Journal of Computational and Applied Mathematics 43 (1992) 243–259
North-Holland

CAM 1252

Projected collocation for higher-order higher-index differential-algebraic equations

Uri M. Ascher *
Department of Computer Science, University of British Columbia, Vancouver, Canada

Linda R. Petzold **
Department of Computer Science, University of Minnesota, Minneapolis, MN 55455, United States

Received 27 June 1991
Revised 4 November 1991

Abstract

Ascher, U.M. and L.R. Petzold, Projected collocation for higher-order higher-index differential-algebraic equations, Journal of Computational and Applied Mathematics 43 (1992) 243–259.

Higher-order, higher-index Hessenberg systems of initial and boundary value differential-algebraic equations (DAEs) are considered. These types of systems arise in a variety of applications, including multibody systems. We extend a class of recently introduced projected implicit Runge–Kutta methods and define a new class of projected piecewise polynomial collocation methods for the solution of these problems. Stabilizing reformulations are considered as well, and a new projected invariant method is proposed.

The higher-order ODE part of the DAE is collocated directly by a piecewise polynomial. A projection modification helps restore all the properties of stability and superconvergence which a corresponding collocation method for an ODE possesses. Higher-order collocation at Radau points is recommended for initial-value problems. The projection methods appear to be particularly promising for the solution of DAE boundary value problems, where the need to maintain stability in the differential part of the system often necessitates the use of methods based on symmetric discretizations, like collocation at Gauss points. Previously defined symmetric methods have severe limitations when applied to these problems, including instability, oscillation and loss of accuracy; the new methods overcome these difficulties.

For higher-index problems we consider reformulation methods of stabilizing index reduction. We propose new methods of projected invariants to handle particularly tough higher-index problems. The advantages offered by these methods are demonstrated numerically.

Keywords: Differential-algebraic systems; higher-index, projected collocation; stability; stiffness; projected invariant.

Correspondence to: Prof. U.M. Ascher, Department of Computer Science, University of British Columbia, Vancouver, B.C., Canada V6T 1Z2. e-mail: ascher@cs.ubc.ca.

* The work of this author was partially supported under NSERC Canada Grant OGP0004306.
** The work of this author was partially supported by the Applied Mathematical Sciences subprogram of the Office of Energy Research, U.S. Department of Energy, by Lawrence Livermore National Laboratory under contract W-7405-Eng-48.

1. Introduction

Most general-purpose codes for solving ODEs, both for initial-value problems (IVPs) and for boundary value problems (BVPs), require the ODE to be in the form of a first-order system. The methods proposed in the literature on directly approximating a higher-order ODE are almost always too specialized to be used in a general solver. An exception is piecewise polynomial collocation [1,4,11], which can be applied directly even to mixed-order systems [3]. In these methods the approximation sought for an ODE of order m collocating at k collocation points in each element of an arbitrary mesh is a piecewise polynomial of order $k+m$ in C^{m-1}. When the collocation points are chosen as zeros of certain orthogonal polynomials, e.g., Gauss or Radau points, methods with desirable properties of stability and superconvergence result. For first-order systems these collocation methods are equivalent to certain implicit Runge–Kutta methods, so for higher-order systems they give an extended family of finite-difference methods. For a given higher-order ODE with $m > 1$, such a direct collocation method is cheaper than a similar method applied to the converted first-order system, and yields similar accuracy at least in the nonstiff case [1].

But for very stiff ODEs, some such collocation methods (including those based on Gauss points) suffer an order reduction (i.e., no superconvergence occurs). Moreover, for symmetric difference schemes certain stability difficulties may occasionally arise (see, e.g., [4]). This phenomenon certainly occurs also for systems of differential-algebraic equations (DAEs), which can be viewed as a limit of corresponding very stiff ODEs (cf. [10,16,17]).

Yet, a DAE in semi-explicit form has a more special structure than its regularization as a stiff ODE, because the approximation to the algebraic solution components can be sought in a space of lower continuity [2,16]. In the collocation context, treating the algebraic part of the system as an ODE of order $m = 0$ works very well for semi-explicit index-1 DAEs.

For higher-index DAEs, however, approximating the algebraic solution components in a lower continuity space alone is not sufficient to recover the stability and superconvergence properties of collocation for nonstiff ODEs. In [5] we have proposed a projection method for first-order DAEs of index 2 in Hessenberg form which does achieve the desired numerical behaviour. The idea is to update the differential solution components in such a way that the algebraic constraints are satisfied at mesh points.

In this paper we extend the methods and results of [5,6] for semi-explicit DAEs composed of a higher-order system of ODEs depending on additional unknown functions (the algebraic solution components) and satisfying additional constraints which do not include the algebraic unknowns. Such systems arise in many applications. One source of applications is a higher-order ODE system with the additional requirement that an invariant be maintained [13]. Other examples include [8] and the very important equations of constrained dynamics (see, e.g., [20]). The methods reported here are ripe for implementation in a rather general-purpose context, and work is underway to extend the package COLSYS [3] based on our results.

In Section 2 we consider the case where an ODE system of order m, $m \geq 1$, is coupled with algebraic constraints such that the DAE has index 2. The essential difference between the approximation method here and the one reported in [5] is in the smoother, higher-order piecewise-polynomial space in which the approximation for the ODE solution lies. We define an essential underlying ODE for this problem and project the $(m-1)$st derivative of the collocation solution at mesh points to retrieve nonstiff stability and superconvergence order.

We also propose a method for *projected invariants* [6,13], for a higher-order ODE with an invariant. This may at times be useful for redefining a DAE which is more amenable to numerical discretization than the given one.

In Section 3 we then consider a class of higher-order DAEs of the higher index $m+1$. We discuss methods for reducing the problem to one of those covered in Section 2 and the idea of an additional projection designed to obey the original constraints. This extends results of [6]. It is desirable to obtain an index-2 reformulation where the constraint matrix and the Jacobian matrix with respect to the algebraic unknowns are balanced, in addition to satisfying the original constraints. A new, direct method of projected invariants, which achieves this goal and applies to the higher-order ODE formulation, is proposed.

The proposed methods are demonstrated numerically in Section 4.

2. Higher-order, index-2 DAEs

In this section we consider the DAE of order m:

$$x^{(m)} = f(z(x),\ y,\ t), \tag{2.1a}$$

$$\mathbf{0} = g(z(x),\ t), \tag{2.1b}$$

where $z_j(t) = x^{(j-1)}(t) := \mathrm{d}^{j-1}x(t)/\mathrm{d}t^{j-1}$, $1 \leqslant j \leqslant m$, and

$$z(x)(t) = \left(x^{\mathrm{T}}(t),\ (x')^{\mathrm{T}}(t), \ldots, (x^{(m-1)})^{\mathrm{T}}(t)\right)^{\mathrm{T}}, \tag{2.2}$$

assuming that $g_{z_m} f_y$ is nonsingular for all t, $0 \leqslant t \leqslant 1$. Converting (2.1a) into first-order form in z, it is clear that (2.1) has index 2.

Standard arguments using Newton's method and the Newton–Kantorovich Theorem (cf. [4]) apply here as in [5], so below we concentrate on the linear (or linearized) case

$$x^{(m)} = f(z(x),\ y,\ t) = \sum_{j=1}^{m} A_j z_j + By + q, \tag{2.3a}$$

$$\mathbf{0} = g(z(x),\ t) = \sum_{j=1}^{m} C_j z_j + r, \tag{2.3b}$$

where A_j, B and C_j are smooth functions of t, $0 \leqslant t \leqslant 1$, $A_j(t) \in \mathbb{R}^{n_x \times n_x}$, $B(t) \in \mathbb{R}^{n_x \times n_y}$, $C_j(t) \in \mathbb{R}^{n_y \times n_x}$, $n_y \leqslant n_x$ and $C_m B$ is nonsingular for each t. All matrices involved are assumed to be uniformly bounded in norm, together with their derivatives, by a constant of moderate size. The inhomogeneities are $q(t) \in \mathbb{R}^{n_x}$ and $r(t) \in \mathbb{R}^{n_y}$.

