let A, B, q and also x_* be sufficiently smooth.*

Let all partitions π be such that $h/h_{min} \leq \rho$, with a global constant ρ . Then, with

$$M \geq N + \mu,$$

the following statements are true:

- (1) *For sufficient fine partitions π and each sequence of arbitrarily placed nodes (17), there exists exactly one $x_\pi^R \in X_\pi$ minimizing the functional $\Phi_{\pi,M}^R$ on X_π , and*

$$\|x_\pi^R - x_*\|_{H_D^1} \leq C_R h^{N-\mu+1}.$$

- (2) *For each integration rule related to the interval $[0, 1]$, with M nodes (17) and positive weights $\gamma_1, \dots, \gamma_M$, that is exact for polynomials with degree less than or equal to $2M - 2$, and sufficient fine partitions π , there exists exactly one $x_\pi^I \in X_\pi$ minimizing the functional $\Phi_{\pi,M}^I$ on X_π , and $x_\pi^I = x_\pi^R$, thus*

$$\|x_\pi^I - x_*\|_{H_D^1} \leq C_R h^{N-\mu+1}.$$

Since Gauss-Legendre and Gauss-Radau integration rules are exact for polynomials up to degree $2M - 1$ and $2M - 2$, respectively, with positive weights, they are well suitable here, but Gauss-Lobatto rules do not meet the requirement of Theorem 2 (2).

Proof (1) In [26], additionally supposing $0 < \tau < \dots < \tau_M < 1$, conditions are derived that ensure the existence and uniqueness of x_π^R minimizing $\Phi_{\pi,M}^R$ on

X_π . It is shown that x_π^R has similar convergence properties as x_π minimizing Φ on X_π . Merely the constant C_R is slightly larger than C in (7). A further careful check of the proofs in [26] shows that the assertion holds also true for $\tau_1 = 0$ and/or $\tau_M = 1$, possibly with a larger constant C_R .

- (2) For each arbitrary $x \in X_\pi$, the expression

$$\theta_j(t) := \left| \sum_{i=1}^M l_{ji}(t)(A(t_{ji})(Dx)'(t_{ji}) + B(t_{ji}x(t_{ji}) - q(t_{ji})) \right|^2, \quad t \in (t_{j-1}, t_j),$$

shows that θ_j is a polynomial with degree less than or equal to $2M - 2$, thus

$$\int_{t_{j-1}}^{t_j} \theta_j(t) dt = h_j \sum_{i=1}^M \gamma_i \theta_j(t_{ji}) = h_j \sum_{i=1}^M \gamma_i |A(t_{ji})(Dx)'(t_{ji}) + B(t_{ji}x(t_{ji}) - q(t_{ji}))|^2$$

Therefore, it follows that $\Phi_{\pi,M}^I(x) = \Phi_{\pi,M}^R(x)$, for all $x \in X_\pi$, and $\Phi_{\pi,M}^I$ coincides with the special functional $\Phi_{\pi,M}^I$ having the same nodes. Eventually, (2) is a particular case of (1). □

As already emphasized above, until now we are aware of only sufficient convergence conditions, which is, in particular, especially applicable for the size of M . So far, often the applications run well with $M = N + 1$ and no significant difference to calculations with a larger M was visible, e.g., [27, Section 6] and [28, Section 4]. Also the experiments in Section 4 below are carried out with $M = N + 1$. The following statement for A and B with polynomial entries allows to choose M independently of the index μ and confirms the choice $M = N + 1$ for constant A and B .

Theorem 3 *Let the DAE (1) be regular with index $\mu \in \mathbb{N}$ and let the boundary condition (2) be accurately stated. Let x_* be a solution of the boundary value problem (1)–(2), and let q and also x_* be sufficiently smooth. Let the entries of A and B be polynomials with degree less than or equal to N_{AB} . Let all partitions π be such that $h/h_{\min} \leq \rho$, with a global constant ρ . Then, with*

$$M \geq N + 1 + N_{AB},$$

the following statements are true:

- (1) *For sufficient fine partitions π and each sequence of arbitrarily placed nodes (17), there exists exactly one $x_\pi^R \in X_\pi$ minimizing the functional $\Phi_{\pi,M}^R$ on X_π , and*

$$\|x_\pi^R - x_*\|_{H_D^1} \leq C_R h^{N-\mu+1}.$$

- (2) *For each integration rule of interpolation type related to the interval $[0, 1]$, with M nodes (17) and positive weights $\gamma_1, \dots, \gamma_M$, that is exact for polynomials with degree less than or equal to $2M - 2$, and sufficient fine partitions π , there exists exactly one $x_\pi^I \in X_\pi$ minimizing the functional $\Phi_{\pi,M}^I$ on X_π , and $x_\pi^I = x_\pi^R$, thus*

$$\|x_\pi^I - x_*\|_{H_D^1} \leq C_R h^{N-\mu+1}.$$

- (3) If A and B are even constant matrices, for sufficient fine partitions π and each sequence of arbitrarily placed nodes (17), there exists exactly one $x_\pi^R \in X_\pi$ minimizing the functional $\Phi_{\pi,M}^R$ on X_π , and

$$\|x_\pi^R - x_*\|_{H_D^1} \leq C_R h^{\max\{0, N-\mu+1\}}.$$

Proof (1): This follows from [26, Proposition 2.2(iv)] and [27, Theorem 4.1]. (2): As in the proof of the previous theorem, this is again a consequence of (1). (3): The statement is a consequence of [26, Proposition 2.2(iv)] and [27, Theorem 4.7]. \square

Remark 7 Observe a further interesting feature. Let A and B be constant matrices. Set $N = 1, M = N + 1$. Then, it holds that

$$\Phi_{\pi,M}^C(x) = \Phi_{\pi,M}^R(x) = \Phi_{\pi,M}^I(x), \quad x \in X_\pi,$$

in which $\Phi_{\pi,M}^I$ is associated with the corresponding Gauss-Legendre or Gauss-Radau rules. This follows from the fact that the 2-point Chebyshev integration nodes are just the Gauss-Legendre nodes.

We underline that, by Theorem 3 (3), the approximate solutions stay bounded also for DAEs with larger index μ , for instance [28, Table 6] confirming that for an index four Jordan DAE.

Having in mind the implementation of such an overdetermined least-squares collocation, for given partition π and a given polynomial degree N , a number of parameters and options must be selected:

- basis functions for X_π ;
- number M of collocation points and their location $0 \leq \tau_1 < \dots < \tau_M \leq 1$;
- setup and solution of the discrete least-squares problem.

We will discuss the first two issues in this context below and refer to [24] for the third one. The main aim is on implementations being as stable as possible, not necessarily best computational efficiency.

3 Collocation nodes, mass matrix and integration weights

3.1 Collocation nodes for $\Phi_{\pi,M}^R$

The functional $\Phi_{\pi,M}^R$ in (16) is based on polynomial interpolation using M nodes (17). It seems reasonable to choose these nodes in such a way that, separately on each subinterval $[t_{j-1}, t_j]$ of the partition, the interpolation error is as small as possible in a certain sense. Without restriction of the generality we can trace back the matter to the interval $[0, 1]$.

Consider functions $q \in C([0, 1], \mathbb{R}^m)$ and define the interpolation operator $R_M : C([0, 1], \mathbb{R}^m) \rightarrow C([0, 1], \mathbb{R}^m)$ by

$$R_M q = \sum_{i=1}^M l_i q(\tau_i),$$

with the Lagrange basis functions (18) such that $(R_M q)(\tau_i) = q(\tau_i), i = 1, \dots, M$, and $R_M q \in Y_M$, where $Y_M \subset C([0, 1], \mathbb{R}^m)$ is the subspace of all functions whose components are polynomials up to degree $M - 1$. Introducing $\omega(\tau) = (\tau - \tau_1)(\tau - \tau_2) \cdots (\tau - \tau_M)$ and using componentwise the divided differences we have the error representation, e.g., [23, Kapitel 5],

$$q(\tau) - (R_M q)(\tau) = \omega(\tau) q[\tau_1, \dots, \tau_M, \tau].$$

For smooth functions $q \in C^M([0, 1], \mathbb{R}^m)$ it follows that

$$\|q - R_M q\|_{L^2}^2 = \int_0^1 \omega(\tau)^2 |q[\tau_1, \dots, \tau_M, \tau]|^2 d\tau \leq \int_0^1 \omega(\tau)^2 d\tau \frac{m}{(M!)^2} \|q^{(M)}\|_{\infty}^2.$$

For the evaluation of $\Phi_{\pi, M}^R$ (16), it seems reasonable to choose the collocation nodes in such a way that this expression is minimized for all functions $q \in C^{(M)}([0, 1], \mathbb{R}^m)$. The optimal set of nodes is determined by the condition

$$\min_{0 \leq \tau_1 < \dots < \tau_M \leq 1} \|\omega\|_{L^2(0,1)}.$$

It is well known that this functional is minimized if the collocation nodes are chosen to be the Gauss-Legendre nodes [23, Chapter 7.5.1 and 4.5.4].

