Numerical Algorithms (2022) 89:965–986
https://doi.org/10.1007/s11075-021-01141-6

ORIGINAL PAPER

Towards a reliable implementation of least-squares collocation for higher index differential-algebraic equations—Part 2: the discrete least-squares problem

Michael Hanke[1] · **Roswitha März**[2]

Received: 14 May 2021 / Accepted: 16 May 2021 / Published online: 15 June 2021

Abstract
In the two parts of the present note we discuss 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. We provide in Part 1 a robust selection of basis functions and collocation points to design the discrete problem whereas we analyze the discrete least-squares problem and substantiate a procedure for its numerical solution in Part 2.

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

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

1 Introduction

This is Part 2 of our work entitled *Towards a reliable implementation of least-squares collocation for higher index differential-algebraic equations*, which is introduced and classified in detail in Part 1. We put together here very briefly the necessary ingredients for fluent reading of the current second part.

✉ 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

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

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

$$G_a x(a) + G_b x(b) = d. \tag{2}$$

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, $G_a, G_b \in \mathbb{R}^{l_{dyn}\times m}$. Thereby, l_{dyn} 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. We assume further that $\ker D \subseteq \ker G_a$ and $\ker D \subseteq \ker G_b$.

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

Let $\mathfrak{P}_K$ denote the set of all polynomials of degree less than or equal to $K \ge 0$.

Given the partition π,

$$\pi : \quad a = t_0 < t_1 < \cdots < t_n = b, \tag{3}$$

with the stepsizes $h_j = t_j - t_{j-1}$, 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 $N \ge 1$ be a fixed integer. We are looking for an approximate solution of our boundary value problem from the ansatz space X_π,

$$\begin{aligned} X_\pi = \{&x \in C_\pi([a,b],\mathbb{R}^m) : Dx \in C([a,b],\mathbb{R}^k),\\ &x_\kappa|_{[t_{j-1},t_j)} \in \mathfrak{P}_N,\ \kappa = 1,\dots,k, \quad x_\kappa|_{[t_{j-1},t_j)} \in \mathfrak{P}_{N-1},\ \kappa = k+1,\dots,m,\ j = 1,\dots,n\}. \end{aligned} \tag{4}$$

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

$$\boldsymbol{\Phi}(x) = \int_a^b |A(t)(Dx)'(t) + B(t)x(t) - q(t)|^2 \mathrm{d}t + |G_a x(a) + G_b x(b) - d|^2. \tag{5}$$

Hanke and März [11, Theorem 1] provides sufficient conditions ensuring the existence and uniqueness of the approximate solution from X_π.

The functional values $\boldsymbol{\Phi}(x)$, 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, perturbations of the functional may have a serious influence on the error of the approximate least-squares solution or even prevent convergence towards the exact solution. Therefore, careful approximations of the integral in $\boldsymbol{\Phi}$ are required. We take over the options provided in [11], in which $M \ge N+1$ so-called collocation points

$$0 \le \tau_1 < \cdots < \tau_M \le 1. \tag{6}$$

are used, further, on the subintervals of the partition π,

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

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

$$W = \begin{bmatrix} W_1 \\ \vdots \\ W_n \end{bmatrix} \in \mathbb{R}^{mMn}, \quad W_j = h_j^{1/2} \begin{bmatrix} w(t_{j1}) \\ \vdots \\ w(t_{jM}) \end{bmatrix} \in \mathbb{R}^{mM}, \tag{7}$$

we turn to an approximate functional of the form

$$\boldsymbol{\Phi}_{\pi,M}(x) = W^T \mathscr{L} W + |G_a x(a) + G_b x(b) - d|^2, \quad x \in X_\pi, \tag{8}$$

with a positive definite symmetric matrix[1]

$$\mathscr{L} = \operatorname{diag}(L \otimes I_m, \ldots, L \otimes I_m). \tag{9}$$

As detailed in [11], we have different options for the positive definite symmetric matrix $L \in \mathbb{R}^{M \times M}$, namely

$$L = L^C = M^{-1} I_M, \tag{10}$$

$$L = L^I = \operatorname{diag}(\gamma_1, \ldots, \gamma_M), \tag{11}$$

$$L = L^R = (\tilde{V}^{-1})^T \tilde{V}, \tag{12}$$

see [11, Section 3] for details concerning the selection of the quadrature weights and the construction of the mass matrix. We emphasize that the matrices L^C, L^I, L^R depend only on M, the node sequence (6), and the quadrature weights, but do not depend on the partition π and its stepsizes at all.

In the context of the experiments below, we denote each of the different versions of the functional by $\boldsymbol{\Phi}^C_{\pi,M}$, $\boldsymbol{\Phi}^I_{\pi,M}$, and $\boldsymbol{\Phi}^R_{\pi,M}$, respectively.

It should be underlined that minimizing each version of the functional $\boldsymbol{\Phi}_{\pi,M}$ on X_π can be viewed as a special least-squares method to solve the *overdetermined collocation system* $W = 0$, $G_a x(a) + G_b x(b)) = d$, with respect to $x \in X_\pi$, that is in detail, the collocation system

$$A(t_{ji})(Dx)'(t_{ji}) + B(t_{ji})x(t_{ji}) = q(t_{ji}), \quad i = 1, \ldots, M, \quad j = 1, \ldots, n, \tag{13}$$

$$G_a x(a) + G_b x(b)) = d. \tag{14}$$

The system (13)–(14) for $x \in X_\pi$ becomes overdetermined since X_π has dimension $mnN + k$, whereas the system consists of $mnM + l_{dyn} > nmN + k + l_{dyn} \geq mnN + k$ scalar equations. We refer to [11, Theorem 2] for sufficient conditions which ensure the existence and uniqueness of the minimizing element

$$x_\pi = \operatorname{argmin}\{\boldsymbol{\Phi}_{\pi,M}(x) : x \in X_\pi\}. \tag{15}$$

Once the basis of the ansatz space X_π has been chosen and the collocation nodes are selected, the discrete problem (15) for a linear boundary value problem (1)–(2) leads to a constrained linear least-squares problem

$$\varphi(c) = |\mathscr{A} c - r|^2_{\mathbb{R}^{nmM+l_{dyn}}} \to \min! \tag{16}$$

[1] $\otimes$ denoting the Kronecker product.

under the linear constraint

$$\mathscr{C}c = 0. \tag{17}$$

The equality constraints consists of the $k(n-1)$ continuity conditions for the elements of X_π while the functional $\varphi(c)$ represents a reformulation of the functional (8). Here, $c \in \mathbb{R}^{n(mN+k)}$ is the vector of coefficients of the basis functions for X_π disregarding the continuity conditions. Furthermore, it holds $r \in \mathbb{R}^{nmM+l}$, $\mathscr{A} \in \mathbb{R}^{(nmM+l)\times n(mN+k)}$, and $\mathscr{C} \in \mathbb{R}^{(n-1)k\times n(mN+k)}$. The matrices $\mathscr{A}$ and $\mathscr{C}$ are very sparse. Owing to the construction, $\mathscr{C}$ has full row rank.

We specify the structure of $\mathscr{A}$ and $\mathscr{C}$ in detail in Section 2 below. Different approaches to solve the constraint optimization problem (16)–(17) have been tested. We report on related experiments in Section 4. The examples which are used on different places are collected in the particular Section 3. The performance of the linear solver is discussed in Section 5. Section 6 shows some additional experiments concerning the boundary conditions weighting. Section 7 contains final remarks.