2.1. Essential underlying ODE

We derive a stability result for this system. As in [5], there exists a smooth, bounded matrix function $R(t) \in \mathbb{R}^{(n_x - n_y) \times n_x}$ whose linearly independent and normalized rows form a basis for the nullspace of B^{T} (R can be taken to be orthonormal). Thus, for each t, $0 \leqslant t \leqslant 1$,

$$RB = 0. \tag{2.4}$$

We assume that there exists a constant K_1 of moderate size such that

$$\|(C_m B)^{-1}\| \leq K_1, \tag{2.5}$$

uniformly in t, and obtain [5, Lemma 2.1] that there is a constant K_2 of moderate size such that

$$\|(S \quad F)\| = \left\| \begin{pmatrix} R \\ C_m \end{pmatrix}^{-1} \right\| \leq K_2. \tag{2.6}$$

The constant K_2 depends, in addition to K_1, also on $\|B\|$, $\|C_m\|$ and $\|R\|$. Let K_3 be a moderate bound on $B, C_1, \ldots, C_m, R$ and their derivatives:

$$\|B^{(j)}\|, \|C_l^{(j)}\|, \|R^{(j)}\| \leq K_3, \quad j = 0, 1. \tag{2.7}$$

Define new variables

$$v_j = z_j, \quad j = 1, \ldots, m-1, \tag{2.8a}$$

$$v_m = Rz_m, \tag{2.8b}$$

$$v = (v_1, v_2, \ldots, v_m)^{\mathrm{T}} \tag{2.8c}$$

(note that $v(t) \in \mathbb{R}^{mn_x - n_y}$). Then, using (2.3b), the inverse transformation is given by

$$z_m = \begin{pmatrix} R \\ C_m \end{pmatrix}^{-1} \begin{pmatrix} v_m \\ -\left(\sum_1^{m-1} C_j z_j + r\right) \end{pmatrix} \equiv S v_m - F\left(\sum_1^{m-1} C_j v_j + r\right), \tag{2.9}$$

where $S(t) \in \mathbb{R}^{n_x \times (n_x - n_y)}$ and

$$F := B(C_m B)^{-1}. \tag{2.10}$$

By our assumptions and (2.6) this mapping is well-conditioned. Both S and F are smooth and bounded. Differentiating (2.8b) yields the *essential underlying ODE* (EUODE)

$$v_j' = v_{j+1}, \quad j = 1, \ldots, m-2, \tag{2.11a}$$

$$v_{m-1}' = S v_m - F\left(\sum_1^{m-1} C_j v_j + r\right), \tag{2.11b}$$

$$v_m' = R \sum_1^{m-1} A_j v_j + (RA_m + R')\left(S v_m - F\left(\sum_1^{m-1} C_j v_j + r\right)\right) + Rq. \tag{2.11c}$$

For a unique solution of (2.3) one needs to impose $mn_x - n_y$ independent boundary conditions

$$B_0 z(0) + B_1 z(1) = \beta. \tag{2.12}$$

These can obviously be written in terms of v:

$$\hat{B}_0 v(0) + \hat{B}_1 v(1) = \hat{\beta}, \tag{2.13}$$

with $\hat{B}_0$ and $\hat{B}_1$ square matrices.

Theorem 2.1. *Let the DAE* (2.3) *have smooth, bounded coefficients, and assume that* (2.5) *holds and that the boundary value ODE* (2.11), (2.13) *is well-conditioned. Then there is a constant K of*

moderate size such that

$$\|z\| \leq K(\|q\| + \|r\| + |\beta|), \tag{2.14a}$$

$$\|y\| \leq K(\|q\| + \|r\| + \|r'\| + |\beta|). \tag{2.14b}$$

Proof. The proof is a straightforward modification of the one for [5, Theorem 2.1]. □

2.2. Higher-order collocation

To define our collocation method, consider a mesh

$$\pi\colon\ 0 = t_0 < t_1 < \cdots < t_N = 1, \quad h_i = t_i - t_{i-1},\ h := \max_{1 \leq i \leq N} h_i, \tag{2.15}$$

and a set of k points, $k \geq m$,

$$0 < \rho_1 < \rho_2 < \cdots < \rho_k \leq 1. \tag{2.16}$$

These points satisfy the orthogonality condition

$$\int_0^1 \phi(s) \prod_{i=1}^{k} (s - \rho_i)\, ds = 0, \quad \forall \phi \in \mathcal{P}_{p-k}, \tag{2.17}$$

for some $p \geq k$. In particular, $p = 2k$ for Gauss points (zeros of Legendre polynomial; this gives a symmetric scheme with $\rho_k < 1$), and $p = 2k - 1$ for Radau points (which satisfy $\rho_k = 1$).

In the unprojected collocation method, we seek an approximate solution $x_\pi \in \mathcal{P}_{k+m,\pi} \cap C^{m-1}[0, 1]$, [1] $y_\pi \in \mathcal{P}_{k,\pi}$, which satisfies the boundary conditions ((2.12) or a nonlinear version thereof) and for each i, $1 \leq i \leq N$,

$$x_\pi^{(m)}(t_j) = f\big(z\big(x_\pi(t_j)\big),\ y_\pi(t_j),\ t_j\big), \tag{2.18a}$$

$$\mathbf{0} = g\big(z\big(x_\pi(t_j)\big),\ t_j\big), \tag{2.18b}$$

$$t_j = t_{i-1} + h_i \rho_j, \quad j = 1, \ldots, k. \tag{2.18c}$$

It can be easily verified that this gives the same number of algebraic equations as there are free parameters in the representation. We can represent the approximate solution on the ith element $[t_{i-1}, t_i]$ as

$$x_\pi(t) = \sum_{j=1}^{m} \frac{(t - t_{i-1})^{j-1}}{(j-1)!} z_j^{i-1} + h_i^m \sum_{j=1}^{k} \psi_j\left(\frac{t - t_{i-1}}{h_i}\right) w_j, \tag{2.19}$$

with $\psi_j(s) \in \mathcal{P}_{k+m}\,[0, 1]$ satisfying (for $1 \leq j \leq k$)

$$\psi_j^{(l)}(0) = 0, \quad l = 0, \ldots, m-1, \qquad \psi_j^{(m)}(\rho_l) = \delta_{jl}, \quad l = 1, \ldots, k. \tag{2.20}$$

Then $z_j^{i-1} = x_\pi^{(j-1)}(t_{i-1})$ (these are known for an IVP and are part of the *global* unknown system for a BVP) and $w_j = x_\pi^{(m)}(t_j)$ are the local unknowns. In addition, $y_\pi(t)$ is defined locally as the polynomial interpolant of its k collocation values:

$$y_\pi(t_j) = y_j, \quad j = 1, \ldots, k. \tag{2.21}$$

[1] We say that v is in $\mathcal{P}_l$ if $v(t)$ is a polynomial of order l (degree $< l$) on an appropriate interval, and that v is in $\mathcal{P}_{l,\pi}$ if $v(t)$ is a piecewise polynomial which is in $\mathcal{P}_l$ on each subinterval (element) of the mesh π.

