Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 221 (2007) 739–760
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Arbitrary order Krylov deferred correction methods for differential algebraic equations

Jingfang Huang [*,1], Jun Jia [1], Michael Minion [2]

Department of Mathematics, University of North Carolina, Chapel Hill, CB#3250, Phillips Hall, Chapel Hill, NC 27599, USA

Received 6 December 2005; received in revised form 20 May 2006; accepted 28 June 2006
Available online 14 August 2006

Abstract

In this paper, a new framework for the construction of accurate and efficient numerical methods for differential algebraic equation (DAE) initial value problems is presented. The methods are based on applying spectral deferred correction techniques as preconditioners to a Picard integral collocation formulation for the solution. The resulting preconditioned nonlinear system is solved using Newton–Krylov schemes such as the Newton-GMRES method. Least squares based orthogonal polynomial approximations are computed using Gaussian type quadratures, and spectral integration is used to avoid the numerically unstable differentiation operator. The resulting Krylov deferred correction (KDC) methods are of arbitrary order of accuracy and very stable. Preliminary results show that these new methods are very competitive with existing DAE solvers, particularly when high precision is desired.

Subject classifications: 65B05; 65F10; 65L80

Keywords: Spectral deferred correction methods; Differential algebraic equations; Spectral integration; Preconditioners; Krylov subspace methods; GMRES

1. Introduction

This paper introduces a new class of methods for the numerical solution of differential algebraic equation (DAE) systems of the form

$$F(y(t), y'(t), t) = 0. \tag{1}$$

* Corresponding author. Fax: +1 919 962 9345.
E-mail addresses: huang@amath.unc.edu (J. Huang), junjia@amath.unc.edu (J. Jia), minion@amath.unc.edu (M. Minion).
[1] The work of this author was supported in part by NSF under Grants DMS0411920 and DMS0327896.
[2] The work of this author was supported in part under contract DE-AC03-76SF00098 by the Director, Department of Energy (DOE) Office of Science; Office of Advanced Scientific Computing Research; Office of Mathematics, Information, and Computational Sciences, as well as the Alexander von Humboldt Foundation.

0021-9991/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2006.06.040

DAEs arise naturally in many applications, for example, the discretization of partial differential equations (PDEs) or model reduction and singular perturbations [8]. Compared with ordinary differential equations (ODEs), the numerical solution of DAEs is, in general, a more challenging subject since the algebraic part of the DAE can often be expressed as the infinite stiffness limit of a singular perturbation problem, and such stiffness typically poses difficulty to traditional numerical ODE methods. Currently, several methods are available for solving DAEs, including the backward differentiation formulas (BDF) based package DASSL developed by Petzold et al. which is applicable to DAE problems of index 0 and 1 [8,29]; and the Runge–Kutta based RADAU by Hairer et al. which can be applied to DAE problems of index up to 3 [18,20]. Detailed discussions of these available solvers as well as a test set can be found in [1], and the readers are referred to the references therein.

In this paper, we discuss a new class of numerical methods for DAE initial value problems which are based on a combination of ideas from spectral deferred correction methods for ODEs and inexact Newton methods (Newton–Krylov methods) for solving nonlinear equations. Deferred and defect correction methods, first proposed by Pereyra and Zadunaisky [26,36,37], build higher-order accurate solutions of initial value ODEs by iteratively approximating an equation for the error or defect to increase the accuracy of a provisional solution. Recently, Dutt et al. [13] presented a new variation on the deferred/defect correction strategy for ODEs which is based on a Picard integral equation form of the correction equation and utilizes spectral integration on Gaussian quadrature nodes. The resulting spectral deferred correction (SDC) schemes can, in principle, achieve arbitrary order of accuracy for both stiff and non-stiff problems, and (unlike linear-multistep methods) the linear stability properties of higher-order versions of the methods are similar to those of lower-order versions. Furthermore, each substep of the SDC method is computationally no more complex than that of a first-order Euler method.

More recently, it was shown in [5,21] that the deferred correction process can be considered as an iterative scheme with the corresponding fixed point being the solution to the collocation formulation of the ODE. In particular, for linear problems, it was shown in [21] that SDC methods are equivalent to solving a preconditioned form of the collocation equation for the error by a Neumann series expansion. Therefore, Krylov subspace methods such as the generalized minimum residual (GMRES) procedure can be utilized to accelerate the convergence. For nonlinear problems, the Krylov-SDC techniques are applied in [21] to a linear implicit formulation of the error equation. Stability and accuracy analyses in [21] demonstrate that the accelerated SDC methods provide improvements in the accuracy, efficiency, and stability of the original SDC approach. Furthermore, results show that the acceleration also effectively eliminates the order reduction previously observed in SDC and other methods [9,12,24,34] for certain stiff ODE systems.

The purpose of this paper is to demonstrate that the accelerated SDC techniques originally developed for ODEs can be generalized to construct solvers for initial value DAEs which have arbitrary order of accuracy while maintaining at each substep the same computational complexity of a simple first-order Euler method. Indeed SDC ideas have recently been applied to certain classes of DAEs [30–32] and PDEs [7,25]. In this paper we present a more general approach based on a Picard integral type formulation for DAEs

$$F\left(y_0 + \int_0^t Y(\tau)\,\mathrm{d}\tau, Y(t), t\right) = 0, \tag{2}$$

where $Y(t) = y'(t)$ is introduced as the new unknown function and $y(t)$ is recovered using quadrature. Although this formulation is quite general, we discuss in Section 4 how it can be modified to increase efficiency for systems with algebraic and differential variables.

Introducing the spectral integration matrix S as in [21], the discretized collocation formulation is given by

$$\mathbf{F}(\mathbf{y}_0 + \Delta t S \otimes \mathbf{Y}, \mathbf{Y}, \mathbf{t}) = \mathbf{0}, \tag{3}$$

which will be symbolically denoted as $\mathbf{H}(\mathbf{Y}) = \mathbf{0}$. In the formula, $\mathbf{t} = [t_1, t_2, \ldots, t_p]^{\mathrm{T}}$ represents the Gaussian nodes, $\mathbf{y}_0 = [y_0, y_0, \ldots, y_0]^{\mathrm{T}}$ the vector of initial values, $\mathbf{Y} = [Y_1, Y_2, \ldots, Y_p]^{\mathrm{T}}$ the desired solution to the collocation formulation which approximates $Y(t_m)$ at the discretized point t_m, and $\otimes$ is the tensor product (i.e. $\Delta t S$ is applied to each component of $\mathbf{Y}$). The direct solution of $\mathbf{H}(\mathbf{Y}) = \mathbf{0}$ when p is large is in general computationally inefficient as the matrix S is dense. Instead, we assume a provisional solution $\tilde{\mathbf{Y}} = [\tilde{Y}_1, \tilde{Y}_2, \ldots, \tilde{Y}_p]^{\mathrm{T}}$ is gi-

ven, and define the discretized error vector as $\boldsymbol{\delta} = \mathbf{Y} - \tilde{\mathbf{Y}} = [\delta_1, \delta_2, \ldots, \delta_p]^{\mathrm{T}}$. A simple substitution yields an error equation for $\boldsymbol{\delta}$

$$\mathbf{F}(\mathbf{y}_0 + \Delta t S \otimes \tilde{\mathbf{Y}} + \Delta t S \otimes \boldsymbol{\delta}, \tilde{\mathbf{Y}} + \boldsymbol{\delta}, \mathbf{t}) = \mathbf{0}. \tag{4}$$

Following the strategy for SDC methods for ODEs, we show how a low-order time-stepping procedure can be applied to Eq. (4) to yield an approximation $\tilde{\boldsymbol{\delta}} = [\tilde{\delta}_1, \tilde{\delta}_2, \ldots, \tilde{\delta}_p]^{\mathrm{T}}$ to the error. Similar to the ODE case in [21], we show that this is equivalent to solving

$$\mathbf{F}(\mathbf{y}_0 + \Delta t S \otimes \tilde{\mathbf{Y}} + \Delta t \tilde{S} \otimes \tilde{\boldsymbol{\delta}}, \tilde{\mathbf{Y}} + \tilde{\boldsymbol{\delta}}, \mathbf{t}) = \mathbf{0}, \tag{5}$$

where $\tilde{S}$ is a low-order, lower-triangular approximation of the spectral integration matrix S. Unfortunately, our numerical experiments demonstrate that using this "un-accelerated" SDC approach is not effective for some problems.

Instead, one can consider the SDC correction step in Eq. (5) as an "implicit" function

$$\tilde{\boldsymbol{\delta}} = \tilde{\mathbf{H}}(\tilde{\mathbf{Y}}), \tag{6}$$

where the provisional solution $\tilde{\mathbf{Y}}$ is the input variable and the output is $\tilde{\boldsymbol{\delta}}$. Here we also show that, under rather general assumptions, the Jacobian of $\tilde{\mathbf{H}}$ is closer to identity than that of $\mathbf{H}$ (see Eq. (25)), hence $\tilde{\mathbf{H}} = \mathbf{0}$ is better conditioned compared with $\mathbf{H}(\mathbf{Y}) = \mathbf{0}$. We demonstrate how available Newton–Krylov packages such as those in [23] are easily adapted and applied directly to Eq. (6) to find the zero of $\tilde{\mathbf{H}}$ (which also solves $\mathbf{H} = \mathbf{0}$). Hence the SDC procedure is analogous to a preconditioner of the collocation equation (3). This direct procedure differs from the linear implicit formulation used for ODEs in [21].

This paper is organized as follows. In Section 2, we review for completeness several fundamental numerical techniques which are combined in Section 3 to construct Krylov subspace accelerated SDC methods for DAEs of the form (1). In Section 4, a discussion of modifying the general form of the Picard equation (2) based on the decomposition of algebraic and differential variables is presented. Also, a convergence analysis of the methods and additional numerical considerations based on the index of the system are included. In Section 5, we present several preliminary numerical results to show the accuracy and efficiency of the new solvers.

2. Fundamentals

In this section, we briefly discuss the fundamentals required to construct our new DAE solvers. Specifically, Gaussian quadrature based orthogonal polynomial expansions, spectral integration, collocation methods for ODE initial value problems, deferred correction formulations, and inexact Newton methods are discussed.

2.1. Orthogonal polynomials and spectral integration

It is well known that when uniform grid points are used in polynomial interpolation, the so-called "Runge" phenomenon can be observed [4], hence higher-order (>8) uniform grid interpolation has traditionally been avoided. It is also known that least squares approximation using orthogonal polynomials is both numerically stable and accurate. For a smooth function $f(t)$ defined on $[-1, 1]$, the coefficient b_k of its Legendre polynomial expansion

$$f(t) = \sum_{k=0}^{\infty} b_k L_k(t)$$

decays exponentially fast, and this fact is widely used in spectral methods [10,15]. The coefficients of the expansion are determined by the integral

$$b_k = \left(k + \frac{1}{2}\right) \int_{-1}^{1} L_k(t) f(t)\, \mathrm{d}t$$

which can be accurately approximated using Gaussian quadrature, i.e.

$$b_k \approx \sum_{i=1}^{p} \left(k+\frac{1}{2}\right) w_i L_k(t_i) f_i,$$

where $f_i = f(t_i)$ and $\{t_i, w_i\}$ are the nodes and weights of the quadrature. Therefore, given any p function values f_i at the Gaussian nodes $\{t_1, t_2, \ldots, t_p\}$, a (numerically stable) linear mapping can be constructed which maps the function values $\mathbf{f} = [f(t_1), f(t_2), \ldots, f(t_p)]^{\mathrm{T}}$ to the Legendre coefficients $\mathbf{b} = [b_0, b_1, \ldots, b_{p-1}]^{\mathrm{T}}$. This can be represented as $\mathbf{b} = \mathscr{L}\mathbf{f}$, where $\mathscr{L}$ is the Legendre transform operator. The degree $p-1$ polynomial

$$L^p(t) = \sum_{k=0}^{p-1} b_k L_k(t)$$

is equivalent to the Lagrange interpolation polynomial which interpolates $f(t_i)$ at the Gaussian nodes.