2 The structure of the discrete problem (16), (17)

Based on the analysis in [11, Section 4] we provide a basis of the ansatz space X_π to begin with. Assume that $\{p_0, \ldots, p_{N-1}\}$ is a basis of $\mathfrak{P}_{N-1}$ defined on the reference interval $[0, 1]$. Then, $\{\bar{p}_0, \ldots, \bar{p}_N\}$ given by

$$\bar{p}_i(\rho) = \begin{cases} 1, & i = 0, \\ \int_0^\rho p_{i-1}(\sigma)\mathrm{d}\sigma, & i = 1, \ldots, N, \end{cases} \quad \rho \in [0, 1], \tag{18}$$

forms a basis of $\mathfrak{P}_N$. The transformation to the interval (t_{j-1}, t_j) of the partition π (3) yields

$$p_{ji}(t) = p_i((t - t_{j-1})/h_j), \quad \bar{p}_{ji}(t) = h_j \bar{p}_i((t - t_{j-1})/h_j). \tag{19}$$

and in particular

$$\bar{p}_{ji}(t_{j-1}) = h_j \bar{p}_i(0) = h_j \begin{cases} 1, & i = 0, \\ 0, & i = 1, \ldots, N, \end{cases}$$

$$\bar{p}_{ji}(t_j) = h_j \bar{p}_i(1) = h_j \begin{cases} 1, & i = 0, \\ \int_0^1 p_{i-1}(\sigma)\mathrm{d}\sigma, & i = 1, \ldots, N. \end{cases}$$

Next we form the matrix functions

$$\bar{\mathscr{P}}_j = \begin{bmatrix} \bar{p}_{j0} & \ldots & \bar{p}_{jN} \end{bmatrix} : I_j \to \mathbb{R}^{1\times(N+1)}, \quad \mathscr{P}_j = \begin{bmatrix} p_{j0} & \ldots & p_{j,N-1} \end{bmatrix} : I_j \to \mathbb{R}^{1\times N},$$

such that

$$\bar{\mathscr{P}}_j(t_{j-1}) = h_j \begin{bmatrix} 1 & 0 & \ldots & 0 \end{bmatrix}, \quad j = 1, \ldots, n, \tag{20}$$

$$\bar{\mathscr{P}}_j(t_j) = h_j \begin{bmatrix} 1 & \int_0^1 p_0(\sigma)\mathrm{d}\sigma & \ldots & \int_0^1 p_{N-1}(\sigma)\mathrm{d}\sigma \end{bmatrix}, \quad j = 1, \ldots, n. \tag{21}$$

Observe that choosing $\{p_0, \ldots, p_{N-1}\}$ to be Legendre polynomials simplifies the latter matrix to

$$\bar{\mathscr{P}}_j(t_j) = h_j \begin{bmatrix} 1 & 1 & 0 & \ldots & 0 \end{bmatrix}, \quad j = 1, \ldots, n,$$

which will prove important.

For $x \in X_\pi$ we set the denotations

$$x(t) = x_j(t) = \begin{bmatrix} x_{j1}(t) \\ \vdots \\ x_{jm}(t) \end{bmatrix} \in \mathbb{R}^m, \quad Dx(t) = Dx_j(t) = \begin{bmatrix} x_{j1}(t) \\ \vdots \\ x_{jk}(t) \end{bmatrix} \in \mathbb{R}^k, \quad t \in I_j.$$

Then, we develop each x_j componentwise

$$\begin{aligned} x_{j\kappa}(t) &= \sum_{l=0}^{N} c_{j\kappa l} \bar{p}_{jl}(t) = \bar{\mathscr{P}}_j(t) c_{j\kappa}, \quad \kappa = 1, \dots, k, \\ x_{j\kappa}(t) &= \sum_{l=0}^{N-1} c_{j\kappa l} p_{jl}(t) = \mathscr{P}_j(t) c_{j\kappa}, \quad \kappa = k+1, \dots, m. \end{aligned}$$

with

$$c_{j\kappa} = \begin{bmatrix} c_{j\kappa 0} \\ \vdots \\ c_{j\kappa N} \end{bmatrix} \in \mathbb{R}^{N+1}, \quad \kappa = 1, \dots, k, \quad c_{j\kappa} = \begin{bmatrix} c_{j\kappa 0} \\ \vdots \\ c_{j\kappa, N-1} \end{bmatrix} \in \mathbb{R}^{N}, \quad \kappa = k+1, \dots, m.$$

Introducing still

$$\Omega_j(t) = \begin{bmatrix} I_k \otimes \bar{\mathscr{P}}_j(t) & 0 \\ 0 & I_{m-k} \otimes \mathscr{P}_j(t) \end{bmatrix} \in \mathbb{R}^{m \times (mN+k)}, \quad c_j = \begin{bmatrix} c_{j1} \\ \vdots \\ c_{jm} \end{bmatrix} \in \mathbb{R}^{mN+k},$$

we represent

$$x_j(t) = \Omega_j(t) c_j, \tag{22}$$

$$Dx_j(t) = D\Omega_j(t) c_j = \begin{bmatrix} I_k \otimes \bar{\mathscr{P}}_j(t) & 0 \end{bmatrix} c_j, \quad t \in I_j, \quad j = 1, \dots, n. \tag{23}$$

Now we collect all coefficients $c_{j\kappa l}$ in the vector c,

$$c = \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix} \in \mathbb{R}^{nmN+nk}.$$

It follows that the matrix $\mathscr{C} \in \mathbb{R}^{k(n-1) \times n(mN+k)}$ in (17) corresponding to the continuity requirement for Dx has the precise form

$$\mathscr{C} = \begin{bmatrix} I_k \otimes \bar{\mathscr{P}}_1(t_1) & -I_k \otimes \bar{\mathscr{P}}_2(t_1) & & & \\ & I_k \otimes \bar{\mathscr{P}}_2(t_2) & -I_k \otimes \bar{\mathscr{P}}_3(t_2) & & \\ & \ddots & \ddots & & \\ & & & \ddots & \ddots \\ & & & I_k \otimes \bar{\mathscr{P}}_{n-1}(t_{n-1}) & -I_k \otimes \bar{\mathscr{P}}_n(t_{n-1}) \end{bmatrix}. \tag{24}$$

By construction the segments Dx_j, $j = 1, \dots, n$, all together form a continuous function Dx on $[a, b]$ exactly when $\mathscr{C} c = 0$.

Regarding the structure of $\mathscr{C} \in \mathbb{R}^{k(n-1)\times n(mN+k)}$ we know that

$$\operatorname{rank}\mathscr{C} = k(n-1), \quad \dim\ker\mathscr{C} = nmN + k = \dim X_\pi,$$

and formula (22) provides an one-to-one relation between X_π and $\ker\mathscr{C} \subset \mathbb{R}$.

Now we turn to the detailed description of the functional value (16). For this aim we factorize $L = \tilde{L}^T\tilde{L}$ and $\mathscr{L} = \tilde{\mathscr{L}}^T\tilde{\mathscr{L}}$ such that

$$\tilde{\mathscr{L}} = \operatorname{diag}(\tilde{L}\otimes I_m, \cdots, \tilde{L}\otimes I_m)$$

and (8) rewrites as

$$\boldsymbol{\Phi}_{\pi,M}(x) = |\tilde{\mathscr{L}}W|^2_{\mathbb{R}^{nmM}} + |G_a x(a) + G_b x(b) - d|^2_{\mathbb{R}^{d_{dyn}}}, \quad x \in X_\pi.$$

We derive applying (22), (23)

$$G_a x(a) + G_b x(b) = G_a D^+ D\Omega_1(t_0)c_1 + G_b D^+ D\Omega_n(t_n)c_n =: \Gamma_a c_1 + \Gamma_b c_n$$

with matrices $\Gamma_a, \Gamma_b \in \mathbb{R}^{l_{dyn}\times(mN+k)}$, and

$$w(t_{ji}) = \underbrace{\left[A(t_{ji})(D\Omega_j)'(t_{ji}) + B(t_{ji})\Omega_j(t_{ji})\right]}_{=\mathscr{A}_{ji}} c_j - q(t_{ji}) = \mathscr{A}_{ji}c_j - q(t_{ji}),$$

with $\mathscr{A}_{ji} \in \mathbb{R}^{m\times(mN+k)}$. According to (7) we set

$$W_j = h_j^{1/2}\begin{bmatrix} w(t_{j1}) \\ \vdots \\ w(t_{jM}) \end{bmatrix} = h_j^{1/2}\begin{bmatrix} \mathscr{A}_{j1} \\ \vdots \\ \mathscr{A}_{jM} \end{bmatrix} c_j - h_j^{1/2}\begin{bmatrix} q(t_{j1}) \\ \vdots \\ q(t_{jM}) \end{bmatrix}$$

and

$$(\tilde{L}\otimes I_m)W_j = \underbrace{h_j^{1/2}(\tilde{L}\otimes I_m)\begin{bmatrix} \mathscr{A}_{j1} \\ \vdots \\ \mathscr{A}_{jM} \end{bmatrix}}_{=:\mathscr{A}_j} c_j - (\tilde{L}\otimes I_m)\underbrace{h_j^{1/2}\begin{bmatrix} q(t_{j1}) \\ \vdots \\ q(t_{jM}) \end{bmatrix}}_{=:W_j^{[q]}} = \mathscr{A}_j c_j - (\tilde{L}\otimes I_m)W_j^{[q]}.$$

