Projected Implicit Runge-Kutta Methods for Differential-Algebraic Equations

by

Uri M. Ascher

Linda R. Petzold

Technical Report 90-20
May 25, 1990

Abstract

In this paper we introduce a new class of numerical methods, *Projected Implicit Runge-Kutta methods*, for the solution of index-two Hessenberg systems of initial and boundary value differential-algebraic equations (DAEs). These types of systems arise in a variety of applications, including the modelling of singular optimal control problems and parameter estimation for differential-algebraic equations such as multibody systems. The new 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. 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 linear problems we define an essential underlying boundary value ODE and prove well-conditioning of the differential (or state-space) solution components. This is then used to prove stability and superconvergence for the corresponding numerical approximations for linear and nonlinear problems.

Uri M. Ascher is at the Department of Computer Science, University of British Columbia, Vancouver, B.C. V6T 1W5, Canada. His work was partially supported under NSERC Canada Grant OGP0004306.

Linda R. Petzold is at the Computing & Mathematics Research Division, Lawrence Livermore National Laboratory, L-316, Livermore, California 94550. Her work 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

Much attention has recently been devoted to the development of numerical methods for differential-algebraic equations (DAEs). It appears that the direct approximation of initial value problems (IVPs) can be satisfactorily achieved, using for example BDF schemes [9] or certain implicit Runge-Kutta methods [17], for fully-implicit index-1 DAEs and for certain restricted classes of higher index problems as well. For boundary value problems (BVPs) the situation is much less clear, even for the fully-implicit index-1 case [16, 1, 2, 12]. The difficulty is that the possible occurrence of increasing solution modes for stable BVPs may render one-sided difference schemes such as BDF unstable, unless upwinding is used. The latter is, however, a possibly expensive and cumbersome procedure in the context of a general purpose code. Symmetric one-leg schemes, which in practice are often very successful for stiff BVPs [3, 4], can have severe limitations including instability, oscillation and loss of accuracy when applied to DAEs [1, 2].

In this paper, we will consider DAEs of the form

$$\mathbf{x}' = \mathbf{g}_1(\mathbf{x}, \mathbf{y}, t) \tag{1.1a}$$

$$\mathbf{0} = \mathbf{g}_2(\mathbf{x}, t) \tag{1.1b}$$

where $(\partial \mathbf{g}_2/\partial \mathbf{x})(\partial \mathbf{g}_1/\partial \mathbf{y})$ is assumed to be nonsingular for all t, $\mathbf{x}$, $\mathbf{y}$ in a neighborhood of the solution. Systems of this form are often referred to as *Hessenberg index-2 DAEs* [9]. Many problems of engineering and scientific interest occur naturally or can easily be rewritten in this form. The time-dependent incompressible Navier-Stokes equations, and the charge-neutral semiconductor device equations are in this form following a spatial discretization. Other problems, with index higher than two, can easily be brought into this form by differentiating the constraints; the original constraints remain in the system and are enforced by means of additional Lagrange multipliers. This approach, which was introduced in Gear [15], is aimed at preserving the stability of the original system by enforcing the constraints [14], and is often convenient to implement because for many systems of physical interest the constraint derivative is readily available. Multibody systems [17] fall into this category. So do certain boundary value problems which arise when modelling chemical reactions [7, 8]. Optimal control or parameter estimation for multibody systems can be written as a boundary value problem of the form (1.1) [19, 5]. Similarly, trajectory prescribed path control problems [9] lead to boundary value problems of the form (1.1).

Although the index-2 system (1.1) can in principle be rewritten as an index-1 system or as an ODE through repeated differentiations of the constraints, this requires an additional amount of differentiability in the problem, the necessary derivatives are often not readily available, and there may be a loss of stability for some problems [14, 11] if the constraints are not enforced. Alternatively, the system (1.1) can be rewritten as a smaller set of *state-space ODEs* [20, 14] for which the constraints are

always satisfied; this approach is often used for the solution of initial-value multibody systems. However, the set of state-space variables in which the resulting system is expressed may need to change depending on the solution, leading to difficulties especially in the BVP case. Thus we are motivated to consider the direct solution of Hessenberg index-2 DAEs.

Consider an implicit Runge-Kutta (IRK) method applied to (1.1)

$$\mathbf{X}_i' = \mathbf{g}_1(\mathbf{X}_i, \mathbf{Y}_i, t_i) \tag{1.2a}$$

$$\mathbf{0} = \mathbf{g}_2(\mathbf{X}_i, t_i), \quad i = 1, 2, \ldots, k \tag{1.2b}$$

$$\hat{\mathbf{x}}_n = \hat{\mathbf{x}}_{n-1} + h_n \sum_{j=1}^{k} b_j \mathbf{X}_j' \tag{1.2c}$$

where $\mathbf{X}_i = \hat{\mathbf{x}}_{n-1} + h_n \sum_{j=1}^{k} a_{ij} \mathbf{X}_j'$. It is well known that IRK methods (1.2) can exhibit order reduction when applied to DAEs. The reduction of order can be particularly severe in the case of superconvergent symmetric methods: for example, constant-stepsize k-stage Gauss collocation, which has order $2k$ for nonstiff ODEs, yields order $k+q$ for $\mathbf{x}$ and $k+q-1$ for $\mathbf{y}$, where $q = 1$ (k odd) or $q = 0$ (k even), when applied to index-2 Hessenberg systems (1.1). In addition, there are restrictions on the mesh to obtain this order for $q = 1$. If we require more continuity of $\mathbf{y}$, i.e. we set

$$\hat{\mathbf{y}}_n = \hat{\mathbf{y}}_{n-1} + h_n \sum_{j=1}^{k} b_j \mathbf{Y}_j'$$

$$\mathbf{Y}_i = \hat{\mathbf{y}}_{n-1} + h_n \sum_{j=1}^{k} a_{ij} \mathbf{Y}_j'$$

then the order in $\mathbf{y}$ drops further to $k+q-2$ [17]. A potentially more severe problem for symmetric methods is instability. Ascher[1] has shown that symmetric methods applied to fully-implicit index-1 DAEs can be unstable, in the sense that the stability is governed by a 'ghost ODE' which is determined in part by time-dependent coupling in the system and may not be stable for well-conditioned systems. Gear [15] has noted that there is a close relationship between semi-explicit index-2 systems and fully-implicit index-1 systems; hence one might expect that symmetric schemes applied to well-conditioned problems of the form (1.1) could sometimes be unstable. We will show that this is indeed the case.

To overcome these difficulties, we introduce a new class of numerical methods, *Projected Implicit Runge-Kutta (PIRK) methods.* To define these methods, let $\hat{\mathbf{x}}_n$ be given by the IRK method (1.2) where $\hat{\mathbf{x}}_{n-1} = \mathbf{x}_{n-1}$. Let

$$\mathbf{x}_n = \hat{\mathbf{x}}_n + G_{12} \lambda_n \tag{1.3}$$

where $G_{12} = \partial \mathbf{g}_1 / \partial \mathbf{y}$, and λ_n is determined by the requirement that

$$\mathbf{g}_2(\mathbf{x}_n, t_n) = 0. \tag{1.4}$$

This defines the PIRK method for $\mathbf{x}$. The solution for $\mathbf{y}$ can be determined from the solution for $\mathbf{x}$, and to the same order of accuracy, via a post-processing step.

While we will deal in this paper only with the Hessenberg index-2 system (1.1), we note that there is a straightforward extension of the PIRK methods to systems with a combination of index-one constraints and index-two constraints, namely

$$\mathbf{x}' = \mathbf{g}_1(\mathbf{x}, \mathbf{y}, \mathbf{z}, t) \tag{1.5a}$$
$$0 = \mathbf{g}_2(\mathbf{x}, \mathbf{y}, t) \tag{1.5b}$$
$$0 = \mathbf{g}_3(\mathbf{x}, t) \tag{1.5c}$$

where $\partial \mathbf{g}_2/\partial \mathbf{y}$ and $(\partial \mathbf{g}_3/\partial \mathbf{x})(\partial \mathbf{g}_1/\partial \mathbf{z})$ are both nonsingular. Then the projected method is given by

$$\mathbf{x}_n = \hat{\mathbf{x}}_n + G_{13}\lambda_n, \tag{1.6}$$

where $G_{13} = \partial \mathbf{g}_1/\partial \mathbf{z}$, and the constraints are required to be satisfied at t_n. The properties of the method remain unchanged for this extended class of problems.

In Section 2, we consider in some detail the question of conditioning of a BVP for the linear index-2 DAE (2.1a), (2.1b) defined below. We derive the underlying ODE which propagates the information in the system and from which the conditioning of the system can be deduced. We give a stability result which shows that, while higher index systems are in general ill-posed in the classical sense [16], the ill-posedness in the DAE (1.1) is concentrated in $\mathbf{y}$, while for $\mathbf{x}$ a well-posed problem may be retrieved. We give a well-conditioned example which shows that a careless index reduction procedure can be disastrous. For this same example, standard (unprojected) Gaussian collocation methods become unstable. This can be seen from the analysis and is also demonstrated by an experiment in Section 4.

In Section 3 we use the analytical tools developed in Section 2 to give a stability analysis for the projected IRK methods which shows that they are stable, with a stability constant close to that of the underlying BVP. We then restrict ourselves to collocation methods and show that their nonstiff superconvergence properties are retrieved with the projection (1.3), (1.4). This result has practical significance because symmetric collocation methods form the basis for the well-known code COLSYS [3] for boundary value ODEs. In Section 4 we present some numerical examples.

We note that if the original IRK method already satisfies (1.4), then $\lambda_n = 0$ and the projected method coincides with the usual one. The main practical contribution of this paper is therefore expected to be in the numerical solution of BVPs, and in particular in the performance improvement of symmetric IRK schemes.

Throughout this paper we use the following notation: Let $|\cdot|$ be the Euclidean vector norm. For a matrix A we denote the induced matrix norm by $\|A\|$. For a function $\mathbf{u}(t)$, $a \le t \le b$, we denote the corresponding max function norm by $\|\mathbf{u}\| := \max\{|\mathbf{u}(t)|, a \le t \le b\}$. The limits a and b will be understood from the context. Also, $\mathcal{P}_k[a,b]$ stands for the class of polynomials of order k on $[a,b]$.

2 Problem conditioning

It is well-known (see e.g. [16], [1]) that DAE problems with index exceeding 1 are in a sense ill-posed. Hence it is important to investigate the conditioning (stability) of such problems carefully. Consider the linear BVP

$$\begin{aligned} \mathbf{x}' &= G_{11}\mathbf{x} + G_{12}\mathbf{y} + \mathbf{q}_1 && \text{(2.1a)} \\ \mathbf{0} &= G_{21}\mathbf{x} + \mathbf{q}_2 && \text{(2.1b)} \\ \beta &= B_0\mathbf{x}(0) + B_1\mathbf{x}(1) && \text{(2.1c)} \end{aligned}$$

where G_{11}, G_{12} and G_{21} are smooth functions of t, $0 \le t \le 1$, $G_{11}(t) \in \mathcal{R}^{m_x \times m_x}$, $G_{12}(t) \in \mathcal{R}^{m_x \times m_y}$, $G_{21}(t) \in \mathcal{R}^{m_y \times m_x}$, $m_y \le m_x$, $G_{21}G_{12}$ is nonsingular for each t (hence the DAE has index 2), and $B_0, B_1 \in \mathcal{R}^{(m_x - m_y)\times m_x}$. All matrices involved are assumed to be uniformly bounded in norm by a constant of moderate size. The inhomogeneities are $\mathbf{q}_1(t) \in \mathcal{R}^{m_x}$, $\mathbf{q}_2(t) \in \mathcal{R}^{m_y}$, $\beta \in \mathcal{R}^{m_x - m_y}$.