To accurately approximate derivatives of a function, spectral methods for differential equations rely on the derivative of the orthogonal polynomial expansion defined as (see, pp. 60–65 in [10])

$$\frac{\mathrm{d}}{\mathrm{d}t} L^p(t) = \sum_{k=0}^{p-2} c_k L_k(t),$$

where

$$c_k = (2k+1) \sum_{\substack{i=k+1 \\ i+k \text{ odd}}}^{p-1} b_i,$$

or more succinctly $\mathbf{c} = \mathscr{D}\mathbf{b}$ where $\mathscr{D}$ is the spectral differentiation operator. It has been shown that $\mathscr{D}$ is numerically ill-conditioned in the sense that the norm of the linear mapping scales like $O(p^2)$ for large p [16,35]. On the other hand, spectral integration involves the integral of $L^p(t)$ defined by

$$\int_{-1}^{t} L^p(\tau)\,\mathrm{d}\tau = \sum_{k=0}^{p} a_k L_k(t),$$

where for $k \geqslant 1$

$$a_k = \frac{b_{k-1}}{2k-1} - \frac{b_{k+1}}{2k+3}, \tag{7}$$

or simply $\mathbf{a} = \mathscr{I}\mathbf{b}$ where $\mathscr{I}$ is the spectral integration operator. The norm of this linear mapping $\mathscr{I}$ (unlike the spectral differentiation operator $\mathscr{D}$) is bounded independent of p. Therefore, spectral integration is more stable than spectral differentiation and is the preferred method in our following discussions.

Instead of a linear mapping $\mathscr{I}$ from $\mathbf{b}$ to $\mathbf{a}$ defined in Eq. (7), spectral integration may also be constructed as a mapping from the function values $\mathbf{f}$ to the integrals

$$d_k = \int_{-1}^{t_k} L^p(\tau)\,\mathrm{d}\tau,$$

or simply $\mathbf{d} = \mathscr{S}\mathbf{f}$. The operator $\mathscr{S}$ is referred to as the spectral integration operator [16]. More generally, given an arbitrary interval $[a, b] \subset \mathbb{R}$ with $b - a = \Delta t$, a linear transformation can be applied to map $[a, b]$ to $[-1, 1]$. We also denote the corresponding Gaussian nodes in $[a, b]$ as $\mathbf{t} = [t_1, t_2, \ldots, t_p]^{\mathrm{T}}$, and the function values as $\mathbf{f} = [f(t_1), f(t_2), \ldots, f(t_p)]^{\mathrm{T}}$. The scaled operator $\mathscr{S}$ is then given by

$$[\Delta t S\mathbf{f}]_k = \int_{a}^{t_k} L^p(\tau)\,\mathrm{d}\tau, \tag{8}$$

where $L^p(t)$ is the polynomial interpolant of $\mathbf{f}$ and S is a matrix independent of Δt. In the following discussions, we refer to S as the *spectral integration matrix*. Note that cost of the direct evaluation of $\Delta t S\mathbf{f}$ is $O(p^2)$ operations which can be reduced to $O(p \log p)$ [2,14]. We are studying this technique for possible acceleration of our current code when p is large.

It is also possible to formulate S using Radau or Lobatto type quadrature nodes instead of Gaussian nodes and to calculate the Legendre polynomial coefficients accordingly. The Radau Ia quadrature nodes use the left end point (i.e. $t_1 = a$), the Radau IIa nodes use the right end point (i.e. $t_p = b$), and the Lobatto quadrature nodes include both end points. Also, Chebyshev polynomials and the corresponding quadrature nodes may be used instead of Legendre polynomial based nodes, which allows the fast Fourier transform (FFT) to be used for acceleration. Detailed analytical and numerical comparisons of different polynomials and nodes for our methods will be reported later. For a discussion of the choice of nodes for the original SDC methods for ODEs, the readers are referred to [24].

2.2. Picard integral equation and collocation formulations

Consider the initial value scalar ODE

$$\varphi'(t) = f(t, \varphi(t)), \quad t \in [a, b], \tag{9}$$

$$\varphi(a) = \varphi_0. \tag{10}$$

For simplicity, we assume $[a, b] = [0, \Delta t]$ in the following discussions, which corresponds to one marching step in a numerical method. The solution $\varphi(t)$ can also be expressed as the solution to the Picard integral formulation of the ODE

$$\varphi(t) = \varphi_0 + \int_0^t f(\tau, \varphi(\tau))\, \mathrm{d}\tau. \tag{11}$$

It is straightforward to apply the spectral integration operator defined above to discretize Eq. (11). Given the quadrature nodes $\mathbf{t} \in [0, \Delta t]$, then the discretized *collocation formulation* of Eq. (11) is

$$\boldsymbol{\varphi} = \boldsymbol{\varphi}_0 + \Delta t S \mathbf{f}(\mathbf{t}, \boldsymbol{\varphi}). \tag{12}$$

where $\boldsymbol{\varphi} = [\varphi_1, \varphi_2, \ldots, \varphi_p]^{\mathrm{T}}$ is the desired solution at the Gaussian nodes, $\boldsymbol{\varphi}_0 = [\varphi_0, \varphi_0, \ldots, \varphi_0]^{\mathrm{T}}$, and the function values are given by

$$\mathbf{f}(\mathbf{t}, \boldsymbol{\varphi}) = [f(t_1, \varphi_1), f(t_2, \varphi_2), \ldots, f(t_p, \varphi_p)]^{\mathrm{T}}.$$

Eq. (12) is typically nonlinear with dimension $p \times p$ as compared to the 1×1 system one encounters from using backward Euler (or BDF) methods. For N dimensional vector ODEs, the number of unknowns becomes pN as compared to N in BDF methods. It is mainly for this reason that higher-order collocation methods are rarely used for ODEs.

2.3. Error equations and spectral deferred corrections

In deferred and defect correction methods first introduced by Pereyra and Zadunaisky [26,36,37], the basic strategy for computing a higher-order accurate solution is to iteratively use a lower-order method to solve an equation for the error or defect and hence improve an approximate solution. In this section, we follow the terminology in [13], and review how spectral deferred corrections can be applied to the ODE initial value problem (9), (10). The details outlined in this section will be extended to DAEs in Section 3.

As with classical deferred and defect correction methods, a single time step of an SDC method begins by first dividing the time step $[0, \Delta t]$ into a set of intermediate sub-steps defined by the points $\mathbf{t} = [t_1, t_2, \ldots, t_p]^{\mathrm{T}}$ with $0 \leqslant t_1 < \cdots < t_p \leqslant \Delta t$ (in SDC methods, $\mathbf{t}$ corresponds to the quadrature nodes of Gaussian type). Next, a provisional approximation $\tilde{\boldsymbol{\varphi}} = [\tilde{\varphi}_1, \tilde{\varphi}_2, \ldots, \tilde{\varphi}_p]^{\mathrm{T}}$ is computed at the intermediate points using a standard numerical method. Applying standard approximation or interpolation theory, the continuous counterpart of $\tilde{\boldsymbol{\varphi}}$ (for example, the Legendre polynomial expansion) is then constructed and is denoted by $\tilde{\varphi}(t)$. Finally, an equation for the error $\delta(t) = \varphi(t) - \tilde{\varphi}(t)$ is derived which is then approximated with a time marching scheme and used to improve the original solution $\tilde{\boldsymbol{\varphi}}$.

In the original deferred and defect correction methods [26,36,37], the correction equation is cast as an ODE which explicitly includes the derivative (or higher derivatives) of $\tilde{\varphi}(t)$. On the other hand, SDC methods utilize the Picard integral equation (11) to construct a corresponding integral equation for $\delta(t)$. Specifically,

$$\delta(t) = \int_0^t [f(\tau, \tilde{\varphi}(\tau) + \delta(\tau)) - f(\tau, \tilde{\varphi}(\tau))]\, \mathrm{d}\tau + \epsilon(t), \tag{13}$$

where

$$\epsilon(t) = \varphi_0 + \int_0^t f(\tau, \tilde{\varphi}(\tau))\, \mathrm{d}\tau - \tilde{\varphi}(t). \tag{14}$$

In SDC methods, after computing approximations $[\epsilon_1, \epsilon_2, \ldots, \epsilon_p]$ using spectral integration (see Section 2.1), a lower-order method is applied to approximate $\delta(t)$ in Eq. (13). For example, the backward Euler type method

$$\tilde{\delta}_{m+1} = \tilde{\delta}_m + \Delta t_m [f(t_{m+1}, \tilde{\varphi}_{m+1} + \tilde{\delta}_{m+1}) - f(t_{m+1}, \tilde{\varphi}_{m+1})] + \epsilon_{m+1} - \epsilon_m. \tag{15}$$

The solution $\tilde{\boldsymbol{\delta}} = [\tilde{\delta}_1, \tilde{\delta}_2, \cdots, \tilde{\delta}_p]^{\mathrm{T}}$ which approximates $\delta(t)$ is then added to the provisional solution $\tilde{\boldsymbol{\varphi}}$ to form a more accurate provisional solution. Spectral integration is then again applied using Eq. (14) to accurately compute the new $\epsilon(t)$, and the iteration procedure continues until the residual is smaller than a prescribed tolerance or a maximum prescribed number of iterations is reached. Note that the computational complexity of solving the implicit equation (15) is the same as that of the simple first-order backward Euler method.

It is easy to see that if the SDC procedure converges to a numerical solution $\boldsymbol{\varphi}$, then this solution satisfies the collocation formulation equation given in Eq. (12). Therefore, the SDC procedure can also be thought of as an iterative method for solving Eq. (12). In [21], a Krylov subspace based procedure for accelerating the convergence of the SDC iterations to the collocation formulation solution is presented, and the analogous procedure for DAEs is discussed in the remainder of this paper.

2.4. Newton–Krylov methods

Consider a general algebraic system $M(x) = 0$ with N equations and unknowns, and suppose an approximate solution x_0 is known. Newton's method can be used to iteratively compute a sequence of quadratically convergent approximations (assuming the Jacobian matrix J_M is nonsingular at the solution)

$$x_{n+1} = x_n - \delta x,$$

where δx is the solution of the linear equation

$$J_M(x_n)\delta x = b$$

with $b = M(x_n)$ and $J_M(x_n)$ the Jacobian matrix of $M(x)$ at x_n. When the matrix J_M is dense, computing the solution of this linear equation with Gaussian elimination requires $\mathrm{O}(N^3)$ operations.

However, for many special matrices, the amount of work required to find the solution can be greatly reduced. Consider the case

$$J_M(x_n) = \pm I - C,$$

where most of the eigenvalues of C are clustered close to 0. Because of the rapid decay of most eigenmodes in $C^q b$, a more efficient approach than Gaussian elimination is to iteratively search for the optimal solution in the Krylov subspace defined by

$$K_q(J_M, b) = \{b, Cb, C^2 b, \ldots, C^q b\}.$$

The iterations in Newton's method and the Krylov subspace methods can then be intertwined, and the resulting methods are usually referred to as the Newton–Krylov methods. The readers are referred to [22,23,33] for detailed discussions.