For $n_y = 0$, the obtained approximate solution is well known to be stable and attain a superconvergence order $O(h^p)$ at mesh points, for an arbitrary mesh [2,11]. But for $n_y > 0$, i.e., for a DAE, we must now define a projected collocation method in order to retrieve similar properties. We do this by modifying for the ith element

$$z_m^i \leftarrow x_\pi^{(m-1)}(t_i) + B(t_i)\lambda, \tag{2.22}$$

with $\lambda = \lambda_i$ determined so as to satisfy the constraint (2.1b) at the mesh point t_i. We also require that the constraint (2.1b) be satisfied at t_0, so if the points ρ_k are symmetric about $\frac{1}{2}$, then the projected collocation method is symmetric as well (cf. [5]).

For a nonlinear problem we consider the projection in the context of a quasilinearization method, with $B = f_y$.

The unknowns λ_i are obviously well-defined (by (2.5)) and can be eliminated locally (in terms of known quantities in case of an IVP or in terms of the mesh unknowns $\{z_i\}$ in case of a BVP). Note that if $\rho_k < 1$, then the projected solution is not in C^{m-1} any more; rather, it is in C^{m-2}. If $\rho_k = 1$, then the requirement that (2.1b) be satisfied at mesh points is already included in the collocation equations (2.18b), so the projected and the unprojected methods coincide.

Theorem 2.2. *Given a well-conditioned, semi-explicit, linear index-2 system* (2.3), (2.12) *with smooth coefficients to be solved numerically by the k-stage projected collocation method described above, then for $h > 0$ sufficiently small,*

(1) *the local error in $x^{(l)}$ is* $O(h_i^{\min(p+1,k+2)})$, $l = 0, \ldots, m-1$;

(2) *there exists a unique projected collocation solution*;

(3) *the projected collocation method is stable, with a moderate stability constant, provided that the BVP has a moderate stability constant K*;

(4) *the global error in $x^{(l)}$ is* $O(h^{\min(p,k+1)})$, $l = 0, \ldots, m-1$;

(5) *the errors in the intermediate variables w_j and y_j are* $O(h^k)$.

Proof. As in [1], or in [4, Section 5.6], note that the higher-order collocation method gives, upon elimination of the local unknowns w_j in each element, a one-step difference scheme which approximates the corresponding first-order form of the ODE to order $O(h_i^{k+1})$. This scheme in fact differs from the same collocation scheme applied directly to the first-order form of the ODE only in higher-order terms. Furthermore, the EUODE (2.11) is the same as the EUODE for the first-order form of this index-2 DAE, and the proposed projection is subsequently seen to correspond to the same one as in [5]. Therefore, the proof of [5, Theorem 3.1] applies here and yields the above-stated results. □

Theorem 2.3. *Let the coefficient functions and the inhomogeneities in* (2.3) *be in $C^{p+m}[0, 1]$. Under the additional assumptions of Theorem* 2.2, *the projected collocation method satisfies for $0 \leqslant t \leqslant 1$,*

$$\left| x_\pi^{(l)}(t) - x^{(l)}(t) \right| = O(h^{\min(k+m-l,p)}), \quad l = 0, \ldots, m, \tag{2.23a}$$

$$\left| y_\pi(t) - y(t) \right| = O(h^k). \tag{2.23b}$$

Moreover, nonstiff superconvergence order holds for the projected collocation method, viz.

$$|z_i^l - x^{(l-1)}(t_i)| = O(h^p), \quad 1 \leq l \leq m, \ 0 \leq i \leq N. \tag{2.24}$$

Proof. The proof is sufficiently similar to that of [5, Theorem 3.2] to allow us to review only the differences between them. We concentrate on the superconvergence result (2.24) first. Using the transformation (2.8) with $x_\pi^{(j-1)}$ replacing z_j, we obtain an approximation v_π to v. For this approximation we obtain superconvergence in the usual way (cf. [1,5]), even though it is not a piecewise polynomial. The transformation back from v to z, which holds at mesh points due to the projection and (2.9), allows us then to conclude (2.24). (For unprojected collocation, it is only this back-transformation which does not hold in general; but this deficiency is sufficient to potentially generate much trouble, as demonstrated in Section 4.) Next, note that (2.23b) and (2.23a) for $l = m-1$, m are obtained directly from Theorem 2.2. The higher-order global convergence results in (2.23a) are now obtained by integrating up from $l = m-1$ for lower-order derivatives and using (2.24). □

As mentioned before, these results for the linear case can be "lifted" to include the nonlinear case using standard arguments. We thus consider given nonlinear boundary conditions

$$b(z(0), z(1)) = 0, \tag{2.25}$$

whose linear incarnation is given in (2.12), and obtain (cf. [5]) the following theorem.

Theorem 2.4. *Let $x(t)$, $y(t)$ be an isolated solution of the DAE problem* (2.1), (2.25) *and assume that f and g have continuous second partial derivatives and that the smoothness assumptions of Theorem* 2.3 *hold for the linearized problem in the neighborhood of $x(t)$, $y(t)$. Then there are positive constants ρ and h_0 such that for all meshes with $h \leq h_0$,*

(1) *there is a unique solution $x_\pi(t)$, $y_\pi(t)$ to the projected collocation equations in a tube $S_\rho(x, y)$ of radius ρ around $x(t)$, $y(t)$;*

(2) *this solution can be obtained by Newton's method, which converges quadratically provided that the initial guess for $x_\pi(t)$, $y_\pi(t)$ is sufficiently close to $x(t)$, $y(t)$;*

(3) *the error estimates* (2.23a)–(2.24) *hold.*

2.3. Projected invariants I

In [6] we have considered the method of *projected invariants* as a way to improve the behaviour of the problem to be discretized in case that C_m is "nicer" than B in (2.3), and an additional differentiation of the constraints can be afforded. Here we consider a different variant of the same approach because we wish to retain the ODE in higher-order form, in order to take advantage of this form in the collocation approximation.

Thus, differentiating (2.1b), we have

$$0 = g_t + \sum_{j=1}^{m} g_{z_j} x^{(j)}. \tag{2.26}$$

This can be substituted into (2.1a) to eliminate y, obtaining an ODE of order m for x. For the linear(ized) case, (2.3a) plus the derivative of (2.3b) are equivalent to

$$x^{(m)} = \sum_{j=1}^{m} \left[HA_j - \binom{m}{j-1} FC^{(m-j+1)} \right] z_j + Hq - Fr^{(m)}, \tag{2.27}$$

where F is defined in (2.10) with $C_m \equiv C$ and $H = I - FC = SR$. For practical reasons, we actually prefer to keep the explicit form (2.1b), (2.26), but for a clearer exposition of the method we now consider (2.27).

One could proceed to simply integrate this ODE, as is often done in robotics. But in some applications, neglecting the constraints after discretization produces poorly conditioned problems (e.g., [8,12]). Hence we consider the index-2 DAE

$$x^{(m)} = \sum_{j=1}^{m} \left[HA_j - \binom{m}{j-1} FC^{(m-j+1)} \right] z_j + Hq - Fr^{(m)} + C^{\mathrm{T}}\mu \tag{2.28}$$

and (2.3b). The technique is defined for nonlinear problems along the same lines, using quasilinearization.

Now, in (2.28), (2.3b) we have a DAE to which the projected collocation method described above is applied. The difference from the original index-2 problem is that B in (2.3a) has been replaced by C^{T} (or another convenient smooth matrix function $D(t)$ which has the dimensions of $B \equiv f_y$ and must satisfy that $g_x D$ is nonsingular for each t). The resulting DAE is sometimes much more amenable to discretization by the same numerical method, as demonstrated in Section 4 (cf. [6, Section 3.5]).