We seek conditions under which this BVP is guaranteed to be well-conditioned (stable) in an appropriate sense. Since $G_{21}G_{12}$ is nonsingular, G_{12} has full rank. Hence there exists a smooth, bounded matrix function $R(t) \in \mathcal{R}^{(m_x - m_y)\times m_x}$ whose linearly independent rows form a basis for the nullspace of G_{12}^T (the existence of such R can be obtained from [13]). Further, R can be taken to be orthonormal.[1] Thus, for each t, $0 \le t \le 1$,

$$RG_{12} = 0. \tag{2.2}$$

We assume that there exists a constant $\hat{K}$ of moderate size such that

$$\|(G_{21}G_{12})^{-1}\| \le \hat{K} \tag{2.3}$$

uniformly in t. From the following lemma we can then conclude that there is a moderate-sized constant $\hat{\hat{K}}$ such that

$$\left\| \begin{pmatrix} R \\ G_{21} \end{pmatrix}^{-1} \right\| \le \hat{\hat{K}}.$$

Lemma 2.1 *There exists a constant K of moderate size such that for orthonormal R satisfying (2.2),*

$$\left\| \begin{pmatrix} R \\ G_{21} \end{pmatrix}^{-1} \right\| \le K\|G_{12}\|\|G_{21}\|\|(G_{21}G_{12})^{-1}\|.$$

[1] To see this, consider the QR-factorization of (G_{12}, R^T). This matrix is smooth and nonsingular, thus there is a (unique) smooth QR-factorization with positive diagonal elements in the triangular part. Taking the last $m_x - m_y$ rows of the resulting Q^T gives a new orthonormal R.

Proof:

Consider first the QR-decomposition of G_{12} at a fixed t:

$$Q^T G_{12} = \begin{pmatrix} U \\ 0 \end{pmatrix}$$

Write $Q^T = \begin{pmatrix} Q_1^T \\ Q_2^T \end{pmatrix}$ where Q_2^T has $m_x - m_y$ rows, and define

$$\hat{R} := Q_2^T.$$

We have $(G_{21}G_{12})^{-1} = (G_{21}Q_1U)^{-1}$, so $\|(G_{21}Q_1)^{-1}\| \le \|G_{12}\|\|(G_{21}G_{12})^{-1}\|$. Hence

$$\left\| \begin{pmatrix} \hat{R} \\ G_{21} \end{pmatrix}^{-1} \right\| = \left\| \left(\begin{pmatrix} \hat{R} \\ G_{21} \end{pmatrix} Q \right)^{-1} \right\| = \left\| \begin{pmatrix} 0 & I \\ G_{21}Q_1 & G_{21}Q_2 \end{pmatrix}^{-1} \right\| \le \tilde{K}\|G_{21}\|\|G_{12}\|\|(G_{21}G_{12})^{-1}\|$$

with $\tilde{K}$ a suitable moderate constant.

Now, $\hat{R}(t)$ is not necessarily continuous. Still, due to the uniqueness of the orthonormal projector, at each t

$$R^T R = \hat{R}^T \hat{R} = I - G_{12}(G_{12}^T G_{12})^{-1} G_{12}^T.$$

Thus, $XR = \hat{R}$, where $X(t)$ is defined by $X = \hat{R}R^T$. The matrix X is nonsingular because Q_1 spans the nullspaces of both $\hat{R}$ and R. Also, $\|X\| \le 1$. Therefore, we can write

$$\left\| \begin{pmatrix} R \\ G_{21} \end{pmatrix}^{-1} \right\| = \left\| \begin{pmatrix} X^{-1}\hat{R} \\ G_{21} \end{pmatrix}^{-1} \right\| = \left\| \begin{pmatrix} \hat{R} \\ G_{21} \end{pmatrix}^{-1} \begin{pmatrix} X & 0 \\ 0 & I \end{pmatrix} \right\| \le K\|G_{12}\|\|G_{21}\|\|(G_{21}G_{12})^{-1}\|.$$

□

Multiplying (2.1a) by R we have

$$R\mathbf{x}' = R(G_{11}\mathbf{x} + \mathbf{q}_1). \tag{2.4}$$

Let

$$\mathbf{v} = R\mathbf{x}, \qquad 0 \le t \le 1. \tag{2.5}$$

Then, using (2.1b), the inverse transformation is given by

$$\mathbf{x} = \begin{pmatrix} R \\ G_{21} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{v} \\ -\mathbf{q}_2 \end{pmatrix} \equiv S\mathbf{v} + \hat{\mathbf{q}} \tag{2.6}$$

where $S(t) \in \mathcal{R}^{m_x \times (m_x - m_y)}$ satisfies

$$RS = I, \qquad G_{21}S = 0. \tag{2.7}$$

Differentiating (2.5) and substituting (2.4), we obtain the *essential underlying ODE*[2]

$$\mathbf{v}' = [(RG_{11} + R')S]\mathbf{v} + [R\mathbf{q}_1 + (RG_{11} + R')\hat{\mathbf{q}}], \tag{2.8}$$

which is subject to $m_x - m_y$ boundary conditions, obtained from (2.1c) using (2.6):

$$(B_0 S(0))\mathbf{v}(0) + (B_1 S(1))\mathbf{v}(1) = \beta - B_0\hat{\mathbf{q}}(0) - B_1\hat{\mathbf{q}}(1). \tag{2.9}$$

Now, if the ordinary BVP (2.8), (2.9) is stable, i.e. if its Green's function is bounded in norm by a constant of moderate size, then a similar conclusion holds for the DAE. We note that the underlying ODE is not unique; (2.8) is unique only up to a nonsingular (bounded, time-dependent) change of variables. However, such a transformation of variables does not alter the boundedness (or lack thereof) of the Green's function, and hence the stability properties are properly reflected. We obtain the following theorem:

Theorem 2.1 *Let the BVP (2.1a)-(2.1c) have smooth, bounded coefficients, and assume that (2.3) holds and that the underlying BVP (2.8)-(2.9) is stable. Then there is a constant K of moderate size such that*

$$\|\mathbf{x}\| \leq K(\|\mathbf{q}_1\| + \|\mathbf{q}_2\| + |\beta|) \tag{2.10a}$$
$$\|\mathbf{y}\| \leq K(\|\mathbf{q}_1'\| + \|\mathbf{q}_2'\| + \|\mathbf{q}_1\| + \|\mathbf{q}_2\| + |\beta|) \tag{2.10b}$$

Proof:

Our assumptions guarantee the well-conditioning of the transformation (2.5), (2.6). Hence, the inhomogeneities appearing in (2.8), (2.9) are bounded in terms of the original ones. The stability of the BVP (2.8), (2.9) guarantees a similar bound for $\|\mathbf{v}\|$. Conclusion (2.10a) is then obtained using (2.6).

Now, given $\mathbf{x}$ we obtain $\mathbf{y}$ through multiplying (2.1a) by G_{21}, yielding

$$\mathbf{y} = (G_{21}G_{12})^{-1}G_{21}(\mathbf{x}' - G_{11}\mathbf{x} - \mathbf{q}_1). \tag{2.11}$$

The bound (2.10b) is obtained from this expression using (2.10a) and (2.3). □

Note that no derivatives of $\mathbf{q}_1$ or $\mathbf{q}_2$ appear in the bound (2.10a). The problem for the "differential variables" $\mathbf{x}$ is well-posed in the classical sense! Only in the expressions (2.10b), (2.11) for the "algebraic variables" $\mathbf{y}$ do we get derivatives of the inhomogeneities.

Remark

[2]We have referred to (2.8) as the *essential* underlying ODE in order to distinguish it from other ODEs which have been referred to in the literature as underlying ODEs (e.g. [14]), emphasizing that (2.8) has a minimal size and yields a well-posed problem for $\mathbf{x}$, as described in Theorem 2.1.

In the context of the incompressible Navier-Stokes equations, the matrices G_{12}^T and G_{21} may be identified with the *div* operator, R may be identified with the *curl* operator, and the essential underlying ODE may be identified with the vorticity-streamfunction formulation. □

Our approach in deriving an underlying ODE should be contrasted with another, perhaps more common one (see, e.g. [18, 14]), which eliminates $\mathbf{y}$ to obtain an ODE for $\mathbf{x}$. Defining for each t

$$F := G_{12}(G_{21}G_{12})^{-1}, \qquad H = FG_{21}, \qquad \tilde{\mathbf{q}} := F\mathbf{q}_2, \tag{2.12}$$

we have $H\mathbf{x} + \tilde{\mathbf{q}} = \mathbf{0}$ and $(I - H)G_{12} = 0$. As in (2.4) we obtain from (2.1a)

$$(I - H)\mathbf{x}' = (I - H)(G_{11}\mathbf{x} + \mathbf{q}_1)$$

(indeed, $R(t)$ used earlier can be considered, at least locally, as $m_x - m_y$ linearly independent rows of $I - H(t)$), but now, instead of using (2.5) we differentiate (2.1b), obtaining

$$\mathbf{0} := G_{12}\mathbf{x}' + G_{12}'\mathbf{x} + \mathbf{q}_2', \tag{2.13}$$

multiply this expression by F, add $F'(G_{21}\mathbf{x} + \mathbf{q}_2) = \mathbf{0}$, and substitute for $H\mathbf{x}'$. This yields

$$\mathbf{x}' = [(I - H)G_{11} - H']\mathbf{x} + [(I - H)\mathbf{q}_1 - \tilde{\mathbf{q}}']. \tag{2.14}$$

The obtained ODE is subject to the boundary conditions (2.1c). However, m_y additional conditions are needed, and these are precisely those necessary to complement (2.13) so that it becomes equivalent to (2.1b). The most common choice is

$$\mathbf{0} = G_{21}(0)\mathbf{x}(0) + \mathbf{q}_2(0), \tag{2.15}$$

but other choices are possible too.

The underlying BVP (2.14), (2.1c), (2.15) has two drawbacks.[3] The first is that, assuming it is stable, $\|\mathbf{x}\|$ is bounded in terms of $\tilde{\mathbf{q}}'$ (as well as the original inhomogeneities). For linear problems, this bound is not as sharp as (2.10a). In particular, it suggests that using any one-step discretization, the roundoff error accumulation could be proportional to Nh_{min}^{-1} where N is the total number of steps (i.e. mesh size) and h_{min} is the minimum step size taken. The bound (2.10a), on the other hand, suggests that roundoff error accumulation for $\mathbf{x}$ will depend only on N, and this indeed turns out to be the case. The proof of the numerical stability of our proposed methods depends on the bound (2.10a) (cf. Section 3).

The other drawback of (2.14) is that its BVP may be unstable, even if the original problem is stable. This phenomenon may arise because the algebraic constraint (2.1b)

[3]These drawbacks do not occur in the problems addressed for example in [18], where the constraints and the multipliers occur linearly and with constant coefficients.

has been replaced by a differential form (2.13),(2.15) and this introduces a propagation of information where there has been none before (cf. [14], [6]).

Example 1

Consider for $0 \le t \le 1$

$$\begin{aligned} x_1' &= (\lambda - \frac{1}{2-t})x_1 + (2-t)\lambda y + \frac{3-t}{2-t}e^t \\ x_2' &= \frac{1-\lambda}{t-2}x_1 - x_2 + (\lambda-1)y + 2e^t \\ 0 &= (t+2)x_1 + (t^2-4)x_2 - (t^2+t-2)e^t, \end{aligned} \tag{2.16}$$

where $x_1(0) = 1$. Here $\lambda > 0$ is a parameter, say $\lambda = 50$. Note that $(G_{21}G_{12})^{-1} = 1/(4-t^2) \le 1/3$. Choose

$$R(t) = (1-\lambda, (2-t)\lambda).$$

Then

$$\begin{pmatrix} R \\ G_{21} \end{pmatrix}^{-1} = (4-t^2)^{-1} \begin{pmatrix} 4-t^2 & (2-t)\lambda \\ t+2 & \lambda-1 \end{pmatrix}, \qquad S = \begin{pmatrix} 1 \\ 1/(2-t) \end{pmatrix}$$

and (2.3) is satisfied with $\hat{K} = O(\lambda)$. The underlying ODE (2.8) for the homogeneous problem is

$$v' = -(\lambda + 1/(2-t))v$$

with

$$v(0) = \lambda + 1.$$

This is a stable IVP. Hence the stability constant of the DAE problem is $O(\lambda)$, which is mild for $\lambda = 50$.