In general, an efficient numerical implementation of a Newton–Krylov method depends on two things:

(a) A formulation of the problem $M(x) = 0$ such that J_M is close to the identity matrix $\pm I$.
(b) An efficient procedure for computing the matrix vector product Cb (or equivalently $J_M b$).

For (a), one common technique to improve the convergence of the method is to apply a "preconditioner" to the original system. Traditionally, such preconditioners are chosen as sparse matrices close to J_M^{-1} [11]. Dense integral operators have also been used as preconditioners (see e.g. [27]), which are efficiently applied to an arbitrary vector using fast convolution algorithms such as the fast multipole method [17]. One of the main themes of this paper is that the SDC procedure (for both ODEs and DAEs), in which lower-order methods are used to produce a higher-order solution, is equivalent to using a lower-order approximation process of a particular equation as a preconditioner for a higher-order method. This presents a different way to understand the deferred correction methods discussed in Section 2.3, and allows one to easily adapt existing Newton–Krylov methods to accelerate the convergence of the method.

In regards to point (b), in this paper the operator M is the collocation formulation of the DAE discussed in next section, and the Jacobian of this operator is not always easy to derive. As this is often the case with large systems, a general forward difference approximation technique is adapted in most Newton–Krylov solvers where for any vector v, $J_M(x)v$ is approximated by

$$D_h M(x : v) = (M(x + hv) - M(x))/h$$

for some properly chosen parameter h (h may be complex). This difference approximation technique as well as the choice of h has been carefully studied previously and the readers are referred to [22] for details.

3. Krylov deferred corrections for differential algebraic equations

In this section, the various building blocks of the Krylov subspace accelerated deferred correction methods for ODEs outlined in the previous section are adapted to construct arbitrary order and stable Krylov deferred correction (KDC) methods for DAEs. Specifically, the Picard integral formulation and collocation discretization, error equation and deferred correction method, and the use of Newton–Krylov acceleration are discussed in order.

3.1. Picard equation and collocation formulation for DAEs

In the original SDC method for ODEs, the Picard integral equation (11) is used to form the similar integral equation (13) for the correction. This formulation has the benefit that it avoids the ill-conditioned differentiation operator as discussed in Section 2.1. However, it is not immediately clear how to generalize this technique to DAEs since a Picard equation form of the DAE is not readily available.

Toward this end, instead of solving for $y(t)$ in Eq. (1) directly, our new formulation uses $y'(t)$ as the unknown variable, which will be denoted by $Y(t)$ in the following discussions. Expressing $y(t)$ as the integral of $Y(t)$, the DAE system $F(y(t), y'(t), t) = 0$ becomes

$$F\left(y_0 + \int_0^t Y(\tau)\,\mathrm{d}\tau, Y(t), t\right) = 0. \tag{16}$$

Although this formulation is quite general, we discuss in the following section cases when this formulation should be modified to separate differential and algebraic variables in order to increase the efficiency of the numerical method.

As in Section 2.2, this Picard type equation can be directly discretized using the spectral integration matrix S to yield

$$\mathbf{F}(\mathbf{y}_0 + \Delta t S \otimes \mathbf{Y}, \mathbf{Y}, \mathbf{t}) = \mathbf{0}, \tag{17}$$

where $\mathbf{Y} = [Y_1, Y_2, \ldots, Y_p]^{\mathrm{T}}$ is the desired solution which approximates $Y(t) = y'(t)$ at the Gaussian nodes. The solution $\mathbf{y} = [y_1, y_2, \ldots, y_p]^{\mathrm{T}}$ is then recovered using the spectral integration matrix

$$\mathbf{y} = \mathbf{y}_0 + \Delta t S \otimes \mathbf{Y}.$$

Eq. (17) is the analog of the collocation formulation for ODEs given in Eq. (12), and in the following discussions we write Eq. (17) symbolically as $\mathbf{H}(\mathbf{Y}) = \mathbf{0}$.

As for the initial values, note that when Gaussian or Radau IIa nodes are used, only $y(0) = y_0$ is required in the collocation formulation. However, when Radau Ia or Lobatto nodes are applied, $Y(0)$ is also required when calculating $\Delta t S \otimes \mathbf{Y}$, which can be derived by solving the equation $F(y(0), Y(0), 0) = 0$.

3.2. Error equations and modified spectral deferred corrections

Following the procedure for SDC method for ODEs, given an approximation or provisional solution to the DAE $\tilde{\mathbf{Y}} = [\tilde{Y}_1, \tilde{Y}_2, \cdots, \tilde{Y}_p]^{\mathrm{T}}$ at the nodes $\mathbf{t}$, one can define an equation for the error $\delta(t) = Y(t) - \tilde{Y}(t)$ by

$$F\left(y_0 + \int_0^t \left(\tilde{Y}(\tau) + \delta(\tau)\right) \mathrm{d}\tau, \tilde{Y}(t) + \delta(t), t\right) = 0, \tag{18}$$

where $\tilde{Y}(t)$ is the polynomial interpolation of $\tilde{\mathbf{Y}}$.

As in the original SDC, we wish to use a low-order method to approximate the error equation (18) and improve the provisional solution $\tilde{Y}(t)$. Note that Eq. (18) gives the identity

$$F\left(y_0 + \int_0^{t_{m+1}} \tilde{Y}(\tau)\, \mathrm{d}\tau + \left(\int_0^{t_m} + \int_{t_m}^{t_{m+1}}\right)\delta(\tau)\, \mathrm{d}\tau, \tilde{Y}(t_{m+1}) + \delta(t_{m+1}), t_{m+1}\right) = 0. \tag{19}$$

A simple time-marching discretization of this equation similar to the explicit (forward) Euler method for ODEs gives a low-order solution $\tilde{\boldsymbol{\delta}} = [\tilde{\delta}_1, \tilde{\delta}_2, \ldots, \tilde{\delta}_p]^{\mathrm{T}}$ by solving

$$F\left(y_0 + [\Delta t S \otimes \tilde{\mathbf{Y}}]_{m+1} + \sum_{l=1}^{m+1} \Delta t_l \tilde{\delta}_{l-1}, \tilde{Y}_{m+1} + \tilde{\delta}_{m+1}, t_{m+1}\right) = 0, \tag{20}$$

where $\Delta t_{l+1} = t_{l+1} - t_l$, t_0 and δ_0 are set to 0. Note that this update formula is in general implicit since no explicit formula for $\tilde{\delta}_{m+1}$ exists. Similarly, a time-marching scheme based on backward Euler method analogous to Eq. (15) is given by

$$F\left(y_0 + [\Delta t S \otimes \tilde{\mathbf{Y}}]_{m+1} + \sum_{l=1}^{m+1} \Delta t_l \tilde{\delta}_l, \tilde{Y}_{m+1} + \tilde{\delta}_{m+1}, t_{m+1}\right) = 0. \tag{21}$$

These two methods differ only in the way how the time integral of $\delta(t)$ is approximated. Eq. (20) is equivalent to the rectangle rule using the left endpoint while Eq. (21) is the rectangle rule using the right endpoint. A discussion of the advantages of one over the other is presented in the next section.

The two time-stepping methods can be written in matrix form as

$$\mathbf{F}(\mathbf{y}_0 + \Delta t S \otimes \tilde{\mathbf{Y}} + \Delta t \tilde{S} \otimes \tilde{\boldsymbol{\delta}}, \tilde{\mathbf{Y}} + \tilde{\boldsymbol{\delta}}, \mathbf{t}) = \mathbf{0}, \tag{22}$$

where $\Delta t \tilde{S}$ is the lower triangular representation of the rectangle rule approximation of the spectral integration operator S. Specifically, for Eq. (20)

$$\Delta t \tilde{S} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ \Delta t_1 & 0 & \cdots & 0 & 0 \\ \Delta t_1 & \Delta t_2 & \cdots & 0 & 0 \\ \cdot & \cdot & \cdots & 0 & 0 \\ \Delta t_1 & \Delta t_2 & \cdots & \Delta t_{p-1} & 0 \end{bmatrix} \tag{23}$$

and for Eq. (21)

$$\Delta t \tilde{S} = \begin{bmatrix} \Delta t_1 & 0 & \cdots & 0 & 0 \\ \Delta t_1 & \Delta t_2 & \cdots & 0 & 0 \\ \cdot & \cdot & \cdots & 0 & 0 \\ \Delta t_1 & \Delta t_2 & \cdots & \Delta t_{p-2} & 0 \\ \Delta t_1 & \Delta t_2 & \cdots & \Delta t_{p-2} & \Delta t_{p-1} \end{bmatrix}. \tag{24}$$

As in the ODE case, the order of accuracy of $\tilde{\mathbf{Y}}$ is increased by iteratively using Eq. (22) to approximate the error, and if $\tilde{\mathbf{Y}}$ converges, it converges to the solution of the collocation equation (17). Therefore, the accuracy of the current methods can be broken down into two separate convergence issues: the convergence of the deferred correction iterations to the collocation solution (as $\tilde{\boldsymbol{\delta}} \to \mathbf{0}$), and the convergence of the solution of the collocation equation to the exact solution (i.e. as $\Delta t \to 0$ or $p \to \infty$). The focus of this paper concerns the acceleration of the first iteration, and we assume the problem is resolved to desired error tolerance by the collocation formulation. Readers are referred to Section 4.1 for further order and accuracy discussions of the collocation methods.

3.3. Acceleration using Newton–Krylov methods

In [21], it is observed that the original SDC method for ODE initial value problems can be considered as a Neumann series expansion for solving the preconditioned system (12), where the preconditioner is the spectral deferred correction process. Also in that paper, a linear implicit method is applied to general nonlinear ODE problems. In this section, we generalize this idea to DAEs, and explain how Newton–Krylov methods can be directly applied to the preconditioned system instead of using the linear implicit formulation.

From the discussion in previous section, a low-order method can be considered as a tool for deriving the approximate error $\tilde{\boldsymbol{\delta}}$ as a function of the given provisional solution $\tilde{\mathbf{Y}}$ from the "implicit" function in (22). In the following, we use

$$\tilde{\boldsymbol{\delta}} = \tilde{\mathbf{H}}(\tilde{\mathbf{Y}})$$

to represent the explicit form of this implicit function. As a reminder, evaluation of $\tilde{\mathbf{H}}$ is nothing more than one iteration of the deferred correction procedure. Notice that when $\tilde{\boldsymbol{\delta}} = \mathbf{0}$, the solution to Eq. (22) is identical to Eq. (17). Therefore, solving the collocation formulation $\mathbf{H}(\mathbf{Y}) = \mathbf{0}$ is equivalent to finding the zero of the implicit equation

$$\tilde{\mathbf{H}}(\mathbf{Y}) = \mathbf{0}.$$

Because the low-order method solves an approximation of the collocation formulation, it is not surprising that the explicit function $\tilde{\mathbf{H}}(\mathbf{Y}) = \mathbf{0}$ is better conditioned compared with the original collocation formulation in (17) as shown by the following analysis. Applying the implicit function theorem, the Jacobian matrix $J_{\tilde{\mathbf{H}}}$ of $\tilde{\mathbf{H}}$ is given by

$$J_{\tilde{\mathbf{H}}} = \frac{\partial \tilde{\boldsymbol{\delta}}}{\partial \mathbf{Y}} = -\left(\frac{\partial \mathbf{F}}{\partial \mathbf{Y}} + \frac{\partial \mathbf{F}}{\partial \mathbf{y}}\Delta t\tilde{S}\right)^{-1}\left(\frac{\partial \mathbf{F}}{\partial \mathbf{Y}} + \frac{\partial \mathbf{F}}{\partial \mathbf{y}}\Delta t S\right) = -I + \left(\frac{\partial \mathbf{F}}{\partial \mathbf{Y}} + \frac{\partial \mathbf{F}}{\partial \mathbf{y}}\Delta t\tilde{S}\right)^{-1}\left(\frac{\partial \mathbf{F}}{\partial \mathbf{y}}\Delta t(\tilde{S} - S)\right). \tag{25}$$