Introducing still the sparse matrix $\mathscr{A} \in \mathbb{R}^{(nmM+l_{dyn})\times(nmN+nk)}$ and the vector $r \in \mathbb{R}^{nmM+l_{dyn}}$,

$$\mathscr{A} = \begin{bmatrix} \mathscr{A}_1 & 0 & \cdots & & 0 \\ 0 & \ddots & & & \vdots \\ \vdots & & \ddots & & \\ & & & \ddots & 0 \\ 0 & & & & \mathscr{A}_n \\ \Gamma_a & 0 & \cdots & 0 & \Gamma_b \end{bmatrix}, \quad r = \begin{bmatrix} (\tilde{L}\otimes I_m)W_1^{[q]} \\ \vdots \\ (\tilde{L}\otimes I_m)W_n^{[q]} \end{bmatrix}$$

we arrive at the representation

$$\begin{aligned}\varphi(c) &= |\mathscr{A}c - r|^2_{\mathbb{R}^{nmM+l_{dyn}}} = \sum_{j=1}^{n} |\mathscr{A}_j c_j - (\tilde{L}\otimes I_m)W_j^{[q]}|^2_{\mathbb{R}^{mM}} + |\Gamma_a c_1 + \Gamma_b c_n - d|^2_{\mathbb{R}^{l_{dyn}}} \\ &= \sum_{j=1}^{n} |(\tilde{L}\otimes I_m)W_j|^2_{\mathbb{R}^{mM}} + |\Gamma_a c_1 + \Gamma_b c_n - d|^2_{\mathbb{R}^{l_{dyn}}} = |\tilde{\mathscr{L}}W|^2_{\mathbb{R}^{nmM}} + |\Gamma_a c_1 + \Gamma_b c_n - d|^2_{\mathbb{R}^{l_{dyn}}} \\ &= \boldsymbol{\Phi}_{\pi,M}(x),\end{aligned}$$

as desired. Eventually, each minimizer $x_\pi \in \operatorname{argmin}\{\boldsymbol{\Phi}_{\pi,M}(x) : x \in X_\pi\}$ corresponds to a minimizer $c_{min} \in argmin\{\varphi(c) : c \in \mathbb{R}^{nmN+nk}, \mathscr{C}c = 0\}$, and vice versa. Recall that [11, Theorem 2] provides sufficient condition for x_π to be unique.

Proposition 1 *Let the functional $\boldsymbol{\Phi}_{\pi,M}$ have the only minimizer x_π on X_π. Then the following assertions are valid:*

(1) *There is exactly one minimizer c_{min} of the functional φ on $\ker\mathscr{C}$.*
(2) *If $\mathscr{B} \in \mathbb{R}^{(nmN+nk)\times(nmN+k)}$ is a basis of $\ker\mathscr{C}$ then $\mathfrak{A} := \mathscr{A}\mathscr{B}$ has full column rank $nmN + k$.*

Proof (1) follows directly from the above representations of the related functionals. (2): $z \in \ker\mathfrak{A}$ implies $\mathscr{B}z \in \ker\mathscr{A}$ and $c_{min}+\mathscr{B}z \in \ker\mathscr{C}$, $\varphi(c_{min}+\mathscr{B}z) = \varphi(c_{min})$. Owing to the uniqueness of the minimizer it follows that $\mathscr{B}z = 0$, and in turn $z = 0$, since $\mathscr{B}$ has full column rank being a basis. □

3 Test examples

The first test problem is often used in the literature to show that standard integration methods fail if applied to higher index DAEs, e.g., [13, 14].

Example 1 The DAE

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

has index-3 and dynamical degree of freedom $l_{dyn} = 0$ 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 $\|x_*\|_{H^1_D((0,1),\mathbb{R}^3)} \approx 1.11$.

The next example is the linearized version of a test problem presented [6] that has also been discussed, e.g., in [12].

Example 2 We consider the DAE

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

where

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

$$B(t) = \begin{bmatrix} 0 & 0 & 0 & -1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & -1 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & -1 & 0\\ 0 & 0 & \sin t & 0 & 1 & -\cos t & -2\rho\cos^2 t\\ 0 & 0 & -\cos t & -1 & 0 & -\sin t & -2\rho\sin t\cos t\\ 0 & 0 & 1 & 0 & 0 & 0 & 2\rho\sin t\\ 2\rho\cos^2 t & 2\rho\sin t\cos t & -2\rho\sin t & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \rho = 5,$$

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 index 3 and dynamical degree of freedom $l_{dyn} = 4$. 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 $\|x_*\|_{H^1_D((0,5),\mathbb{R}^7)} \approx 9.4$.

The following example is a boundary value problem in contrast to Example 2 which is an initial value problem.

Example 3 On the interval [0, 1], consider the DAE

$$\begin{bmatrix} 1&0&0&0&0&0\\ 0&1&0&0&0&0\\ 0&0&0&0&0&0\\ 0&0&1&0&0&0\\ 0&0&0&1&0&0\\ 0&0&0&0&1&0 \end{bmatrix} \frac{d}{dt} \begin{bmatrix} x_1\\ x_2\\ y_1\\ y_2\\ y_3\\ y_4 \end{bmatrix} + \begin{bmatrix} 0&-\lambda&0&0&0&0\\ -\lambda&0&0&0&0&0\\ -1&0&1&0&0&0\\ 0&0&0&1&0&0\\ 0&0&0&0&1&0\\ 0&0&0&0&0&1 \end{bmatrix} \begin{bmatrix} x_1\\ x_2\\ y_1\\ y_2\\ y_3\\ y_4 \end{bmatrix} = \begin{bmatrix} 0\\0\\0\\0\\0\\0 \end{bmatrix}, \quad \lambda > 0,$$

subject to the boundary conditions

$$x_1(0) = x_1(1) = 1.$$

This DAE can be brought into the proper form (1) by setting

$$A = \begin{bmatrix} 1&0&0&0&0&0\\ 0&1&0&0&0&0\\ 0&0&0&0&0&0\\ 0&0&1&0&0&0\\ 0&0&0&1&0&0\\ 0&0&0&0&1&0 \end{bmatrix}, \quad D = \begin{bmatrix} 1&0&0&0&0&0\\ 0&1&0&0&0&0\\ 0&0&1&0&0&0\\ 0&0&0&1&0&0\\ 0&0&0&0&1&0 \end{bmatrix}, \quad B = \begin{bmatrix} 0&-\lambda&0&0&0&0\\ -\lambda&0&0&0&0&0\\ -1&0&1&0&0&0\\ 0&0&0&1&0&0\\ 0&0&0&0&1&0\\ 0&0&0&0&0&1 \end{bmatrix}.$$

This DAE has the tractability index $\mu = 4$ and dynamical degree of freedom $l = 2$. The solution reads

$$\begin{aligned} x_{*,1}(t) &= \frac{e^{-\lambda t}(e^{\lambda} + e^{2\lambda t})}{1 + e^{\lambda}} \\ x_{*,2}(t) &= \frac{e^{-\lambda t}(-e^{\lambda} + e^{2\lambda t})}{1 + e^{\lambda}} \\ y_{*,1}(t) &= \frac{e^{-\lambda t}(e^{\lambda} + e^{2\lambda t})}{1 + e^{\lambda}} \\ y_{*,2}(t) &= \lambda \frac{e^{-\lambda t}(-e^{\lambda} + e^{2\lambda t})}{1 + e^{\lambda}} \\ y_{*,3}(t) &= \lambda^2 \frac{e^{-\lambda t}(e^{\lambda} + e^{2\lambda t})}{1 + e^{\lambda}} \\ y_{*,4}(t) &= \lambda^3 \frac{e^{-\lambda t}(-e^{\lambda} + e^{2\lambda t})}{1 + e^{\lambda}} \end{aligned}$$

4 Approaches to solve the constraint optimization problem (16)–(17)

Different approaches to solve the constraint optimization problem (16)–(17) have been tested, namely the direct elimination method, the weighting of the constraints, and a special deferred correction procedure as specified in the following three subsections.