On the other hand, the best polynomial approximation to a given function q in the L^2 -norm is obtained if the Fourier approximation with respect to the Legendre polynomials is computed. However, to the best of our knowledge, there are no estimations of the interpolation error in $L^2((0, 1), \mathbb{R}^m)$ known.⁵ However, in the uniform norm and with arbitrary node sequences, for each $q \in C([0, 1], \mathbb{R}^m)$, the estimate

$$\|R_M q - q\|_{\infty} \leq (1 + \Lambda_M) \text{dist}_{\infty}(q, Y_M)$$

holds true where $\text{dist}_{\infty}(q, Y_M) = \min\{\|q - y\|_{\infty} \mid y \in Y_M\}$ and Λ_M is so-called Lebesgue constant defined by

$$\Lambda_M = \max_{\tau \in [0,1]} \sum_{i=1}^M |l_i(\tau)|$$

in which l_i are again the Lagrange basis functions (18).

The Lebesgue constant Λ_M^L for Gauss-Legendre nodes has the property $\Lambda_M^L = O(\sqrt{M})$. If instead Chebyshev nodes are used, the corresponding Lebesgue constant Λ_M^C behaves like $\Lambda_M^C = O(\log M)$ ([12, p 206] and the references therein). For uniform polynomial approximations, these nodes are known to be optimal [9, Theorem

⁵It holds that $\|R_M q\|_{\infty} \leq \max_{\tau \in [0,1]} \sum_{i=1}^M |l_i(\tau)| |q(\tau_i)| \leq \Lambda_M \|q\|_{\infty}$ which means that the interpolation operator R_M is bounded in $C([0, 1], \mathbb{R}^m)$, and the Lebesgue constant is a bound of the operator norm. In contrast, R_M it is unbounded in $L^2((0, 1), \mathbb{R}^m)$!

Table 1 Lebesgue constants for Chebyshev nodes (Λ_M^C), Gauss-Legendre nodes (Λ_M^L), Gauss-Lobatto nodes (Λ_M^{Lo}), Gauss-Radau nodes (Λ_M^R), and uniform nodes including the boundaries (Λ_M^U) and without boundaries (Λ_M^O)

M	Λ_M^C	Λ_M^L	Λ_M^{Lo}	Λ_M^R	Λ_M^U	Λ_M^O
5	1.989	3.322	1.636	4.035	2.708	10.375
10	2.429	5.193	2.121	6.348	17.849	204.734
15	2.687	6.649	2.386	8.126	283.211	5107.931
20	2.870	7.885	2.576	9.627	5889.584	138852.138

7.6]. Table 1 shows some values for the Lebesgue constants. Note that the Lebesgue constants Λ_M^U for equidistant nodes grow exponentially (see, e.g., [44]).⁶

Remark 8 Computation of nodes and weights for Gauss-type integration formulae

In the following we will make heavy use of Gauss-Legendre, Gauss-Radau, and Gauss-Lobatto integration nodes and their corresponding weights. Since we do not have them available in tabular form for large M with sufficient accuracy, they will be computed on the fly. A severe concern is the accuracy of the nodes and weights. In the case of Gauss-Legendre integration rules, the computed nodes and weights have been provided by the Gnu Scientific Library routine `glfixed.c` [14]. It makes use of tabulated values for $M = 1(1)20, 32, 64, 96, 100, 128, 256, 512, 1024$ with an accuracy of 27 digits. Other values are computed on the fly with an accuracy being a small multiple of the machine rounding unit using an adapted version of the Newton method.

For computing the Gauss-Lobatto nodes and weights, the methods of [37] (using the Newton method) as well as [17] (a variant of the method in [21]) have been implemented. Table 2 contains some comparisons to the tabulated values in [37] that have 20 digits. The method of [37] provides slightly more accurate values than that of [17]. Therefore, the former has been used further on.

We did not find sufficiently accurate tabulated values for the Gauss-Radau nodes and weights. Therefore, the method of [16] has been implemented. We assume that the results obtained have an accuracy similar to the values for the Gauss-Lobatto nodes and weights using the method in [17].

3.2 The mass matrix

In the following, we will make extensive use of Legendre polynomials. For the readers' convenience, the necessary properties are collected in Appendix A.1.

⁶For each M , there is a set of interpolation nodes τ_i^* that minimizes the corresponding Lebesgue constant Λ_M^* . This constant is only slightly smaller than Λ_M^C [30].

Table 2 Accuracy of the computed nodes and weights of the Gauss-Lobatto integration rules

M	Method of [37]				Method of [17]			
	(A)	(B)	(C)	(D)	(A)	(B)	(C)	(D)
6	1.11e-16	1.11e-16	4.73e-16	5.87e-16	1.11e-16	3.33e-16	4.73e-16	9.99e-15
12	1.11e-16	8.33e-17	2.01e-15	6.14e-16	3.33e-16	4.44e-16	6.04e-15	5.86e-14
24	0	3.47e-17	0	9.64e-16	2.22e-16	2.22e-16	8.37e-15	1.23e-13
48	1.11e-16	3.47e-17	3.41e-14	5.23e-15	3.33e-16	1.22e-15	1.71e-13	2.76e-12
96	5.55e-16	2.95e-17	4.73e-16	1.76e-14	4.44e-16	4.44e-16	2.76e-13	4.05e-12

For each method, the absolute error of the nodes (A), the absolute error of the weights (B), the maximum componentwise relative error of nodes (C) and weights (D) are shown. The machine accuracy (machine epsilon) is 2.22×10^{-16}

Let us turn to $\Phi_{\pi, M}^R$ (16) again. A critical ingredient for determining its properties is the mass matrix L^R in (20). Denote as before by $l_i(\tau), i = 1, \dots, M$, the Lagrange basis functions for the node sequence (17), that is, cf. (18),

$$l_i(\tau) = \frac{\prod_{\kappa \neq i} (\tau - \tau_\kappa)}{\prod_{\kappa \neq i} (\tau_i - \tau_\kappa)}.$$

For evaluating L^R , we will use the normalized shifted Legendre polynomials $\hat{P}_\nu = (\nu + 1)^{1/2} \tilde{P}_\nu$ (cf Appendix A.1). Assume the representation

$$l_i(\tau) = \sum_{\nu=1}^M \alpha_{i\nu} \hat{P}_{\nu-1}(\tau). \tag{23}$$

A short calculation shows

$$L_{ij}^R = \sum_{\lambda=1}^M \alpha_{i\lambda} \alpha_{j\lambda}.$$

Letting $a^i = (\alpha_{i1}, \dots, \alpha_{iM})^T$ we obtain $L_{ij}^R = (a^i)^T a^j$. Collecting the vectors a^i in a matrix $A = (a^1, \dots, a^M)$ it holds $L^R = A^T A$. The definition of the coefficients $\alpha_{i\nu}$ provides us with $\tilde{V} a^i = e^i$ where e^i denotes the i th unit vector and

$$\tilde{V} = \begin{bmatrix} \hat{P}_0(\tau_1) & \dots & \hat{P}_{M-1}(\tau_1) \\ \vdots & & \vdots \\ \hat{P}_0(\tau_M) & \dots & \hat{P}_{M-1}(\tau_M) \end{bmatrix}. \tag{24}$$

This gives $A = \tilde{V}^{-1}$.

$V = \tilde{V}^T$ is a so-called Vandermonde-like matrix [15]. It is nonsingular under the condition (17) [41, Theorem 3.6.11]. In [15], representations and estimations of the condition number with respect to the Frobenius norm of such matrices are derived. In particular, [15, Table 1] shows impressingly small condition numbers if the collocation nodes are chosen to be the zeros of \hat{P}_M , that is the Gauss-Legendre

Table 3 Spectral condition numbers of the Vandermonde-like matrices for different node choices

M	GLe	GR	GLo	Ch	cNC	oNC
5	1.55e+0	2.79e+0	3.23e+0	2.16e+0	3.76e+0	3.04e+0
10	2.11e+0	3.96e+0	4.28e+0	3.00e+0	2.39e+1	5.23e+1
15	2.57e+0	4.85e+0	5.11e+0	3.66e+0	3.98e+2	1.14e+3
20	2.94e+0	5.60e+0	5.83e+0	4.21e+0	8.62e+3	3.10e+4
50	4.62e+0	8.86e+0	9.00e+0	6.60e+0	1.13e+12	1.97e+13
100	6.52e+0	1.25e+1	1.26e+1	9.32e+0	*	*

The columns represent Gauss-Legendre nodes (GLe), Gauss-Radau nodes (GR), Gauss-Lobatto nodes (GLo), Chebyshev nodes (Ch), Newton-Cotes type nodes including the boundary (cNC) and without the boundary (oNC). An asterisk * indicates an overflow condition

nodes. Moreover, this condition number is optimal among all scalings of the Legendre polynomials [15]. A consequence of the Christoffel-Darboux formula is that the rows of \tilde{V} are orthogonal for Gauss-Legendre nodes.⁷ Thus, we have the representation $\tilde{V} = \mathcal{D}U$ with an orthogonal matrix U and a diagonal matrix \mathcal{D} with positive diagonal entries.⁸

It is known that the Gauss-Legendre nodes are not the very best set of nodes. However, a comparison of Tables 1 and 2 in [15] as well as [18, Table 4] indicates that the gain of choosing optimal nodes for Legendre polynomials compared to the choice of Gauss-Legendre nodes is rather minor.

