Available online at www.sciencedirect.com
SCIENCE DIRECT
Journal of Computational Physics 214 (2006) 633–656
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Accelerating the convergence of spectral deferred correction methods

Jingfang Huang *, Jun Jia, Michael Minion

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

Received 9 May 2005; received in revised form 6 October 2005; accepted 7 October 2005
Available online 29 November 2005

Abstract

In the recent paper by Dutt, Greengard and Rokhlin, a variant of deferred or defect correction methods is presented which couples Gaussian quadrature with the Picard integral equation formulation of the initial value ordinary differential equation. The resulting *spectral deferred correction* (SDC) methods have been shown to possess favorable accuracy and stability properties even for versions with very high order of accuracy. In this paper, we show that for linear problems, the iterations in the SDC algorithm are equivalent to constructing a preconditioned Neumann series expansion for the solution of the standard collocation discretization of the ODE. This observation is used to accelerate the convergence of SDC using the GMRES Krylov subspace method. For nonlinear problems, the GMRES acceleration is coupled with a linear implicit approach. Stability and accuracy analyses show the accelerated scheme provides an improvement in the accuracy, efficiency, and stability of the original SDC approach. Furthermore, preliminary numerical experiments show that accelerating the convergence of SDC methods can effectively eliminate the order reduction previously observed for stiff ODE systems.

MSC: 65B05; 65F10; 65L05

Keywords: Spectral deferred correction methods; Stiff equations; Krylov subspace methods; GMRES

1. Introduction

In this paper, we discuss the numerical solution of the ordinary differential equation (ODE) initial value problem

$$\varphi'(t) = F(t, \varphi(t)), \quad t \in [0, T], \tag{1}$$

$$\varphi(0) = \varphi_0, \tag{2}$$

where $\varphi_0, \varphi(t) \in \mathbb{C}^N$ and $F : \mathbb{R} \times \mathbb{C}^N \to \mathbb{C}^N$. Many numerical techniques for approximating this type of equation have been developed in the last century and the readers are referred to [3,4,10,15,16,23,32] for detailed

* 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).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2005.10.004

discussions. In this paper, we present a technique which is designed to improve the performance of the spectral deferred correction (SDC) methods introduced by Dutt, Greengard, and Rokhlin in 2000 [9]. The SDC strategy introduced in [9] is a variation on the classical defect or deferred correction methods [5,25,28,31,34,35] which allows for the construction of stable explicit and implicit methods with extremely high order of accuracy. As with classical deferred and defect correction methods cited above, a single time step of an SDC method begins by first dividing the time step $[t_n, t_{n+1}]$ into a set of intermediate sub-steps defined by the points $\vec{t} = [t_0, t_1, \ldots, t_p]$ with $t_n = t_0 < t_1 < \ldots < t_p \leqslant t_{n+1}$. For simplicity, we assume $t_n = t_0 = 0$ in the following discussions. Next, a provisional approximation $\vec{\varphi}^{[0]} = [\varphi^{[0]}(t_0), \varphi^{[0]}(t_1), \cdots, \varphi^{[0]}(t_p)]$ is computed at the intermediate points using a standard numerical method, e.g., the explicit Euler method for non-stiff problems or the implicit Euler method for stiff problems as in [9]. Applying standard approximation or interpolation theory, the continuous counter part of $\vec{\varphi}^{[0]}$ can be constructed and is represented as $\varphi^{[0]}(t)$. Using $\varphi^{[0]}(t)$, an equation for the error $\delta(t) = \varphi(t) - \varphi^{[0]}(t)$ is then constructed. This correction equation for $\delta(t)$ can be approximated using a similar low order method, and an improved numerical solution is constructed. This procedure can then be repeated resulting in a sequence of approximate solutions.