On the other hand, in (2.12) we get

$$H = \begin{pmatrix} \lambda & (t-2)\lambda \\ \frac{\lambda-1}{2-t} & 1-\lambda \end{pmatrix} \qquad (I-H)G_{11} = \begin{pmatrix} \frac{\lambda-1}{2-t} & (t-2)\lambda \\ \frac{\lambda-1}{(2-t)^2} & -\lambda \end{pmatrix}$$

and the homogeneous (2.14) is

$$\mathbf{x}' = \begin{pmatrix} \frac{\lambda-1}{2-t} & (t-3)\lambda \\ 0 & -\lambda \end{pmatrix} \mathbf{x}.$$

The boundary conditions corresponding to (2.1c), (2.15) are $x_1(0) = x_2(0) = 1$. This is an unstable IVP, whose stability constant blows up exponentially in λ. □

The possibility that examples like the one just described may arise should not be too surprising if one takes into account the arbitrariness of the location $t = 0$ in (2.15) (cf. [2]). Indeed, as discussed in the introduction this is the source of potential trouble in the straightforward index reduction technique, which is eliminated when

applying the multiplier technique. In case of the BVP (2.1a)-(2.1c) the multiplier technique yields the system

$$\begin{aligned} \mathbf{x}' &= G_{11}\mathbf{x} + G_{12}\mathbf{y} + \mathbf{q}_1 + G_{12}\mu && (2.17a)\\ \mathbf{0} &= G_{21}\mathbf{x} + \mathbf{q}_2 && (2.17b)\\ \mathbf{0} &= G_{21}(G_{11}\mathbf{x} + G_{12}\mathbf{y} + \mathbf{q}_1) + G_{21}'\mathbf{x} + \mathbf{q}_2' && (2.17c) \end{aligned}$$

and (2.1c). This system does not suffer the potential instability demonstrated in Example 1. To see this we use (2.17c) to eliminate $\mathbf{y}$,

$$\mathbf{y} = -(G_{21}G_{12})^{-1}[(G_{21}G_{11} + G_{21}')\mathbf{x} + G_{21}\mathbf{q}_1 + \mathbf{q}_2'] \qquad (2.18)$$

and substitute in (2.17a) to obtain

$$\mathbf{x}' = [(I - H)G_{11} - H']\mathbf{x} + [(I - H)\mathbf{q}_1 - \tilde{\mathbf{q}}'] + G_{12}\mu. \qquad (2.19)$$

Multiplying (2.19) by R yields the original underlying BVP (2.8), (2.9). Also, multiplying (2.19) by G_{21} yields

$$(G_{21}\mathbf{x} + \mathbf{q}_2)' = G_{21}G_{12}\mu$$

from which we conclude, using a differentiation of (2.17b), that $\mu = \mathbf{0}$ in exact arithmetic (cf. [15]). [4]

We have shown that the second drawback of the formulation (2.14), (2.1c), (2.15) is "corrected" by the formulation (2.17). However, the first drawback mentioned following (2.15) is not eliminated here either.

The above conditioning arguments do not extend directly for nonlinear problems, because now R, whose derivative is used in (2.8), depends on the solution as well. However, a linearization may be considered. Thus, consider the BVP

$$\begin{aligned} \mathbf{x}' &= \mathbf{g}_1(\mathbf{x}, \mathbf{y}, t) && (2.20a)\\ \mathbf{0} &= \mathbf{g}_2(\mathbf{x}, t) && (2.20b)\\ \mathbf{0} &= \mathbf{b}(\mathbf{x}(0), \mathbf{x}(1)) && (2.20c) \end{aligned}$$

with the same dimensions as in (2.1a)-(2.1c). Here $\mathbf{g}_1, \mathbf{g}_2$ and $\mathbf{b}$ are smooth functions of their arguments. The approach is standard: we define linear operators about appropriate functions $\mathbf{u}(t)$ and $\mathbf{v}(t)$ (not the same as $\mathbf{v}$ of (2.5) of course) as

$$\begin{aligned} \mathcal{L}_1[\mathbf{u}, \mathbf{v}](\mathbf{x}, \mathbf{y}) &\equiv \mathbf{x}' - G_{11}\mathbf{x} - G_{12}\mathbf{y} && (2.21a)\\ \mathcal{L}_2[\mathbf{u}](\mathbf{x}) &\equiv G_{21}\mathbf{x} && (2.21b)\\ \mathcal{L}_3[\mathbf{u}](\mathbf{x}) &\equiv B_0\mathbf{x}(0) + B_1\mathbf{x}(1) && (2.21c) \end{aligned}$$

[4] Actually, in [15] the multiplier is $-G_{21}^T\mu$, yielding $\mu = \mathbf{0}$ as well. Using this, the resulting system is similar, but we prefer $G_{12}\mu$.

where $G_{11}(t) := (\partial \mathbf{g}_1/\partial \mathbf{x})$, $G_{12}(t) := (\partial \mathbf{g}_1/\partial \mathbf{y})$, $G_{21}(t) := (\partial \mathbf{g}_2/\partial \mathbf{x})$, evaluated at $\mathbf{u}, \mathbf{v}$ and t. The matrices B_0 and B_1 are similarly defined as the derivatives of $\mathbf{b}$ evaluated at $\mathbf{u}(0)$ and $\mathbf{u}(1)$. Then for an isolated solution $\mathbf{x}$, $\mathbf{y}$ of the nonlinear problem (2.20a)-(2.20c), the variational problem

$$\mathcal{L}_1[\mathbf{x}, \mathbf{y}](\mathbf{z}, \mathbf{w}) \equiv \mathbf{0}, \qquad \mathcal{L}_2[\mathbf{x}](\mathbf{z}) \equiv \mathbf{0}, \qquad \mathcal{L}_3[\mathbf{x}](\mathbf{z}) = \mathbf{0},$$

has only the trivial solution $\mathbf{z} \equiv \mathbf{0}$, $\mathbf{w} \equiv \mathbf{0}$. Further assuming the conditions of Theorem 2.1 to hold (for this we must have that the solution $\mathbf{x}$, $\mathbf{y}$ is sufficiently smooth), we obtain its conclusions, and in particular (2.10a) holds for the linearized problem in an appropriate neighborhood of this isolated solution.

3 Projected IRK methods

Consider the DAE problem (2.20a)-(2.20c)

$$\begin{aligned} \mathbf{x}' &= \mathbf{g}_1(\mathbf{x}, \mathbf{y}, t) \\ \mathbf{0} &= \mathbf{g}_2(\mathbf{x}, t) \\ \mathbf{0} &= \mathbf{b}(\mathbf{x}(0), \mathbf{x}(1)). \end{aligned}$$

Let $b = (b_1, \ldots, b_k)^T$, $c = (c_1, \ldots, c_k)^T$, $\mathcal{A} = (a_{ij})_{i,j=1}^k$ be the coefficients of a k-stage Implicit Runge-Kutta (IRK) scheme (see, e.g., [9]). We assume that $0 \le c_1 \le c_2 \le \cdots \le c_k \le 1$ and that $\mathcal{A}$ is nonsingular (which excludes Lobatto schemes but leaves in all other IRK schemes of practical interest). Denote the internal stage order by k_I ($k_I \ge 1$ for consistency) and the nonstiff order at mesh points by k_d ($k_d \le 2k$). For collocation schemes, in particular, $k_I = k$, $c_1 > 0$ and the c_i are distinct.

Given a mesh

$$\begin{aligned} \pi &: 0 = t_0 < t_1 < \cdots < t_N = 1 \\ h_n &: = t_n - t_{n-1} \\ h &: = \max\{h_n,\ 1 \le n \le N\}, \end{aligned} \tag{3.1}$$

a projected IRK method for (2.20a)-(2.20c) samples (2.20c), requires

$$\mathbf{0} = \mathbf{g}_2(\mathbf{x}_0, 0) \tag{3.2}$$

and approximates (2.20a), (2.20b) on each mesh subinterval $[t_{n-1}, t_n]$, $1 \le n \le N$, by

$$\mathbf{X}_i' = \mathbf{g}_1(\mathbf{X}_i, \mathbf{Y}_i, t_i) \tag{3.3a}$$

$$\mathbf{0} = \mathbf{g}_2(\mathbf{X}_i, t_i), \quad i = 1, 2, \ldots, k \tag{3.3b}$$

$$t_i = t_{n-1} + h_n c_i \tag{3.3c}$$

$$\mathbf{X}_i = \mathbf{x}_{n-1} + h_n \sum_{j=1}^{k} a_{ij} \mathbf{X}'_j \tag{3.3d}$$

$$\mathbf{0} = \mathbf{g}_2(\mathbf{x}_n, t_n), \tag{3.3e}$$

$$\mathbf{x}_n = \mathbf{x}_{n-1} + h_n \sum_{j=1}^{k} b_j \mathbf{X}'_j + G_{12}^n \lambda_n \tag{3.3f}$$

where $G_{12}^n = \frac{\partial \mathbf{g}_1}{\partial \mathbf{y}}(\mathbf{x}_n, \mathbf{y}_n, t_n)$. In practice we evaluate G_{12}^n at some suitable approximations $\bar{\mathbf{x}}_n, \bar{\mathbf{y}}_n$, discussed in Section 3.3.

Observe that if we drop the requirement (3.3e) and set $\lambda_n = \mathbf{0}$ then an IRK method is obtained as discussed in [10], [9], [17]. Thus, if $\hat{\mathbf{x}}_n$ is the result of one IRK step starting from $\mathbf{x}_{n-1}$, then $\mathbf{x}_n$ is given by

$$\mathbf{x}_n = \hat{\mathbf{x}}_n + G_{12}^n \lambda_n \tag{3.4}$$

and can be viewed as the projection of $\hat{\mathbf{x}}_n$ onto the algebraic manifold at the next mesh point t_n. [5]

Consider now the linear DAE (2.1a), (2.1b), i.e. $\mathbf{g}_1 = G_{11}\mathbf{x} + G_{12}\mathbf{y} + \mathbf{q}_1$, $\mathbf{g}_2 = G_{21}\mathbf{x} + \mathbf{q}_2$. Note that the unknowns $\mathbf{X}_i$, $\mathbf{Y}_i$, $i = 1, 2, \ldots, k$ and λ_n are internal to the subinterval $[t_{n-1}, t_n]$ and can be eliminated locally. The resulting discretizations do not contain anything directly related to $\mathbf{y}$, similarly to the manipulations for the continuous case in Section 2. Since the IRK scheme is locally well-defined, we have

Lemma 3.1 *For projected IRK schemes (3.3), we have the following:*

1. *The projected IRK scheme (3.3) is locally well-defined.*
2. *If $c_k = 1$ then the IRK scheme and the projected IRK scheme coincide.*

Proof:

Let $R(t)$ be defined as in (2.2). From (3.4) we have

$$\mathbf{v}_n := R^n \mathbf{x}_n = R^n \hat{\mathbf{x}}_n. \tag{3.5}$$

The linear version of (3.3e) further reads $\mathbf{0} = G_{21}^n \mathbf{x}_n + \mathbf{q}_2^n$, so the equivalent of (2.6) can be written:

$$\mathbf{x}_n = \begin{pmatrix} R^n \\ G_{21}^n \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{v}_n \\ -\mathbf{q}_2^n \end{pmatrix} = S^n R^n \hat{\mathbf{x}}_n + \hat{\mathbf{q}}^n.$$

Since $\hat{\mathbf{x}}_n$ is well-defined, so is $\mathbf{x}_n$. This proves the first claim.

If $c_k = 1$ then (3.3e) is contained in (3.3b), so we can take $\lambda_n = \mathbf{0}$ in (3.3f). This choice is unique because the two schemes are locally well-defined. Thus, they coincide. □

[5] Hereinafter we denote a function value $\phi(t_n)$ by ϕ^n.

The projected IRK method is well-defined also globally, i.e. a solution to the global discretization on $[0,1]$ using (3.2), (3.3) for the linear DAE of index 2 exists. This, however, does not follow directly by comparing to the unprojected method: indeed, a global existence does not necessarily hold for the latter. We now show not only that a solution exists, but more strongly that the scheme is stable, with a stability bound mimicking (2.10a). Convergence of order $O(h^{min(k_I+1,k_d)})$ follows. For various special cases, including the most interesting ones, we then proceed to recover superconvergence results as well. This section closes with a treatment of collocation methods for nonlinear problems.

3.1 Existence, stability and basic convergence

We now give a basic existence, stability and convergence theorem for the linear case.