4.1 Direct elimination method

The matrix $\mathscr{C}$ has full row rank $(n-1)k$.

The solution manifold of (17), that is $\ker \mathscr{C}$, forms an $(nmN + k)$-dimensional subspace of $\mathbb{R}^{nmN+nk}$ which can be characterized by

$$\mathscr{C}c = 0 \text{ if and only if } c = \mathscr{B}z \text{ for some } z \in \mathbb{R}^{nmN+k}.$$

Here, $\mathscr{B} \in \mathbb{R}^{n(mN+k)\times(nmN+k)}$ is an orthogonal basis of $\ker \mathscr{C}$. With this representation, the constrained minimization problem can be reduced to the unconstrained one

$$\tilde{\varphi}(z) = |\mathscr{A}\mathscr{B}z - r|^2_{\mathbb{R}^{nmN+l_{dyn}}} \to \min!$$

Owing to Proposition 1, the matrix product $\mathscr{A}\mathscr{B}$ has full column rank $nmN + k$.

The implemented algorithm is that of [5] (see also [4, Section 5.1.2]) which is sometimes called the *direct elimination method*. In our tests below the direct method seems to be the most robust one.

4.2 Weighting of the constraints to solve the optimization problem (16)–(17).

In this approach, a sufficiently large parameter $\omega > 0$ is chosen and the problem (16)–(17) is replaced by the free minimization problem

$$\varphi_\omega(c) = |\mathscr{A}c - r|^2_{\mathbb{R}^{nmN+l_{dyn}}} + \omega|\mathscr{C}c|^2_{\mathbb{R}^{k(n-1)}} \to \min!$$

It is known that[2] the minimizer c_ω of φ_ω converges towards the solution of (16)–(17) for $\omega \to \infty$ (cf. [9, Section 12.1.5]). Two different orderings of the equations have been implemented. One is

$$\mathscr{G} = \begin{bmatrix} \omega\mathscr{C} \\ \mathscr{A} \end{bmatrix}, \quad \bar{r} = \begin{bmatrix} 0 \\ r \end{bmatrix}$$

while the other uses a block-bidiagonal structure as it is common for collocation methods for ODEs, cf [1]. It is known that the order of the equations in the weighting method may have a large impact on the accuracy of the solutions [16]. In our test examples, however, we did not observe a difference in the behavior of both orderings.

The results of the weighting method depend substantially on the choice of the parameter ω. In order to have an accurate approximation of the exact solution c_* of the problem (16)–(17), a large value of ω should be used (in the absence of rounding errors). However, if ω becomes too large, the algorithm may lack numerical stability. A discussion of this topic has been given in [16]. In particular, it turns out that the algorithm used for the QR decomposition and the pivoting strategies have a strong influence on the success of this method. In our implementation, we use the sparse QR implementation of [8]. On the other hand, an accuracy of the solution being much lower than the approximation error of x_π is not necessary.[3] Therefore, a number of experiments have been done in order to obtain some insight into what reasonable choices might be.

Experiment 1 Influence of the choice of the weighting parameter ω

We use Example 2. Two sets of parameters are selected: (i) $N = 5$, $n = 160$ and (ii) $N = 20$, $n = 20$. The choice (i) corresponds to low degree polynomials with a corresponding large number of subintervals while (ii) uses higher degree polynomials with a corresponding small number of subintervals. Both cases have been selected according to [11, Table 20] in such a way that a high accuracy can be obtained while at the same time having only a small influence of the problem conditioning. The other parameters chosen in this experiment are: $M = N + 1$, Gauss-Legendre collocation nodes and Legendre polynomials as basis functions. The error in dependence of ω

[2] Assuming a fullrank condition on $\mathscr{A}$!

[3] The Eigen library has its own implementation of a sparse QR factorization. The latter turned out to be very slow compared to SPQR.

is measured both with respect to the exact solution and with respect to a reference solution obtained by the direct solution method. The results are provided in Tables 1 and 2. The results for Example 3 below are quite similar. The results indicate that an optimal ω may vary considerably depending on the problem parameters. However, the accuracy against the exact solution is rather insensitive of ω.

4.3 Deferred correction procedure

The direct solution method by eliminating the constraints has often the deficiency of generating a lot of fill-in in the intermediate matrices. An approach to overcome this situation has been proposed in [16]. The solutions of the weighting approach are iteratively enhanced by a defect correction process. This method is implemented in the form presented in [2, 3]. This form is called the *deferred correction procedure* for

Table 1 Influence of parameter ω for the constraints in Example 2 using $N = 5$ and $n = 160$

	(A)			(B)		
ω	$L^\infty(0,5)$	$L^2(0,5)$	$H_D^1(0,5)$	$L^\infty(0,5)$	$L^2(0,5)$	$H_D^1(0,5)$
1e-09	2.25e+00	4.54e+00	9.37e+00	2.25e+00	4.54e+00	8.04e+00
1e-08	2.00e+00	4.59e+00	9.04e+00	2.00e+00	4.59e+00	9.04e+00
1e-07	3.55e-01	5.83e-01	1.05e+00	3.55e-01	5.83e-01	1.05e+00
1e-06	1.06e-05	1.66e-05	2.84e-05	1.06e-05	1.66e-05	2.84e-05
1e-05	1.02e-07	1.60e-07	3.07e-07	1.02e-07	1.60e-07	2.75e-07
1e-04	2.26e-08	1.49e-08	1.41e-07	5.51e-09	6.27e-09	3.51e-08
1e-03	2.26e-08	1.53e-08	1.41e-07	5.51e-09	7.09e-09	3.54e-08
1e-02	2.15e-08	1.39e-08	1.40e-07	4.44e-09	3.36e-09	3.31e-08
1e-01	2.13e-08	1.39e-08	1.40e-07	4.28e-09	3.28e-09	3.29e-08
1e+00	2.00e-08	1.31e-08	1.31e-07	2.99e-09	2.99e-09	2.29e-08
1e+01	1.73e-08	1.12e-08	1.13e-07	1.27e-10	7.51e-11	7.53e-10
1e+02	1.73e-08	1.12e-08	1.12e-07	3.64e-10	3.84e-11	3.85e-10
1e+03	1.73e-08	1.12e-08	1.12e-07	2.36e-09	3.05e-10	3.06e-09
1e+04	2.15e-08	1.15e-08	1.16e-07	1.82e-08	2.91e-09	2.92e-08
1e+05	1.18e-07	3.27e-08	3.28e-07	1.26e-07	3.18e-08	3.20e-07
1e+06	6.69e-06	5.08e-07	5.08e-06	6.68e-06	5.08e-07	5.09e-06
1e+07	6.28e-05	5.09e-06	5.09e-05	6.28e-05	5.09e-06	5.09e-05
1e+08	9.94e-05	2.82e-05	2.83e-04	9.94e-05	2.82e-05	2.83e-04
1e+09	3.33e+01	7.87e+00	7.91e+01	3.33e+01	7.87e+00	7.91e+01
1e+10	8.61e+01	5.91e+01	5.93e+02	8.61e+01	5.91e+01	5.93e+02

The error of the solution with respect to the exact solution (A) and with respect to a discrete reference solution obtained by a direct method (B) is given in the norms of $L^2((0,5),\mathbb{R}^7)$, $L^\infty((0,5),\mathbb{R}^7)$ and $H_D^1((0,5),\mathbb{R}^7)$

Table 2 Influence of parameter ω for the constraints in Example 2 using $N = 20$ and $n = 20$.