To construct the correction equation, the classical methods cited above rely on differentiation of $\varphi^{[0]}(t)$ to form an ODE for $\delta(t)$, where $\varphi^{[0]}(t)$ is the interpolating polynomial of $\vec{\varphi}^{[0]}$. On the other hand, SDC methods utilize the Picard integral equation formulation of the ODE

$$\varphi(t) = \varphi_0 + \int_0^t F(\tau, \varphi(\tau))\,\mathrm{d}\tau \tag{3}$$

to construct a corresponding integral equation for $\delta(t)$. Specifically

$$\delta(t) = \int_0^t [F(\tau, \varphi^{[0]}(\tau) + \delta(\tau)) - F(\tau, \varphi^{[0]}(\tau))]\,\mathrm{d}\tau + \epsilon(t), \tag{4}$$

where

$$\epsilon(t) = \varphi_0 + \int_0^t F(\tau, \varphi^{[0]}(\tau))\,\mathrm{d}\tau - \varphi^{[0]}(t). \tag{5}$$

The discretizaton of these equations will be discussed in more detail in the following section, but for now note that the discretization of $\epsilon(t)$ is simply a numerical integration. It is for this reason that the points $\vec{t}$ which define the sub-steps in SDC methods are chosen to be Gaussian quadrature nodes so the numerically stable spectral integration technique can be applied [12]. The integral equation formulation for $\delta(t)$ in Eq. (4) coupled with spectral integration rules allows SDC methods to overcome the loss of stability of classical deferred/defect correction methods as the order of the method increases. For a detailed discussion of the different choices of quadrature nodes see [24].

Deferred correction methods based on the Picard integral formulation and spectral integration are of interest for several reasons, most notably because of the relative ease with which one can theoretically construct methods with arbitrarily high order of accuracy. Preliminary numerical tests presented in [9] suggest that SDC methods are competitive with the best existing ODE initial value problem solvers, especially for stiff problems or where high accuracy is required. Furthermore the stability regions of the implicit methods are close to optimal and do not degrade with increased orders of accuracy [24]. Semi-implicit and multi-implicit variations of SDC methods have also been presented which enable the construction of very high order methods for equations with both stiff and non-stiff components [6,26,27]. Also noted in these papers, however, is the fact that when SDC methods are applied to very stiff equations, the effective order of accuracy of the method is reduced for values of the time step above a certain threshold. This type of order reduction (which is also present in many popular types of Runge–Kutta methods [7,8,30]) means that, although the methods are stable for larger time steps, one must use a very small time step for the method to converge with full order.

The main results in this paper stem from considering the limit of the correction iterations for an SDC method for a fixed step of size $\Delta t = t_{n+1} - t_n$. Observe that, if the correction iteration in the SDC method converges, then $\epsilon(t)$ given in Eq. (5) will approach zero at the Gaussian nodes $\vec{t} \in [t_n, t_{n+1}]$. Hence the resulting limit solution will satisfy the collocation (or pseudo-spectral) approximation of the Picard equation (3) given by

$$\vec{\varphi} = \vec{\varphi}_0 + \Delta t \mathbf{S}\vec{F}, \tag{6}$$

where

$$\vec{F} = [F(t_0, \varphi(t_0)), F(t_1, \varphi(t_1)), \ldots, F(t_p, \varphi(t_p))]^{\mathrm{T}},$$

$\vec{\varphi}_0$ is a vector of initial conditions

$$\vec{\varphi}_0 = [\varphi(t_0), \varphi(t_0), \ldots, \varphi(t_0)]^{\mathrm{T}}$$

and $\mathbf{S}$ is the spectral integration matrix [11,12] corresponding to the Gaussian nodes $\vec{t}$ discussed in detail in Section 2.3.

Since Eq. (6) couples the solution values at each of the sub-steps defined by $\vec{t}$, a direct solution of this equation requires a system of size Np be solved, as opposed to a system of size N which arises from a single sub-step of an SDC method (or in other typical methods like BDF). However, since the limit of the SDC iterations (when it converges) is the collocation solution, then one iteration of the correction step in an SDC method can also be thought of as one step in an iterative procedure for solving Eq. (6) directly. Once this observation is made, it is natural to attempt to accelerate this convergence if possible.