Theorem 3.1 *Given a stable, semi-explicit, linear Hessenberg index two system (2.1a)-(2.1c) to be solved numerically by the k-stage Projected IRK method, then for h sufficiently small*

1. *The local error in x is $O(h_n^{\min(k_d+1,k_I+2)})$.*
2. *There exists a unique projected IRK solution.*
3. *The projected IRK method is stable, with a moderate stability constant, provided that the BVP has a moderate stability constant K.*
4. *The global error in x is $O(h^{\min(k_d,k_I+1)})$.*
5. *The errors in the intermediate variables $\mathbf{X}_i'$ and $\mathbf{Y}_i$ are $O(h^{\min(k_d,k_I)})$, while those in $\mathbf{X}_i$ are $O(h^{\min(k_d,k_I+1)})$.*

Proof:

We first prove the "local" claims. Thus, fix a step counter n. The true solution to the DAE satisfies

$$\begin{aligned}
&\mathbf{x}'(t_i) = \mathbf{g}_1(\mathbf{x}(t_i), \mathbf{y}(t_i), t_i)\\
&\mathbf{0} = \mathbf{g}_2(\mathbf{x}(t_i), t_i), \qquad i = 1, 2, \ldots, k\\
&\mathbf{x}(t_n) = \mathbf{x}(t_{n-1}) + h_n \sum_{i=1}^{k} b_i \mathbf{x}'(t_{n-1} + c_i h) - \delta_{k+1}^{x(n)}\\
&\mathbf{0} = \mathbf{g}_2(\mathbf{x}(t_n), t_n),
\end{aligned}$$

and

$$\mathbf{x}(t_i) = \mathbf{x}(t_{n-1}) + h_n \sum_{j=1}^{k} a_{ij} \mathbf{x}'(t_{n-1} + c_j h_n) - \delta_i^{x(n)}, \quad i = 1, 2, \ldots, k,$$

with $\delta_i^{x(n)} = O(h_n^{k_I+1})$, $\delta_{k+1}^{x(n)} = O(h_n^{k_d+1})$.

Let $G_{21}^{(i)} = \partial \mathbf{g}_2/\partial \mathbf{x}$, $G_{12}^{(i)} = \partial \mathbf{g}_1/\partial \mathbf{y}$, and $G_{11}^{(i)} = \partial \mathbf{g}_1/\partial \mathbf{x}$, where the partial derivatives are evaluated at t_i. Lemma 3.1 assures us that (3.3) is well-defined. Subtracting the above equations from (3.3), we obtain

$$\mathbf{E}_i^{x\prime} = G_{11}^{(i)}\mathbf{E}_i^x + G_{12}^{(i)}\mathbf{E}_i^y \tag{3.6a}$$

$$0 = G_{21}^{(i)}\mathbf{E}_i^x, \qquad i = 1, 2, \ldots, k \tag{3.6b}$$

$$\mathbf{e}_n^x = \mathbf{e}_{n-1}^x + h_n \sum_{i=1}^{k} b_i E_i^{x\prime} + \delta_{k+1}^x + G_{12}^{(n)}\lambda_n \tag{3.6c}$$

$$0 = G_{21}^{(n)}\mathbf{e}_n^x \tag{3.6d}$$

$$\mathbf{E}_i^x = \mathbf{e}_{n-1}^x + h_n \sum_{j=1}^{k} a_{ij}\mathbf{E}_j^{x\prime} + \delta_i^x, \tag{3.6e}$$

where $\mathbf{E}_i^{x\prime} = \mathbf{X}_i' - \mathbf{x}'(t_i)$, $\mathbf{E}_i^x = \mathbf{X}_i - \mathbf{x}(t_i)$, and $\mathbf{e}_n^x = \mathbf{x}_n - \mathbf{x}(t_n)$.

Now, to eliminate $\mathbf{E}_i^y$, multiply (3.6a) by $R^{(i)}$, where $R^{(i)}G_{12}^{(i)} = 0$ as in (2.2). Similarly, eliminate λ_n by multiplying (3.6c) by $R^{(n)}$. This yields

$$R^{(i)}\mathbf{E}_i^{x\prime} = R^{(i)}G_{11}^{(i)}\mathbf{E}_i^x \tag{3.7a}$$

$$R^{(n)}\mathbf{e}_n^x = R^{(n)}\mathbf{e}_{n-1}^x + h_n \sum_{i=1}^{k} b_i R^{(n)}\mathbf{E}_i^{x\prime} + R^{(n)}\delta_{k+1}^x \tag{3.7b}$$

Let $\mathbf{E}_i^v = R^{(i)}\mathbf{E}_i^x$ and $\mathbf{e}_n^v = R^{(n)}\mathbf{e}_n^x$. Then (3.6b) and (3.6d) imply that

$$\mathbf{E}_i^x = \begin{pmatrix} R^{(i)} \\ G_{21}^{(i)} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{E}_i^v \\ 0 \end{pmatrix} = S^{(i)}\mathbf{E}_i^v, \qquad i = 1, 2, \ldots, k \tag{3.8a}$$

$$\mathbf{e}_n^x = \begin{pmatrix} R^{(n)} \\ G_{21}^{(n)} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{e}_n^v \\ 0 \end{pmatrix} = S^{(n)}\mathbf{e}_n^v, \tag{3.8b}$$

where $R^{(i)}S^{(i)} = I$ and $R^{(n)}S^{(n)} = I$.

Now, (3.6e), (3.8a) and (3.8b) give

$$S^{(i)}\mathbf{E}_i^v = S^{(n-1)}\mathbf{e}_{n-1}^v + h_n \sum_{j=1}^{k} a_{ij}\mathbf{E}_j^{x'} + \delta_i^x. \tag{3.9}$$

Let

$$\underline{\mathbf{E}}^v = (\mathbf{E}_1^v, \mathbf{E}_2^v, \ldots, \mathbf{E}_k^v)^T$$
$$\underline{\mathbf{E}}^{x'} = (\mathbf{E}_1^{x'}, \mathbf{E}_2^{x'}, \ldots, \mathbf{E}_k^{x'})^T$$
$$\underline{\delta}^x = (\delta_1^x, \delta_2^x, \ldots, \delta_k^x)^T,$$

and

$$\underline{\mathbf{e}}^v_{n-1} = \mathbf{1} \otimes \mathbf{e}^v_{n-1},$$

where $\mathbf{1} = (1, 1, \ldots, 1)^T$. Then we can rewrite (3.9) to obtain

$$\underline{S}\underline{\mathbf{E}}^v = (I_k \otimes S^{(n-1)})\underline{\mathbf{e}}^v_{n-1} + h_n(\mathcal{A} \otimes I_{m_x})\underline{\mathbf{E}}^{x'} + \underline{\delta}^x,$$

where $\underline{S} = \mathrm{diag}(S^{(1)}, S^{(2)}, \ldots, S^{(k)})$. Solving for $\underline{\mathbf{E}}^{x'}$, we have

$$h_n\underline{\mathbf{E}}^{x'} = (\mathcal{A}^{-1} \otimes I_{m_x})(\underline{S}\underline{\mathbf{E}}^v - (I_k \otimes S^{(n-1)})\underline{\mathbf{e}}^v_{n-1} - \underline{\delta}^x). \tag{3.10}$$

Substituting (3.10) into (3.7a), we have

$$\underline{R}(\mathcal{A}^{-1} \otimes I_{m_x})(\underline{S}\underline{\mathbf{E}}^v - (I_k \otimes S^{(n-1)})\underline{\mathbf{e}}^v_{n-1} - \underline{\delta}^x) = h_n\underline{R}\underline{G}_{11}\underline{S}\underline{\mathbf{E}}^v, \tag{3.11}$$

where

$$\underline{R} = \mathrm{diag}(R_1, R_2, \ldots, R_k)$$
$$\underline{G}_{11} = \mathrm{diag}(G^{(1)}_{11}, G^{(2)}_{11}, \ldots, G^{(k)}_{11}).$$

Now, solve for $\underline{\mathbf{E}}^v$ in (3.11) to obtain

$$(\underline{R}(\mathcal{A}^{-1} \otimes I_{m_x})\underline{S} - h_n\underline{R}\underline{G}_{11}\underline{S})\underline{\mathbf{E}}^v = \underline{R}(\mathcal{A}^{-1} \otimes I_{m_x})((I_k \otimes S^{(n-1)})\underline{\mathbf{e}}^v_{n-1} + \underline{\delta}^x). \tag{3.12}$$

Note that by Taylor's series,

$$\begin{aligned}\underline{R}(\mathcal{A}^{-1} \otimes I_{m_x})\underline{S} &= (\mathcal{A}^{-1} \otimes I_{m_x})\underline{R}\underline{S} + h_n\mathcal{B} \otimes (R'S) + O(h_n^2) \\ &= \mathcal{A}^{-1} \otimes I_{m_x} + h_n\mathcal{B} \otimes (R'S) + O(h_n^2),\end{aligned}$$

where $(\mathcal{B})_{ij} = (\mathcal{A}^{-1})_{ij}(c_i - c_j)$, and that

$$\underline{R}(\mathcal{A}^{-1} \otimes I_{m_x})(I_k \otimes S^{(n-1)}) = \mathcal{A}^{-1} \otimes I_{m_x} + h_n(\mathcal{C}\mathcal{A}^{-1}) \otimes (R'S) + O(h_n^2),$$

where $\mathcal{C} = \mathrm{diag}(c_i)$. All matrices which are not superscripted are assumed here to be evaluated at t_n. Thus we have from (3.12),

$$\begin{aligned}&(I_{km_x} + h_n(\mathcal{A}\mathcal{B}) \otimes (R'S) - h_n(\mathcal{A} \otimes I_{m_x})\underline{R}\underline{G}_{11}\underline{S} + O(h_n^2))\underline{\mathbf{E}}^v \\ &\quad = (I_{km_x} + h_n(\mathcal{A}\mathcal{C}\mathcal{A}^{-1}) \otimes (R'S) + O(h_n^2))\underline{\mathbf{e}}^v_{n-1} \\ &\quad +(I_k \otimes R^{(n-1)} + O(h_n))\underline{\delta}^x.\end{aligned} \tag{3.13}$$

Solving for $\underline{\mathbf{E}}^v$ from (3.13) and substituting into (3.10), we obtain

$$\begin{aligned}h_n\underline{\mathbf{E}}^{x'} &= (\mathcal{A}^{-1} \otimes I_{m_x})(\underline{S}(I_{km_x} - h_n(\mathcal{A}\mathcal{B}) \otimes (R'S) + h_n(\mathcal{A} \otimes I_{m_x})\underline{R}\underline{G}_{11}\underline{S} \\ &\quad +O(h_n^2))((I_{km_x} + h_n(\mathcal{A}\mathcal{C}\mathcal{A}^{-1}) \otimes (R'S) + O(h_n^2))\underline{\mathbf{e}}^v_{n-1} \\ &\quad +(I_k \otimes R^{(n-1)} + O(h_n))\underline{\delta}^x) - (I_k \otimes S^{(n-1)})\underline{\mathbf{e}}^v_{n-1} - \underline{\delta}^x).\end{aligned}$$

Collecting terms, we obtain

$$\begin{aligned} h_n\underline{\mathbf{E}}^{x'} &= (\mathcal{A}^{-1}\otimes I_{m_x})(\underline{S}(I_{km_x}-h_n(\mathcal{A}\mathcal{B})\otimes(R'S)+h_n(\mathcal{A}\otimes I_{m_x})\underline{RG}_{11}\underline{S} \\ &\quad +h_n(\mathcal{A}\mathcal{C}\mathcal{A}^{-1})\otimes(R'S))-I_k\otimes S^{(n-1)}+O(h_n^2))\underline{\mathbf{e}}^v_{n-1} \\ &\quad +(\mathcal{A}^{-1}\otimes I_{m_x})(\underline{S}(I_k\otimes R^{(n-1)})-I_{km_x}+O(h_n))\underline{\delta}^x. \end{aligned}$$

Note that $\underline{S}=I_k\otimes S^{(n-1)}+h_n\mathcal{C}\otimes S'+O(h_n^2)$. Thus

$$\begin{aligned} h_n\underline{\mathbf{E}}^{x'} &= (\mathcal{A}^{-1}\otimes I_{m_x})(h_n\mathcal{C}\otimes S'-h_n\underline{S}((\mathcal{A}\mathcal{B}-\mathcal{A}\mathcal{C}\mathcal{A}^{-1})\otimes(R'S)) \\ &\quad +h_n\underline{S}(\mathcal{A}\otimes I_{m_x})\underline{RG}_{11}\underline{S}+O(h_n^2))\underline{\mathbf{e}}^v_{n-1} \\ &\quad +(\mathcal{A}^{-1}\otimes I_{m_x})(\underline{S}(I_k\otimes R^{(n-1)})-I_{km_x}+O(h_n))\underline{\delta}^x. \end{aligned} \tag{3.14}$$