When $\frac{\partial \mathbf{F}}{\partial \dot{\mathbf{Y}}}$ is non-singular (e.g. $\frac{\partial \mathbf{F}}{\partial \dot{\mathbf{Y}}} = I$ for ODE systems), since $\tilde{S}$ is an approximation of S, when Δt is small, $J_{\tilde{\mathbf{H}}}$ is close to $-I$. This was the first requirement for the efficient application of Newton–Krylov methods discussed in Section 2.4. For comparison, the Jacobian matrix of $\mathbf{H} = \mathbf{0}$ is given by

$$J_{\mathbf{H}} = \frac{\partial \mathbf{H}}{\partial \mathbf{Y}} = \left(\frac{\partial \mathbf{F}}{\partial \mathbf{Y}} + \frac{\partial \mathbf{F}}{\partial \mathbf{y}}\Delta t S\right).$$

For higher-index DAE problems in which $\frac{\partial \mathbf{F}}{\partial \dot{\mathbf{Y}}}$ is singular, the KDC techniques can be applied to some of the unknowns as will be discussed in next section. Using the implicit function theorem, it can be shown for this case as well that the resulting Jacobian matrix is again closer to the identity matrix compared with $J_{\mathbf{H}}$. Also, when any eigenvalue λ of the matrix

$$C = J_{\tilde{\mathbf{H}}} + I = \left(\frac{\partial \mathbf{F}}{\partial \mathbf{Y}} + \frac{\partial \mathbf{F}}{\partial \mathbf{y}}\Delta t\tilde{S}\right)^{-1}\left(\frac{\partial \mathbf{F}}{\partial \mathbf{y}}\Delta t(\tilde{S} - S)\right) \tag{26}$$

satisfies $\|\lambda\| \geqslant 1$ (this may happen to higher-index DAE systems independent of the choice of Δt), the un-accelerated SDC methods (consider the Neumann series for linear problems) become divergent, on the other hand, the Newton–Krylov methods converge efficiently as long as the number of such eigenvalues is small.

Finally, we recall that the second requirement for the efficient application of Newton–Krylov methods discussed in Section 2.4 is an efficient procedure for computing the function $\tilde{\mathbf{H}}$. As noted earlier, this is simply a deferred correction iteration described succinctly in Eq. (22).

4. Convergence, index, and implementations

Unlike the construction of numerical techniques for ODE initial value problems, which is considered a mature subject in many respects, the efficient and accurate solutions of DAE systems are more challenging in both theory and implementation, especially for higher-index systems. The purpose of this section is to present preliminary results on the analytical and numerical properties of the KDC methods, including the convergence analysis and essential techniques describing how a DAE system, depending on its index, can be discretized so that the new KDC methods are efficient and accurate.

4.1. Convergence analysis

There are in fact two distinct notions of convergence which pertain to KDC methods. The first is the convergence of the Krylov accelerated deferred correction steps to the solution of the collocation equation (17). The second is the convergence of the solution of the collocation discretization (17) to the solution of the DAE given by the Picard equation (16). This latter convergence is usually thought of as occurring as the time-step (i.e. the integration interval in the Picard equation) goes to zero. It is important to point out that the first type of convergence mainly affects the efficiency but not the accuracy of the scheme, as long as the iterations eventually converge.

Both the convergence of Newton–Krylov methods and the convergence of collocation schemes for DAEs have been extensively studied previously, and we summarize several results from existing literature in the following. These results, when coupled together, show the local convergence of the KDC methods.

Convergence of Newton–Krylov methods. It is well known that under rather standard assumptions, the original Newton's method converges quadratically when the initial guess is "close" to the real solution. However, for Newton–Krylov methods (inexact Newton methods), each linear correction equation is only solved approximately, and the *local* convergence order is no longer quadratic (but still convergent). It can be shown that super-linear local convergence can be obtained for specially chosen parameters in the Newton–Krylov schemes (see Theorem 6.1.2 in [23]). For arbitrary initial approximations, continuation/homotopy methods are necessary to accomplish *global* convergence. Interested readers are referred to [22,23] for further discussions.

It is important to note that the efficiency of Newton–Krylov methods in general and KDC methods in particular can be significantly affected by the form of the preconditioner. In the present context, the splitting of algebraic and differential variables in Section 4.2 and the use of semi-implicit schemes as discussed in Section 4.4 can have a large impact on the convergence of the Krylov deferred correction iterations. In general, the effective choice of preconditioner requires deep insight into the structure of the DAE system and is hence problem dependent. Interested readers are referred to [6] and the references therein for a discussion of general preconditioning techniques.

Convergence and orders of collocation formulations. In [18,19], it was shown that collocation formulations are equivalent to certain Runge–Kutta methods (see p. 27 in [19]), and the convergence and order of accuracy are determined by the collocation points and properties of the DAE problem to be solved, in particular the index (see p. 18 in [18]). As a general guidance, the collocation formulation for ODE systems (index 0 DAE systems) using p Gaussian points is order $2p$, A-stable and L-stable, symplectic and symmetric, and hence the "optimal" choice. For higher-index DAE systems, order reduction can be observed, the extent of which depends on the index and the type of nodes. Radau IIa nodes are in general the best choice (and are used here) in that the reduced order is higher than other choices. In particular, for Radau IIa nodes, one should expect for index one problems order $2p - 1$ for both differential and algebraic variables , but for index 2 problems, order $2p - 1$ for differential variables and order p for algebraic variables. Interested readers are referred to [18] for further details on the convergence (as well as divergence) of collocation formulations for different DAE problems including partial results for index 3.

4.2. Differential and algebraic variables

One implicit assumption of this paper is that the error equation (20) or (21) for $\tilde{\boldsymbol{\delta}}$ is more efficient to solve compared with a direct solution of the collocation formulation (17) or the original DAE system (1). Although this assumption is generally true, due to the existence of algebraic equations, it may not be the case for at least some of the unknowns in DAE systems. Therefore, in many instances one can treat algebraic variables (whose derivative never appears) and differential ones differently in the discretization and yield a more efficient overall method. An immediate consequence of such modification is that the number of variables in the KDC system is reduced. Our preliminary numerical experiments show that this leads to a more efficient method compared with the general formulation introduced in Section 3 where KDC strategies are applied to both variables, however the accuracy of the solutions are similar for all problems we tested. Rigorous analyses for both formulations as well as implementation details are currently being pursued. In the following, we present some examples illustrating the issues regarding the modified formulations.

Purely algebraic equation systems. For the purpose of intuitive insight, consider first the purely algebraic system

$$F(y,t)=0.$$

As the derivative never appears in this system (hence y is referred to as an algebraic variable), introducing the error equation and spectral integration can neither improve the efficiency nor accuracy. In fact, as spectral integration couples solutions at different node points, simply using Newton's method for the algebraic system at required nodes will in this case be more efficient than using the KDC approach.

Index 1 problems. Next consider the index 1 DAE problem

$$\begin{cases} y' = f(y,z), \\ 0 = g(y,z), \end{cases}$$

where f and g are sufficiently differentiable and the inverse of $\frac{\partial g}{\partial z}$ is bounded. In this case, as the derivative of z never appears in the system (hence z is called the algebraic variable), it is more efficient to apply spectral integration only to the differential variable y by making $\{Y,z\}$ the unknowns. The corresponding discretized collocation formulation becomes

$$\begin{cases} \mathbf{Y} = \mathbf{f}(\mathbf{y}_0 + \Delta t S \otimes \mathbf{Y}, \mathbf{z}), \\ \mathbf{0} = \mathbf{g}(\mathbf{y}_0 + \Delta t S \otimes \mathbf{Y}, \mathbf{z}), \end{cases}$$

and the error equation is given by

$$\begin{cases} \tilde{\mathbf{Y}} + \tilde{\boldsymbol{\delta}} = \mathbf{f}(\mathbf{y}_0 + \Delta t S \otimes \tilde{\mathbf{Y}} + \Delta t \tilde{S} \otimes \tilde{\boldsymbol{\delta}}, \mathbf{z}), \\ \mathbf{0} = \mathbf{g}(\mathbf{y}_0 + \Delta t S \otimes \tilde{\mathbf{Y}} + \Delta t \tilde{S} \otimes \tilde{\boldsymbol{\delta}}, \mathbf{z}). \end{cases}$$

The Newton–Krylov procedure is then only applied to $\tilde{\boldsymbol{\delta}}$ corresponding to the Y-component, and the preconditioned system is denoted by $\tilde{\boldsymbol{\delta}} = \tilde{\mathbf{H}}(\tilde{\mathbf{Y}})$. An immediate advantage of this modified formulation is that the number of unknowns is reduced in $\tilde{\mathbf{H}}(\tilde{\mathbf{Y}}) = \mathbf{0}$ which reduces the cost of the Newton–Krylov solver.

Higher-index problems. For higher-index problems, we recommend that the differential variables and the algebraic ones be treated differently. As a general rule, the spectral integration technique and the Krylov subspace methods should only be applied to the differential variables. Finding the optimal discretization for a given DAE may require careful analysis of the structure of the DAE system, and hence is problem dependent. The use of KDC methods is not a substitute for such analysis, nevertheless, our experience is that the KDC methods do converge (although perhaps slowly) for a wide class of DAEs.

4.3. Scaled Newton's method and the index

When KDC methods are applied to DAE systems, each marching step (from Gaussian node t_m to t_{m+1}) in one SDC sweep requires the solution of a (generally) nonlinear system. When Newton's method is used, unlike

the ODE case, further numerical issues arise for higher-index DAE systems, including finding the correct scaling when the Jacobian matrix is poorly scaled, e.g.

$$J = \begin{bmatrix} O(1) & O(1) \\ O(h) & O(h) \end{bmatrix}.$$

Also, multiple iteration control strategies for different variables of the solution may be required since the error and convergence behavior for different variables may vary widely. These techniques are generally problem (index) dependent, and our experience has shown that these techniques are necessary to achieve rapid converge of the KDC iterations as well. As these techniques are already well-documented in the context of the BDF and Runge–Kutta based methods (see, e.g. Section 7 in [18] and Section 5.2 in [8]), we neglect the details in this paper.

4.4. Semi-implicit KDC schemes as preconditioners

Comparing Eqs. (20) and (21), one may wonder if and when the explicit Euler method is useful since both forms require the solution of a nonlinear system for the unknowns. There are many cases that applying an explicit scheme to some of the equations in the DAE system may improve the efficiency of the numerical method. One such case is the well-known non-stiff ODE systems. A more relevant example is the index 1 DAE system discussed in Section 4.2. When an explicit method can be applied to the first equation, z then becomes the only unknown at next time step, and hence Newton's method becomes more efficient. It is also possible to use a "semi-implicit" discretization analogous to the ODE case [28] where some terms in the DAE system are treated with using Eq. (20) and some with (21). In the KDC methods framework, these techniques can be considered as different preconditioning strategies, and hence may improve the efficiency of the method with no loss in accuracy. We are currently investigating these strategies in the broader context of partial differential algebraic systems.