In Section 3, we show that for linear problems, the SDC method is equivalent to solving a particular preconditioned equation for the error corresponding to Eq. (6). Moreover, an explicit form for the provisional solution after the kth SDC iteration $\vec{\varphi}^{[k]}$ in terms of a Neumann series expansion is derived. Specifically,

$$\vec{\varphi}^{[k]} - \vec{\varphi}^{[0]} \approx \vec{b} + \mathbf{C}\vec{b} + \mathbf{C}^2\vec{b} + \cdots + \mathbf{C}^k\vec{b} \tag{7}$$

for a specific matrix $\mathbf{C}$ and vector $\vec{b}$. A consequence of the derivation of Eq. (7) is that it provides a precise statement of when and how rapidly the correction iterations in the SDC method converge. This observation can also be used to accelerate the convergence of the original SDC methods by searching for the optimal solution in the Krylov subspace $\mathbf{K}(\mathbf{C}, \vec{b}) = \mathrm{span}\{\vec{b}, \mathbf{C}\vec{b}, \mathbf{C}^2\vec{b}, \ldots, \mathbf{C}^k\vec{b}\}$ using the generalized minimum residual (GMRES) or other Krylov subspace based iterative methods. For nonlinear problems, the above acceleration can be coupled with a linear implicit approach to improve the performance of SDC methods.

In this paper, the new class of accelerated methods is studied analytically and numerical comparisons with the original SDC methods for both linear and nonlinear problems are presented. Stability and accuracy analyses for the accelerated schemes are given. We observe that for non-stiff problems, GMRES accelerated SDC methods improve both the accuracy and stability of the original SDC methods. In fact, in several numerical examples we tested, for a given time step of size Δt, the accelerated methods quickly converge to the collocation solution while the correction iterations of the original SDC are divergent. For stiff problems [20], we show the accelerated methods improve the accuracy of the original SDC, and under certain assumptions, remove the order reduction phenomenon observed in the original SDC.

The structure of this paper is as follows. In Section 2, we briefly describe the original spectral deferred correction methods. In Section 3, we show that for linear problems, the original SDC is equivalent to the preconditioned Neumann series expansion given in Eq. (7). In Section 4, we describe how the convergence of the original SDC can be accelerated for both linear and non-linear problems. In Section 5, we present the stability and accuracy analyses for the GMRES accelerated SDC methods. In Section 6, we demonstrate the improved accuracy and stability of the accelerated methods using several linear and nonlinear examples. Finally in Section 7, we discuss possible extensions and further generalizations of the approach.

2. The spectral deferred correction methods

In this section, we summarize the details of the spectral deferred correction methods from [9] which are necessary to present the derivation of the accelerated methods in Sections 3 and 4.

2.1. The Picard integral equation and error equation

Consider the Picard integral equation representation of the ODE initial value problem given in Eq. (3). Suppose an approximate solution $\varphi^{[0]}(t)$ to Eqs. (1) and (2) is given, and define the error $\delta(t)$ as before

$$\delta(t) = \varphi(t) - \varphi^{[0]}(t). \tag{8}$$

Substituting (8) into (3) yields

$$\varphi^{[0]}(t) + \delta(t) = \varphi_0 + \int_0^t F(\tau, \varphi^{[0]}(\tau) + \delta(\tau))\,\mathrm{d}\tau. \tag{9}$$

To reduce notational clutter here and in the following the time dependence of the second argument of F will be implicitly assumed, e.g., $F(t,\varphi^{[0]}(t))$ is written simply as $F(t,\varphi^{[0]})$. Now consider the residual function

$$\epsilon(t) = \varphi_0 + \int_0^t F(\tau, \varphi^{[0]})\,\mathrm{d}\tau - \varphi^{[0]}(t), \tag{10}$$

which simply gives the error in the Picard equation (3). Rearranging Eq. (9) and using Eq. (10) gives a Picard-type integral equation for the error

$$\delta(t) = \int_0^t [F(\tau, \varphi^{[0]} + \delta) - F(\tau, \varphi^{[0]})]\,\mathrm{d}\tau + \epsilon(t). \tag{11}$$

Note that unlike the classical deferred or defect correction methods in [28,34,35], the equation for $\delta(t)$ is not written here as an ODE.