Substituting into (3.7b), and noting that $\underline{S}(I_k\otimes R^{(n-1)})=I_{km_x}+O(h_n)$, we have

$$\begin{aligned} \mathbf{e}^v_n &= R^{(n)}S^{(n-1)}\mathbf{e}^v_{n-1} \\ &\quad +(b^T\mathcal{A}^{-1}\otimes R^{(n)})(h_n\mathcal{C}\otimes S'-h_n\underline{S}((\mathcal{A}\mathcal{B}-\mathcal{A}\mathcal{C}\mathcal{A}^{-1})\otimes(R'S)) \\ &\quad +h_n\underline{S}(\mathcal{A}\otimes I_{m_x})\underline{RG}_{11}\underline{S}+O(h_n^2))\underline{\mathbf{e}}^v_{n-1}+O(h_n\underline{\delta}^x)+O(\delta^x_{k+1}). \end{aligned}$$

Noting that $b^T\mathcal{A}^{-1}\mathcal{C}\mathbf{1}=1$ (by the order conditions $k_I\geq 1$) and rewriting, we have

$$\begin{aligned} \mathbf{e}^v_n &= \mathbf{e}^v_{n-1}+h_nR'S\mathbf{e}^v_{n-1}+h_nRS'\mathbf{e}^v_{n-1}-h_nb^T\mathcal{B}\mathbf{1}R'S\mathbf{e}^v_{n-1} \\ &\quad +h_nb^T\mathcal{C}\mathcal{A}^{-1}\mathbf{1}R'S\mathbf{e}^v_{n-1}+h_nRG_{11}S\mathbf{e}^v_{n-1} \\ &\quad +O(h_n^2\mathbf{e}^v_{n-1})+O(h_n\underline{\delta}^x)+O(\delta^x_{k+1}). \end{aligned}$$

Noting that $\mathcal{C}\mathcal{A}^{-1}-\mathcal{B}=\mathcal{A}^{-1}\mathcal{C}$, and that $(RS)=I$ implies that $(RS)'=0$, we have the desired result, namely

$$\mathbf{e}^v_n=(I+h_nR'S\mathbf{e}^v_{n-1}+h_nRG_{11}S+O(h_n^2))\mathbf{e}^v_{n-1}+O(h_n\underline{\delta}^x)+O(\delta^x_{k+1}). \tag{3.15}$$

Since $\underline{\delta}^x=O(h_n^{k_I+1})$ and $\delta^x_{k+1}=O(h_n^{k_d+1})$, we have proved our claim about the local error.

The other, "global", claims now follow using standard arguments: Comparing (3.15) to the minimal underlying ODE (2.8), and noting that by (3.2) and (3.3f) for $n=N$ the boundary conditions (2.9) for $\mathbf{v}$ are reproduced as well, we have obtained a one-step difference method of accuracy $\min(k_d,k_I+1)$ for the ODE problem (2.8), (2.9). We can write the resulting system of algebraic equations for $\vec{\mathbf{v}}=(\mathbf{v}_0^T,\mathbf{v}_1^T,\ldots,\mathbf{v}_N^T)^T$ as

$$\mathbf{A}\vec{\mathbf{v}}=\vec{\mathbf{g}}$$

where $\mathbf{A}$ is an almost-block-diagonal matrix approximating the multiple shooting matrix and $\vec{\mathbf{g}}$ is an appropriate right hand side. For h sufficiently small, it follows that $\mathbf{A}$ is nonsingular, hence the discrete solution exists. Moreover, upon rescaling $\mathbf{A}$ into a divided difference form (calling the result $\mathbf{A}$ again) we obtain the stability bound

$$\|\mathbf{A}^{-1}\|\leq K+O(h)$$

with K as in (2.10a). The estimate $\mathbf{e}_n^v = O(h^{\min(k_d,k_I+1)})$ follows similarly. (For details see, e.g., Section 5.2.1 of [4]). For the errors in the intermediate stages, it follows from (3.10) and (3.12), noting that (3.12) implies $\underline{\mathbf{E}}^v = (I + O(h))\underline{\mathbf{e}}_{n-1}^v + O(\underline{\delta}^x)$, that

$$\underline{\mathbf{E}}^x = O(h^{\min(k_d,k_I+1)}) \tag{3.16a}$$

$$\underline{\mathbf{E}}^{x'} = O(h^{\min(k_d,k_I)}). \tag{3.16b}$$

It then follows from (3.6a) that also

$$\mathbf{E}^y = O(h^{\min(k_d,k_I+1)}). \tag{3.17}$$

□

3.2 Superconvergence for projected collocation methods

Consider now the special case where the unprojected IRK scheme is a collocation scheme. Then there are generally discontinuous functions $\mathbf{x}_\pi : [0,1] \to \mathcal{R}^{m_x}$, $\mathbf{y}_\pi : [0,1] \to \mathcal{R}^{m_y}$ such that for each element $[t_{n-1}, t_n]$, $\mathbf{x}_\pi \in \mathcal{P}_{k+1}[t_{n-1}, t_n)$, $\mathbf{y}_\pi \in \mathcal{P}_k[t_{n-1}, t_n)$,

$$\mathbf{x}_\pi(t_{n-1}) = \mathbf{x}_{n-1}, \qquad \mathbf{x}'_\pi(t_i) = \mathbf{X}'_i, \qquad \mathbf{x}_\pi(t_i) = \mathbf{X}_i,$$

$$\mathbf{y}_\pi(t_i) = \mathbf{Y}_i, \quad 1 \le i \le k \tag{3.18a}$$

$$\mathbf{x}_n = \mathbf{x}_\pi(t_n) + G_{12}^n \lambda_n \tag{3.18b}$$

$$\mathbf{0} = G_{21}^n \mathbf{x}_n + \mathbf{q}_2^n. \tag{3.18c}$$

Let

$$\mathbf{v}_\pi(t) := R(t)\mathbf{x}_\pi(t), \qquad 0 \le t \le 1. \tag{3.19}$$

Then (3.3a), (3.3b) yield for each i, $1 \le i \le k$,

$$\mathbf{X}_i = \mathbf{x}_\pi(t_i) = S^i \mathbf{v}_\pi(t_i) + \hat{\mathbf{q}}^i \tag{3.20a}$$

$$\mathbf{v}'_\pi(t_i) = [(RG_{11} + R')S]^i \mathbf{v}_\pi(t_i) + [R\mathbf{q}_1 + (RG_{11} + R')\hat{\mathbf{q}}]^i, \tag{3.20b}$$

i.e. $\mathbf{v}_\pi$ collocates the ODE (2.8). It satisfies the BC (2.9) as well. Moreover, by (3.18)

$$\mathbf{v}_\pi(t_n) = R^n \mathbf{x}_\pi(t_n) = R^n \mathbf{x}_n \tag{3.21}$$

so $\mathbf{v}_\pi \in C[0,1]$ (unlike $\mathbf{x}_\pi$). However, unless $R(t)$ is constant $\mathbf{v}_\pi$ is not a piecewise polynomial of order $k+1$ in general. Still, we can produce an analogue of the usual superconvergence argument (see, e.g. pp. 219-222 in [4]) based on the convergence results of Section 3.1.

Theorem 3.2 *Under the assumptions of Theorem 3.1, the projected collocation method satisfies for* $0 \le t \le 1$

$$|\mathbf{x}_\pi(t) - \mathbf{x}(t)| = O(h^{\min(k+1,k_d)}) \tag{3.22a}$$
$$|\mathbf{x}'_\pi(t) - \mathbf{x}'(t)| = O(h^k) \tag{3.22b}$$
$$|\mathbf{y}_\pi(t) - \mathbf{y}(t)| = O(h^k). \tag{3.22c}$$

Moreover, let the coefficient functions and the inhomogeneities in (2.1a), (2.1b) be in $C^{k_d+1}[0,1]$. *Then the nonstiff superconvergence order holds for the projected collocation method, viz.*

$$|\mathbf{x}_n - \mathbf{x}(t_n)| = O(h^{k_d}), \qquad 0 \le n \le N. \tag{3.23}$$

Proof:

By (3.16) the estimates (3.22a), (3.22b) hold at collocation points (note $k \le k_d$). Since

$$\mathbf{y}_\pi(t_i) - \mathbf{y}(t_i) = (G_{21}G_{12})^{-1}G_{21}(\mathbf{x}'_\pi(t_i) - \mathbf{x}'(t_i) - G_{11}(\mathbf{x}_\pi(t_i) - \mathbf{x}(t_i))$$

we have a similar result for $\mathbf{y}$. The estimates (3.22a), (3.22b) and (3.22c) are obtained upon noting that $\mathbf{x}'_\pi$ and $\mathbf{y}_\pi$ are (unattached) polynomials of order k on each mesh subinterval.

We next show superconvergence. Consider the collocation (3.20b) of the ODE (2.8). Let $G(t,s)$ be the Green's function of (2.8), (2.9) and define

$$\mathcal{L}\mathbf{v} \equiv \mathbf{v}' - [(RG_{11} + R')S]\mathbf{v}.$$

Then for each t in $[0,1]$,

$$\mathbf{e}^v_\pi(t) := \mathbf{v}_\pi(t) - \mathbf{v}(t) = \int_0^1 G(t,s)\mathcal{L}(\mathbf{v}_\pi(s) - \mathbf{v}(s))ds = \sum_{n=1}^{N}\int_{t_{n-1}}^{t_n} G(t,s)\mathcal{L}\mathbf{e}^v_\pi(s)ds.$$

For an interval (t_{n-1}, t_n) we have $\mathcal{L}\mathbf{e}^v_\pi(t_i) = 0$, $i = 1,2,\ldots,k$, so write

$$G(t,s)\mathcal{L}\mathbf{e}^v_\pi(s) = \mathbf{w}(s)\Pi_{i=1}^k(s - t_i).$$

We claim that if $t \notin (t_{n-1}, t_n)$ then the function $\hat{\mathbf{w}}(s) := (h_n/h)^{k-1}\mathbf{w}(s)$ has $k_d - k$ bounded derivatives and can therefore be written as

$$\hat{\mathbf{w}}(s) = \phi(s) + O(h_n^{k_d-k})$$

for some (vector) polynomial $\phi \in \mathcal{P}_{k_d-k}(t_{n-1}, t_n)$. Once we show this we obtain, noting that $\int_{t_{n-1}}^{t_n} \phi(s)\Pi_{i=1}^{k}(s-t_i)ds = \mathbf{0}$, the estimate

$$\int_{t_{n-1}}^{t_n} G(t,s)\mathcal{L}\mathbf{e}_\pi^v(s)ds = O(h_n^{k_d+2-k}h^{k-1}).$$

Then, for a mesh point t we have $t \notin (t_{n-1}, t_n)$ for all n, $1 \le n \le N$, so summing up the estimates for each n yields

$$\mathbf{e}_\pi^v(t) = O(h_n^{k_d+1-k}h^{k-1}),$$

and the result (3.23) follows.

It remains to show that the derivatives of $\hat{\mathbf{w}}(s)$ are bounded. The assumed smoothness of the problem coefficients yields boundedness of s-derivatives of $G(t,s)$, so it remains to show that k_d derivatives of $(h_n/h)^{k-1}\mathcal{L}\mathbf{e}_\pi^v(s)$ are bounded. Since $\mathbf{v}(s)$ and its k_d+1 derivatives are again bounded by assumption, it remains to consider k_d+1 derivatives of $(h_n/h)^{k-1}\mathbf{v}_\pi(s)$.