3. Higher-order, higher-index DAEs

In this section we consider the DAE of order m:

$$x^{(m)} = f(z(x), y, t), \tag{3.1a}$$

$$0 = g(x, t), \tag{3.1b}$$

which is index $m+1$ if $g_x f_y$ is nonsingular for all t, $0 \leqslant t \leqslant 1$, as we shall assume. We will often concentrate on the linear problem

$$x^{(m)} = f(z(x), y, t) = \sum_{j=1}^{m} A_j z_j + By + q, \tag{3.2a}$$

$$0 = g(x, t) = Cx + r, \tag{3.2b}$$

with dimensions and smoothness assumptions as in Section 2 here and in [6, Section 2]. The DAE (3.1) is supplied with $m(n_x - n_y)$ boundary conditions

$$b(z(0), z(1)) = 0, \tag{3.3}$$

which are assumed to be independent of the set [2]

$$0 = g^{(l)}(z(x(0)), 0), \quad l = 0, \ldots, m-2, \tag{3.4}$$

[2] We use the notation $g^{(l)}(z(x), t) = (\mathrm{d}^l/\mathrm{d}t^l) g(x(t), t)$.

so that it makes sense to consider an isolated solution of the index-2 BVP obtained by

$$\mathbf{0} = g^{(m-1)}(z(x), t), \tag{3.5}$$

together with (3.1a), (3.3), (3.4).

3.1. Reduction to index 2

One simple way of dealing with the higher-index DAE is to apply $m-1$ differentiations to the constraints, as in (3.5) above, and then proceed to apply a projected collocation method as in Section 2. The projected invariants method I of Section 2.3 may obviously be applied here as well. It may be argued that the ability to differentiate the constraints (3.1b) $m-1$ times, at least in principle, without having to deal with distribution functions, is part of the essential assumptions that one must make on the problem.

An investigation of the *stability* of such a transformation to index 2 was taken up in [6] (see, in particular, Section 3.2 there). It is not difficult to see that by replacing (3.1), (3.3) by (3.1a), (3.5), (3.3), (3.4) one introduces an algebraic instability, whereby errors may grow like a polynomial of degree $m-2$. No such instability is introduced if one uses Baumgarte's stabilization [7], whereby (3.5) is replaced by

$$\sum_{j=0}^{m-1} \alpha_j \frac{\mathrm{d}^j}{\mathrm{d}t^j} g(x(t), t) = \mathbf{0}, \tag{3.6}$$

with

$$\sigma(\tau) = \sum_{j=0}^{m-1} \alpha_j \tau^j \tag{3.7}$$

having only negative roots.

Thus, applying projected collocation to the problem (3.1a), (3.6), (3.3), (3.4) or to its projected invariants reformulation yields a stable, superconvergent numerical method for the original problem (3.1).

3.2. Enforcing the original constraints

While the method described in Section 3.1 yields stability and superconvergence, the original constraints (3.1b) are only satisfied approximately (albeit to a superconvergence order at mesh points). If the precise satisfaction of these constraints is desired, then it can be achieved quite simply by projecting the mesh values of x_π at mesh points (cf. [19]).

Thus, just as we redefine z_m^i in (2.22) to satisfy the index-2 constraints at mesh points, we can redefine

$$z_1^i \leftarrow x_\pi(t_i) + D(t_i)\mu, \tag{3.8}$$

with x_π the projected collocation solution of Section 3.1 at the ith element, and with $\mu = \mu_i$ determined so as to satisfy the constraint (3.1b) at the mesh point t_i. The matrix function $D(t)$ has the dimensions of $B \equiv f_y$ and must satisfy that $g_x D$ is nonsingular for each mesh point t_i. A recommended choice is $D := C^{\mathrm{T}} \equiv g_x^{\mathrm{T}}$ (cf. [6]).

Theorem 3.1. *The projected collocation procedure of Section* 3.1 *plus the additional projection* (3.8) *is well-defined and the obtained solution satisfies the results of Theorem* 2.4.

Proof. The implicit function theorem implies that μ is well-defined in (3.8), given the requirement on D. Moreover, since the exact solution satisfies both (3.1) and (3.6) and since the error is of superconvergence order before applying (3.8), the correction satisfies $\mu_i = O(h^p)$, so the superconvergence order is maintained following (3.8). □

This procedure can be generalized in an obvious way to updating z_j^i so as to satisfy the $(j-1)$st derivative of the constraints (3.1b), $j = 1, \ldots, m$.

Another way to proceed, indeed the first we had considered, would call for a direct projected collocation of (3.1), where at the end of each step the collocated solution (satisfying (2.18) with $z(x_\pi)$ replaced by x_π) is projected by

$$z_j^i \leftarrow x_\pi^{(j-1)}(t_i) + B(t_i)\lambda_j, \quad j = 1, \ldots, m,$$

to satisfy $g^{(l)}(x(t_i), t_i) = 0$, $l = 0, \ldots, m-1$. However, we were not able to prove the full set of results of Theorem 2.4 in this case, so it shall not be considered further.

3.3. Projected invariants II

Consider the ODE

$$x^{(m)} = \hat{f}(z(x), t), \tag{3.9}$$

subject to (3.3), which happens to satisfy (3.1b) as well. An approximate solution of (3.9), (3.3) following discretization, however, does not generally satisfy (3.1b) any more. We wish to obtain an approximate solution of this BVP subject to the constraint (3.1b). The ODE (3.9) has been obtained by (or is equivalent to) the elimination of y from (3.1a) using the mth derivative of (3.1b) as in Section 2.3, or from some independent source.

In all of the methods proposed hitherto for approximating (3.1), a combination of the derivatives of the original constraint, rather than the constraint itself, is collocated. When g_x varies rapidly in t, this may dictate the choice of a smaller step size in order to maintain stability (see [6] and Section 4). Essentially, it is difficult in such a situation to choose a matrix D in the index-2 formulation which would balance the constraint matrix well. To avoid small step sizes, we wish to develop a projected invariants method which projects directly onto the original constraint. This was done in [6] by writing (3.9) as a first-order system in z and modifying

$$z_1' = z_2 + D\mu,$$

requiring (3.1b), where μ is a Lagrange multiplier function and $D(t)$ is as in Section 2.3.

Here, however, we wish to retain the higher-order form of the ODE. Note also that simply adding $D\mu$ to (3.9) (requiring (3.1b)) will not work, because the resulting DAE has index $m+1$.

Hence, we replace the method of [6] as follows. Let

$$z_1 = x + \phi, \qquad z_j = x^{(j-1)}, \quad j = 2, \ldots, m. \tag{3.10}$$

Then a projected invariants formulation is obtained as

$$x^{(m)} = \hat{f}(x+\phi, x', \ldots, x^{(m-1)}, t), \tag{3.11a}$$

$$\phi' = D(t)\mu, \tag{3.11b}$$

$$\mathbf{0} = g(x+\phi, t), \tag{3.11c}$$

with

$$\phi(0) = \mathbf{0}. \tag{3.12}$$

It is easy to verify that for an isolated solution of (3.9), (3.3), there corresponds an isolated solution of (3.11), (3.12), (3.3) with $\phi \equiv \mathbf{0}$, $\mu \equiv \mathbf{0}$.

In (3.11), (3.3), (3.12) we have an index-2 BVP involving no derivatives of the original constraints, with the matrix function D at our disposal (in particular, we can choose $D = g_x^{\mathrm{T}}$), and to which our projected collocation methods may be successfully applied. When the above reformulation considerations are important, this is our method of choice. Note, though, that the DAE system has increased in size.