	(A)			(B)		
ω	$L^\infty(0,5)$	$L^2(0,5)$	$H_D^1(0,5)$	$L^\infty(0,5)$	$L^2(0,5)$	$H_D^1(0,5)$
1e-09	2.44e+00	4.91e+00	7.59e+00	2.44e+00	4.91e+00	7.59e+00
1e-08	4.40e-02	7.51e-02	1.31e-01	4.40e-02	7.51e-02	1.31e-01
1e-07	6.38e-08	9.91e-08	1.85e-07	6.38e-08	9.91e-08	1.85e-07
1e-06	1.35e-08	2.38e-08	3.80e-08	1.35e-08	2.38e-08	3.80e-08
1e-05	2.76e-09	3.68e-09	6.77e-09	2.76e-09	3.68e-09	6.77e-09
1e-04	1.86e-10	2.77e-10	5.11e-10	1.86e-10	2.77e-10	5.13e-10
1e-03	5.12e-11	1.59e-11	5.68e-11	4.59e-11	1.60e-11	6.23e-11
1e-02	2.49e-11	4.53e-12	4.29e-11	4.25e-11	5.62e-12	5.43e-11
1e-01	3.63e-11	4.57e-12	4.59e-11	5.97e-11	6.32e-12	6.35e-11
1e+00	6.01e-11	5.37e-12	5.40e-11	8.58e-11	7.61e-12	7.64e-11
1e+01	1.51e-10	1.64e-11	1.64e-10	1.53e-10	1.60e-11	1.61e-10
1e+02	4.67e-10	4.35e-11	4.37e-10	4.39e-10	4.14e-11	4.16e-10
1e+03	1.29e-08	8.11e-10	8.15e-09	1.29e-08	8.13e-10	8.17e-09
1e+04	1.50e-07	8.22e-09	8.26e-08	1.50e-07	8.22e-09	8.26e-08
1e+05	6.26e-07	4.26e-08	4.28e-07	6.26e-07	4.26e-08	4.28e-07
1e+06	1.10e-05	7.53e-07	7.57e-06	1.10e-05	7.53e-07	7.57e-06
1e+07	3.43e-05	3.17e-06	3.19e-05	3.43e-05	3.17e-06	3.19e-05
1e+08	1.85e-04	1.22e-05	1.23e-04	1.85e-04	1.22e-05	1.23e-04
1e+09	1.77e-05	3.69e-06	3.22e-05	1.77e-05	3.69e-06	3.22e-05
1e+10	6.74e+00	2.38e+00	1.47e+01	6.74e+00	2.38e+00	1.47e+01

The error of the solution with respect to the exact solution (A) and with respect to a discrete reference solution obtained by a direct method (B) is given in the norms of $L^2((0,5),\mathbb{R}^7)$, $L^\infty((0,5),\mathbb{R}^7)$ and $H_D^1((0,5),\mathbb{R}^7)$

constrained least-squares problems by the authors. As a stopping criterion, the estimate (i) in [3, p. 254] has been implemented. Additionally, a bound for the maximal number of iterations can be provided. Under reasonable conditions, at most 2 iterations should be sufficient for obtaining maximal (with respect to the sensitivity of the problem) accuracy for the discrete solution.

The iterative solver using defect corrections may overcome the difficulties connected with a suitable choice of the parameter ω in the weighting method. According to Experiment 1, we would expect the optimal ω to be in the order of magnitude $10^{-3} \dots 10^{+2}$ with an optimum around 10^{-2}. This is in contrast to the recommendations given in [3] where a choice of $\omega \approx \varepsilon_{\text{mach}}^{-1/3}$ is recommended for the deferred correction algorithm. We test the performance of the deferred correction solver in the next experiment. Here, the tolerance in the convergence check is set to 10^{-15}. The iterations are considered not to converge if the convergence check has failed after two iterations.

Experiment 2 We check the performance of the deferred correction solver in dependence of the weight parameter ω. Both Examples 2 and 3 are used. The results are presented in Tables 3, 4, 5 and 6. The results indicate that a larger value for ω seems to be preferable.

5 Performance of the linear solvers

In this section, we intend to provide some insight into the behavior of the linear solvers. This concerns both the accuracy as well as the computational resources (computation time, memory consumption). All these data are highly implementation dependent. Also the hardware architecture plays an important role.

The linear solvers have been implemented using the standard strategy of subdividing them into a factorization step and a solve step. The price to pay is a larger memory consumption. However, their use in the context of, e.g., a modified Newton method may decrease the computation time considerably.

The tests have been run on a Linux laptop Dell Latitude E5550. While the program is a pure sequential one, the MKL library may use shared memory parallel versions of their BLAS and LAPACK routines. The CPU of the machine is an Intel(R) Core(TM) i7-5600U CPU @ 2.60GHz providing two cores, each of them capable of hyperthreading. For the test runs, cpu throttling has been disabled such that all cores ran at roughly 3.2 GHz.

The parameter for the weighting solver is $\omega = 1$ while the corresponding parameter for the deferred correction solver is $\omega = \epsilon_{\text{mach}}^{-1/3} \approx 1.65 \times 10^5$. These parameters have been chosen since they seem to be best suited for the examples. The test cases (combination of N and n) have been selected by choosing the best combinations in [11, Tables 20 and 21], respectively.

Experiment 3 First, we consider Example 2. For all values of N, $M = N + 1$ Gauss-Legendre nodes have been used. The characteristics of the test cases using Legendre basis functions are provided in Table 7. For the special properties of the Legendre

Table 3 Influence of the parameter ω on the accuracy of the discrete solution for Example 2 using $N = 5$ and $n = 160$. The error of the solution with respect to the exact solution (A) and with respect to a discrete reference solution obtained by a direct method (B) is given in the norms of $L^2((0,5),\mathbb{R}^7)$, $L^\infty((0,5),\mathbb{R}^7)$ and $H_D^1((0,5),\mathbb{R}^7)$. 2 iterations are applied

	(A)			(B)		
ω	$L^\infty(0,5)$	$L^2(0,5)$	$H_D^1(0,5)$	$L^\infty(0,5)$	$L^2(0,5)$	$H_D^1(0,5)$
0.01[a]	2.13e-08	1.39e-08	1.40e-07	4.30e-09	3.30e-09	3.31e-08
10	1.73e-08	1.12e-08	1.12e-07	5.43e-11	1.62e-11	1.63e-10
$\varepsilon_{\text{mach}}^{-1/3}$	1.73e-08	1.12e-08	1.12e-07	5.42e-11	1.62e-11	1.63e-10

[a]Iteration did not converge

Table 4 Influence of the parameter ω on the accuracy of the discrete solution for Example 2 using $N = 20$ and $n = 20$

	(A)			(B)		
ω	$L^\infty(0,5)$	$L^2(0,5)$	$H_D^1(0,5)$	$L^\infty(0,5)$	$L^2(0,5)$	$H_D^1(0,5)$
0.01[a]	2.20e-11	3.35e-12	3.37e-11	6.25e-11	6.67e-12	6.70e-11
10	1.79e-11	1.98e-12	1.99e-11	6.26e-11	6.91e-12	6.95e-11
$\varepsilon_{\text{mach}}^{-1/3}$	1.11e-11	1.71e-12	1.72e-11	6.26e-11	6.94e-12	6.97e-11

[a]Iteration did not converge

The error of the solution with respect to the exact solution (A) and with respect to a discrete reference solution obtained by a direct method (B) is given in the norms of $L^2((0,5),\mathbb{R}^7)$, $L^\infty((0,5),\mathbb{R}^7)$ and $H_D^1((0,5),\mathbb{R}^7)$. 2 iterations are applied

Table 5 Influence of the parameter ω on the accuracy of the discrete solution for Example 3 using $N = 20$ and $n = 5$.

	(A)			(B)		
ω	$L^\infty(0,5)$	$L^2(0,5)$	$H_D^1(0,5)$	$L^\infty(0,5)$	$L^2(0,5)$	$H_D^1(0,5)$
0.01	8.25e-08	6.17e-09	8.72e-09	2.52e-06	1.55e-07	2.20e-07
10	2.73e-07	1.41e-08	2.00e-08	2.63e-06	1.61e-07	2.27e-07
$\varepsilon_{\text{mach}}^{-1/3}$	3.84e-09	3.61e-10	5.11e-10	2.45e-06	1.56e-07	2.20e-07

The error of the solution with respect to the exact solution (A) and with respect to a discrete reference solution obtained by a direct method (B) is given in the norms of $L^2((0,5),\mathbb{R}^6)$, $L^\infty((0,5),\mathbb{R}^6)$ and $H_D^1((0,5),\mathbb{R}^6)$. 2 iterations are applied

Table 6 Influence of the parameter ω on the accuracy of the discrete solution for Example 3 using $N = 5$ and $n = 20$.