Using (3.19), write

$$\mathbf{v}_\pi^{(j)}(s) = \sum_{l=0}^{j} \binom{j}{l} R^{(j-l)}(s)\mathbf{x}_\pi^{(l)}(s).$$

The derivatives of R are bounded by assumption, and for $\mathbf{x}_\pi$ we have (3.22b) holding. Since $\mathbf{x}_\pi' \in \mathcal{P}_{(}t_{n-1}, t_n)$, this yields

$$\mathbf{x}_\pi^{(l)}(s) = \begin{cases} \mathbf{x}^{(l)}(s) + O(h^k h_n^{1-l}) & 1 \le l \le k \\ 0 & l > k \end{cases}$$

Substitution into the expression for $\mathbf{v}_\pi^{(j)}$ yields the claimed bound and completes the proof. □

Remark

The good stability properties of Theorem 3.1 are obtained essentially under the assumption that the underlying ODE (2.8) is nonstiff (relative to the maximum step size h used; indeed we have relied on proximity to the standard multiple shooting method for showing the third claim there). But symmetric difference schemes have proved useful for stiff boundary value ODEs (see, e.g., Section 10.3.2 of [4]). Anticipating possible stiffness, one may worry that the projected IRK method (3.2)-(3.3) does not look symmetric even when the unprojected method is.

However, viewing a (qualifying) projected IRK method from a collocation point of view puts such worries to rest: In addition to collocating the ODE and the algebraic constraints at collocation points, we also collocate the algebraic constraints at all mesh points. This is clearly symmetric in t provided that the points $c_1, c_2, \ldots, c_k$

are symmetric about 1/2. The unsymmetric appearance of (3.3) is due to an implementation choice: we have specified that on each mesh subinterval $\mathbf{x}_\pi$ is continuous to the right but (generally) not to the left. The method itself remains symmetric. □

3.3 Projected collocation for nonlinear problems

For a nonlinear DAE problem (2.20a)-(2.20c), equations (3.3a), (3.3b), (3.3e) are in general nonlinear. (For (3.3f) we apply a fixed-point iteration.) Consider a damped Newton iteration step: Given current iterate values $\bar{\mathbf{x}}_{n-1}$, $\bar{\mathbf{x}}_n$, $\bar{\mathbf{X}}_i$, $\bar{\mathbf{Y}}_i$, $\bar{\mathbf{X}}'_i$, $\bar{\mathbf{y}}_n$, values of the next iterate are given by $\mathbf{x}_n = \bar{\mathbf{x}}_n + \mu\delta\mathbf{x}_n$, $\mathbf{X}_i = \bar{\mathbf{X}}_i + \mu\delta\mathbf{X}_i$, etc., where $0 < \mu \leq 1$ is the damping factor ($\mu = 1$ gives Newton's method) and

$$\delta\mathbf{X}'_i = -(\bar{\mathbf{X}}'_i - \mathbf{g}_1(\bar{\mathbf{X}}_i, \bar{\mathbf{Y}}_i, t_i)) + G_{11}(\bar{\mathbf{X}}_i, \bar{\mathbf{Y}}_i, t_i)\delta\mathbf{X}_i + G_{12}(\bar{\mathbf{X}}_i, \bar{\mathbf{Y}}_i, t_i)\delta\mathbf{Y}_i \quad (3.24a)$$

$$\mathbf{0} = \mathbf{g}_2(\bar{\mathbf{X}}_i, t_i) + G_{21}(\bar{\mathbf{X}}_i, t_i)\delta\mathbf{X}_i, \qquad i = 1, 2, \ldots, k \quad (3.24b)$$

$$\delta\mathbf{X}_i = \delta\mathbf{x}_{n-1} + h_n \sum_{j=1}^{k} a_{ij}\delta\mathbf{X}'_j \quad (3.24c)$$

$$\mathbf{0} = \mathbf{g}_2(\bar{\mathbf{x}}_n, t_n) + G_{21}(\bar{\mathbf{x}}_n, t_n)\delta\mathbf{x}_n, \quad (3.24d)$$

$$\delta\mathbf{x}_n = \delta\mathbf{x}_{n-1} + h_n \sum_{j=1}^{k} b_j\delta\mathbf{X}'_j + G_{12}(\bar{\mathbf{x}}_n, \bar{\mathbf{y}}_n, t_n)\delta\lambda_n \quad (3.24e)$$

If the IRK method is a collocation method then it is more convenient to present the same Newton method in terms of quasilinearization. Thus, given a current iterate $\bar{\mathbf{x}}_\pi(t)$, $\bar{\mathbf{y}}_\pi(t)$ with $\bar{\mathbf{x}}_\pi(t_{n-1}) = \bar{\mathbf{x}}_{n-1}$, $\bar{\mathbf{x}}_\pi(t_i) = \bar{\mathbf{X}}_i$, etc., the next iterate $\mathbf{x}_\pi(t)$, $\mathbf{y}_\pi(t)$ is given as

$$\begin{aligned} \mathbf{x}_\pi(t) &= \bar{\mathbf{x}}_\pi(t) + \mu\delta\mathbf{x}_\pi(t) \\ \mathbf{y}_\pi(t) &= \bar{\mathbf{y}}_\pi(t) + \mu\delta\mathbf{y}_\pi(t), \end{aligned}$$

where $\delta\mathbf{x}_\pi$, $\delta\mathbf{y}_\pi$ is the projected collocation solution to the linearized problem (cf. (2.21))

$$\mathcal{L}_1[\bar{\mathbf{x}}_\pi, \bar{\mathbf{y}}_\pi](\delta\mathbf{x}, \delta\mathbf{y}) = -(\bar{\mathbf{x}}'_\pi(t) - \mathbf{g}_1(\bar{\mathbf{x}}_\pi(t), \bar{\mathbf{y}}_\pi(t), t)) \quad (3.25a)$$

$$\mathcal{L}_2[\bar{\mathbf{x}}_\pi](\delta\mathbf{x}) = -\mathbf{g}_2(\bar{\mathbf{x}}_\pi(t), t) \quad (3.25b)$$

$$\mathcal{L}_3[\bar{\mathbf{x}}_\pi](\delta\mathbf{x}) = -\mathbf{b}(\bar{\mathbf{x}}_\pi(0), \bar{\mathbf{x}}_\pi(1)). \quad (3.25c)$$

It is not difficult to see that the two approaches yield the same results, i.e. the operations of discretization and linearization commute. For this method we may now use standard arguments, combining results from the stability theorems 2.1, 3.1 and the convergence theorems 3.1 and 3.2 to obtain

Theorem 3.3 *Let $\mathbf{x}(t)$, $\mathbf{y}(t)$ be an isolated solution of the DAE problem (2.20a)-(2.20c) and assume that $\mathbf{g}_1$ and $\mathbf{g}_2$ have continuous second partial derivatives and*

that the smoothness assumptions of Theorem 3.2 hold for the linearized problem in the neighborhood of $\mathbf{x}(t)$, $\mathbf{y}(t)$. *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* $\mathbf{x}_\pi(t)$, $\mathbf{y}_\pi(t)$ *to the projected collocation equations (3.3) in a tube* $S_\rho(\mathbf{x}, \mathbf{y})$ *of radius* ρ *around* $\mathbf{x}(t)$, $\mathbf{y}(t)$.

2. *This solution can be obtained by Newton's method, which converges quadratically provided that the initial guess for* $\mathbf{x}_\pi(t)$, $\mathbf{y}_\pi(t)$ *is sufficiently close to* $\mathbf{x}(t)$, $\mathbf{y}(t)$.

3. *The error estimates (3.22a)-(3.23) hold.*

Proof:

The proof follows standard lines (see, e.g. Ch. 5 of [4]). We first consider the projected collocation method for the linearized problem at the exact, isolated solution

$$\begin{aligned}
\mathcal{L}_1[\mathbf{x},\mathbf{y}](\hat{\mathbf{x}},\hat{\mathbf{y}})(t) &= \mathbf{g}_1(\mathbf{x}(t),\mathbf{y}(t),t) \\
&\quad - G_{11}(\mathbf{x}(t),\mathbf{y}(t),t)\mathbf{x}(t) - G_{12}(\mathbf{x}(t),\mathbf{y}(t),t)\mathbf{y}(t) \quad (3.26a)\\
\mathcal{L}_2[\mathbf{x}](\hat{\mathbf{x}})(t) &= \mathbf{g}_2(\mathbf{x}(t),t) + G_{21}(\mathbf{x}(t),t)\mathbf{x}(t) \quad (3.26b)\\
\mathcal{L}_3[\mathbf{x}](\hat{\mathbf{x}}) &= \mathbf{b}(\mathbf{x}(0),\mathbf{x}(1)) + B_0\mathbf{x}(0) + B_1\mathbf{x}(1). \quad (3.26c)
\end{aligned}$$

Denote the collocation solution to this linear problem for $\hat{\mathbf{x}} = \mathbf{x}$ and $\hat{\mathbf{y}} = \mathbf{y}$ by $\hat{\mathbf{x}}_\pi$, $\hat{\mathbf{y}}_\pi$. Since Theorem 2.1 holds for (3.26), Theorems 3.1 and 3.2 hold for $\hat{\mathbf{x}}_\pi(t)$, $\hat{\mathbf{y}}_\pi(t)$ and $\hat{\mathbf{x}}_\pi(t) - \mathbf{x}(t)$, $\hat{\mathbf{y}}_\pi(t) - \mathbf{y}(t)$. The collocation operator has a bounded inverse at $\mathbf{x}$, $\mathbf{y}$, hence by the assumed smoothness also at $\hat{\mathbf{x}}_\pi$, $\hat{\mathbf{y}}_\pi$, and

$$|\hat{\mathbf{x}}_\pi(t) - \mathbf{x}(t)|,\ |\hat{\mathbf{y}}_\pi(t) - \mathbf{y}(t)| = O(h^k), \qquad 0 \leq t \leq 1.$$

For h sufficiently small, this means that the Newton-Kantorovich Theorem applies, yielding existence of a projected collocation solution to the nonlinear problem and quadratic convergence of Newton's iterates to it. Also

$$|\mathbf{x}_\pi(t) - \mathbf{x}(t)|, |\mathbf{y}_\pi(t) - \mathbf{y}(t)| = O(h^k), \qquad 0 \leq t \leq 1.$$

Finally, write (2.20a)-(2.20c) for $\tilde{\mathbf{x}}$, $\tilde{\mathbf{y}}$ as

$$\begin{aligned}
\mathcal{L}_1[\mathbf{x},\mathbf{y}](\tilde{\mathbf{x}},\tilde{\mathbf{y}})(t) &= \mathbf{g}_1(\mathbf{x}(t),\mathbf{y}(t),t) - G_{11}(\mathbf{x}(t),\mathbf{y}(t),t)\mathbf{x}(t) - G_{12}(\mathbf{x}(t),\mathbf{y}(t),t)\mathbf{y}(t) \\
&\quad + O(|\mathbf{x}(t) - \tilde{\mathbf{x}}(t)|^2 + |\mathbf{y}(t) - \tilde{\mathbf{y}}(t)|^2) \quad (3.27a)\\
\mathcal{L}_2[\mathbf{x}](\tilde{\mathbf{x}})(t) &= \mathbf{g}_2(\mathbf{x}(t),t) + G_{21}(\mathbf{x}(t),t)\mathbf{x}(t) + O(|\mathbf{x}(t) - \tilde{\mathbf{x}}(t)|^2) \quad (3.27b)\\
\mathcal{L}_3[\mathbf{x}](\tilde{\mathbf{x}}) &= \mathbf{b}(\mathbf{x}(0),\mathbf{x}(1)) + B_0\mathbf{x}(0) + B_1\mathbf{x}(1) \\
&\quad + O(|\mathbf{x}(0) - \tilde{\mathbf{x}}(0)|^2 + |\mathbf{x}(1) - \tilde{\mathbf{x}}(1)|^2) \quad (3.27c)
\end{aligned}$$

and note that $\mathbf{x}_\pi$, $\mathbf{y}_\pi$ collocate (3.27) while $\hat{\mathbf{x}}_\pi$, $\hat{\mathbf{y}}_\pi$ collocate (3.26). Using the stability of the linear collocation operator, this yields

$$\max_{0\le t\le 1} |\mathbf{x}_\pi(t) - \hat{\mathbf{x}}_\pi(t)| \le \text{const.} \max_{0\le t\le 1} |\mathbf{x}_\pi(t) - \mathbf{x}(t)|^2 = O(h^{2k})$$

(and a similar result for $\mathbf{y}_\pi$). Since the results of Theorem 3.2 hold for $\hat{\mathbf{x}}_\pi$, they hold also for $\mathbf{x}_\pi$, because $k_d \le 2k$. This completes the proof. □

4 Numerical examples

The projected and unprojected collocation methods based on Gauss and on Radau points have been implemented. Some test runs are reported and discussed in this section. Recall that the Gauss schemes are symmetric, with $c_k < 1$ and $k_d = 2k$; the midpoint scheme is obtained for $k = 1$. The Radau schemes are not symmetric, with $c_k = 1$ and $k_d = 2k - 1$; the backward Euler scheme is obtained for $k = 1$. Thus, the unprojected and projected Radau schemes coincide, but the unprojected and projected Gauss schemes are different from each other. As discussed in the introduction, the unprojected Gauss methods may yield poorer results both because of a possibly larger stability constant and because of a possible reduction in the accuracy order. The error in the unprojected Gauss method is also less smooth than the error in the projected one, making it more difficult to control the unprojected error by local mesh adjustments. All this will be demonstrated below.