We note that by replacing (3.11a) with (3.1a) and the mth derivative of (3.1b), we obtain the system

$$x^{(m)} = f(x+\phi, x', \ldots, x^{(m-1)}, y, t), \tag{3.13a}$$

$$\phi' = D(t)\mu, \tag{3.13b}$$

$$\mathbf{0} = g^{(m)}(x+\phi, x', \ldots, x^{(m-1)}, t), \tag{3.13c}$$

$$\mathbf{0} = g(x+\phi, t), \tag{3.13d}$$

which has the same analytical and collocation solutions as (3.11) (in case that (3.9) has been obtained by m constraint differentiations from (3.1) of course), but may lead to a more efficient implementation. The equations (3.11) have the disadvantage that $\hat{f}$ may be expensive to compute because it requires the evaluation of certain projector matrices. In (3.13), although these matrices or equivalent linear algebra computations occur implicitly in the solution of the nonlinear discretized system, they need only be updated/decomposed as often as convergence of the nonlinear iteration requires.

4. Implementation and numerical examples

The above described numerical methods have been implemented using floating-point arithmetic with a 14-hexadecimal-digits mantissa. Thus, in the numerical results tabulated below, roundoff error may be assumed to contaminate values below 10^{-12}, say. For each mesh element $[t_{i-1}, t_i]$ we solve for (i.e., eliminate) the local unknowns w_j and y_j, $j = 1, \ldots, k$, in terms of the mesh values z_{i-1}^j, $j = 1, \ldots, m$, and apply some of the various projections, as the case may be. The local linear system has size $kn_x \times kn_x$ (because the y_j can be eliminated in advance), in contrast to a size of $mkn_x \times mkn_x$ which would be needed had the ODE been converted to a first-order form prior to collocation. For an IVP, the mesh values z_i are computed using a local Newton method. For a BVP we apply a quasilinearization method globally, obtaining a large linear system for all the z_i simultaneously, as usual for a one-step or a multiple-shooting method (see, e.g., [4]). This system is solved using a standard method.

Example 4.1 (Ascher and Petzold [6]). This example is a linear model, at least in form, of a "mechanical system"

$$p' = v, \tag{4.1a}$$

$$M(t)v' = f(p, v, t) - C^{\mathrm{T}}(t)\lambda + q(t), \tag{4.1b}$$

$$0 = C(t)p + r(t), \tag{4.1c}$$

where $M(t)$ is symmetric positive definite. We choose

$$f = \begin{pmatrix} 0 & 0 \\ 0 & \dfrac{\alpha}{(2+t)\nu} \end{pmatrix} v, \qquad C = (1,\ t-2),$$

$$M(t) = \frac{1}{(2+t)\nu^2} \begin{pmatrix} \dfrac{\nu^2 + (\nu-1)^2}{2-t} & -\nu(2\nu-1) \\ -\nu(2\nu-1) & 2(2-t)\nu^2 \end{pmatrix},$$

resulting in

$$B = M^{-1}C^{\mathrm{T}} = \begin{pmatrix} (4-t^2)\nu \\ (\nu-1)(t+2) \end{pmatrix},$$

where ν and α are parameters. The inhomogeneous terms $q(t)$, $r(t)$ and the initial conditions have been chosen so that the solutions for both components of p and v are e^t, and $\lambda = e^t/(2-t)$.

This example yields difficulties in many methods, including traditional stabilization techniques like [15], when $\nu h \gg 1$ (cf. [6]). It can be easily verified that in this case the EUODE is stiff when $\alpha = 1$ and nonstiff when $\alpha = 2$. We consider its solution in two different formulations. In the first formulation, that of Section 3.1, the constraint is differentiated once to yield the system of index 2:

$$p'' = M^{-1}f - B\lambda + q(t), \tag{4.2a}$$

$$0 = Cp' + C'p + r', \tag{4.2b}$$

with $p_1(0) = p_2(0) = p_1'(0) = p_2'(0) = 1$. A variation of this consists of (4.2) plus the original constraint (4.1c), which we solve by projecting the (projected) collocation solution of (4.2) at the end of each step to satisfy (4.1c), as described in Section 3.2. We use C^{T} for the latter projection matrix.

In Tables 4.1 and 4.2 we present numerical results for various cases of projected and unprojected Gauss collocation (denoted Proj-v) and Radau collocation for (4.2), as well as results when applying an additional projection on the original constraint (denoted Proj-p). We write "y" if the projection is used and "n" if not. Only uniform meshes (with N subintervals) are used, and we record maximum error at mesh points in p_1 and v_1 and calculate the convergence rates as the mesh is refined in each case as well. Under "drift" we also list the defect in the original constraint (4.1c).

The claimed convergence results are clearly demonstrated in Table 4.1 for $\nu = 1$. The loss of superconvergence in the unprojected Gauss collocation is restricted to p', as discussed already