5. Preliminary numerical results

In this section, we show some preliminary numerical results for linear and nonlinear DAE problems with different index. The new methods are currently implemented in matlab, and Radau IIa nodes are used in spectral integration. Whenever possible, the Picard integral discretization is applied only to differential variables as discussed in Section 4.2.

5.1. Linear DAE system

For the first example, we consider a simple linear DAE system of index 2 (see p. 267 in [3] where α is set to 10)

$$\begin{pmatrix} y_1'(t) \\ y_2'(t) \\ 0 \end{pmatrix} = \begin{pmatrix} 10 - \frac{1}{2-t} & 0 & 10(2-t) \\ \frac{9}{2-t} & -1 & 9 \\ t+2 & t^2-4 & 0 \end{pmatrix} \begin{pmatrix} y_1(t) \\ y_2(t) \\ y_3(t) \end{pmatrix} + \begin{pmatrix} \frac{3-t}{2-t} e^t \\ 2e^t \\ e^t(2-t-t^2) \end{pmatrix} \tag{27}$$

whose analytical solution is given by $\mathbf{y}(t) = (e^t, e^t, -e^t/(2-t))$.

We first demonstrate the order of accuracy of the KDC methods by computing the solution from $t_0 = 0$ to $t_{\text{final}} = 1$ using Radau IIa points with $p = 3$, 4, and 5. The full GMRES orthogonalization procedure is applied to the resulting preconditioned linear collocation formulations. The convergence of the error at $t = 1$ versus the time step for the KDC methods is shown in Fig. 1 for $y_1(t)$ (left) and $y_3(t)$ (right), respectively. The data in Fig. 1 confirm that the KDC method using p Radau IIa nodes is converging with approximate order $2p - 1$ for the differential variables (y_1 and y_2), but only order p for the algebraic variable y_3 (see discussion in Section 4.1).

As a comparison with BDF based methods of orders 2, 3, and 4 (see [3], p. 268, Fig. 10.2) where a step-size smaller than 10^{-3} is required for 10 digits of accuracy in y_1, the new KDC methods using $p = 5$ only requires a step-size of approximately $10^{-0.9}$ for 14 digits accuracy, with 440 function evaluations.

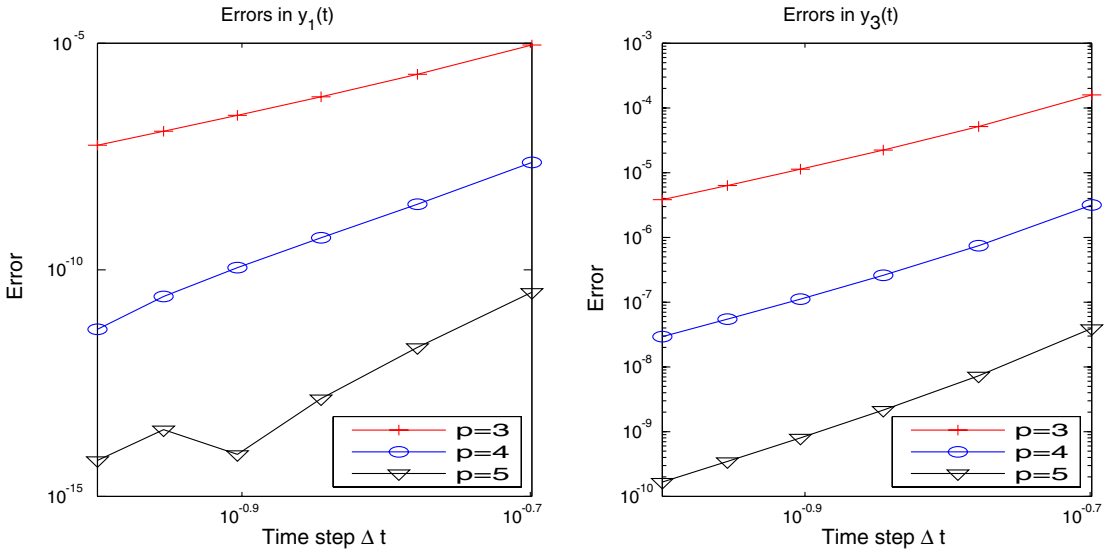

Fig. 1. Convergence test of KDC methods with full GMRES and Radau IIa points.

When the GMRES method is applied to the $N \times N$ linear system $Ax = b$, the memory required increases linearly with the iteration number k, and the number of multiplications scales like $\frac{1}{2}k^2N$. Hence, for large k, the GMRES procedure becomes very expensive and requires excessive memory storage. For these reasons, instead of a full orthogonalization procedure, GMRES can be restarted every k_0 steps where $k_0 < N$ is some fixed integer parameter. The restarted version is denoted here as GMRES(k_0). We next study the effect of using GMRES(k_0) on the efficiency of KDC methods.

For the linear system above, we use 16 Radau IIa nodes and set $\Delta t = t_{\text{final}} = 1$. The total number of unknowns is $N = 16 \times 3$, and hence $k_0 = 48$ is equivalent to the full GMRES procedure. In Fig. 2, we study the convergence of the KDC method using different k_0 applied to the preconditioned collocation formulation of Eq. (27). Numerical results show that keeping more data in storage (larger k_0) gives better convergence results. However, $k_0 = 20$ results in similar convergence to $k_0 = 48$. In the figure, we plot how the residual

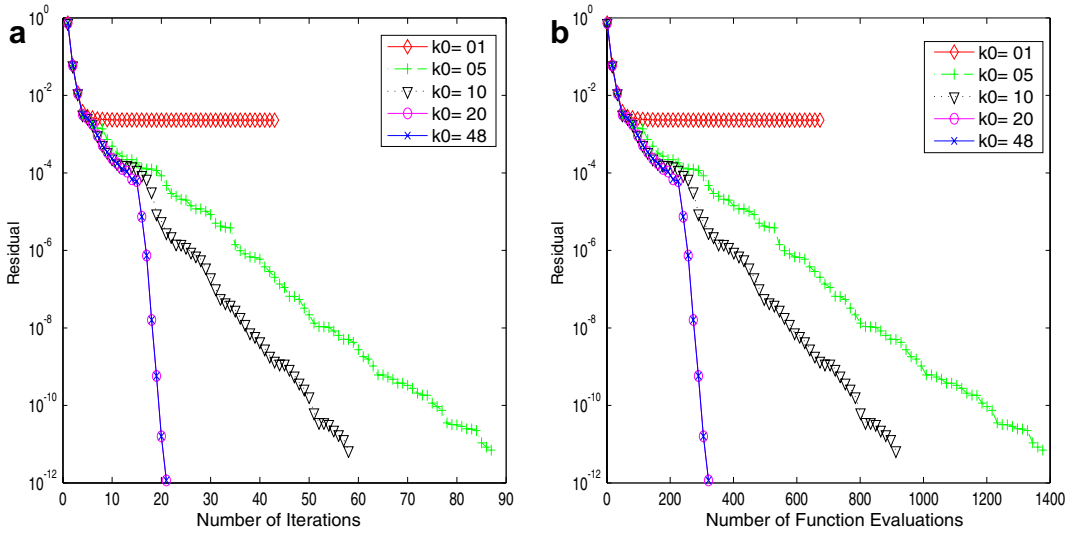

Fig. 2. Convergence of GMRES(k_0) for different k_0 for for a KDC method applied to the linear system as a function of (a) GMRES steps and (b) number of function evaluations.

decays, as the true error is not readily available from the GMRES subroutine. For this problem, 16 points resolve the solution to 14 significant digits so up to a constant factor, the residual is almost identical to the true error. Also, because Eq. (27) is linear, each GMRES iteration (SDC sweep) needs exactly $p = 16$ function evaluations. This explains why the two plots are identical (except for a factor *16* in x-axis).

Finally for this example, note that the KDC method using 9 Radau IIa nodes allows a step-size of 1 (one time step from 0 to 1) for 12 digits of accuracy in y_1 and y_2, with a total number of 162 function evaluations (compared to more than 1000 for the BDF methods in [3]).

5.2. Transistor amplifier problem

In our second example, we consider the transistor amplifier problem in [1] which is a stiff DAE system of index 1 consisting of eight equations given by

$$M\frac{\mathrm{d}y}{\mathrm{d}t} = f(y), \quad y(0) = y_0, \quad y'(0) = y'_0,$$

with $y \in \mathbb{R}^8$ and $0 \leqslant t \leqslant 0.2$. The matrix M is of rank 5 and is given by

$$M = \begin{pmatrix} -C_1 & C_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ C_1 & -C_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -C_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -C_3 & C_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & C_3 & -C_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -C_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -C_5 & C_5 \\ 0 & 0 & 0 & 0 & 0 & 0 & C_5 & -C_5 \end{pmatrix}.$$

The function f is defined as

$$f(y) = \begin{pmatrix} -\frac{U_e(t)}{R_0} + \frac{y_1}{R_0} \\ -\frac{U_b}{R_2} + y_2\left(\frac{1}{R_1} + \frac{1}{R_2}\right) - (\alpha - 1)g(y_2 - y_3) \\ -g(y_2 - y_3) + \frac{y_3}{R_3} \\ -\frac{U_b}{R_4} + \frac{y_4}{R_4} + \alpha g(y_2 - y_3) \\ -\frac{U_b}{R_6} + y_5\left(\frac{1}{R_5} + \frac{1}{R_6}\right) - (\alpha - 1)g(y_5 - y_6) \\ -g(y_5 - y_6) + \frac{y_6}{R_7} \\ -\frac{U_b}{R_8} + \frac{y_7}{R_8} + \alpha g(y_5 - y_6) \\ \frac{y_8}{R_9} \end{pmatrix},$$

where g and U_e are auxiliary functions given by

$$g(x) = \beta(\mathrm{e}^{\frac{x}{U_F}} - 1) \quad \text{and} \quad U_e(t) = 0.1\sin(200\pi t).$$

As in Fig. 2 for the linear case, we first consider the effect of using different GMRES(k_0). For this nonlinear system, we use the Newton-GMRES method in which GMRES(k_0) is applied in each Newton iteration to reduce the residual by a factor of η (see Section 2.4). In the experiment, 16 Radau IIa nodes are used, the step-size is 0.0025 which resolves the solution to eight digits of accuracy, and η is set to 0.3. In (a) of Fig. 3, the residual after each GMRES step (one SDC sweep) is presented for different choices of k_0 and in (b) the residual versus number of function evaluations. It can be seen that $k_0 = 10$ provides results similar to the full GMRES procedure which requires $k_0 = 128$. This is similar to the linear case shown in Fig. 2.