In Table 3 we provide condition numbers of \tilde{V} with respect to the Euclidean norm for different choices of nodes. Note that the condition number of L^R is the square of that of \tilde{V} .

The condition numbers for all Gauss-type and Chebyshev nodes are remarkably small.

3.3 Computation of quadrature weights for general $\Phi_{\pi,M}^I$

In order to apply $\Phi_{\pi,M}^I$ (15), a numerical quadrature formulae is necessary. For standard nodes sequences (Gauss-Legendre, Gauss-Lobatto, Gauss-Radau) their

⁷The Christoffel-Darboux formula for Legendre polynomials reads: If $i \neq \kappa$, then

$$\sum_{v=0}^{M-1} \hat{P}_v(\tau_i) \hat{P}_v(\tau_\kappa) = \frac{\mu_{M-1}}{\mu_M} \frac{\hat{P}_M(\tau_i) \hat{P}_{M-1}(\tau_\kappa) - \hat{P}_M(\tau_\kappa) \hat{P}_{M-1}(\tau_i)}{\tau_i - \tau_\kappa}$$

where μ_M and μ_{M-1} are the leading coefficients of \hat{P}_M and \hat{P}_{M-1} , respectively. For the Gauss-Legendre nodes, it holds $\hat{P}_M(\tau_i) = 0$. Hence, the right-hand sides vanish.

⁸The diagonal elements $d_i, i = 1, \dots, M$ of \mathcal{D} can be evaluated analytically using the Christoffel-Darboux formula again:

$$d_i = \sum_{v=0}^{M-1} \hat{P}_v^2(\tau_i) = \frac{\mu_{M-1}}{\mu_M} \left(\hat{P}'_M(\tau_i) \hat{P}_{M-1}(\tau_i) - \hat{P}_{M-1}(\tau_i) \hat{P}'_M(\tau_i) \right) = \frac{\mu_{M-1}}{\mu_M} \hat{P}'_M(\tau_i) \hat{P}_{M-1}(\tau_i).$$

computation has been described above. However, for general node sequences, the weights must be evaluated. This can be done following the derivations in [41, p. 175]: Let $\hat{P}_v(\tau)$ denote the normalized shifted Legendre polynomials as before. In particular, it holds then

$$\int_0^1 \hat{P}_0(\tau) d\tau = 1, \quad \int_0^1 \hat{P}_v(\tau) d\tau = 0, \quad v = 1, 2, \dots$$

For a given function $q \in C[0, 1]$, the integral is approximated by the integral of its polynomial interpolation. Using the representation (23) of the Lagrange basis functions we obtain

$$\begin{aligned} \int_0^1 q(\tau) d\tau &\approx \int_0^1 \sum_{i=1}^M q(\tau_i) \sum_{v=1}^M \alpha_{iv} \hat{P}_{v-1}(\tau) d\tau \\ &= \sum_{i=1}^M q(\tau_i) \sum_{v=1}^M \alpha_{iv} \int_0^1 \hat{P}_{v-1}(\tau) d\tau \\ &= \sum_{i=1}^M q(\tau_i) \alpha_{i1}. \end{aligned}$$

Consequently, for the weights it holds $\gamma_i = \alpha_{i1}, i = 1, \dots, M$. The definition (23) shows that the vector $\gamma = (\gamma_1, \dots, \gamma_M)^T$ of weights fulfills the linear system

$$V\gamma = e^1$$

where $V = \tilde{V}^T$ with \tilde{V} from (24) and $e^1 = (1, 0, \dots, 0)^T$ is the first unit vector.

The discussion of the condition number of V shows that we can expect reliable and accurate results at least for reasonable node sequences.

For general node sequences, the weights may become negative. This happens, for example, for uniformly distributed nodes and $M > 7$ (Newton-Cotes formulae) [41, p. 148]. So for $\Phi_{\pi, M}^I$, only node sequences leading to positive quadrature weights γ_i are admitted in order to prevent L^I from not being positive definite.

4 Choice of basis functions for the ansatz space X_π

The ansatz space X_π (5) consists of piecewise polynomials having the degree $N - 1$ for the algebraic components and the degree N for the differentiated ones on each subinterval of the partition π (4). For collocation methods for boundary value problems for ordinary differential equations this question has led to the choice of a Runge-Kutta basis for stability reasons (see [2]). This has been later on also used successfully for boundary value problems for index-1 DAEs [3, 31, 32, 34]. However, this ansatz makes heavily use of the collocation nodes which are at the same time used as the nodes for the Runge-Kutta basis. In our case, the number M of collocation nodes and the degree N of the polynomials for the differentiated components do not coincide since $M > N$ such that the reasoning applied in the case of ordinary differential equations does not transfer to the least-squares case.

Taking into account the computational expense for solving the discretized system, bases with local support are preferable. Ideally, the support of each basis function consists of only one subinterval of (4).⁹ Note that the Runge-Kutta basis has this property. We consider the Runge-Kutta basis and further local basis with orthogonal polynomials. A drawback of this strategy is the fact that the continuity of the piecewise polynomials approximating the differentiated components must be ensured explicitly. This in turn will lead to a discrete least-squares problem with equality constraints. Details can be found in Part 2 [24].

Looking for a local basis we turn to the reference interval $[0, 1]$. Once a basis on this reference interval is available it can be defined on any subinterval (t_{j-1}, t_j) by a simple linear transformation.

Assume that $\{p_0, \dots, p_{N-1}\}$ is a basis of the set of polynomials of degree less than N defined on the reference interval $[0, 1]$. Then, a basis $\{\bar{p}_0, \dots, \bar{p}_N\}$ for the ansatz functions for the differentiated components is given by

$$\bar{p}_i(\rho) = \begin{cases} 1, & i = 0, \\ \int_0^\rho p_{i-1}(\sigma) d\sigma, & i = 1, \dots, N, \end{cases} \quad \rho \in [0, 1], \quad (25)$$

and the transformation to the interval (t_{j-1}, t_j) of the partition π (4) yields

$$\begin{aligned} p_{ji}(t) &= p_i((t - t_{j-1})/h_j), \\ \bar{p}_{ji}(t) &= h_j \bar{p}_i((t - t_{j-1})/h_j). \end{aligned} \quad (26)$$

Additionally to this transformation, the continuity of the piecewise polynomials must be ensured. This gives rise to the additional conditions

$$\bar{p}_{ji}(t_j) = \bar{p}_{j+1,i}(t_j), \quad i = 0, \dots, N, \quad j = 1, \dots, n-1, \quad (27)$$

which must be imposed explicitly.¹⁰

4.1 The Runge-Kutta basis

In order to define the Runge-Kutta basis, let the N interpolation points ρ_i with (9) be given. Then, the Lagrange basis functions are chosen,

$$p_i(\rho) = \frac{\prod_{\kappa \neq i+1} (\rho - \rho_\kappa)}{\prod_{\kappa \neq i+1} (\rho_i - \rho_\kappa)}, \quad i = 0, \dots, N-1.$$

Remark 9 Note that the interpolation nodes are only used to define the local basis functions. Thus, their selection is completely independent of the choice of collocation nodes. In view of the estimations (10) and (11) and the argumentation there we prefer Gauss-Legendre interpolation nodes. This choice is also supported by Experiments 2 and 5 below.

⁹This excludes for example B-spline bases!

¹⁰This is in contrast to choices of basis functions that fulfil the basis conditions. An example of such basis functions is B-splines.