Table 4.1
Example 4.1, projected collocation

(ν, α)	Method	k	N	Proj-v?	Proj-p?	error(p_1)	rate(p)	error(v_1)	rate(v)	Drift
(1, 1)	Radau	2	5		n	$0.28\cdot10^{-3}$		$0.85\cdot10^{-4}$		$0.34\cdot10^{-4}$
			10		n	$0.34\cdot10^{-4}$	3.0	$0.10\cdot10^{-4}$	3.1	$0.38\cdot10^{-5}$
			20		n	$0.42\cdot10^{-5}$	3.0	$0.12\cdot10^{-5}$	3.0	$0.44\cdot10^{-6}$
	Gauss	2	5	n	n	$0.43\cdot10^{-5}$		$0.81\cdot10^{-3}$		$0.29\cdot10^{-5}$
			10	n	n	$0.27\cdot10^{-6}$	4.0	$0.20\cdot10^{-3}$	2.0	$0.18\cdot10^{-6}$
			20	n	n	$0.17\cdot10^{-7}$	4.0	$0.50\cdot10^{-4}$	2.0	$0.11\cdot10^{-7}$
	Gauss	2	5	y	n	$0.43\cdot10^{-5}$		$0.37\cdot10^{-5}$		$0.29\cdot10^{-5}$
			10	y	n	$0.27\cdot10^{-6}$	4.0	$0.23\cdot10^{-6}$	4.0	$0.18\cdot10^{-6}$
			20	y	n	$0.17\cdot10^{-7}$	4.0	$0.14\cdot10^{-7}$	4.0	$0.11\cdot10^{-7}$
	Radau	3	5		n	$0.76\cdot10^{-7}$		$0.68\cdot10^{-7}$		$0.43\cdot10^{-7}$
			10		n	$0.24\cdot10^{-8}$	5.0	$0.22\cdot10^{-8}$	5.0	$0.13\cdot10^{-8}$
			20		n	$0.75\cdot10^{-10}$	5.0	$0.68\cdot10^{-10}$	5.0	$0.42\cdot10^{-10}$
	Gauss	3	5	n	n	$0.18\cdot10^{-8}$		$0.33\cdot10^{-5}$		$0.36\cdot10^{-9}$
			10	n	n	$0.29\cdot10^{-10}$	6.0	$0.21\cdot10^{-6}$	4.0	$0.56\cdot10^{-11}$
			20	n	n	$0.45\cdot10^{-12}$	6.0	$0.13\cdot10^{-7}$	4.0	$0.86\cdot10^{-13}$
	Gauss	3	5	y	n	$0.18\cdot10^{-8}$		$0.18\cdot10^{-8}$		$0.36\cdot10^{-9}$
			10	y	n	$0.29\cdot10^{-10}$	6.0	$0.28\cdot10^{-10}$	6.0	$0.57\cdot10^{-11}$
			20	y	n	$0.45\cdot10^{-12}$	6.0	$0.44\cdot10^{-12}$	6.0	$0.88\cdot10^{-13}$
(50, 1)	Radau	2	10		n	$0.63\cdot10^{-4}$		$0.17\cdot10^{-2}$		$0.43\cdot10^{-5}$
			20		n	$0.12\cdot10^{+1}$	−14	$0.41\cdot10^{+2}$	−14	$0.99\cdot10^{-2}$
			40		n	$0.12\cdot10^{-5}$	20	$0.30\cdot10^{-4}$	20	$0.15\cdot10^{-6}$
			80		n	$0.10\cdot10^{-6}$	3.6	$0.24\cdot10^{-5}$	3.7	$0.99\cdot10^{-8}$
	Radau	2	10		y	$0.63\cdot10^{-4}$		$0.17\cdot10^{-2}$		$0.44\cdot10^{-15}$
			20		y	$0.96\cdot10^{0}$	−14	$0.32\cdot10^{+2}$	−14	$0.44\cdot10^{-15}$
			40		y	$0.11\cdot10^{-5}$	20	$0.30\cdot10^{-4}$	20	$0.89\cdot10^{-15}$
			80		y	$0.97\cdot10^{-7}$	3.5	$0.24\cdot10^{-5}$	3.7	$0.89\cdot10^{-15}$
	Gauss	2	10	n	n	$0.48\cdot10^{+6}$		$0.35\cdot10^{+8}$		$0.18\cdot10^{-6}$
			20	n	n	$0.25\cdot10^{+20}$	−45	$0.53\cdot10^{+22}$	−47	$0.82\cdot10^{+4}$
			40	n	n	$0.94\cdot10^{+13}$	21	$0.69\cdot10^{+16}$	20	$0.31\cdot10^{-1}$
			80	n	n	$0.16\cdot10^{+12}$	5.9	$0.44\cdot10^{+15}$	4.0	$0.73\cdot10^{-3}$
	Gauss	2	10	y	n	$0.35\cdot10^{-3}$		$0.18\cdot10^{-1}$		$0.18\cdot10^{-6}$
			20	y	n	$0.51\cdot10^{-5}$	6.1	$0.26\cdot10^{-3}$	6.1	$0.11\cdot10^{-7}$
			40	y	n	$0.71\cdot10^{-7}$	6.2	$0.23\cdot10^{-5}$	6.8	$0.71\cdot10^{-9}$
			80	y	n	$0.33\cdot10^{-8}$	4.4	$0.99\cdot10^{-7}$	4.5	$0.44\cdot10^{-10}$
	Radau	3	10		n	$0.17\cdot10^{-6}$		$0.46\cdot10^{-5}$		$0.13\cdot10^{-8}$
			20		n	$0.20\cdot10^{-6}$	−0.3	$0.65\cdot10^{-5}$	−0.5	$0.42\cdot10^{-10}$
			40		n	$0.17\cdot10^{-9}$	10	$0.59\cdot10^{-8}$	10	$0.13\cdot10^{-11}$
			80		n	$0.38\cdot10^{-11}$	5.5	$0.11\cdot10^{-9}$	5.7	$0.41\cdot10^{-13}$
	Gauss	3	10	n	n	$0.13\cdot10^{+2}$		$0.31\cdot10^{+4}$		$0.56\cdot10^{-11}$
			20	n	n	$0.73\cdot10^{+16}$	−49	$0.77\cdot10^{+19}$	−51	$0.24\cdot10^{+2}$
			40	n	n	$0.14\cdot10^{+9}$	26	$0.10\cdot10^{+13}$	23	$0.14\cdot10^{-5}$
			80	n	n	$0.86\cdot10^{+5}$	10.7	$0.65\cdot10^{+10}$	7.3	$0.19\cdot10^{-8}$
	Gauss	3	10	y	n	$0.89\cdot10^{-7}$		$0.23\cdot10^{-5}$		$0.56\cdot10^{-11}$
			20	y	n	$0.35\cdot10^{-7}$	1.3	$0.14\cdot10^{-5}$	0.7	$0.89\cdot10^{-13}$
			40	y	n	$0.10\cdot10^{-10}$	11.7	$0.38\cdot10^{-9}$	11.8	$0.89\cdot10^{-15}$
			80	y	n	$0.12\cdot10^{-12}$	6.5	$0.38\cdot10^{-11}$	6.6	$0.13\cdot10^{-14}$

Table 4.2
Example 4.1, projected collocation continued

(ν, α)	Method	k	N	Proj-v?	Proj-p?	error(p_1)	rate(p)	error(v_1)	rate(v)	Drift
(50, 2)	Radau	2	10		n	$0.35\cdot10^{-3}$		$0.18\cdot10^{-2}$		$0.20\cdot10^{-5}$
			20		n	$0.24\cdot10^{-3}$	0.6	$0.23\cdot10^{-3}$	3.0	$0.15\cdot10^{-5}$
			40		n	$0.72\cdot10^{-4}$	1.7	$0.28\cdot10^{-4}$	3.0	$0.31\cdot10^{-6}$
			80		n	$0.35\cdot10^{-5}$	4.4	$0.35\cdot10^{-5}$	3.0	$0.10\cdot10^{-7}$
	Radau	2	10		y	$0.38\cdot10^{-3}$		$0.19\cdot10^{-2}$		$0.44\cdot10^{-15}$
			20		y	$0.23\cdot10^{-3}$	0.7	$0.26\cdot10^{-3}$	2.8	$0.44\cdot10^{-15}$
			40		y	$0.69\cdot10^{-4}$	1.7	$0.22\cdot10^{-4}$	3.6	$0.44\cdot10^{-15}$
			80		y	$0.34\cdot10^{-5}$	4.3	$0.33\cdot10^{-5}$	2.7	$0.89\cdot10^{-15}$
	Gauss	2	10	y	n	$0.41\cdot10^{-4}$		$0.11\cdot10^{-4}$		$0.18\cdot10^{-6}$
			20	y	n	$0.26\cdot10^{-5}$	4.0	$0.69\cdot10^{-6}$	4.0	$0.11\cdot10^{-7}$
			40	y	n	$0.16\cdot10^{-6}$	4.0	$0.43\cdot10^{-7}$	4.0	$0.71\cdot10^{-9}$
			80	y	n	$0.10\cdot10^{-7}$	4.0	$0.27\cdot10^{-8}$	4.0	$0.44\cdot10^{-10}$
	Radau	3	10		n	$0.37\cdot10^{-6}$		$0.78\cdot10^{-7}$		$0.13\cdot10^{-8}$
			20		n	$0.11\cdot10^{-7}$	5.0	$0.24\cdot10^{-8}$	5.0	$0.42\cdot10^{-10}$
			40		n	$0.36\cdot10^{-9}$	5.0	$0.77\cdot10^{-10}$	5.0	$0.13\cdot10^{-11}$
			80		n	$0.11\cdot10^{-10}$	5.0	$0.20\cdot10^{-11}$	5.2	$0.40\cdot10^{-13}$
	Gauss	3	10	y	n	$0.18\cdot10^{-8}$		$0.32\cdot10^{-9}$		$0.56\cdot10^{-11}$
			20	y	n	$0.28\cdot10^{-10}$	6.0	$0.59\cdot10^{-11}$	5.8	$0.89\cdot10^{-13}$
			40	y	n	$0.36\cdot10^{-12}$	6.3	$0.44\cdot10^{-12}$	3.8	$0.49\cdot10^{-14}$
			80	y	n	$0.11\cdot10^{-12}$	1.6	$0.30\cdot10^{-12}$	0.6	$0.18\cdot10^{-14}$

in [5]. But the unprojected Gauss collocation performs much worse (indeed, unacceptably poorly) when $\nu = 50$ or larger (cf. [5, Example 1]). Corresponding results are not recorded in Table 4.2 because the involved local matrix became numerically rank-deficient. The projected Gauss collocation results are comparable to or better than the corresponding results for Radau collocation with the same number of collocation points, as the theory asserts. When νh is large, the projected methods do not always perform acceptably well, especially when $\alpha = 1$ and when using Radau collocation.