2.2. Euler methods on Gaussian quadrature nodes

Deferred correction methods proceed by iteratively solving the error Eq. (11) using a low order method to improve the provisional solution $\vec{\varphi}^{[0]}$. To describe the time stepping procedure, suppose as before that the time step interval $[t_n, t_{n+1}]$ has been subdivided using the points $t_0,t_1,t_2,\ldots,t_p$ such that

$$t_n = t_0 < t_1 < t_2 \cdots < t_p \leqslant t_{n+1}. \tag{12}$$

Note that Eq. (11) gives the identity

$$\delta(t_{m+1}) = \delta(t_m) + \int_{t_m}^{t_{m+1}} [F(\tau, \varphi^{[0]} + \delta) - F(\tau, \varphi^{[0]})]\,\mathrm{d}\tau + \epsilon(t_{m+1}) - \epsilon(t_m). \tag{13}$$

Letting δ_m denote the numerical approximation to $\delta(t_m)$ (and likewise for $\varphi_m^{[0]}$ and ϵ_m), a simple discretization of Eq. (13) similar to the explicit Euler (forward Euler) method for ODEs is

$$\delta_{m+1} = \delta_m + \Delta t_m (F(t_m, \varphi_m^{[0]} + \delta_m) - F(t_m, \varphi_m^{[0]})) + \epsilon_{m+1} - \epsilon_m, \tag{14}$$

where $\Delta t_m = t_{m+1} - t_m$. Similarly, an implicit scheme for the solution based on the backward Euler method is

$$\delta_{m+1} = \delta_m + \Delta t_m \left(F(t_{m+1}, \varphi_{m+1}^{[0]} + \delta_{m+1}) - F(t_{m+1}, \varphi_{m+1}^{[0]}) \right) + \epsilon_{m+1} - \epsilon_m. \tag{15}$$

Denoting the "low order" approximation of $\delta(t)$ by $\vec{\delta}^{[1]} = [\delta_1, \delta_2, \ldots, \delta_p]$, a refined solution is given by $\vec{\varphi}^{[1]} = \vec{\varphi}^{[0]} + \vec{\delta}^{[1]}$. In order to complete the discretization, we must specify how the terms ϵ_m are computed.

2.3. The spectral integration matrix

First note that there are various ways to choose the points $t_0,t_1,t_2,\ldots,t_p$ to define the sub-steps in the SDC method. When Gaussian quadrature nodes are used, $\{t_1,\ldots t_p\}$ are interior points in $[t_n, t_{n+1}]$ and the endpoints are not used. On the other hand the Radau Ia quadrature nodes $t_0,t_1,t_2,\ldots,t_p$ use the left end point while the Radau IIa nodes $t_1,t_2,\ldots,t_p$ have $t_p = t_{n+1}$. Finally, the Lobatto quadrature rule requires the use of both end points.

Using the Gaussian nodes as an example, suppose we are given the scalar function values $\vec{\varphi} = \{\varphi_1, \varphi_2, \ldots, \varphi_p\}$ at the nodes, then the Legendre polynomial expansion $L^p(\vec{\varphi}, t)$ can be constructed to approximate $\vec{\varphi}$ where the coefficients are computed using Gaussian quadrature rules. This gives a numerically stable and efficient way to find the equivalent interpolating polynomial of degree $p-1$. Integrating this interpolating polynomial analytically from t_0 to t_m, a linear mapping Q is derived which maps the function values $\vec{\varphi}$ to the integral of the interpolating polynomial

$$[\vec{\varphi}]_m = \int_{t_0}^{t_m} L^p(\vec{\varphi}, \tau)\,\mathrm{d}\tau.$$

This can be written in matrix form

$$Q\vec{\varphi} = \Delta t S \vec{\varphi}, \tag{16}$$

where S will be referred to as the integration matrix, and is independent of Δt. Note that in the more general case where $\varphi(t) \in \mathbb{C}^N$, Eq. (16) must be interpreted as being applied component-wise to $\vec{\varphi}$, i.e., $\vec{\varphi}$ is a vector of length Np and

$$Q\vec{\varphi} = \Delta t (I_p \otimes S)\vec{\varphi}, \tag{17}$$

where I_p is the identity matrix of size $p \times p$. In the following, we use script font to denote this tensor product, i.e. $\mathbf{S}$ denotes the $Np \times Np$ block diagonal matrix $I_p \otimes S$.