The numerical computation of \bar{p} is more involved. If not precalculated, the integrals must be available in a closed formula. This can surely be done by expressing the Lagrange basis functions in the monomial representation such that the integration can be carried out analytically. Once these coefficients are known, the evaluation of the values of the basis functions at a given $\rho \in [0, 1]$ is easily done using the Horner method. However, this approach amounts to the inversion of the Vandermonde matrix using the nodes (9). This matrix is known to be extremely ill-conditioned. In particular, its condition number grows exponentially with N [5, 19]. Therefore, an orthogonal basis might be better suited. This leads to a representation

$$p_i(\rho) = \sum_{\kappa=1}^N \alpha_{i\kappa} Q_\kappa(\rho) \tag{28}$$

for some polynomials Q_1, \dots, Q_N . If these polynomials fulfil a three-term recursion,¹¹ the evaluation of function values can be performed using the Clenshaw algorithm [13] which is only slightly more expensive than the Horner method. In order to use this approach, the integrals of p_0, \dots, p_{N-1} must be easily representable in terms of the chosen basis. Here, the Legendre and Chebyshev polynomials are well-suited (cf below Appendix A.1 and (30) as well as Appendix A.2 and (32)).

4.2 Orthogonal polynomials

A reasonable choice for the basis is orthogonal polynomials. We will consider Legendre polynomials first. A motivation is provided in the following example.

Example 1 Consider the index-1 DAE

$$x = q(t), \quad t \in [0, 1].$$

Let $\{\hat{P}_0, \dots, \hat{P}_{N-1}\}$ be the normalized shifted Legendre polynomials. Then letting $x = \sum_{i=1}^N \alpha_i \hat{P}_{i-1}$ for some vector $\alpha = (\alpha_1, \dots, \alpha_N)^T$, the least-squares functional

$$\Phi(x) = \int_0^1 (x(t) - q(t))^2 dt$$

corresponding to this DAE is minimized if $\alpha = b$ and $b = (b_1, \dots, b_N)^T$ where $b_i = \int_0^1 q(t) \hat{P}_{i-1}(t) dt$ which is just the best approximation of the solution in $H_D^1((0, 1), \mathbb{R}) = L^2((0, 1), \mathbb{R})$.

Similar relations hold for the differential equation $x' = f$ if the basis functions for the differentiated components are constructed according to (25). Hence, these basis functions seem to qualify well for index-1 DAEs.

The necessary ingredients for the efficient implementation of the Legendre polynomials are collected in Appendix A.1.

¹¹Which they do if the polynomials are orthogonal with respect to some scalar product [9, Theorem 6.2].

Another common choice is Chebyshev polynomials of the first kind. They have been used extensively in the context of spectral methods because of their excellent approximation properties (cf [11, 43], see also [10]). The relations used for their implementation can be found in Appendix A.2.¹²

4.3 Comparison of different basis representations

The choice of the basis function representations is dominated by the question of obtaining a most robust implementation. The computational complexity of the representations presented above is not that much different such that this aspect plays a minor role.

The check for robustness can be subdivided into two questions:

1. Which representation is most robust locally?
2. Which representation is most robust globally?

In the following experiments, N will be varied. The functional used is $\Phi_{\pi, M}^R$. The number of collocation nodes is $M = N + 1$. Table 3 motivates the choice of the Gauss-Legendre nodes as collocation nodes. In order to compute the norms of $L^2((0, 1), \mathbb{R}^m)$ and $H_D^1((0, 1), \mathbb{R}^m)$, Gaussian quadrature with $N + 2$ integration nodes on each subinterval of π is used.

4.3.1 Local behavior of the basis representations

In order to answer the first question, it is reasonable to experiment first with a higher index example that does not have any dynamic components (that is, $l = 0$) on a grid π consisting only of one subinterval (that is, $n = 1$). In that case, we check the ability to interpolate functions and to numerically differentiate them.

For $n = 1$, there are no continuity conditions (27) involved. Therefore, the discrete problem becomes a linear least-squares problem. We will solve it by a Householder QR factorization with column pivoting as implemented in the Eigen library.

The following example is used in [27, 28].

Example 2

$$\begin{aligned}x_2'(t) + x_1(t) &= q_1(t), \\t\eta x_2'(t) + x_3'(t) + (\eta + 1)x_2(t) &= q_2(t), \\t\eta x_2(t) + x_3(t) &= q_3(t), \quad t \in [0, 1].\end{aligned}$$

¹²Let us note in passing that the first routine for solving two-point boundary value problems in the NAG library (NAG® is a registered trademark of The Numerical Algorithms Group) besides shooting methods was just a least-squares collocation method corresponding to $n = 1$ and using a version of the functional $\Phi_{\pi, M}^C$. The NAG routine D02AFF and its predecessor D02TGF (and its driver D02JBF) use Chebyshev polynomials as basis functions and Gauss-Legendre nodes as collocation points [1, 20]. This routine appeared as early as 1970 in Mark 8 of the library and survived to date (as of Mark 27 of 2019) [42].

This is an index-3 example with dynamical degree of freedom $l = 0$ such that no additional boundary or initial conditions are necessary for unique solvability. We choose the exact solution

$$\begin{aligned} x_{*,1}(t) &= e^{-t} \sin t, \\ x_{*,2}(t) &= e^{-2t} \sin t, \\ x_{*,3}(t) &= e^{-t} \cos t \end{aligned}$$

and adapt the right-hand side q accordingly. For the exact solution, it holds $\|x_*\|_{L^2((0,1),\mathbb{R}^3)} \approx 0.673$, $\|x_*\|_{L^\infty((0,1),\mathbb{R}^3)} = 1$, and $\|x_*\|_{H_D^1((0,1),\mathbb{R}^3)} \approx 1.11$.

Experiment 1 Robustness of the representation of the Runge-Kutta basis

In a first experiment we intend to clarify the differences between different representations of the Runge-Kutta basis. The interpolation nodes (9) have been fixed to be the Gauss-Legendre nodes (cf (10)). The Runge-Kutta basis has been represented with respect to the monomial, Legendre, and Chebyshev bases. The results are shown in Fig. 1 (see Appendix). This test indicates that the monomial basis is much less robust than the others for $N > 10$ while the other representations behave very similar.

Experiment 2 Robustness of the Runge-Kutta basis with respect to the node sequence

In this experiment we are interested in understanding the influence of the interpolation nodes. For that, we compared the uniform nodes sequence to the Gauss-Legendre and Chebyshev nodes. The uniform nodes are given by $\rho_i = (i - \frac{1}{2})/N$.

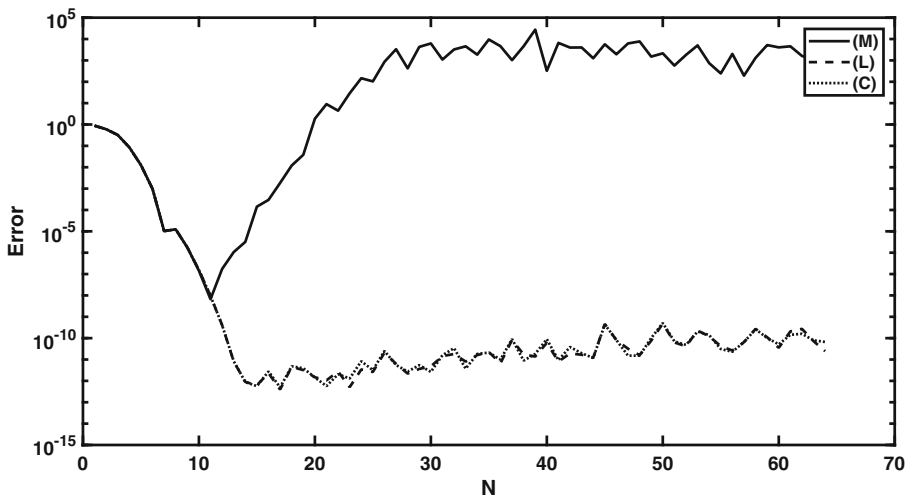


Fig. 1 Error of the approximate solution in Experiment 1 measured in the norm of $H_D^1((0, 1), \mathbb{R}^m)$. The abbreviations (M) for the monomial basis, (L) for the Legendre basis, and (C) for the Chebyshev basis are used

In accordance with the results of the previous experiment, the representation of the Runge-Kutta basis in Legendre polynomials has been chosen. The results are shown in Fig. 2. Not unexpectedly, uniform nodes are inferior to the other choices at least for $N > 13$. On the other hand, there is no significant difference between Gauss-Legendre and Chebyshev nodes.

Experiment 3 Robustness of different polynomial representations

In this experiment we intend to compare the robustness of different bases. Therefore, we have chosen the Runge-Kutta basis with Gauss-Legendre interpolation nodes, the Legendre polynomials, and the Chebyshev polynomials. The results are shown in Fig. 3. All representations show similar behavior.

A general note is in order. The exact solution has approximately the norm 1. The machine accuracy is $\epsilon_{\text{mach}} \approx 2.22 \times 10^{-16}$ in all computations. The best accuracy obtained is approximately 10^{-12} . Considering that there is a twofold differentiation involved in the problem of the example we would expect a much lower accuracy. This surprising behavior has also been observed in other experiments and when using the norms of $L^2((0, 1), \mathbb{R}^m)$ and $L^\infty((0, 1), \mathbb{R}^m)$.