	(A)			(B)		
ω	$L^\infty(0,5)$	$L^2(0,5)$	$H_D^1(0,5)$	$L^\infty(0,5)$	$L^2(0,5)$	$H_D^1(0,5)$
0.01[a]	1.41e-06	4.59e-07	6.49e-07	3.75e-08	4.23e-09	5.98e-09
10	1.39e-06	4.59e-07	6.49e-07	1.42e-08	2.63e-09	3.71e-09
$\varepsilon_{\text{mach}}^{-1/3}$	1.39e-06	4.59e-07	6.49e-07	1.71e-08	2.83e-09	4.00e-09

[a]Iteration did not converge

The error of the solution with respect to the exact solution (A) and with respect to a discrete reference solution obtained by a direct method (B) is given in the norms of $L^2((0,5),\mathbb{R}^6)$, $L^\infty((0,5),\mathbb{R}^6)$ and $H_D^1((0,5),\mathbb{R}^6)$. 2 iterations are applied

Table 7 Case characteristics for Experiment 3 using the Legendre basis.

Case	N	n	dimA	dimC	nun	nnzC	$\boldsymbol{\Phi}^R_{\pi,M}$ nnzA	$\boldsymbol{\Phi}^C_{\pi,M}$ nnzA
1	3	320	8964	1914	8640	5742	101124	101124
2	5	80	3364	474	3280	1422	58964	59044
3	10	5	389	24	380	72	12749	12334
4	20	5	739	24	730	72	47509	46534

The number of nonzero elements in the matrices $\mathscr{A}$ and $\mathscr{C}$ are provided as reported by the functions of the Eigen library. The columns denote: the number of rows of $\mathscr{A}$ (dimA), the number of rows of $\mathscr{C}$ (dimC), the number of unknowns (nun), the number of nonzero elements of $\mathscr{C}$ (nnzC), the number of nonzero elements of $\mathscr{A}$ (nnzA) for the functional $\boldsymbol{\Phi}^R_{\pi,M}$ and $\boldsymbol{\Phi}^C_{\pi,M}$, respectively

polynomials, the matrix $\mathscr{C}$ representing the constraints is extremely sparse featuring only three nonzero elements per row. The computational results are shown in Table 8. In the next computations, the Chebyshev basis has been used which leads to a slightly more occupied matrix $\mathscr{C}$. The results are provided in Tables 9 and 10.

Table 8 Computing times, permanent workspace needed, and error for the cases described in Table 7. The computing times are provided in milliseconds

		$\boldsymbol{\Phi}^R_{\pi,M}$					$\boldsymbol{\Phi}^C_{\pi,M}$				
Case	Solver	nWork	tass	tfact	tslv	Error	nWork	tass	tfact	tslv	Error
1	direct	221829	16	23	4	6.74e-04	221829	14	14	4	6.44e-04
	weighted	309438	16	17	6	6.74e-04	309438	16	17	6	6.44e-04
	deferred	309438	17	17	17	6.74e-04	309438	17	17	17	6.44e-04
2	direct	115932	14	21	4	9.02e-07	116168	6	10	2	8.50e-07
	weighted	155334	16	16	5	9.19e-07	155370	8	8	3	1.65e-06
	deferred	155334	17	16	14	9.02e-07	155370	8	8	7	8.50e-07
3	direct	24233	2	4	1	8.80e-08	24967	1	2	0	6.59e-08
	weighted	26810	3	3	1	8.63e-08	27028	1	1	0	7.95e-08
	deferred	26810	3	3	2	8.80e-08	27028	1	1	1	6.59e-08
4	direct	90277	11	14	2	4.47e-12	90052	9	14	2	5.17e-12
	weighted	96544	13	10	3	1.80e-07	97857	11	10	2	4.86e-08
	deferred	96544	13	10	6	2.12e-12	97857	11	10	6	2.08e-12

They are the average of 100 runs of each case. The error is measured in the norm of $H^1_D((0,5),\mathbb{R}^7)$. The column headings denote: The upper bound on the number of nonzero elements of the QR factors as reported by SPQR (nWork), the time for the matrix assembly (tass), the time for the factorization (afact), and the time for the solution (tslv) for both functionals $\boldsymbol{\Phi}^R_{\pi,M}$ and $\boldsymbol{\Phi}^C_{\pi,M}$

Table 9 Case characteristics for Experiment 3 using the Chebyshev basis

Case	N	n	dimA	dimC	nun	nnzC	$\boldsymbol{\Phi}^R_{\pi,M}$ nnzA	$\boldsymbol{\Phi}^C_{\pi,M}$ nnzA
1	3	320	8964	1914	8640	7656	100164	101124
2	5	80	3364	474	3280	2370	58884	59044
3	10	5	389	24	380	168	12794	12594
4	20	5	739	24	730	288	47509	47119

The number of nonzero elements in the matrices $\mathscr{A}$ and $\mathscr{C}$ are provided as reported by the functions of the Eigen library. The columns denote: the number of rows of $\mathscr{A}$ (dimA), the number of rows of $\mathscr{C}$ (dimC), the number of unknowns (nun), the number of nonzero elements of $\mathscr{C}$ (nnzC), the number of nonzero elements of $\mathscr{A}$ (nnzA) for the functional $\boldsymbol{\Phi}^R_{\pi,M}$ and $\boldsymbol{\Phi}^C_{\pi,M}$, respectively

The previous example is an initial value problem. This structure may have consequences on the performance of the linear solvers. Therefore, in the next experiment, we consider a boundary value problem.

Experiment 4 We repeat Experiment 3 with Example 3. The problem characteristics and computational results are provided in Tables 11, 12, 13, and 14. It should be noted

Table 10 Computing times, permanent workspace needed, and error for the cases described in Table 9

		$\boldsymbol{\Phi}^R_{\pi,M}$					$\boldsymbol{\Phi}^C_{\pi,M}$				
Case	Solver	nWork	tass	tfact	tslv	Error	nWork	tass	tfact	tslv	Error
1	direct	236741	14	21	5	6.74e-04	227571	14	23	5	6.44e-04
	weighted	324429	17	19	6	1.15e-03	325351	17	19	6	6.93e-04
	deferred	324429	18	20	19	6.74e-04	325351	17	19	19	6.44e-04
2	direct	119389	14	22	4	9.02e-07	119012	6	11	2	8.50e-07
	weighted	149444	17	17	6	1.05e-06	149470	8	8	3	8.95e-07
	deferred	149444	18	18	16	9.02e-07	149470	8	8	8	8.50e-07
3	direct	24499	2	4	1	8.80e-08	25415	1	2	0	6.59e-08
	weighted	25806	3	3	1	9.62e-08	26076	1	2	0	8.00e-08
	deferred	25806	3	4	3	8.80e-08	26076	1	2	1	6.59e-08
4	direct	90965	10	15	2	4.41e-12	92081	9	15	2	3.71e-12
	weighted	87839	13	11	2	3.51e-12	100022	11	11	2	2.86e-12
	deferred	87839	14	12	6	2.80e-12	100022	12	11	6	1.96-12

The computing times are provided in milliseconds. They are the average of 100 runs of each case. The error is measured in the norm of $H^1_D((0,5),\mathbb{R}^7)$. The column headings denote: The upper bound on the number of nonzero elements of the QR factors as reported by SPQR (nWork), the time for the matrix assembly (tass), the time for the factorization (afact), and the time for the solution (tslv) for both functionals $\boldsymbol{\Phi}^R_{\pi,M}$ and $\boldsymbol{\Phi}^C_{\pi,M}$

Table 11 Case characteristics for Experiment 4 using the Legendre basis

Case	N	n	dimA	dimC	nun	nnzC	$\boldsymbol{\Phi}^R_{\pi,M}$ nnzA	$\boldsymbol{\Phi}^C_{\pi,M}$ nnzA
1	4	320	9602	1595	9280	4785	86403	80643
2	5	160	5762	795	5600	1422	63363	63363
3	10	5	332	20	325	60	6933	6663
4	20	5	632	20	625	60	25793	25263

The number of nonzero elements in the matrices $\mathscr{A}$ and $\mathscr{C}$ are provided as reported by the functions of the Eigen library. The columns denote: the number of rows of $\mathscr{A}$ (dimA), the number of rows of $\mathscr{C}$ (dimC), the number of unknowns (nun), the number of nonzero elements of $\mathscr{C}$ (nnzC), the number of nonzero elements of $\mathscr{A}$ (nnzA) for the functional $\boldsymbol{\Phi}^R_{\pi,M}$ and $\boldsymbol{\Phi}^C_{\pi,M}$, respectively

that the deferred correction solver returned normally (tolerance as before 10^{-15}) after at most two iterations in all cases. However, in some cases, the results are completely off. This happens, for example, in Tables 12 and 14, cases 1 and 2, for $\boldsymbol{\Phi}^C_{\pi,M}$.