At each step n, $1 \le n \le N$, we have in (3.3a), (3.3b), (3.3d), $k(m_x+m_y)$ algebraic equations expressing $\mathbf{X}'_1, \mathbf{X}'_2, \ldots, \mathbf{X}'_k$ and $\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_k$ in terms of $\mathbf{x}_{n-1}$. In case of the projected method we may substitute (3.3f) into (3.3e) to obtain m_y additional equations for λ_n. These equations are linear when the DAE is linear (or linearized) and can be solved locally (i.e. we perform static elimination). Then (3.3f) is used to obtain a relation of the form

$$\mathbf{x}_n = \Gamma_n \mathbf{x}_{n-1} + \mathbf{r}_n \qquad\qquad 1 \le n \le N$$

and this is solved together with the boundary conditions

$$\mathbf{0} = \mathbf{b}(\mathbf{x}_0, \mathbf{x}_N)$$

appended by (3.2).

In our implementation we have actually relied on using an $R(t)$ satisfying (2.2) at collocation and mesh points. This R does not have to be smooth. We use it to eliminate first at each step n the unknowns $\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_k$ and λ_n. If $R(t)$ is not given by the user then we compute it by selecting $m_x - m_y$ linearly independent rows of the projector $I - H$ where H is defined in (2.12). The same rows are selected for each t so long as they form a well-conditioned $R(t)$.

All results reported in the tables below are for uniform meshes with N subintervals. We use the notation err_i - absolute error in the ith component of $\mathbf{x}$; *rate* - corresponding convergence rate; $.a - b \equiv 0.a \times 10^{-b}$.

Example 1 revisited

This example, although linear, is particularly nasty: While $\|G_{12}\| \sim \lambda$, G_{21} and $G_{21}G_{12}$ are independent of λ. It can be verified that the ghost ODE which governs the stability of the unprojected midpoint scheme [2] is

$$\hat{y}' = \frac{\lambda}{2-t}\hat{y}$$

which is unstable exponentially in λ.

In Table 4.1 we list the maximum error at mesh points in x_1 for some sample runs. The exact solution is

$$\mathbf{x}(t) = \begin{pmatrix} e^t \\ e^t \end{pmatrix}, \quad y(t) = -\frac{e^t}{2-t}$$

and the error in x_2 exhibits similar behaviour to that in x_1.

λ	scheme	k	N	projected?	err_1
1	Gauss	1	10	no	.20-2
			20		.49-3
	Gauss	1	10	yes	.32-2
			20		.80-3
10	Gauss	1	20	no	.12+1
			40		.37
			80		.98-1
			160		.25-1
	Gauss	1	20	yes	.35-2
			40		.81-3
50	Gauss	1	80	no	.96+11
			160		.85+11
	Gauss	1	40	yes	.58-2
			80		.12-2
			160		.27-3
	Radau	1	40		.13-1
	Gauss	3	40	no	.18+8
			80		.79+6
			160		.44+5
	Gauss	3	20	yes	.71-7
			40		.74-9
	Radau	3	20		.25-5
			40		.67-8

Table 4.1: Maximum errors at mesh points for Example 1

From these errors it is clear that the projected Gauss schemes do not suffer the instability which troubles the unprojected Gauss schemes as λ increases. The relative

size of errors in projected Gauss schemes and Radau schemes is as expected, in view of their orders. □

Example 2

The nonlinear BVP

$$
\begin{aligned}
x_1' &= x_3 - y_2 x_1 \\
x_2' &= x_4 - y_2 x_2 \\
x_3' &= -y_1 x_1 + e^t(1 + \sin t) \\
x_4' &= -y_1 x_2 + \frac{1}{1+t}\left(\frac{2}{(1+t)^2} + \sin t\right) \\
x_1 x_2^3 + e^{x_2} &= \frac{e^t}{(1+t)^3} - e^{1/(1+t)} \\
x_3 x_2^3 + (3x_1 x_2^3 + e^{x_2}) x_4 &= \frac{e^t}{(1+t)^3} - \frac{3e^t}{(1+t)^4} - \frac{e^{1/(1+t)}}{(1+t)^2},
\end{aligned}
$$

where $x_1(0) = 1$, $x_1(1) = e$, has the exact solution $\mathbf{x}^T = (e^t, e^t, (1+t)^{-1}, -(1+t)^{-2})$, $\mathbf{y}^T = (\sin t, 0)$. We use

$$
R(t) = \begin{pmatrix} x_2 & 0 & -x_1 & 0 \\ 0 & x_2 & 0 & -x_1 \end{pmatrix}.
$$

Maximum errors at mesh points in x_1 and in x_3 and calculated corresponding convergence rates are listed in Table 4.2.

The expected convergence rate for a k-stage Radau scheme is $2k-1$; that for a projected k-stage Gauss scheme is $2k$; and that for an unprojected k-stage Gauss scheme is k if k is even, $k+1$ if k is odd. These rates are all demonstrated in Table 4.2.

Moreover, if the mesh is arbitrarily nonuniform (in particular, if it does not hold that $h_n = h_{n-1}(1 + O(h_n))$ for almost all n odd or for almost all n even) then the expected rate of convergence for a k-stage unprojected Gauss scheme drops from $k+1$ to k when k is odd (see, e.g., Section 10.3.2 of [4]). Additional experimentation verifies that this is indeed the case for the current example. In particular, the convergence rate for the midpoint scheme drops to $O(h)$ when using a mesh with $h_n = h$ for n odd and $h_n = h/2$ for n even, and the obtained approximation is less accurate than the one obtained using the coarser, uniform mesh with step size h. No corresponding drop in accuracy occurs for the projected scheme.

In Figures 4.1 and 4.2 we plot the errors in x_1 using a uniform mesh with $N = 10$ for the unprojected midpoint scheme and for the projected midpoint scheme, respectively. Both schemes are $O(h^2)$, but from the plots it is clear that the unprojected error is much less smooth than the projected error, making it tougher to estimate and control it. □

scheme	k	N	projected?	err_1	$rate_1$	err_3	$rate_3$
Gauss	1	5	no	.61-1		.94-1	
		10		.17-1	1.9	.34-1	1.5
		20		.46-2	1.9	.87-2	2.0
	1	5	yes	.40-2		.38-1	
		10		.91-3	2.1	.91-2	2.1
		20		.22-3	2.0	.22-2	2.0
	2	5	no	.66-3		.26-1	
		10		.17-3	2.0	.65-2	2.0
		20		.42-4	2.0	.16-2	2.0
	2	5	yes	.62-5		.38-4	
		10		.40-6	3.9	.22-5	4.1
		20		.25-7	4.0	.13-6	4.1
	3	5	no	.20-3		.67-3	
		10		.16-4	3.7	.44-4	3.9
		20		.11-5	3.9	.28-5	4.0
	3	5	yes	.90-8		.73-7	
		10		.13-9	6.1	.12-8	6.0
		20		.20-11	6.0	.18-10	6.0
Radau	1	5		.76-1		.27	
		10		.40-1	.93	.13	1.1
		20		.20-1	.99	.61-1	1.1
	2	5		.45-3		.17-2	
		10		.55-4	3.0	.21-3	3.0
		20		.68-5	3.0	.26-4	3.0

Table 4.2: Maximum errors and convergence rates at mesh points for Example 2

Example 3

The famous pendulum problem

$$\begin{aligned} x_1'' &= -yx_1 \\ x_2'' &= -yx_2 - g \\ 0 &= x_1^2 + x_2^2 - L^2 \end{aligned}$$

(where (x_1, x_2) are cartesian coordinates of an infinitesimal ball of mass 1, L is the pendulum length, y is the tension in the bar and g is the gravitational force) can be converted to an index-2 DAE by one constraint differentiation:

$$\begin{aligned} x_1' &= x_3 \\ x_2' &= x_4 \\ x_3' &= -yx_1 \\ x_4' &= -yx_2 - g \\ 0 &= x_1x_3 + x_2x_4 \end{aligned} \tag{4.1}$$

where $0 \le t \le T$, and initial conditions satisfy $x_2(0) = -\sqrt{L^2 - x_1^2(0)}$.

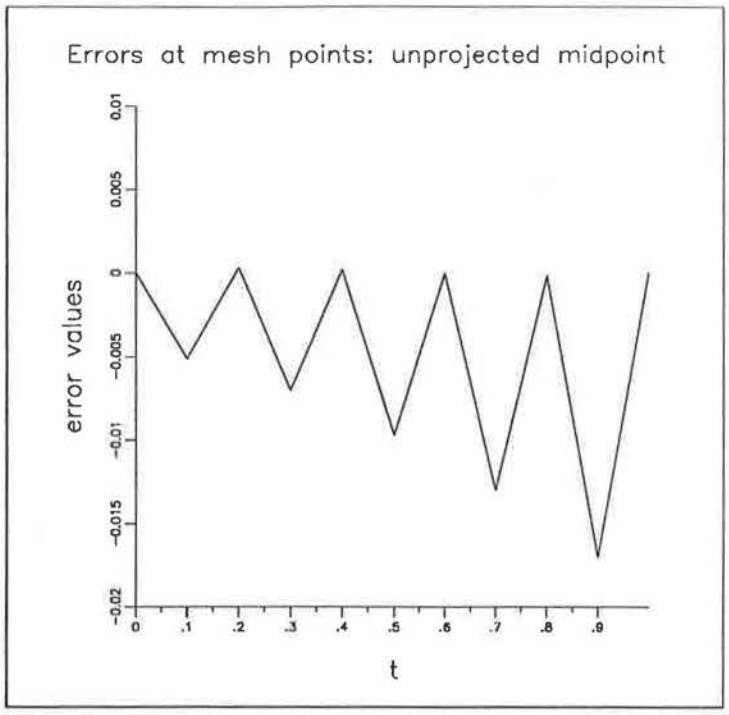

Figure 4.1: Absolute error in x_1 for Example 2 using unprojected midpoint

First, consider the IVP with $L = 1$, $g = 1$, $x_1(0) = 1$, $x_4(0) = -1$, and $T = 1$. As is customary, we calculate the "drift"

$$\text{drift} = x_1^2(1) + x_2^2(1) - L^2$$

in addition to the errors at $t = T$, based on the "exact" values given in [9]. We list errors in x_1 and in x_3 in Table 4.3.

We make two observations: First, there appears to be no drift (up to machine accuracy) in the results for Gauss schemes; in contrast, for the Radau schemes the drift is of the order of accuracy in x_3. Second, the unprojected Gauss schemes suffer no reduced accuracy in x_1 (and also in the unlisted x_2), even for $k = 2$.

Before explaining these observations, we report on another set of experiments with this DAE. This time we want to find the period when the ball is dropped from a horizontal position. Thus, we treat T as an unknown constant and specify $x_1(0) = L$, $x_2(0) = 0$, $x_4(0) = 0$, $x_1(T) = 0$ for a quarter of the period. (The resulting BVP has many solutions; we are looking for the one with the smallest positive T.) We may rescale the independent variable by $\tau = t/T$, adding the ODE $dT/d\tau = 0$ to the system, so $m_x = 5$, $m_y = 1$, $0 \le \tau \le 1$. Choosing the scaling $L = 1$, $g = 13.750371633$ yields a period of almost exactly 2 [14], so we can compare errors in T.