The next example is an index-3 one which has $l = 4$ dynamical degrees of freedom. It is the linearized version of an example presented [6] that has also been considered in [27].

Example 3 Consider the DAE

$$A(Dx)'(t) + B(t)x(t) = q(t), \quad t \in [0, 5],$$

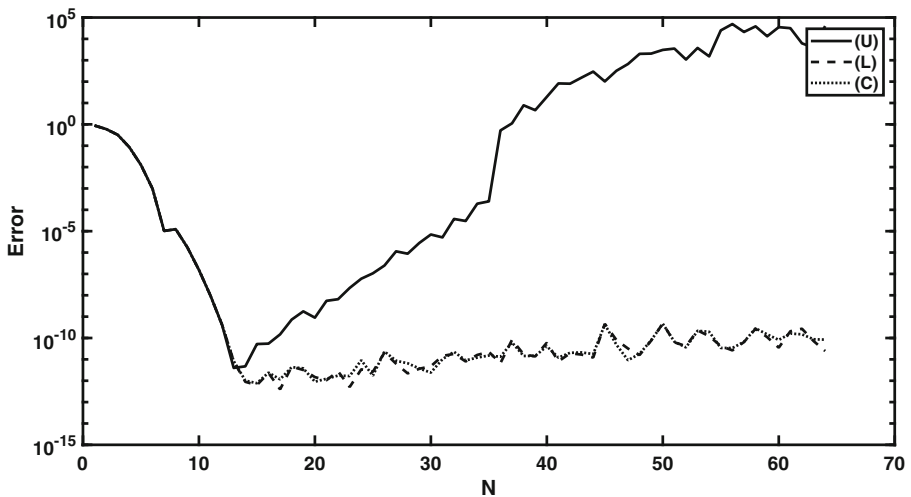


Fig. 2 Error of the approximate solution in Experiment 2 measured in the norm of $H_D^1((0, 1), \mathbb{R}^m)$. The abbreviations (U) for uniform nodes, (L) for the Gauss-Legendre nodes, and (C) for the Chebyshev nodes are used

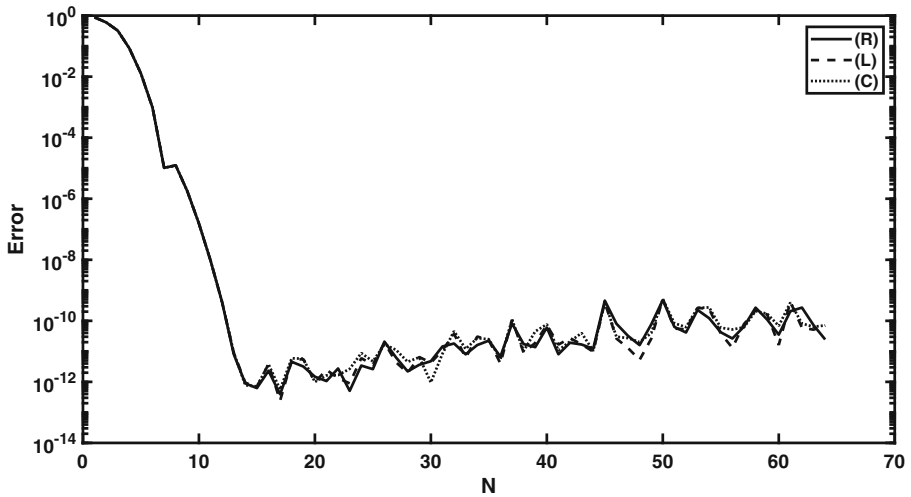


Fig. 3 Error of the approximate solution in Experiment 3 measured in the norm of $H_D^1((0, 1), \mathbb{R}^m)$. The abbreviations (R) for the Runge-Kutta basis in Legendre representation, (L) for the Legendre basis, and (C) for the Chebyshev basis are used

where

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, D = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

and the smooth coefficient matrix

$$B(t) = \begin{bmatrix} 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & \sin t & 0 & 1 & -\cos t & -2\rho \cos^2 t \\ 0 & 0 & -\cos t & -1 & 0 & -\sin t & -2\rho \sin t \cos t \\ 0 & 0 & 1 & 0 & 0 & 0 & 2\rho \sin t \\ 2\rho \cos^2 t & 2\rho \sin t \cos t & -2\rho \sin t & 0 & 0 & 0 & 0 \end{bmatrix},$$

$\rho = 5,$

subject to the initial conditions

$$x_2(0) = 1, \quad x_3(0) = 2, \quad x_5(0) = 0, \quad x_6(0) = 0.$$

This problem has the tractability index 3 and dynamical degree of freedom $l = 4$. The right-hand side q has been chosen in such a way that the exact solution becomes

$$\begin{aligned} x_{*,1} &= \sin t, & x_{*,4} &= \cos t, \\ x_{*,2} &= \cos t, & x_{*,5} &= -\sin t, \\ x_{*,3} &= 2 \cos^2 t, & x_{*,6} &= -2 \sin 2t, \\ x_{*,7} &= -\rho^{-1} \sin t. \end{aligned}$$

For the exact solution, it holds $\|x_*\|_{L^2((0,5),\mathbb{R}^7)} \approx 5.2$, $\|x_*\|_{L^\infty((0,5),\mathbb{R}^7)} = 2$, and $\|x_*\|_{H_D^1((0,5),\mathbb{R}^7)} \approx 9.4$.

The following experiments with Example 3 are carried out under the same conditions as before when using Example 2.

Experiment 4 Robustness of the representation of the Runge-Kutta basis

In this experiment we intend to clarify the differences between different representations of the Runge-Kutta basis. The interpolation points have been fixed to be the Gauss-Legendre nodes. The Runge-Kutta basis has been represented with respect to the monomial, Legendre, and Chebyshev bases. The results are shown in Fig. 4. This test indicates that the monomial basis is much less robust than the others for $N > 15$ while the other representations behave very similar.

Experiment 5 Robustness of the Runge-Kutta basis with respect to the node sequence

In this experiment we are interested in understanding the influence of the interpolation nodes. For that, we compared the uniform nodes sequence to the Gauss-Legendre

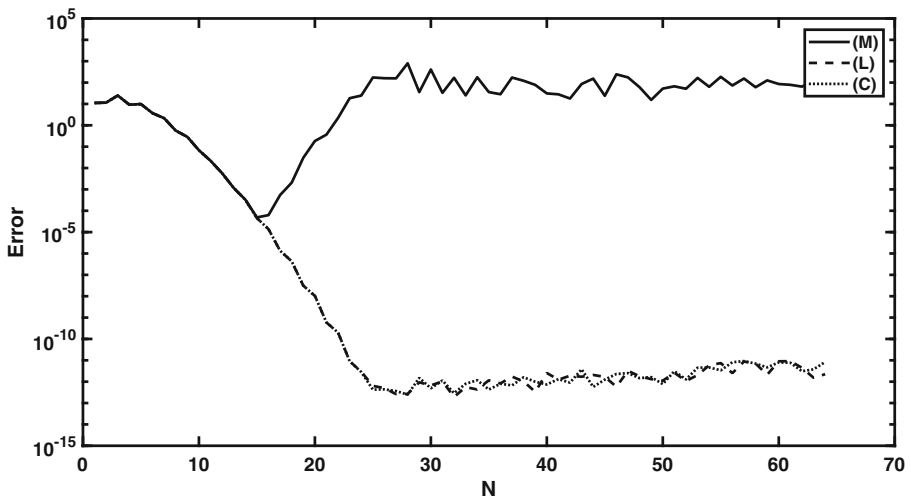


Fig. 4 Error of the approximate solution in Experiment 4 measured in the norm of $H_D^1((0, 1), \mathbb{R}^m)$. The abbreviations (M) for the monomial basis, (L) for the Legendre basis, and (C) for the Chebyshev basis are used

and Chebyshev nodes. The uniform nodes are given by $\rho_i = (i - \frac{1}{2})/N$. In accordance with the results of the previous experiment, the representation of the Runge-Kutta basis in Legendre polynomials has been chosen. The results are shown in Fig. 5. Not unexpectedly, uniform nodes are inferior to the other choices at least for $N > 20$. However, there is no real difference between Gauss-Legendre and Chebyshev nodes.

Experiment 6 Robustness of different polynomial representations

In this experiment we intend to compare the robustness of different bases. Therefore, we have chosen the Runge-Kutta basis with Gauss-Legendre interpolation nodes, the Legendre polynomials, and the Chebyshev polynomials. The results are shown in Fig. 6. All representations show similar behavior.