For general DAE problems, it is not known how to choose the "optimal" k_0 since the choice depends on the number of Gaussian type nodes and the eigenvalue distribution of the underlying problem. For the remainder

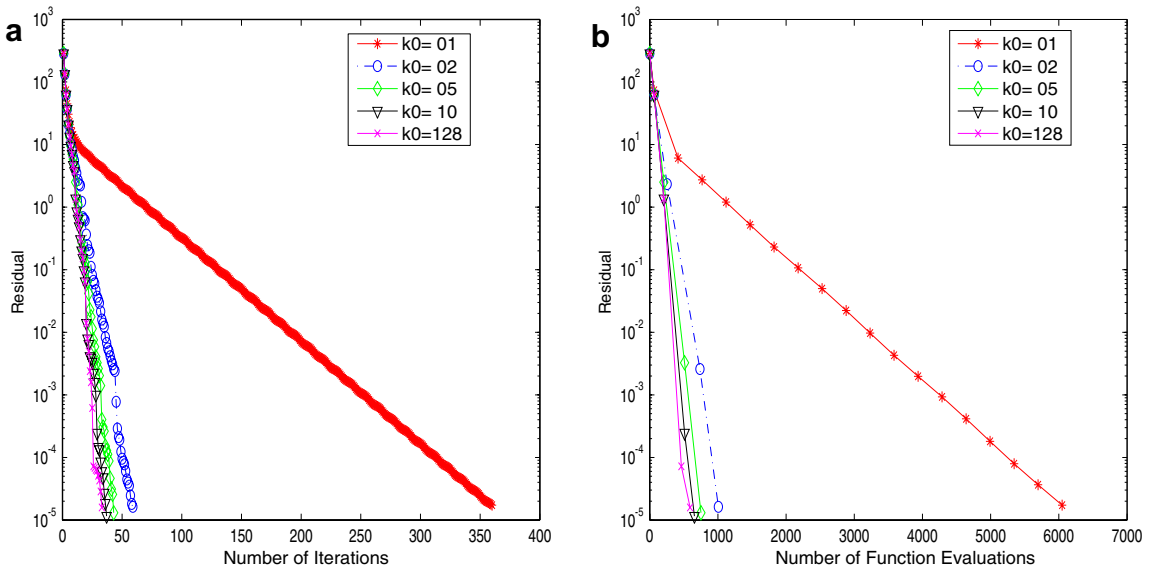

Fig. 3. Convergence of GMRES(k_0) for different k_0 for a KDC method applied to the nonlinear system as a function of (a) GMRES steps and (b) number of function evaluations.

of this paper, we use a simple scheme which selects k_0 to be the smaller of $\{c_1, p + N + c_2\}$ where p is the number of node points, N is the number of equations, c_1 is a large constant determined by the computer memory constraints and efficiency considerations, and c_2 is a small constant currently chosen as 5. However, this strategy is by no means optimal and better schemes are still being pursued (see the discussion below).

We next provide a comparison of the performance of KDC methods with the MEBDFI and RADAU packages (see [1] for discussions of the two methods). For this problem, adaptive step-size and scheme order selections are essential for optimal efficiency as demonstrated in Fig. 4, where for 11 digits of accuracy, the step-sizes used by MEBDFI vary from 10^{-4} to 10^{-14} and those by RADAU from 10^{-3} to 10^{-10}. Nevertheless,

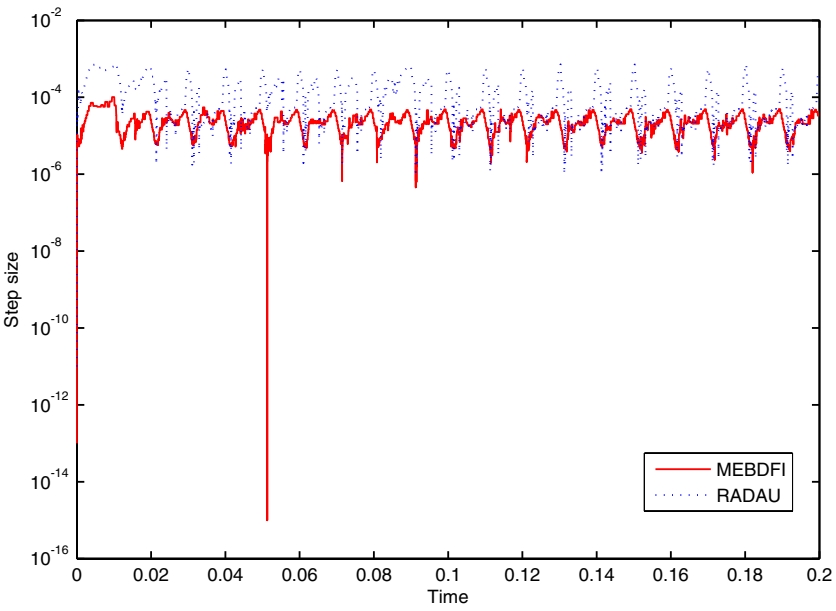

Fig. 4. Adaptive step-sizes of MEBDFI and RADAU for transistor problem (10 digits of accuracy).

we compare the performance of the adaptive MEBDFI and RADAU packages with KDC methods using uniform time steps. We solve the transistor problem from $t = 0.1$ to $t = 0.2$ to avoid the initial sharp step-size transition region. Fig. 5 shows a comparison of the RADAU and MEBDFI packages with KDC methods with $p = 5$, 10, and 20 and a range of fixed time steps.

To give an indication of the disadvantage of using a fixed time step for this example, for the numerical solution using 20 Radau IIa nodes and 200 uniform steps, we compute at each step the Legendre polynomial approximation to the solution (see Section 2.1). We set the error tolerance to 10^{-14} in the Newton–Krylov iterations and hence the solution error mainly comes from the discretization in the collocation formulation (17). In the left panel of Fig. 6, we plot the magnitude of the coefficient c_{10} for each step and in right panel c_{19}. It can be seen that for most steps, 11 terms in the expansion resolves the solution to 12 digits, however 20 Radau IIa points must be used to resolve the solution to 12 digits in all steps. This indicates that adaptive selection of the number of nodes (or alternatively the size of the time step) would significantly increase the efficiency of the KDC methods for this example.

We are currently studying the issue of adaptively choosing the step-size and scheme order for KDC methods. This effort must also consider other parameters related to the Newton–Krylov methods (e.g. k_0 and η above). Further possibilities include increasing the number of node points (reflecting the degree of the approximating polynomial) during the Newton–Krylov procedure, and even using different numbers of nodes for different variables of the solution vector.

5.3. Andrews' squeezing mechanism

Andrews' squeezing mechanism describes the motion of 7 rigid bodies connected by joints without friction, which is modeled by a non-stiff second order DAE system of index 3, consisting of 21 differential and 6 algebraic equations, as given by

$$K\frac{\mathrm{d}y}{\mathrm{d}t} = \Phi(y), \quad y(0) = y_0, \quad y'(0) = y'_0,$$

where

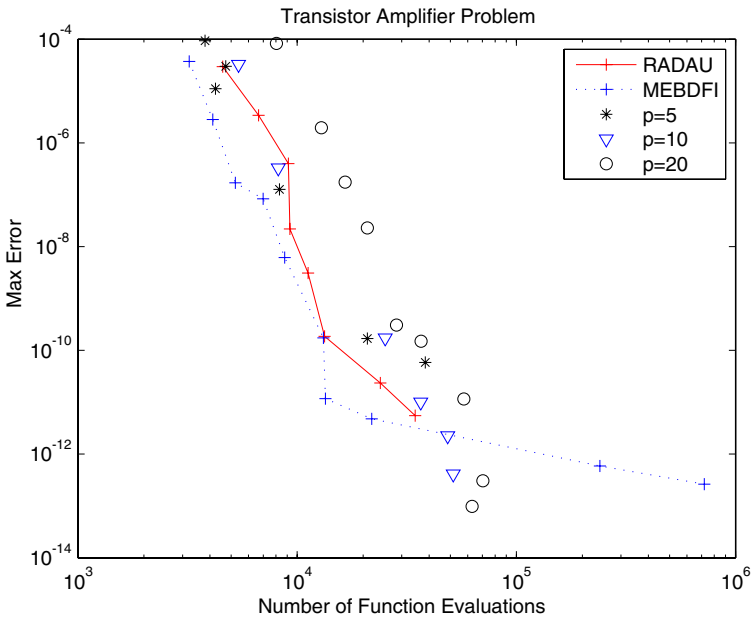

Fig. 5. Efficiency comparison of the fixed order uniform step KDC method with adaptive RADAU and MEBDFI.

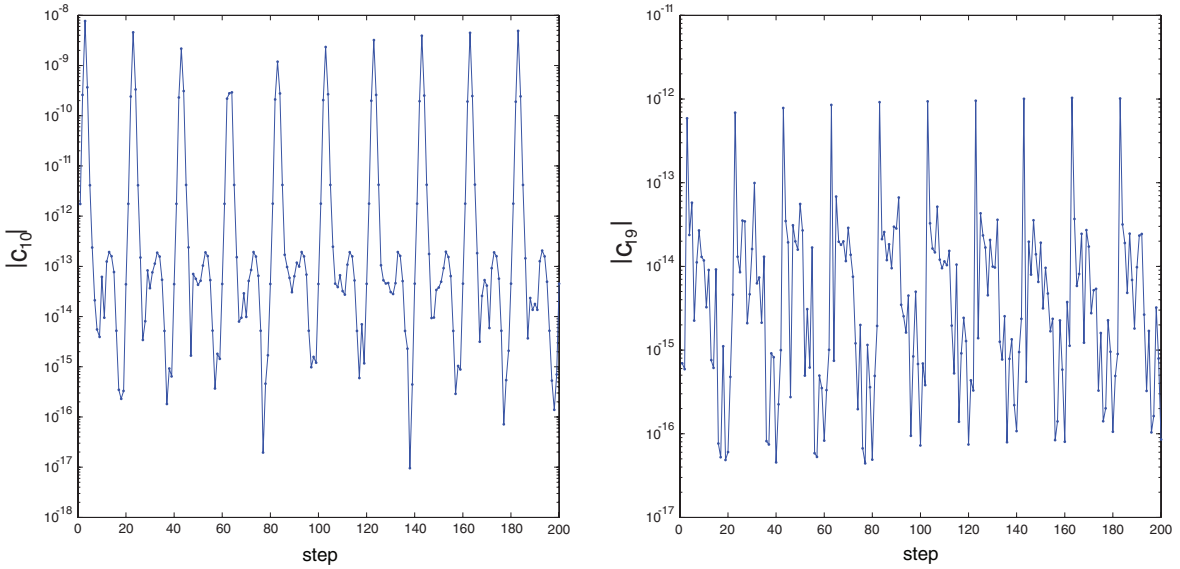

Fig. 6. Magnitude of the Legendre coefficients c_{10} and c_{19}.

$$y = \begin{pmatrix} q \\ \dot{q} \\ \ddot{q} \\ \lambda \end{pmatrix}, \quad K = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \Phi(y) = \begin{pmatrix} \dot{q} \\ \ddot{q} \\ M(q)\ddot{q} - f(q,\dot{q}) + G^{\mathrm{T}}(q)\lambda \\ g(q) \end{pmatrix}.$$

The indices of the unknowns q, $\dot{q}$, $\ddot{q}$ and λ in y are 1, 2, 3, and 3, respectively. We refer interested readers to [1] for explicit forms of functions mentioned above.

As explained in [21], when the original SDC methods are applied to ODE problems, for sufficiently small time step-size Δt, each correction procedure can reduce the residual by a factor of Δt unless machine precision is reached. However, for most DAE systems we tested, especially for higher-index DAE problems, our numerical experiments show that the residual from un-accelerated SDC methods may no longer converge to zero. In Fig. 7, for Andrews' squeezing problem, we use 10 Radau IIa nodes and plot how the residual changes after each deferred correction step. Different step-sizes ($\Delta t = 10^{-3}$, 10^{-4}, 10^{-5}, and 10^{-6}) are tested, and it can be seen that the residual starts increasing after a few iterations regardless how small the step-size is. In Fig. 8, as a comparison, we show the residual after each accelerated Krylov deferred correction, in which the Krylov subspace methods are only applied to the differential variables. We notice that for all step-sizes, the residual quickly converges to machine precision.