For this example, the additional projection on the original constraint does not appear to be particularly helpful. (We have recorded results only for cases with the largest drift, where the projection is expected to have its largest effect.)

The second formulation of this example is in the projected invariants form (2.28), (3.5) described in Sections 2.3 and 3.1. Here (2.27) reads

$$p'' = (HA_2 - 2FC')p' + HM^{-1}q - Fr'', \tag{4.3}$$

where

$$HA_2 - 2FC' = \begin{pmatrix} 0 & (\alpha-2)\nu \\ 0 & \dfrac{(\alpha-2)\nu + 2}{2-t} \end{pmatrix},$$

and (2.3b) is (4.2b). For this reformulation it is apparent that the ODE (4.3) is stiff for $\alpha = 1$ (and $\nu h \gg 1$) and nonstiff when $\alpha = 2$. Computed results are recorded in Table 4.3. They are all very good and agree with the theory.

Table 4.3
Example 4.1, projected invariants I

(ν, α)	Method	k	N	Proj-v?	Proj-p?	error(p_1)	rate(p)	error(v_1)	rate(v)	Drift
(50, 1)	Radau	2	10		n	$0.42\cdot10^{-5}$		$0.20\cdot10^{-4}$		$0.34\cdot10^{-5}$
			20		n	$0.50\cdot10^{-6}$	3.1	$0.35\cdot10^{-5}$	2.6	$0.42\cdot10^{-6}$
			40		n	$0.61\cdot10^{-7}$	3.0	$0.53\cdot10^{-6}$	2.7	$0.53\cdot10^{-7}$
			80		n	$0.75\cdot10^{-8}$	3.0	$0.74\cdot10^{-7}$	2.8	$0.66\cdot10^{-8}$
	Gauss	2	10	y	n	$0.37\cdot10^{-6}$		$0.29\cdot10^{-4}$		$0.18\cdot10^{-6}$
			20	y	n	$0.25\cdot10^{-7}$	3.9	$0.19\cdot10^{-5}$	3.9	$0.11\cdot10^{-7}$
			40	y	n	$0.16\cdot10^{-8}$	4.0	$0.12\cdot10^{-6}$	4.0	$0.71\cdot10^{-9}$
			80	y	n	$0.98\cdot10^{-10}$	4.0	$0.75\cdot10^{-8}$	4.0	$0.44\cdot10^{-10}$
	Radau	3	10		n	$0.19\cdot10^{-8}$		$0.14\cdot10^{-6}$		$0.13\cdot10^{-8}$
			20		n	$0.85\cdot10^{-10}$	4.5	$0.64\cdot10^{-8}$	4.5	$0.42\cdot10^{-10}$
			40		n	$0.32\cdot10^{-11}$	4.7	$0.24\cdot10^{-9}$	4.8	$0.13\cdot10^{-11}$
			80		n	$0.11\cdot10^{-12}$	4.8	$0.79\cdot10^{-11}$	4.9	$0.38\cdot10^{-13}$
	Gauss	3	10	y	n	$0.16\cdot10^{-8}$		$0.78\cdot10^{-7}$		$0.56\cdot10^{-11}$
			20	y	n	$0.32\cdot10^{-10}$	5.6	$0.16\cdot10^{-8}$	5.6	$0.88\cdot10^{-13}$
			40	y	n	$0.54\cdot10^{-12}$	5.9	$0.27\cdot10^{-10}$	5.9	$0.18\cdot10^{-14}$
			80	y	n	$0.76\cdot10^{-14}$	6.1	$0.44\cdot10^{-12}$	6.0	$0.27\cdot10^{-14}$
(50, 2)	Radau	2	10		n	$0.27\cdot10^{-4}$		$0.18\cdot10^{-5}$		$0.36\cdot10^{-5}$
			20		n	$0.34\cdot10^{-5}$	3.0	$0.22\cdot10^{-6}$	3.0	$0.43\cdot10^{-6}$
			40		n	$0.42\cdot10^{-6}$	3.0	$0.27\cdot10^{-7}$	3.0	$0.53\cdot10^{-7}$
			80		n	$0.52\cdot10^{-7}$	3.0	$0.33\cdot10^{-8}$	3.0	$0.66\cdot10^{-8}$
	Gauss	2	10	y	n	$0.39\cdot10^{-7}$		$0.29\cdot10^{-6}$		$0.18\cdot10^{-6}$
			20	y	n	$0.24\cdot10^{-8}$	4.0	$0.18\cdot10^{-7}$	4.0	$0.11\cdot10^{-7}$
			40	y	n	$0.15\cdot10^{-9}$	4.0	$0.11\cdot10^{-8}$	4.0	$0.71\cdot10^{-9}$
			80	y	n	$0.94\cdot10^{-11}$	4.0	$0.71\cdot10^{-10}$	4.0	$0.44\cdot10^{-10}$
	Radau	3	10		n	$0.27\cdot10^{-9}$		$0.25\cdot10^{-8}$		$0.13\cdot10^{-8}$
			20		n	$0.79\cdot10^{-11}$	5.1	$0.78\cdot10^{-10}$	5.0	$0.42\cdot10^{-10}$
			40		n	$0.24\cdot10^{-12}$	5.0	$0.25\cdot10^{-11}$	5.0	$0.13\cdot10^{-11}$
			80		n	$0.87\cdot10^{-14}$	4.8	$0.78\cdot10^{-13}$	5.0	$0.40\cdot10^{-13}$
	Gauss	3	10	y	n	$0.49\cdot10^{-10}$		$0.48\cdot10^{-10}$		$0.56\cdot10^{-11}$
			20	y	n	$0.77\cdot10^{-12}$	6.0	$0.75\cdot10^{-12}$	6.0	$0.88\cdot10^{-13}$
			40	y	n	$0.13\cdot10^{-13}$	5.9	$0.12\cdot10^{-13}$	5.9	$0.18\cdot10^{-14}$

For this example we do not expect the projected invariants method II of Section 3.3 to have any advantage over method I of Section 2.3 (although it certainly is not worse), because C' is constant and not problematic. Thus, the combined constraint matrix for (4.1c) and (4.2b) is balanced well by using a projection with C^T. This situation changes in the next example.

Example 4.2. In (3.2), let $m=2$, $A_1=0$, $A_2=\alpha I$, $B^T=C=(\sin \nu t, \cos \nu t)$, with initial conditions specified and the inhomogeneities chosen so that the solution is the same as in Example 4.1. Also as before, α and ν are parameters.