As a conclusion, we can see that the results of the Experiments 1–3 and 4–6 are largely consistent.

4.3.2 Global behavior of the basis representations

We are interested in understanding the global error, which corresponds to error propagation in the case of initial value problems. In order to understand the error propagation properties we will investigate the accuracy of the computed solution with respect to an increasing number of subintervals n . This motivates to use a rather low order N of polynomials. In the previous section we observed that there is no difference in the local properties between different basis representations for low degrees N of the ansatz polynomials.

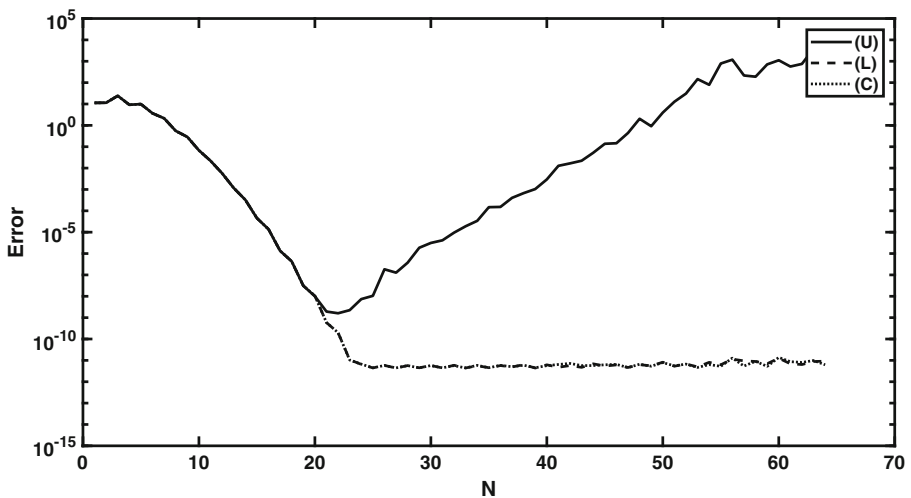


Fig. 5 Error of the approximate solution in Experiment 5 measured in the norm of $H_D^1((0, 1), \mathbb{R}^m)$. The abbreviations (U) for uniform nodes, (L) for the Gauss-Legendre nodes, and (C) for the Chebyshev nodes are used

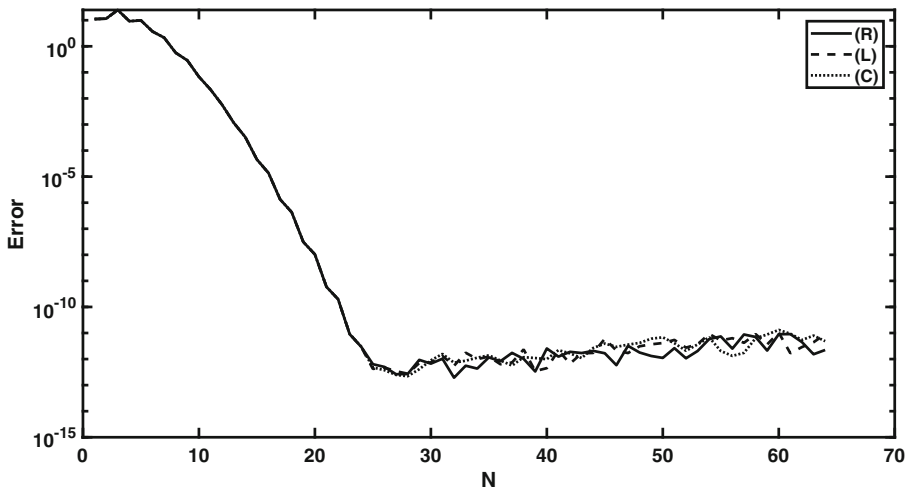


Fig. 6 Error of the approximate solution in Experiment 6 measured in the norm of $H_D^1((0, 1), \mathbb{R}^m)$. The abbreviations (R) for the Runge-Kutta basis in Legendre representation, (L) for the Legendre basis, and (C) for the Chebyshev basis are used

In the following experiments, the functionals used are $\Phi_{\pi, M}^R$ and $\Phi_{\pi, M}^C$. The number of collocation nodes is again $M = N + 1$. The basis functions are the shifted Legendre polynomials.

The discrete problem for $n > 1$ is an equality constraint linear least-squares problem. The equality constraint consists just of the continuity requirements for the differentiated components of the elements in X_π . The problem is solved by a direct solution method as described in Part 2 [24]. In short, the equality constraints are eliminated by a sparse QR-decomposition with column pivoting as implemented in the code SPQR [8]. The resulting least-squares problem has then been solved by the same code.

Experiment 7 Influence of selection of collocation nodes, approximation degree N , and number n of subintervals.

In this experiment, we use Example 3 and vary the choice of collocation nodes as well as the degree N of the polynomial basis and the number n of subintervals. We compare Gauss-Legendre, Radau IIA and Lobatto collocation nodes. Since this example is a pure initial value problem, the use of the Radau IIA collocation nodes is especially justified.¹³ The results using $\Phi_{\pi, M}^R$ are collected in Table 4, those using $\Phi_{\pi, M}^C$ in Table 5. We observe no real difference between the different sets of collocation points. The results seem to confirm the conjecture that, in case of smooth

¹³Such methods are proven in time-stepping procedures for ordinary initial value problems because of their stability properties. Radau IIA methods are also used for many DAEs with index $\mu \leq 2$ since the generated approximations on the grid points satisfy the obvious constraint.

Table 4 Error of the approximate solution using Legendre basis functions and Gauss-Legendre (G), Radau IIA (R), and Lobatto (L) collocation nodes when using the functional $\Phi_{\pi, M}^R$ in Example 3. The norm is that of $H_D^1((0, 5), \mathbb{R}^7)$

n	$N = 3$			$N = 5$			$N = 10$			$N = 20$		
	G	R	L	G	R	L	G	R	L	G	R	L
5	5.37e-03	5.86e-03	5.55e-03	1.37e-05	1.52e-05	1.38e-05	3.41e-12	4.08e-12	3.61e-12	8.97e-11	5.31e-11	1.04e-10
10	2.15e-03	2.33e-03	2.20e-03	1.68e-06	1.77e-06	1.69e-06	3.98e-11	2.53e-11	2.51e-11	4.78e-10	7.58e-10	1.00e-09
20	9.95e-04	1.04e-03	1.00e-03	2.08e-07	2.14e-07	2.08e-07	2.04e-10	2.53e-10	1.80e-10	3.18e-09	2.97e-09	3.30e-09
40	4.80e-04	4.91e-04	4.81e-04	2.58e-08	2.62e-08	2.58e-08	1.34e-09	1.67e-09	1.64e-09	2.56e-08	2.31e-08	3.29e-08
80	2.36e-04	2.39e-04	2.36e-04	3.34e-09	3.32e-09	3.63e-09	1.19e-08	1.33e-08	1.60e-08	1.99e-07	2.03e-07	2.14e-07
160	1.17e-04	1.17e-04	1.17e-04	9.16e-09	9.16e-09	1.18e-08	8.66e-08	1.01e-07	1.13e-07	1.74e-06	1.45e-06	1.94e-06
320	5.81e-05	5.83e-05	5.81e-05	8.06e-08	6.79e-08	8.74e-08	7.90e-07	8.25e-07	9.82e-07	1.39e-05	1.27e-05	1.38e-05

Table 5 Error of the approximate solution using Legendre basis functions and Gauss-Legendre (G), Radau IIA (R), and Lobatto (L) collocation nodes when using the functional $\Phi_{\pi, M}^C$ in Example 3. The norm is that of $H_D^1((0, 5), \mathbb{R}^7)$

n	$N = 3$			$N = 5$			$N = 10$			$N = 20$		
	G	R	L	G	R	L	G	R	L	G	R	L
5	5.22e-03	7.20e-03	7.81e-03	1.30e-05	1.50e-05	1.44e-05	2.89e-12	4.32e-12	1.82e-12	5.15e-11	3.67e-11	4.24e-11
10	2.06e-03	2.85e-03	3.46e-03	1.59e-06	1.75e-06	1.76e-06	3.24e-11	1.95e-11	1.79e-11	3.23e-10	1.57e-10	1.91e-10
20	9.49e-04	1.27e-03	1.67e-03	1.96e-07	2.11e-07	2.19e-07	2.19e-10	1.66e-10	1.06e-10	2.39e-09	1.55e-09	7.72e-10
40	4.58e-04	6.04e-04	8.27e-04	2.42e-08	2.60e-08	2.73e-08	1.59e-09	1.01e-09	7.38e-10	1.80e-08	1.65e-08	4.39e-09
80	2.25e-04	2.95e-04	4.12e-04	3.12e-09	3.51e-09	3.58e-09	1.16e-08	8.67e-09	5.25e-09	1.45e-07	1.41e-07	2.56e-08
160	1.11e-04	1.46e-04	2.06e-04	9.57e-09	1.01e-08	8.23e-09	9.33e-08	7.27e-08	3.96e-08	1.10e-06	1.20e-06	1.52e-07
320	5.54e-05	7.24e-05	1.03e-04	7.95e-08	8.73e-08	7.13e-08	7.47e-07	5.86e-07	3.33e-07	8.82e-06	9.78e-06	1.11e-06

problems, a higher degree N is preferable over a larger n or, equivalently, a smaller stepsize h . In addition, for the highest degree polynomials ($N = 20$), the use of $\Phi_{\pi, M}^C$ seems to produce more accurate results than that of $\Phi_{\pi, M}^R$.