The convergence of the KDC methods and the divergence of the un-accelerated SDC techniques can be explained by studying Eq. (26). In [21], it is shown that the SDC methods for ODEs are equivalent to a Neumann series expansion for the preconditioned system. For ODE problems, as $\frac{\partial \mathbf{F}}{\partial \mathbf{Y}} = I$, the matrix C in Eq. (26) is of order $O(\Delta t)$ as long as Δt is sufficiently small. In this case, the residual always converges to zero, and each spectral deferred correction increases the residual order by Δt. For DAE problems, due to the existence of algebraic equations, $\frac{\partial \mathbf{F}}{\partial \mathbf{Y}}$ may be singular, or some eigenvalues in C may be greater than 1 in magnitude, which explains the increasing residual for the un-accelerated SDC methods. When Krylov subspace methods are applied, however, the iteration is still convergent despite the existence of such eigenvalues. Therefore, considering the deferred/defect correction procedures as preconditioners for the collocation formulation and introducing the Newton–Krylov techniques analytically guarantees the local convergence and significantly improves the convergence rate for DAE problems.

Finally for this problem, we want to mention that in the KDC methods, the residuals may increase during iterations as shown in Fig. 8. We believe this is due to the inaccuracy in the forward difference approximation used to compute the multiplication of the Jacobian times a vector in the Newton–Krylov methods and the

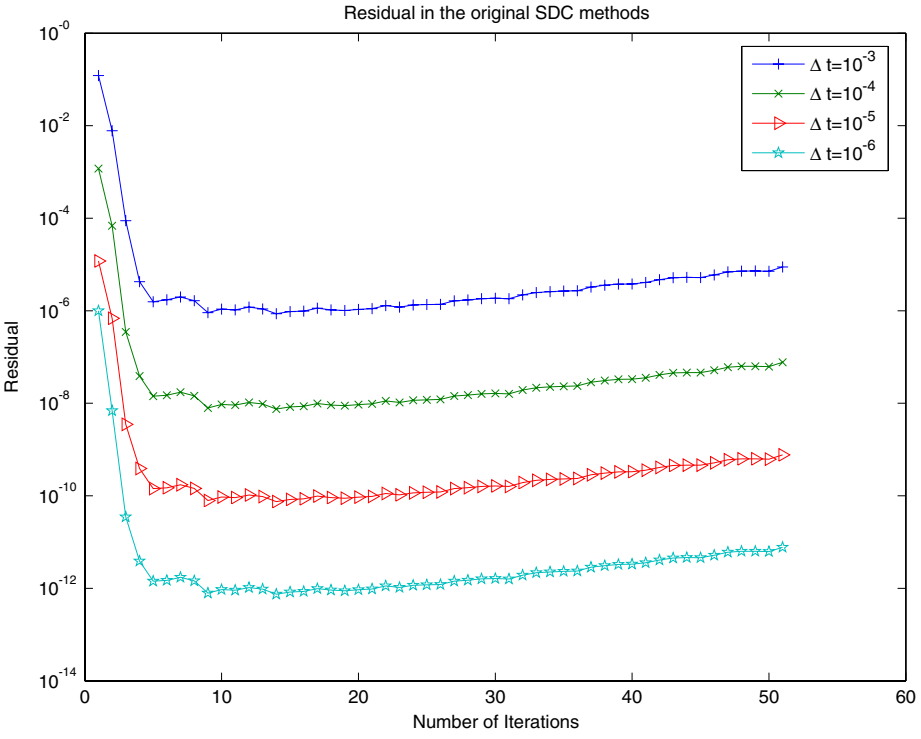

Fig. 7. Residual in the un-accelerated SDC method increases after first few corrections.

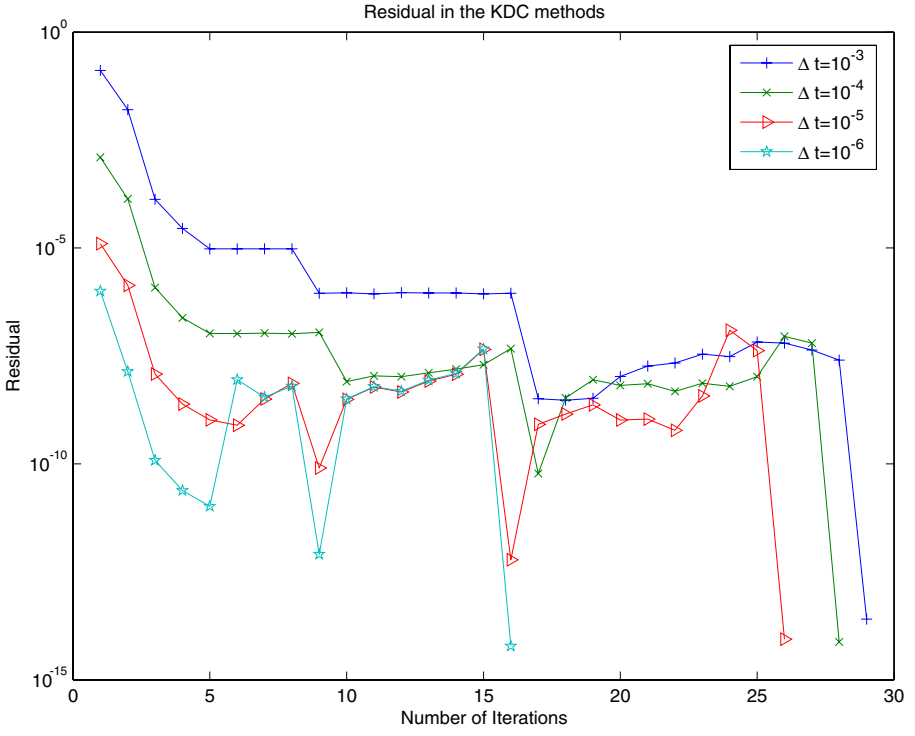

Fig. 8. Residual in the KDC methods converges to machine precision.

nonlinearity of the system. Currently, we are adapting the Newton–Krylov methods to improve the accuracy of the forward difference approximations. An immediate advantage is that (almost) decreasing residuals will provide better error control strategies.

5.4. Wheelset problem

In our last example, we consider the wheelset problem described by a DAE system of dimension 17 (also referred to as an implicit differential equation (IDE) system)

$$\frac{\mathrm{d}p}{\mathrm{d}t} = v, \tag{28}$$

$$M(p)\begin{pmatrix} \frac{\mathrm{d}v}{\mathrm{d}t} \\ \frac{\mathrm{d}\beta}{\mathrm{d}t} \end{pmatrix} = \begin{pmatrix} f(u) - (\partial g_1(p,q)/\partial p)^{\mathrm{T}} C\lambda \\ d(u) \end{pmatrix}, \tag{29}$$

$$0 = g_1(p,q), \tag{30}$$

$$0 = g_2(p,q), \tag{31}$$

where $u = (p, v, \beta, q, \lambda)^{\mathrm{T}} \in \mathbb{R}^{17}$, $p, v \in \mathbb{R}^5$, $\beta \in \mathbb{R}$, $q \in \mathbb{R}^4$, $\lambda \in \mathbb{R}^2$ and C is a scalar constant. Furthermore, $M : \mathbb{R}^5 \to \mathbb{R}^6 \times \mathbb{R}^6$, $f : \mathbb{R}^{17} \to \mathbb{R}^5$, $d : \mathbb{R}^{17} \to \mathbb{R}$, $g_1 : \mathbb{R}^9 \to \mathbb{R}^2$ and $g_2 : \mathbb{R}^9 \to \mathbb{R}^4$. This problem shows some typical properties of simulation problems in contact mechanics, i.e., friction, contact conditions, stiffness, etc. It is an index 3 IDE system but can be reduced to index 2. Interested readers are referred to [1] for the initial conditions, the function forms of M, f, d, g_1 and g_2, as well as more detailed discussions of the problem. In the following, similar to [1], we present test results based on the index-2 formulation where Eq. (30) is replaced by

$$0 = (\partial g_1(p,q)/\partial p)v.$$

We then apply the Picard discretization and Krylov acceleration only to the differential variables p, v, β.

For this test, we march with uniform time-step from $t_0 = 0$ to $t_{\mathrm{final}} = 0.002$ (a region in which relative uniform step-sizes are used by the compared methods). A comparison of the performance of KDC methods using 4 and 8 nodes and various fixed time steps with the DASSL, MEBDFI and PSIDE codes is shown in Fig. 9. Our numerical experiments show that the MEBDFI method requires 159 function evaluations for 5 digits of accuracy and 581 for 12 digits. On the other hand, the new KDC method with 4 Radau IIa nodes requires 40 function evaluations for 5 digits and 348 for 12 digits. The KDC method with 8 nodes uses 152 function evaluations for 12 digits. A more informative comparison (i.e. a longer integration interval and comparison of cpu-times) will be reported when an adaptive time-step KDC method has been implemented.

Finally, because of the excessive storage requirements of GMRES(k_0), we present here a comparison of alternative Krylov subspace methods applied to the wheelset problem. Specifically, we consider the biconjugate gradients stabilized (BiCGStab) method and transpose-free quasi-minimal residual (TFQMR) algorithm (see [6] for a summary of existing Newton–Krylov methods). The storage required in both methods is independent of iteration number k, and the number of multiplications grows only linearly as a function of k. In Fig. 10, for the wheelset problem, we compare the convergence of the full GMRES procedure with BiCGStab and TFQMR in terms of (a) number of iterations and (b) number of function evaluations. In the simulation, we use $p = 4$ Radau IIa nodes and set $\Delta t = t_{\mathrm{final}}$. It can be seen that both BiCGStab and TFQMR converge to the prescribed accuracy after numbers of iterations fewer than full GMRES, with similar numbers of function evaluations. Similar numerical results for $p = 8$ are shown in Fig. 11. Comparisons of different Krylov subspace methods as well as the optimal choices of different parameters (step-size, k_0, η, etc.) are being studied.

6. Conclusion and generalization

In this paper, spectral deferred correction methods are introduced as preconditioners for the collocation formulations of general differential algebraic systems, and Newton–Krylov methods are applied to the resulting nonlinear equations. The resulting KDC methods can in principle be of arbitrarily high order of accuracy while maintaining the computational complexity during the time-marching of simple first-order methods. Pre-

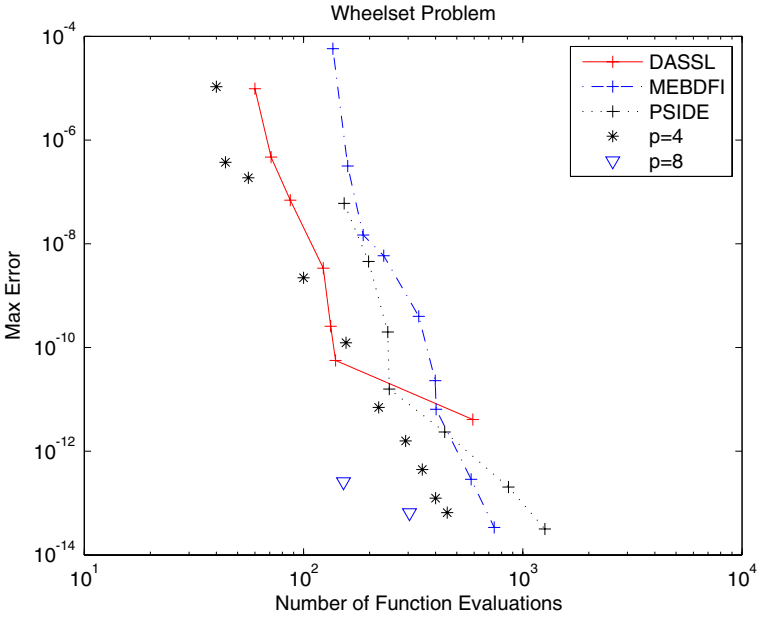

Fig. 9. Efficiency comparison of the uniform step KDC method with adaptive DASSL, MEBDFI and PSIDE.