We first investigate the problem. Choosing $R=(\cos \nu t, -\sin \nu t)$ we have $S=R^T$, $F=B$, and the EUODE is

$$u''=\alpha u'+\nu^2 u.$$

Table 4.4
Example 4.2, projected collocation and projected invariants methods

ν	Reformulation	Method	k	N	error(x_1)	error(x_1')
1	Direct	Radau	2	10	$0.57\cdot10^{-5}$	$0.78\cdot10^{-5}$
				40	$0.95\cdot10^{-7}$	$0.11\cdot10^{-6}$
	PiI	Radau	2	10	$0.57\cdot10^{-5}$	$0.78\cdot10^{-5}$
				40	$0.95\cdot10^{-7}$	$0.11\cdot10^{-6}$
	PiI+	Radau	2	10	$0.10\cdot10^{-4}$	$0.88\cdot10^{-5}$
				40	$0.16\cdot10^{-6}$	$0.13\cdot10^{-6}$
	PiII	Radau	2	10	$0.38\cdot10^{-5}$	$0.20\cdot10^{-4}$
				40	$0.64\cdot10^{-7}$	$0.29\cdot10^{-6}$
	Direct	Gauss	2	10	$0.48\cdot10^{-6}$	$0.77\cdot10^{-6}$
				40	$0.19\cdot10^{-8}$	$0.29\cdot10^{-8}$
	PiI	Gauss	2	10	$0.48\cdot10^{-6}$	$0.77\cdot10^{-6}$
				40	$0.19\cdot10^{-8}$	$0.29\cdot10^{-8}$
	PiI+	Gauss	2	10	$0.43\cdot10^{-6}$	$0.80\cdot10^{-6}$
				40	$0.17\cdot10^{-8}$	$0.31\cdot10^{-8}$
	PiII	Gauss	2	10	$0.66\cdot10^{-7}$	$0.75\cdot10^{-6}$
				40	$0.30\cdot10^{-9}$	$0.29\cdot10^{-8}$
100	Direct	Radau	2	10	$0.26\cdot10^{+4}$	$0.44\cdot10^{+5}$
				40	$0.77\cdot10^{-8}$	$0.46\cdot10^{-6}$
	PiI	Radau	2	10	$0.26\cdot10^{+4}$	$0.44\cdot10^{+5}$
				40	$0.77\cdot10^{-8}$	$0.46\cdot10^{-6}$
	PiI+	Radau	2	10	0.88	$0.54\cdot10^{+2}$
				40	$0.17\cdot10^{-7}$	$0.14\cdot10^{-4}$
	PiII	Radau	2	10	$0.43\cdot10^{-5}$	$0.67\cdot10^{-4}$
				40	$0.42\cdot10^{-7}$	$0.16\cdot10^{-5}$
	Direct	Gauss	2	10	$0.10\cdot10^{+1}$	$0.12\cdot10^{+3}$
				40	$0.42\cdot10^{-4}$	$0.25\cdot10^{-2}$
	PiI	Gauss	2	10	$0.10\cdot10^{+1}$	$0.12\cdot10^{+3}$
				40	$0.42\cdot10^{-4}$	$0.25\cdot10^{-2}$
	PiI+	Gauss	2	10	$0.15\cdot10^{+2}$	$0.16\cdot10^{+4}$
				40	$0.32\cdot10^{-8}$	$0.13\cdot10^{-5}$
	PiII	Gauss	2	10	$0.10\cdot10^{-4}$	$0.42\cdot10^{-4}$
				40	$0.12\cdot10^{-7}$	$0.46\cdot10^{-6}$

Thus, if $\alpha \approx 1$ and $\nu \gg 1$, then the problem is unstable (as an IVP). In the following we choose $\alpha = -\nu^2$, which gives a stable, stiff EUODE when ν^2 is large. Note also that $\|C'\| = \nu \|R\| = \nu \|C\| = \nu$.

For the results in Table 4.4 we have applied projected collocation at Gauss or Radau points as specified, using for some instances coarse meshes, as follows: (i) directly to (3.2a) and (3.2b) once differentiated, (ii) the projected invariants method I (denoted PiI) of Section 2.3, (iii) the projected invariants method I of Section 2.3 coupled with the additional projection back onto the original constraints of Section 3.2 (denoted PiI +), and (iv) the projected invariants method II (denoted PiII) of Section 3.3. (So, the original constraints are enforced both in PiI + and in PiII, the difference being that the derivatives of the constraint appear in PiII only through the ODE.)

For $\nu = 1$ there is no stiffness and C varies slowly. All four methods yield essentially the same results (with the expected superconvergence rates). For $\nu = 100$ and $\nu h \ll 1$, the four methods also yield comparable results (not recorded). But when $h = 0.1$, the direct projected collocation method does not do very well. The first projected invariants method I, which has made a big difference in the previous example, does not help here at all, because $B = C^T$ gives no trouble to begin with. The additional projection of x to enforce the original constraints gives somewhat erratic "improvements". On the other hand, the second projected invariants method II, which projects directly onto the original constraints, does much better for a coarse h (cf. [6, Examples 1 and 4]). This is the most robust reformulation and discretization method among those which we have considered.

References

[1] U. Ascher, Collocation for two-point boundary value problems revisited, *SIAM J. Numer. Anal.* **23** (1986) 596–609.

[2] U. Ascher, On numerical differential algebraic problems with application to semiconductor device simulation, *SIAM J. Numer. Anal.* **26** (1989) 517–538.

[3] U. Ascher, J. Christiansen and R. Russell, Collocation software for boundary value ODEs, *ACM Trans. Math. Software* **7** (1981) 209–222.

[4] U. Ascher, R. Mattheij and R. Russell, *Numerical Solution of Boundary Value Problems for Ordinary Differential Equation* (Prentice-Hall, Englewood Cliffs, NJ, 1988).

[5] U. Ascher and L. Petzold, Projected implicit Runge–Kutta methods for differential-algebraic equations, *SIAM J. Numer. Anal.* **28** (1991) 1097–1120.

[6] U. Ascher and L. Petzold, Stability of computational methods for constrained dynamics systems, TR 91-3, Dept. Comput. Sci., Univ. British Columbia, Vancouver, 1991; also: *SIAM J. Sci. Statist. Comput.*, to appear.

[7] J. Baumgarte, Stabilization of constraints and integrals of motion in dynamical systems, *Comput. Methods Appl. Mech. Engrg.* **1** (1) (1972) 1–16.

[8] B.P. Boudreau, A steady-state diagenetic model for dissolved carbonate species and pH in the porewaters of oxic and suboxic sediments, *Geochim. Cosmochim. Acta* **51** (1987).

[9] B.P. Boudreau and D.E. Canfield, A provisional diagenetic model for pH in anoxic porewaters: Application to the FOAM site, *J. Marine Res.* **46** (1988) 429–455.

[10] K.F. Brenan, S.L. Campbell and L.R. Petzold, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations* (Elsevier, New York, 1989).

[11] C. de Boor and B. Swartz, Collocation at Gaussian points, *SIAM J. Numer. Anal.* **10** (1973) 582–606.

[12] K. Führer and B. Leimkuhler, Numerical solution of differential-algebraic equations for constrained mechanical motion, *Numer. Math.* **59** (1991) 55–69.

[13] C.W. Gear, Maintaining solution invariants in the numerical solution of ODEs, *SIAM J. Sci. Statist. Comput.* **7** (1986) 734–743.

[14] C.W. Gear, Differential-algebraic equation index transformations, *SIAM J. Sci. Statist. Comput.* **9** (1988) 39–47.

[15] C.W. Gear, G.K. Gupta and B. Leimkuhler, Automatic integration of Euler–Lagrange equations with constraints, *J. Comput. Appl. Math.* **12&13** (1985) 77–90.

[16] E. Griepentrog and R. Marz, *Differential-Algebraic Equations and their Numerical Treatment*, Teubner-Texte Math. **88** (Teubner, Leipzig, 1986).

[17] E. Hairer, Ch. Lubich and M. Roche, *The Numerical Solution of Differential-Algebraic Systems by Runge–Kutta Methods*, Lecture Notes in Math. **1409** (Springer, Berlin, 1989).

[18] J.S. Logsdon and L.T. Biegler, Accurate solution of differential-algebraic optimization problems, *Indust. Engrg. Chem. Res.* **28** (1989) 1628–1639.

[19] Ch. Lubich, On projected Runge–Kutta methods for differential-algebraic equations, manuscript, 1990.

[20] R.A. Wehage and E.J. Haug, Generalized coordinate partitioning for dimension reduction in analysis of constrained dynamic systems, *J. Mech. Design* **104** (1982) 247–255.