5 Final remarks and conclusions

In summary, in the present paper, we investigated questions related to an efficient and reliable realization of a least-squares collocation method. These questions are particularly important since a higher index DAE is an essentially ill-posed problem in naturally given spaces, which is why we must be prepared for highly sensitive discrete problems. In order to obtain an overall procedure that is as robust as possible, we provided criteria which led to a robust selection of the collocation points and of the basis functions, whereby the latter is also useful for the shape of the resulting discrete problem. Additionally, a number of new, more detailed, error estimates have been given that support some of the design decisions. The following particular items are worth highlighting in this context:

- The basis for the approximation space should be appropriately shifted and scaled orthogonal polynomials. We could not observe any larger differences between the behavior of Legendre and Chebyshev polynomials.
- The collocation points should be chosen to be the Gauss-Legendre, Lobatto, or Radau nodes. This leads to discrete problems whose conditioning using the discretization by interpolation ($\Phi_{\pi, M}^R$) is not much worse than that resembling collocation methods for ordinary differential equations ($\Phi_{\pi, M}^C$). A particular efficient and stable implementation is obtained if Gauss-Legendre or Radau nodes are used since, in this case, diagonal weighting ($\Phi_{\pi, M}^I$) coincides with the interpolation approach.
- A critical ingredient for the implementation of the method is the algorithm used for the solution of the constrained linear least-squares problems. Given the expected bad conditioning of the least-squares problem, a QR factorization with column pivoting must lie at the heart of the algorithm. At the same time, the sparsity structure must be used as best as possible. This issue will be discussed in Part 2.
- It seems as if, for problems with a smooth solution, a higher degree N of the ansatz polynomials with a low number of subintervals n in the mesh is preferable over a smaller degree with a larger number of subintervals with respect to accuracy. Some first theoretical justification has been provided for this claim.
- So far, in all experiments of this and previously published papers, we did not observe any serious differences in the accuracy obtained in dependence on the choice of $M > N$ for fixed n . The results for $M = N + 1$ are not much different from those obtained for a larger M .
- While superconvergence in classical collocation for ODEs and index-1 DAEs is a very favorable phenomenon, we could not find anything analogous in our experiments.

- The simple collocation procedure using $\Phi_{\pi,M}^C$ performs surprisingly well. In fact, the results are, in our experiments, in par with those using $\Phi_{\pi,M}^R = \Phi_{\pi,M}^I$. However, we have no theoretical justification for this as yet.
- Our method is designed for variable grids. However, so far we have only worked with constant step size. In order to be able to adapt the grid and the polynomial degree, or even select appropriate grids, it is important to understand the structure of the error, that is, how the global error depends on local errors. This is a very important open problem, for which we have no solution yet.

In conclusion, we note that earlier implementations, among others the one from the very first paper in this matter [28] which started from proven ingredients for ODE codes, are from today’s point of view and experience a rather bad version for the least-squares collocation. Nevertheless, the test results calculated with it were already very impressive. This strengthens our belief that a careful implementation of the method will give rise to a very efficient solver for higher index DAEs.

Appendix: Some facts about classical orthogonal polynomials

In the derivations, classical orthogonal polynomials have been heavily used. For the reader’s convenience important properties are collected below.

A.1 Legendre polynomials

The Legendre polynomials P_ν , $\nu = 0, 1, \dots$, are defined by the recurrence relation

$$\begin{aligned} P_0(\tau) &= 1, \\ P_1(\tau) &= \tau, \\ (\nu + 1)P_{\nu+1}(\tau) &= (2\nu + 1)\tau P_\nu(\tau) - \nu P_{\nu-1}(\tau), \quad \nu = 1, 2, \dots \end{aligned} \tag{29}$$

Some properties of the Legendre polynomials are

1. $P_\nu(-1) = (-1)^\nu$, $P_\nu(1) = 1$, $\nu = 0, 1, \dots$,
2. $\int_{-1}^1 P_0(\tau) d\tau = 2$, $\int_{-1}^1 P_\nu(\tau) d\tau = 0$, $\nu = 1, 2, \dots$
3. $\int_{-1}^1 P_\nu(\tau) P_\mu(\tau) d\tau = \frac{2}{2\nu+1} \delta_{\nu\mu}$, $\nu, \mu = 0, 1, \dots$, where $\delta_{\nu\mu}$ denotes the Kronecker δ -symbol,
4. $P'_{\nu+1}(\tau) - P'_{\nu-1}(\tau) = (2\nu + 1)P_\nu(\tau)$, $\nu = 1, 2, \dots$

The latter property is useful for representing integrals,

$$\begin{aligned} \int_{-1}^\tau P_\nu(\sigma) d\sigma &= \frac{1}{2\nu + 1} \left(P_{\nu+1}(\tau) - P_{\nu-1}(\tau) - (-1)^{\nu+1} + (-1)^{\nu-1} \right) \\ &= \frac{1}{2\nu + 1} (P_{\nu+1}(\tau) - P_{\nu-1}(\tau)). \end{aligned} \tag{30}$$

Moreover, $\int_{-1}^\tau P_0(\sigma) d\sigma = \tau + 1$.

For a stable evaluation of the Legendre polynomials, we use a representation proposed in [35],

$$P_{\nu+1}(\tau) = \frac{\nu}{\nu+1}(\tau P_{\nu}(\tau) - P_{\nu-1}(\tau)) + \tau P_{\nu}(\tau).$$

In the implementation, all polynomials must be evaluated simultaneously for each given τ . The evaluation of the recursions is cheap. Linear combinations of the basis function can be conveniently and stably evaluated using the Clenshaw algorithm [13, p. 56] [4, 39].

The shifted Legendre polynomials \tilde{P}_{ν} are given by $\tilde{P}_{\nu}(\rho) = P_{\nu}(2\rho - 1)$, $\nu = 0, 1, \dots$. They fulfill the orthogonality relations

$$\int_0^1 \tilde{P}_{\nu}(\rho)\tilde{P}_{\mu}(\rho)d\rho = \frac{1}{2\nu+1}\delta_{\nu\mu}.$$

Moreover, we introduce the normalized shifted Legendre polynomials \hat{P}_{ν} by

$$\hat{P}_{\nu}(\rho) = (2\nu+1)^{1/2}\tilde{P}_{\nu}(\rho).$$

A.2 Chebyshev polynomials

The Chebyshev polynomials of the first kind T_{ν} , $\nu = 0, 1, \dots$, are defined by the recurrence relation

$$\begin{aligned} T_0(\tau) &= 1, \\ T_1(\tau) &= \tau, \\ T_{\nu+1}(\tau) &= 2\tau T_{\nu}(\tau) - T_{\nu-1}(\tau), \quad \nu = 1, 2, \dots \end{aligned} \tag{31}$$

Some properties of the Chebyshev polynomials are

1. $T_{\nu}(-1) = (-1)^{\nu}$, $T_{\nu}(1) = 1$, $\nu = 0, 1, \dots$
2. $T_{\nu}(\tau) = \frac{1}{2} \left(\frac{1}{\nu+1} T'_{\nu+1}(\tau) - \frac{1}{\nu-1} T'_{\nu-1}(\tau) \right)$, $\nu = 2, 3, \dots$

Similarly as before, we obtain the simple representation

$$\int_{-1}^{\tau} T_{\nu}(\sigma)d\sigma = \frac{1}{2(\nu^2-1)} \left((\nu-1)T_{\nu+1}(\tau) - (\nu+1)T_{\nu-1}(\tau) + (-1)^{\nu-1}2 \right). \tag{32}$$

The orthogonality property of the Chebyshev polynomials reads

$$\int_{-1}^1 T_{\nu}(\tau)T_{\mu}(\tau)\frac{d\tau}{\sqrt{1-x^2}} = \begin{cases} 0, & \nu \neq \mu, \\ \pi, & \nu = \mu = 0, \\ \frac{\pi}{2}, & \nu = \mu \neq 0. \end{cases}$$

The normalized Chebyshev polynomials \bar{T}_{ν} are given by

$$\bar{T}_{\nu}(\tau) = \begin{cases} \sqrt{\frac{1}{\pi}}T_0(\tau), & \nu = 0, \\ \sqrt{\frac{2}{\pi}}T_{\nu}(\tau), & \nu = 1, 2, \dots \end{cases}$$

Linear combinations of Chebyshev polynomials can be stably computed by the Clenshaw algorithm [13, p. 57ff], [4, 39].