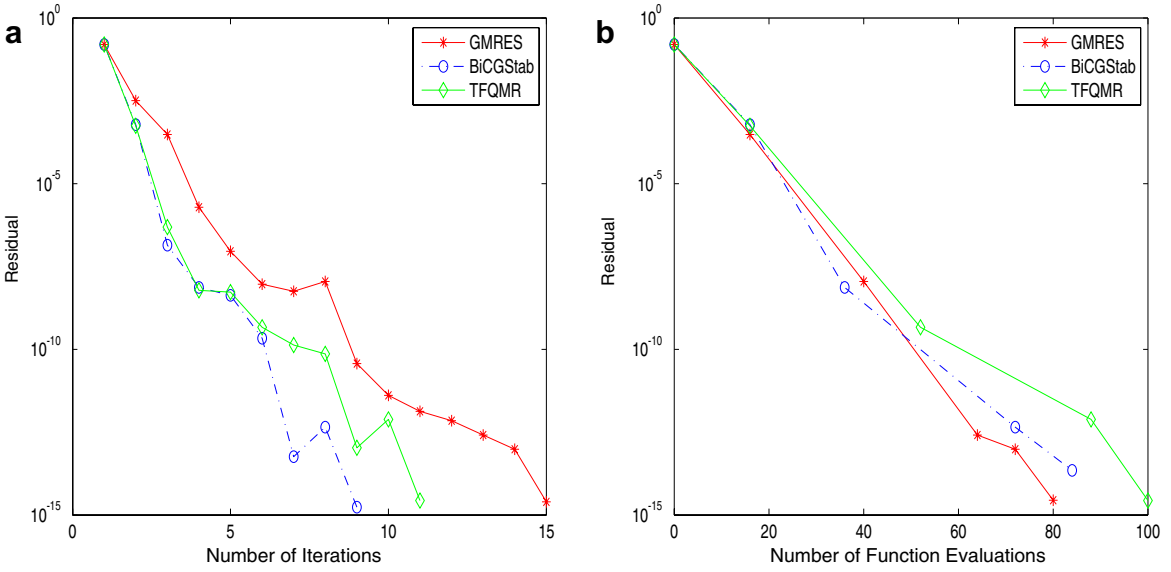

Fig. 10. Convergence of KDC iterations with 4 Radau IIa points for different Newton–Krylov methods.

liminary numerical results show that KDC methods can achieve similar accuracy to existing DAE methods while using a much larger time step. However, in order to fully explore the efficiency of KDC methods, a direct comparison of execution time with existing methods needs to be completed. This requires optimized strategies for the selection of adaptive step-size, order of the method, proper Newton–Krylov methods, simplified Newton's method, as well as several different parameters. These issues are currently being studied by the authors.

The key idea of the new technique is that ''lower-order methods'', if less costly to solve, provide good preconditioners for higher-order methods. This idea can be generalized to other problems including ODE boundary value problems, PDE problems, as well as multi-scale problems which can be reformulated as DAE systems.

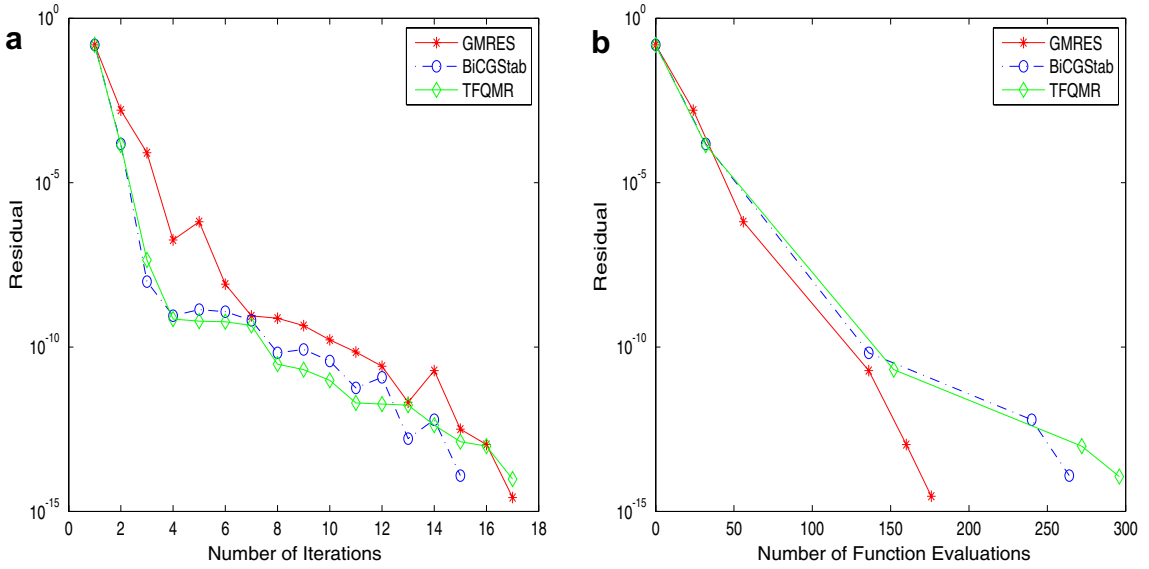

Fig. 11. Convergence of KDC iterations with 8 Radau IIa points for different Newton–Krylov methods.

Acknowledgments

We express our gratitude to Prof. Carl T. Kelly of North Carolina State University for discussions of the inexact Newton methods, and to Professor Casey Miller of the University of North Carolina for many helpful suggestions. Part of the work was finished while J.H. was a visiting scholar at Tsinghua University, Beijing, China, and their support is thankfully acknowledged.

References

[1] Available from: <http://pitagora.dm.uniba.it/~testset/>.
[2] B.K. Alpert, V. Rokhlin, A fast algorithm for the evaluation of Legendre expansions, SIAM J. Sci. Stat. Comput. 12 (1) (1991) 158–179.
[3] U.M. Ascher, L.R. Petzold, Computer Methods for Ordinary Differential Equations and Differential–Algebraic Equations, SIAM, Philadelphia, 1998.
[4] K. Atkinson, An Introduction to Advanced Numerical Analysis, second ed., Wiley, New York, 1989.
[5] W. Auzinger, H. Hofstatter, W. Kreuzer, E. Weinmuller, Modified defect correction algorithms for ODEs. Part I: General theory, Numer. Algorithms 36 (2004) 135–156.
[6] R. Barrett et al., Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, second ed., SIAM, Philadelphia, 1994.
[7] A. Bourlioux, A.T. Layton, M.L. Minion, High-order multi-implicit spectral deferred correction methods for problems of reactive flow, J. Comput. Phys. 189 (2003) 351–376.
[8] K.E. Brenan, S.L. Campbell, L.R. Petzold, Numerical Solution of Initial-Value Problems in Differential–Algebraic Equations, SIAM, Philadelphia, 1995.
[9] M.P. Calvo, C. Palencia, Avoiding the order reduction of Runge–Kutta methods for linear initial boundary value problems, Math. Comput. 71 (2002) 1529–1543.
[10] C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang, Spectral Methods in Fluid Dynamics, Springer, Berlin, 1988.
[11] K. Chen, A. Iserles, P.G. Ciarlet (Eds.), Matrix Preconditioning Techniques and Applications, Cambridge University Press, Cambridge, 2005.
[12] K. Dekker, J.G. Verwer, Stability of Runge–Kutta Methods for Stiff Nonlinear Differential EquationsCWI Monographs, North-Holland, Amsterdam, 1984.
[13] A. Dutt, L. Greengard, V. Rokhlin, Spectral deferred correction methods for ordinary differential equations, BIT 40 (2) (2000) 241–266.
[14] A. Dutt, M. Gu, V. Rokhlin, Fast algorithms for polynomial interpolation, integration, and differentiation, SIAM J. Numer. Anal. 33 (5) (1996) 1689–1711.

[15] D. Gottlieb, S.S. Orszag, Numerical Analysis of Spectral Methods, SIAM, Philadelphia, 1977.
[16] L. Greengard, Spectral integration and two-point boundary value problems, SIAM J. Numer. Anal. 28 (1991) 1071–1080.
[17] L. Greengard, V. Rokhlin, A fast algorithm for particle simulations, J. Comput. Phys. 73 (1987) 325–348.
[18] E. Hairer, C. Lubich, M. Roche, The Numerical Solution of Differential–Algebraic Systems by Runge–Kutta Methods, Springer, Berlin, 1989.
[19] E. Hairer, C. Lubich, G. Wanner, Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations, Springer, Berlin, 2002.
[20] E. Hairer, G. Wanner, Solving Ordinary Differential Equations II, Springer, Berlin, 1996.
[21] J. Huang, J. Jia, M. Minion, Accelerating the convergence of spectral deferred correction methods, J. Comput. Phys. 214 (2) (2006) 633–656.
[22] C.T. Kelly, Solving Nonlinear Equations with Newton's Method, SIAM, Philadelphia, 2003.
[23] C.T. Kelly, Iterative Methods for Linear and Nonlinear Equations, SIAM, Philadelphia, 1995.
[24] A.T. Layton, M.L. Minion, Implications of the choice of quadrature nodes for Picard integral deferred corrections methods for ordinary differential equations, BIT 45 (2) (2005) 341–373.
[25] A.T. Layton, M.L. Minion, Conservative multi-implicit spectral deferred correction methods for reacting gas dynamics, J. Comput. Phys. 194 (2) (2004) 697–714.
[26] V. Pereyra, Iterated deferred correction for nonlinear boundary value problems, Numer. Math. 11 (1968) 111–125.
[27] P. Kolm, S. Jiang, V. Rokhlin, Quadruple and octuple layer potentials in two dimensions. I. Analytical apparatus, Appl. Comput. Harmon. Anal. 14 (1) (2003).
[28] M.L. Minion, Semi-implicit spectral deferred correction methods for ordinary differential equations, Comm. Math. Sci. 1 (2003) 471–500.
[29] L.R. Petzold, A Description of DASSL: A Differential–Algebraic System Solver, SAND82-8637, Sandia National Lab, 1982.
[30] P.K. Vijalapura, J. Strain, S. Govindjee, Fractional step methods for index-1 differential–algebraic equations, J. Comput. Phys. 203 (1) (2005) 305–320.
[31] A. Rangan, Adaptive solvers for partial differential and differential–algebraic equations, Ph.D. Thesis, University of California at Berkeley, 2003.
[32] A. Rangan, Deferred correction methods for low index differential algebraic equations, preprint.
[33] Y. Saad, M.H. Schultz, GMRES: a generalized minimal residual algorithm for solving non-symmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (1986) 856–869.
[34] J.M. Sanz-Serna, J.G. Verwer, W.H. Hundsdorfer, Convergence and order reduction of Runge–Kutta schemes applied to evolutionary problems in partial differential equations, Numer. Math. 50 (1986) 405–418.
[35] L.N. Trefethen, M.R. Trummer, An instability phenomenon in spectral methods, SIAM J. Numer. Anal. 24 (1987).
[36] P.E. Zadunaisky, A method for the estimation of errors propagated in the numerical solution of a system of ordinary differential equations, The Theory of Orbits in the Solar System and in Stellar Systems, in: Proceedings of International Astronomical Union, Symposium 25, 1964.
[37] P.E. Zadunaisky, On the estimation of errors propagated in the numerical integration of ordinary differential equations, Numer. Math. 27 (1976) 21–40.