Table 12 Computing times, permanent workspace needed, and error for the cases described in Table 11

		$\boldsymbol{\Phi}^R_{\pi,M}$					$\boldsymbol{\Phi}^C_{\pi,M}$				
Case	Solver	nWork	tass	tfact	tslv	Error	nWork	tass	tfact	tslv	Error
1	direct	437085	15	28	8	1.53e-04	397127	15	28	8	1.16e-04
	weighted	235746	17	15	5	8.22e-05	341713	17	21	7	2.07e-05
	deferred	235746	18	15	14	**5.53e-02**	341713	17	22	26	**9.09e+02**
2	direct	348742	21	42	12	2.59e-05	348742	19	42	13	1.55e-05
	weighted	153062	12	9	3	9.29e-07	153062	11	9	3	7.75e-06
	deferred	153062	12	9	8	**1.38e-01**	153062	11	9	10	**1.47e-01**
3	direct	11617	1	2	0	8.84e-10	12155	1	2	0	1.31e-09
	weighted	12400	2	2	1	1.25e-09	12141	1	1	0	4.99e-09
	deferred	12400	2	2	1	4.18e-11	12141	1	1	1	5.08e-09
4	direct	46847	7	8	2	2.84e-07	46883	3	4	1	3.57e-07
	weighted	42947	8	5	2	1.17e-07	42859	3	3	1	2.27e-07
	deferred	42947	8	5	4	5.27e-09	42859	3	3	2	1.51e-07

The computing times are provided in milliseconds. They are the average of 100 runs of each case. The error is measured in the norm of $H^1_D((0,1),\mathbb{R}^6)$. The column headings denote: The upper bound on the number of nonzero elements of the QR factors as reported by SPQR (nWork), the time for the matrix assembly (tass), the time for the factorization (afact), and the time for the solution (tslv) for both functionals $\boldsymbol{\Phi}^R_{\pi,M}$ and $\boldsymbol{\Phi}^C_{\pi,M}$. Data in bold indicate results where the solver terminated normally but with a result being completely off

Table 13 Case characteristics for Experiment 4 using the Chebyshev basis

case	N	n	dimA	dimC	nun	nnzC	$\boldsymbol{\Phi}^R_{\pi,M}$ nnzA	$\boldsymbol{\Phi}^C_{\pi,M}$ nnzA
1	4	320	9602	1595	9280	6380	86404	82564
2	5	160	5762	795	5600	3975	63365	63365
3	10	5	332	20	325	140	6937	6787
4	20	5	632	20	625	240	25812	25542

The number of nonzero elements in the matrices $\mathscr{A}$ and $\mathscr{C}$ are provided as reported by the functions of the Eigen library. The columns denote: the number of rows of $\mathscr{A}$ (dimA), the number of rows of $\mathscr{C}$ (dimC), the number of unknowns (nun), the number of nonzero elements of $\mathscr{C}$ (nnzC), the number of nonzero elements of $\mathscr{A}$ (nnzA) for the functional $\boldsymbol{\Phi}^R_{\pi,M}$ and $\boldsymbol{\Phi}^C_{\pi,M}$, respectively

It should be noted that a considerable amount of memory for the QR factorizations is consumed by the internal representation of the Q-factor in SPQR. This can be avoided if the factorization and solution steps are intervowen.

Table 14 Computing times, permanent workspace needed, and error for the cases described in Table 13

		$\boldsymbol{\Phi}^R_{\pi,M}$					$\boldsymbol{\Phi}^C_{\pi,M}$				
case	solver	nWork	tass	tfact	tslv	error	nWork	tass	tfact	tslv	error
1	direct	441870	15	29	8	1.41e-04	417313	15	28	8	9.90e-05
	weighted	223697	18	15	5	8.22e-05	409624	17	24	10	1.29e-05
	deferred	223697	18	15	13	**5.53e-02**	409624	17	24	33	**9.09e+02**
2	direct	353512	21	43	13	3.26e-05	353512	19	43	13	1.90e-05
	weighted	150781	12	9	3	7.43e-07	150781	11	9	3	6.01e-06
	deferred	150781	12	9	9	**1.38e-01**	150781	11	9	9	**1.47e-01**
3	direct	11857	1	2	0	2.71e-09	12397	1	2	0	2.99e-09
	weighted	12226	2	2	1	1.62e-09	11977	1	1	0	5.31e-09
	deferred	12226	2	2	2	7.85e-11	11977	1	1	1	5.20e-10
4	direct	47405	7	8	2	4.06e-08	45915	3	4	1	1.43e-07
	weighted	42901	8	5	2	6.43e-08	42817	4	3	1	2.06e-07
	deferred	42901	8	5	4	1.26e-10	42817	4	3	2	2.69e-09

The computing times are provided in milliseconds. They are the average of 100 runs of each case. The error is measured in the norm of $H^1_D((0,1),\mathbb{R}^6)$. The column headings denote: The upper bound on the number of nonzero elements of the QR factors as reported by SPQR (nWork), the time for the matrix assembly (tass), the time for the factorization (afact), and the time for the solution (tslv) for both functionals $\boldsymbol{\Phi}^R_{\pi,M}$ and $\boldsymbol{\Phi}^C_{\pi,M}$. Data in bold indicate results where the solver terminated normally but with a result being completely off

6 Sensitivity of boundary condition weighting

As already known for boundary value problems for ODEs and index-1 DAEs, a special problem is the scaling of the boundary condition, and hence, here the inclusion of the boundary conditions (2). Their scaling is independent of the scaling of the DAE (1). Therefore, it seems to be reasonable to provide an additional possibility for the scaling of the boundary conditions. We decided to enable this by introducing an additional parameter α to be chosen by the user. So, $\boldsymbol{\Phi}$ from (5) is replaced by the functional

$$\tilde{\boldsymbol{\Phi}}(x) = \int_a^b |A(t)(Dx)'(t) + B(t)x(t) - q(t)|^2 \mathrm{dt} + \alpha |G_a x(a) + G_b x(b) - d|^2.$$

Analogously, the discretized versions $\boldsymbol{\Phi}^R_{\pi,M}$, $\boldsymbol{\Phi}^I_{\pi,M}$ and $\boldsymbol{\Phi}^C_{\pi,M}$ are replaced by their counterparts $\tilde{\boldsymbol{\Phi}}^R_{\pi,M}$, $\tilde{\boldsymbol{\Phi}}^I_{\pi,M}$ and $\tilde{\boldsymbol{\Phi}}^C_{\pi,M}$ with weighted boundary conditions. The convergence theorems will hold true for these modifications of the functional, too.

Experiment 5 Influence of α on the accuracy

We use the example and settings of Experiment 1. The results are provided in Table 15.

Experiment 6 Influence of α on the accuracy

We repeat the previous experiment with Example 3. The discretization parameters are (i) $N = 5, n = 20$ and (ii) $N = 20, n = 5$. All other settings correspond to those of Experiment 5. The results are presented in Table 16.

The results of Experiments 5 and 6 indicate that the final accuracy is rather insensitive to the choice of α. It should be noted that the coefficient matrices in Examples 2 and 3 are well-scaled.

7 Final remarks

In summary, 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 Part 1, 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. We refer to the corresponding *Final remarks and conclusions* in [11].