Funding Open access funding provided by Royal Institute of Technology.

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. Albasiny, E.L. In: Childs, B., Scott, M., Daniel, J.W., Denman, E., Nelson, P. (eds.): A Subroutine for Solving a System of Differential Equations in Chebyshev Series, pp. 280–286. Springer-Verlag, Berlin Heidelberg New York (1979)
2. Ascher, U., Bader, G.: A new basis implementation for a mixed order boundary-value ode solver. *SIAM J. Sci. Statist. Comput.* **8**, 483–500 (1987)
3. Ascher, U., Spiteri, R.: Collocation software for boundary-value differential-algebraic equations. *SIAM J. Sci. Comput.* **15**, 938–952 (1994)
4. Barrio, R.: Rounding error bounds for Clenshaw and Forsythe algorithms for the evaluation of orthogonal polynomial series. *J. Comp. Appl. Math.* **138**, 185–204 (2002)
5. Beckermann, B.: The condition number of real Vandermonde, Krylov and positive definite Hankel matrices. *Numer. Math.* **85**, 553–577 (2000)
6. Campbell, S.L., Moore, E.: Constraint preserving integrators for general nonlinear higher index DAEs. *Num. Math.* **69**, 383–399 (1995)
7. Davis, T.A.: *Direct Methods for Sparse Linear Systems. Fundamentals of Algorithms.* SIAM, Philadelphia (2006)
8. Davis, T.A.: Algorithm 915, SuiteSparseQR: Multifrontal multithreaded rank-revealing sparse QR factorization. *ACM Trans. Math. Softw.* **38**(1), 8:1–8:22 (2011)
9. Deuffhard, P., Hohmann, A.: *Numerische Mathematik 1. Eine algorithmisch orientierte Einführung.* De Gruyter, Berlin, Boston, 5 auflage edition (2019)
10. Driscoll, T.A., Hale, N., Trefethen, L.N.: *Chebfun Guide.* Pafnuty Publications, Oxford (2014)
11. Fornberg, B.: *A Practical Guide to Pseudospectral Methods.* Cambridge University Press, Cambridge (1996)
12. Fornberg, B., Sloan, D.M.: A review of pseudospectral methods for solving partial differential equations. *Acta Numerica* **3**(1), 203–267 (1994)
13. Fox, L., Parker, I.B.: *Chebyshev Polynomials in Numerical Analysis.* Oxford Mathematical Handbooks. Oxford University Press, London (1968)
14. Galassi, M., et al: *GNU Scientific Library Reference Manual.* Network Theory Ltd. 3rd edition edition. Version 2.6. (2009)
15. Gautschi, W.: The condition of Vandermonde-like matrices involving orthogonal polynomials. *Linear Algebra Appl.* **52**(/53), 293–300 (1983)
16. Gautschi, W.: Gauss-Radau formulae for Jacobi and Laguerre weight functions. *Math. Comput. Simul.* **54**, 403–412 (2000)
17. Gautschi, W.: High-order Gauss-Lobatto formulae. *Numer. Algorithm.* **25**, 213–222 (2000)
18. Gautschi, W.: Optimally scaled and optimally conditioned Vandermonde and Vandermonde-like matrices. *BIT Numer. Math.* **51**, 103–125 (2011)
19. Gautschi, W., Inglese, G.: Lower bounds for the condition number of Vandermonde matrices. *Numer. Math.* **52**, 241–250 (1988)
20. Gladwell, I.: The development of the boundary-value codes in the ordinary differential equations chapter of the NAG library. In: Childs, B., Scott, M., Daniel, J.W., Denman, E., Nelson, P. (eds.) *Codes for Boundary-Value Problems in Ordinary Differential Equations*, volume 76 of *Lecture Notes in Computer Science*, pp. 122–143. Springer-Verlag, Berlin, Heidelberg New York (1979)
21. Golub, G.H., Welsch, J.H.: Calculation of Gauss quadrature rules. *Math. Comp.* **23**, 221–230 (1969)

22. Guennebaud, G., Jacob, B., et al.: Eigen v3. <http://eigen.tuxfamily.org> (2010)
23. Hämmerlin, G., Hoffman, K.-H.: Numerical Mathematics. Springer Verlag, New York (1991)
24. Hanke, M., März, R.: Towards a reliable implementation of least-squares collocation for higher-index linear differential-algebraic equations. Part 2: The discrete least-squares problem. Numerical Algorithms. submitted
25. Hanke, M., März, R.: Convergence analysis of least-squares collocation methods for nonlinear higher-index differential-algebraic equations. J. Comp. Appl. Math. **387**, 112514 (2021)
26. Hanke, M., März, R.: A reliable direct numerical treatment of differential-algebraic equations by overdetermined collocation An operator approach. J. Comp. Appl. Math. **387**, 112520 (2021)
27. Hanke, M., März, R., Tischendorf, C.: Least-squares collocation for higher-index linear differential-algebraic equations: Estimating the stability threshold. Math. Comp. **88**(318), 1647–1683 (2019). <https://doi.org/10.1090/mcom/3393>
28. Hanke, M., März, R., Tischendorf, C., Weinmüller, E., Wurm, S.: Least-squares collocation for linear higher-index differential-algebraic equations. J. Comput. Appl. Math. **317**, 403–431 (2017). <https://doi.org/10.1016/j.cam.2016.12.017>
29. Hildebrand, F.B.: Introduction to Numerical Analysis. McGraw-Hill, New York (1956)
30. Ibrahimoglu, B.A.: Lebesgue functions constants in polynomial interpolation. J. Lebesgue Inequal. Appl. **93**, 1–15 (2016)
31. Kitzhofer, G., Koch, O., Pulverer, G., Simon, C.h., Weinmüller, E.: BVPSUITE a new MATLAB solver for singular implicit boundary value problems. ASC Report 35 Vienna University of Technology (2009)
32. Koch, O., März, R., Praetorius, D., Weinmüller, E., et al: Collocation methods for index-1 daes with singularities of the first kind. Math. Comp. **79**, 281–304 (2010)
33. Lamour, R., März, R., C.Tischendorf. In: Ilchmann, A., Reis, T. (eds.): Differential-Algebraic Equations: A Projector Based Analysis. Differential-Algebraic Equations Forum. Springer-Verlag, Berlin Heidelberg New York Dordrecht London (2013)
34. Lamour, R., März, R., Weinmüller, E. In: Ilchmann, A., Reis, T. (eds.): Surveys in Differential-Algebraic Equations III, chapter Boundary-Value Problems for Differential-Algebraic Equations: A Survey: Differential-Algebraic Equations Forum, pp. 177–309. Springer Heidelberg, New York (2015)
35. Lebedev, V.-I., Barburin, O.V.: Calculation of principal values, weights and nodes of the Gauss quadrature formulae of integrals. U.S.S.R Comput. Math. Math. Phys. **5**(3), 81–92 (1965)
36. März, R. In: Ilchmann, A., Reis, T. (eds.): Surveys in Differential-Algebraic Equations II, chapter Differential-Algebraic Equations from a Functional-Analytic Viewpoint: A Survey: Differential-Algebraic Equations Forum, pp. 163–285. Springer Heidelberg, New York (2015)
37. Michaels, H.H.: Abscissas weight coefficients for Lobatto quadrature. Math Comp **17**, 237–244 (1963)
38. Robbins, H.: A remark on Stirling’s formula. Amer. Math. Monthly **62**(1), 26–29 (1955)
39. Smoktunowicz, A.: Backward stability of Clenshaw’s algorithm. BIT Numer. Math. **42**(3), 600–610 (2002)
40. Stallman, R.M., GCC Developers Community, et al: Using the Gnu Compiler collection. CreateSpace, Scotts Valley (2009)
41. Stoer, J.: Numerische Mathematik 1. Springer-Lehrbuch. Springer-Verlag, Berlin, Heidelberg, New York, London, Paris Tokyo HongKong (1989). fünfte, verbesserte auflage edition
42. The Numerical Algorithms Group (NAG). The nag library for fortran (2019)
43. Trefethen, L.N.: Spectral methods in MATLAB. SIAM, Philadelphia (2000)
44. Trefethen, L.N., Weideman, J.A.C.: Two results on polynomial interpolation in equally spaced points. J Approx. Theory **65**, 247–260 (1991)