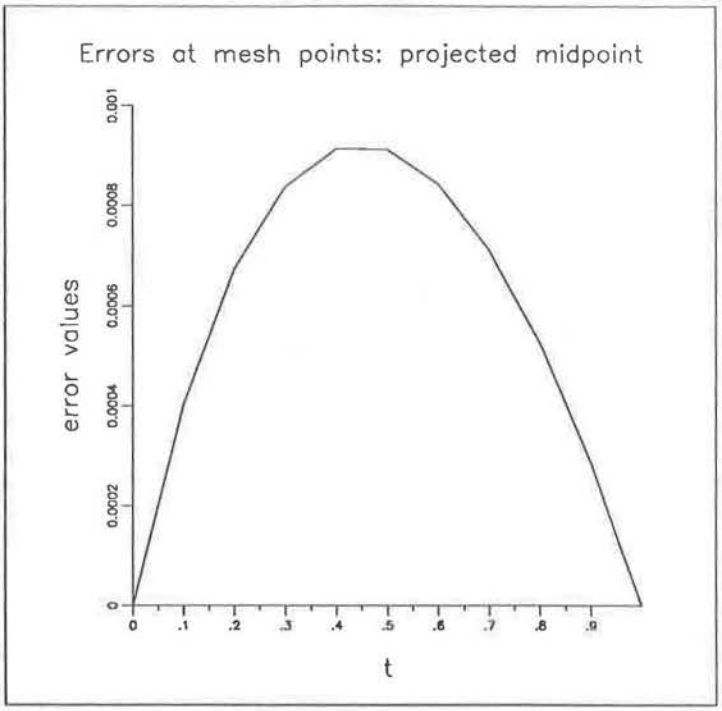

Figure 4.2: Absolute error in x_1 for Example 2 using projected midpoint

Unlike in other nonlinear examples reported here, where Newton's method converged quickly from very rough starts, here we needed to use a continuation step (specifically, a problem instance for $x_2(0) = -.1$ was solved first). Typical accuracy was as reported for the IVP: for instance, using a Gauss scheme with $k = 2$ and $N = 10$, the error in the period was .17-4. Again there was no drift for Gauss schemes, and the unprojected scheme gave as accurate a value for T as the projected one. □

Analysis and one more example

The previous example has two relevant properties: it is obtained by an (unstabilized) differentiation of the constraint of an index-3 DAE of the form

$$\mathbf{z}'' = \mathbf{g}_1(\mathbf{z}, \mathbf{z}', \mathbf{y}, t) \tag{4.2a}$$

$$\mathbf{0} = \mathbf{g}_2(\mathbf{z}, t), \tag{4.2b}$$

and $\mathbf{g}_2$ has only up to quadratic terms in $\mathbf{z}$. (Quadratic constraints occur also in other applications in mechanics and in chemical reactions [7].)

scheme	k	N	projected?	err_1	err_3	drift
Radau	1	10		.28-1	.20	-.19
		20		.17-1	.10	-.10
		40		.96-2	.51-1	-.52-1
	2	10		.10-3	.25-3	-.15-3
		20		.13-4	.31-4	-.19-4
		40		.17-5	.39-5	-.24-5
Gauss	1	10	no	.38-2	.94-3	0
		20		.95-3	.23-3	0
	1	10	yes	.36-2	.12-2	0
		20		.93-3	.30-3	0
	2	10	no	.34-5	.85-4	0
		20		.21-6	.21-4	0
	2	10	yes	.35-5	.11-5	0
		20		.22-6	.69-7	0

Table 4.3: Maximum errors and drift at $t = 1$ for Example 3

The lack of drift in the Gauss schemes occurs because of the special form of $\mathbf{g}_2$ and the fact that we use collocation at Gaussian points. The derivative of $\mathbf{g}_2$ is integrated exactly in this case. For instance, consider the constraint

$$\phi(t) = x_1^2 + x_2^2 - L^2$$

whose differentiated form is being collocated. We have

$$x_\pi^1(t_n)^2 + x_\pi^2(t_n)^2 - L^2 = \phi_\pi(t_n) = \phi_\pi(t_{n-1}) + \int_{t_{n-1}}^{t_n} \phi_\pi'(s)ds.$$

Now, $\int_{t_{n-1}}^{t_n} \phi_\pi'(s)ds$ is equal to its Gaussian quadrature formula, because the integrand is a polynomial of order $2k$, and this is the precision of Gaussian quadrature. But at each collocation point t_j, $\phi_\pi'(t_j) = 0$. Therefore, $\int_{t_{n-1}}^{t_n} \phi_\pi'(s)ds = 0$, which yields in turn

$$\phi_\pi(t_n) = \phi_\pi(t_{n-1}) = \cdots = \phi_\pi(0) = 0.$$

In order to understand the superconvergence obtained unexpectedly for $\mathbf{z}$ using an unprojected Gauss scheme, it is sufficient to consider the linear DAE

$$\mathbf{z}' = \mathbf{u} \tag{4.3a}$$
$$\mathbf{u}' = G_{11}\mathbf{z} + \hat{G}_{11}\mathbf{u} + G_{12}\mathbf{y} + \mathbf{q}_1 \tag{4.3b}$$
$$\mathbf{0} = \hat{G}_{21}\mathbf{z} + G_{21}\mathbf{u} + \mathbf{q}_2 \tag{4.3c}$$

where $G_{21}G_{12}$ is nonsingular, $G_{21}(t) \in \mathcal{R}^{m_y \times m_z}$, $m_y \le m_z$. Let $\hat{R}(t) \in \mathcal{R}^{(m_z - m_y)\times m_z}$ be a smooth, bounded, full-row-rank function satisfying

$$\hat{R}G_{12} = 0.$$

Identifying (4.3) as a special case of (4.2) with $\mathbf{x}^T = (\mathbf{z}^T, \mathbf{u}^T)$, $m_x = 2m_z$, we may choose $R(t)$ satisfying (2.2) as

$$R(t) = \begin{pmatrix} I & 0 \\ 0 & \hat{R}(t) \end{pmatrix}.$$

In the proof of Theorem 3.2 we have shown superconvergence for $\mathbf{v}_\pi = R\mathbf{x}_\pi$ regardless of whether the method was projected or not (i.e., the same $\mathbf{v}_\pi$ also satisfies $\mathbf{v}_\pi = R\hat{\mathbf{x}}_\pi$, where $\hat{\mathbf{x}}_\pi$ is the unprojected collocation solution). But here,

$$\mathbf{v} = \begin{pmatrix} \mathbf{z} \\ \hat{R}\mathbf{u} \end{pmatrix}$$

i.e. $\mathbf{z}$ simply consists of the first m_z components of $\mathbf{v}$. Therefore, we have superconvergence for $\mathbf{z}$ with the unprojected method as well. (Note that this argument does not depend on any special property of the chosen collocation points.)

We illustrate the above arguments using the following example (referred to as Example 4 in Table 4.4

$$\begin{aligned}
x_1' &= x_3 \\
x_2' &= x_4 \\
x_3' &= -y_1 x_1 + e^t(1+\sin t) \\
x_4' &= -y_1 x_2 + \frac{1}{1+t}\left(\frac{2}{(1+t)^2} + \sin t\right) \\
x_3 x_2^3 + (3x_1 x_2^3 + e^{x_2})x_4 &= \frac{e^t}{(1+t)^3} - \frac{3e^t}{(1+t)^4} - \frac{e^{1/(1+t)}}{(1+t)^2},
\end{aligned}$$

where $x_1(0) = 1$, $x_3(0) = 1$, and $x_1(1) = e$. The solution is the same as that of Example 2. In fact, both examples were derived by one differentiation of the constraint in the index-3 DAE

$$\begin{aligned}
z_1'' &= -y_1 z_1 + e^t(1+\sin t) \\
z_2'' &= -y_1 z_2 + \frac{1}{1+t}\left(\frac{2}{(1+t)^2} + \sin t\right) \\
0 &= z_1 z_2^3 + e^{z_2} - \frac{e^t}{(1+t)^3} + e^{1/(1+t)}
\end{aligned}$$

(this in turn is a modification of the pendulum equations where the constraint is made to be nonquadratic and inhomogeneities are fixed to know the solution) the difference being that in Example 2 we have added a stabilizing multiplier (cf. Sections 1 and 2). Results for this unstabilized problem are displayed in Table 4.4.

Observe first that, since the constraint is not quadratic, a nonzero drift appears when using the Gauss schemes. The errors in $x_1 = z_1$ are very close for the projected

scheme	k	N	projected?	err_1	$rate_1$	err_3	$rate_3$	drift
Gauss	1	5	no	.43-2		.13		.29-1
		10		.13-2	1.8	.42-1	1.7	.74-2
		20		.30-3	2.1	.11-1	2.0	.19-2
	1	5	yes	.55-2		.37-1		.29-1
		10		.13-2	2.1	.88-2	2.1	.74-2
		20		.30-3	2.1	.21-2	2.0	.29-2
	2	5	no	.16-4		.28-1		.11-3
		10		.99-6	4.0	.70-2	2.0	.75-5
		20		.63-7	4.0	.17-2	2.0	.47-6
	2	5	yes	.47-5		.36-4		.11-3
		10		.27-6	4.1	.20-5	4.2	.75-5
		20		.16-7	4.1	.11-6	4.1	.47-6
Radau	2	5		.54-3		.17-2		.16-2
		10		.64-4	3.1	.20-3	3.1	.21-3
		20		.78-5	3.0	.25-4	3.0	.26-4

Table 4.4: Maximum errors and convergence rates at mesh points for Example 4

and unprojected schemes (similarly for $x_2 = z_2$, and therefore also for the drift). Using the unprojected Gauss scheme, a full superconvergence order is obtained for x_1 but not for x_3, as expected. In Table 4.2, on the other hand, there is no superconvergence for x_1 (nor for x_2) either. Thus we obtain the curious result that the errors, e.g. in x_1 using $k = 2$, are much better using the unstabilized formulation than those obtained using the stabilized one. Note, however, that in general the well-conditioning of the unstabilized problem does not follow from the well-conditioning (appropriately defined) of the index-3 problem, while that of the stabilized problem does. Also, preferable to using the unprojected schemes in both formulations is using the projected schemes. □

References

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

[2] U. Ascher, *On symmetric schemes and differential-algebraic equations*, SIAM J. Sci. Stat. Comput. 10 (1989), 937-949.

[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, 1988.

[5] H.G. Bock, E. Eich and J. Schloder, *Numerical solution of constrained least squares boundary value problems in differential-algebraic equations*, Proc. Halle Conf. on Numerical Treatment of Differential Equations, Teubner-Texte Math 104, Leipzig, 1987.

[6] J. Baumgarte, *Stabilization of constraints and integrals of motion in dynamical systems*, Comp. Math. Appl. Mech. Eng. 1 (1976), 1-16.

[7] B.P. Boudreau, *A steady-state diagenetic model for dissolved carbonate species and pH in the porewaters of oxic and suboxic sediments*, Geochimica et Cosmochimica Acta 51 (1987), 1985-1196.

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

[9] K. Brenan, S. Campbell and L. Petzold, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, North-Holland, 1989.

[10] K. Brenan and L. Petzold, *The numerical solution of higher index differential/algebraic equations by implicit Runge-Kutta methods*, SIAM J. Numer. Anal. 26 (1989).

[11] S. Campbell and B. Leimkuhler, *Differentiation of Constraints in Differential Algebraic Equations*, J. Mechanics of Structures and Machines, to appear.

[12] K. Clark and L. Petzold, *Numerical Solution of Boundary Value Problems in Differential-Algebraic Systems*, SIAM J. Sci. Stat. Comput. 10 (1989), 915-936.

[13] V. Dolezal, *The existence of a continuous basis of a certain linear subspace of E_r, which depends on a parameter*, Časopis pro pěstováni matematiky, roč. 89, Praha, (1964), 466-468.

[14] K. Führer and B. Leimkuhler, *Formulation and Numerical Solution of the Equations of Constrained Mechanical Motion*, DFVLR-FB 89-08, Munich, 1989.

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

[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 vol. 1409, Springer-Verlag, 1989.

[18] W.H. Hundsdorfer, *Stability results for θ-methods applied to a class of stiff differential-algebraic equations*, CWI Report NM-R8708, Amsterdam, 1987.

[19] J. S. Logsdon and L. T. Biegler, *Accurate solution of differential-algebraic optimization problems*, Industrial and Engineering Chemistry Research 28 (1989) 1628-1639..

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