A critical ingredient for the implementation of the method is the algorithm used for the solution of the discrete linear least-squares problem. Given the expected bad conditioning of the least-squares problem, a QR factorization with column pivoting

Table 15 Influence of weight parameter α for the boundary conditions in Example 2

	$N = 5, n = 160$			$N = 20, n = 20$		
α	$L^\infty(0,5)$	$L^2(0,5)$	$H_D^1(0,5)$	$L^\infty(0,5)$	$L^2(0,5)$	$H_D^1(0,5)$
1e-10	3.18e+00	7.03e+00	1.21e+01	1.60e+00	3.10e+00	5.09e+00
1e-09	9.33e-07	2.33e-06	3.84e-06	1.60e+00	3.10e+00	5.09e+00
1e-08	1.58e-07	3.52e-07	6.16e-07	1.05e-07	1.94e-07	3.54e-07
1e-07	1.27e-07	1.39e-08	3.26e-08	5.06e-09	1.10e-08	2.00e-08
1e-06	7.17e-08	2.20e-09	1.68e-08	9.60e-10	2.29e-09	4.10e-09
1e-05	9.60e-08	1.59e-09	1.58e-08	7.64e-11	2.07e-10	3.80e-10
1e-04	6.99e-08	1.59e-09	1.60e-08	5.00e-11	4.07e-11	9.26e-11
1e-03	9.83e-08	1.82e-09	1.83e-08	3.91e-11	6.41e-12	5.46e-11
1e-02	1.15e-07	2.28e-09	2.29e-08	6.37e-11	6.26e-12	6.25e-11
1e-01	6.43e-08	1.27e-09	1.27e-08	5.11e-11	6.61e-12	6.64e-1
1e+00	6.04e-08	1.13e-09	1.13e-08	6.66e-11	7.50e-12	7.54e-11
1e+01	2.15e-07	3.40e-09	3.42e-08	7.97e-11	9.85e-12	9.89e-11
1e+02	4.12e-07	5.66e-09	5.68e-08	6.78e-11	8.10e-12	8.14e-11
1e+03	4.51e-06	5.74e-08	5.76e-07	9.60e-11	9.81e-12	9.85e-11
1e+04	2.31e-05	2.93e-07	2.95e-06	2.24e-09	1.52e-10	1.52e-09
1e+05	4.68e-04	5.94e-06	5.97e-05	2.91e-08	1.35e-09	1.36e-08
1e+06	2.12e+03	5.16e+01	5.19e+02	2.34e-07	1.68e-08	1.68e-07
1e+07	6.53e+03	1.03e+02	1.04e+03	2.97e-06	1.77e-07	1.77e-06
1e+08	4.60e+02	1.78e+01	1.79e+02	4.76e-06	3.72e-07	3.73e-06
1e+09	2.05e+01	3.27e+00	3.24e+01	4.56e+01	4.90e+00	4.91e+01

The error of the solution is given in the norms of $L^2((0,5),\mathbb{R}^7)$, $L^\infty((0,5),\mathbb{R}^7)$ and $H_D^1((0,5),\mathbb{R}^7)$

must lie at the heart of the algorithm. At the same time, the sparsity structure must be used as best as possible. In our tests, the direct solver seems to be the most robust one. With respect to efficiency and accuracy, the deferred correction solver is preferable. However, it failed in certain tests.

The results for $M = N + 1$ are not much different from those obtained for a larger M, for which we do not yet have an explanation.

In conclusion, we note that earlier implementations, among others the one from the very first paper in this matter [13] 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 gives rise to a very efficient solver for higher index DAEs.

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) [15], the Eigen matrix library (version 3.3.7) [10], SuiteSparse (version 5.6.0) [7], in particular its sparse QR factorization [8], Intel®

Table 16 Influence of weight parameter α for the boundary conditions in Example 3

	$N = 5, n = 20$			$N = 20, n = 5$		
α	$L^\infty(0,1)$	$L^2(0,1)$	$H_D^1(0,1)$	$L^\infty(0,1)$	$L^2(0,1)$	$H_D^1(0,1)$
1e-10	4.21e-02	7.02e-02	9.13e-02	1.03e-06	8.55e-08	1.21e-07
1e-09	4.46e-04	7.38e-04	9.60e-04	1.00e-06	6.11e-08	8.64e-08
1e-08	4.40e-06	6.71e-06	8.80e-06	1.14e-06	6.48e-08	9.16e-08
1e-07	1.47e-06	4.87e-07	6.88e-07	9.84e-07	6.02e-08	8.51e-08
1e-06	1.39e-06	4.59e-07	6.49e-07	1.67e-06	1.10e-07	1.56e-07
1e-05	1.40e-06	4.59e-07	6.49e-07	1.19e-06	8.21e-08	1.16e-07
1e-04	1.40e-06	4.59e-07	6.49e-07	8.55e-07	6.48e-08	9.17e-08
1e-03	1.40e-06	4.59e-07	6.49e-07	1.44e-06	1.04e-07	1.47e-07
1e-02	1.40e-06	4.59e-07	6.49e-07	5.14e-07	4.77e-08	6.75e-08
1e-01	1.40e-06	4.59e-07	6.49e-07	1.69e-06	8.49e-08	1.20e-07
1e+00	1.40e-06	4.59e-07	6.49e-07	2.45e-06	1.56e-07	2.20e-07
1e+01	1.40e-06	4.59e-07	6.49e-07	1.83e-06	1.09e-07	1.54e-07
1e+02	1.40e-06	4.59e-07	6.49e-07	1.91e-05	8.14e-07	1.15e-06
1e+03	1.40e-06	4.59e-07	6.49e-07	1.40e-04	1.10e-06	1.55e-06
1e+04	1.41e-06	4.59e-07	6.49e-07	1.27e-03	5.34e-05	7.56e-05
1e+05	1.39e-06	4.59e-07	6.49e-07	3.69e-04	1.94e-05	2.75e-05
1e+06	1.63e-06	4.66e-07	6.59e-07	3.98e-04	3.42e-05	4.83e-05
1e+07	1.99e+02	5.07e+01	7.18e+01	2.11e-03	3.53e-04	4.99e-04
1e+08	1.99e+02	5.07e+01	7.18e+01	1.22e-01	2.83e-02	4.01e-02
1e+09	1.99e+02	5.07e+01	7.18e+01	4.86e-01	2.05e-01	2.90e-01

The error of the solution is given in the norms of $L^2((0,1),\mathbb{R}^6)$, $L^\infty((0,1),\mathbb{R}^6)$ and $H_D^1((0,1),\mathbb{R}^6)$

MKL (version 2019.5-281), all in double precision with a rounding unit of $\epsilon_{\text{mach}} \approx 2.22 \times 10^{-16}$.[4] The code is optimized using the level -O3.[5]

Funding Open access funding provided by Royal Institute of Technology.

[4]Intel is a registered trademark of Intel Corporation.

[5]The 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.

References

1. 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)
2. Barlow, J.L.: Solution of sparse weighted and equality constrained least squares problems. In: Page, C., LePage, R. (eds.) Computing Science and Statistics, pp. 53–62. Springer, New York (1992)
3. Barlow, J.L., Vemulapati, U.B.: A note on deferred correction for equality constrained least squares problems. SIAM J. Numer Anal. **29**(1), 249–256 (1992)
4. Björck, Å.: Numerical Methods for Least Squares Problems. SIAM, Philadelphia (1996)
5. Björck, Å., Golub, G.H.: Iterative refinement of linear least squares solutions by Householder transformations. BIT **7**, 322–337 (1967)
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. Golub, G.H., van Loan, C.h. Matrix Computations, 2nd edn. The Johns Hopkins University Press, Baltimore and London (1989)
10. Guennebaud, G., Jacob, B., et al.: Eigen v3. http://eigen.tuxfamily.org (2010)
11. Hanke, M., März, R.: Towards a reliable implementation of least-squares collocation for higher-index linear differential-algebaic equations. Part 1: Basics and ansatz choices. Numerical Algorithms. submitted
12. Hanke, M., März, R., Tischendorf, C.: Least-squares collocation for higher-index linear differential-algebaic equations Estimating the stability threshold. Math. Comp. **88**(318), 1647–1683 (2019). https://doi.org/10.1090/mcom/3393
13. 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
14. Lamour, R., März, R., Tischendorf, C. 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)
15. Stallman, R.M., GCC Developers Community, et al: Using the Gnu Compiler collection. CreateSpace Scotts Valley (2009)
16. van Loan, C.h.: On the method of weighting for equality-constrained least-squares problems. SIAM J. Numer. Anal. **22**(5), 851–864 (1985)