For traditional deferred/defect correction methods, there are two factors which prevent the use of extremely high order methods: The first problem relates to the instability of interpolation at equispaced nodes where the Runge phenomenon can be observed when the number of interpolation points p is large. The second problem is that numerical differentiation in the original ODE formulation (Eqs. (1) and (2)) introduces instabilities [33]. Spectral deferred correction methods avoid both of these difficulties by introducing Gaussian-type nodes and using the Picard integral equation. The procedure is explained in next section. In the current numerical implementation, the Legendre polynomial based Radau IIa quadrature nodes are used and the matrix S is precomputed using Mathematica requesting more than 20 digits in accuracy. Detailed comparisons of different choices of nodes will be reported in the future (see also [24]).

2.4. The spectral deferred correction algorithm

Given an approximate solution $\vec{\varphi}^{[0]} = [\varphi_1^{[0]}, \ldots, \varphi_p^{[0]}]$, consider the error equation given by (11). Discretizing the integral in (10) in the same manner as in (6) using the spectral integration matrix yields

$$\vec{\epsilon} = \vec{\varphi}_0 + \Delta t \mathbf{S}\vec{F} - \vec{\varphi}^{[0]}, \tag{18}$$

where $\vec{\epsilon} = [\epsilon(t_1), \ldots, \epsilon(t_p)]$ is the residual at the intermediate points. Once the residual is calculated, an approximation $\vec{\delta}^{[1]}$ to the error Eq. (11) is computed using p steps of the Eq. (14) for non-stiff problems or Eq. (15) for stiff problems. The provisional solution is then updated with $\vec{\varphi}^{[1]} = \vec{\varphi}^{[0]} + \vec{\delta}^{[1]}$, and this procedure can be repeated. The algorithm for SDC is given by the following:

Pseudo-code: spectral deferred correction method

Comment [Compute initial approximation]

For non-stiff/stiff problems, use the forward/backward Euler method to compute an approximate solution $\varphi_m^{[0]} \approx \varphi(t_m)$ at the sub-steps $t_1, \ldots, t_p$ on the interval $[t_n, t_{n+1}]$.

Comment [Compute successive corrections.]

do j = 1, ..., J

(1) Compute the approximate residual function $\vec{\epsilon}$ using $\vec{\varphi}^{[j-1]}$ and Eq. (18).

(2a) For non-stiff problems, compute $\vec{\delta}^{[j]}$ using p steps of Eq. (14).

(2b) For stiff problems, compute $\vec{\delta}^{[j]}$ using p steps of Eq. (15).

(3) Update the approximate solution $\vec{\varphi}^{[j]} = \vec{\varphi}^{[j-1]} + \vec{\delta}^{[j]}$.

endo

It can be shown that each correction procedure in this algorithm can improve the order of the method by one, as long as such improvement has not gone beyond the degree of the underlying interpolating polynomial and the quadrature rules [13,18,19]. For linear ODE problems, a proof is provided in Section 3.2 utilizing the Neumann series expansion.

Remark 2.1. For the initial approximation, an alternative method is to use a constant approximation. This has also been implemented and tested. Detailed analysis and comparisons will be reported in the future.

2.5. The collocation formulation limit of SDC

Consider the iterative correction procedure detailed in the last section. If the SDC procedure is convergent, then by Eq. (18) the limit satisfies $\vec{\epsilon} = 0$, which is equivalent to

$$\vec{\varphi} = \vec{\varphi}_0 + \Delta t \mathbf{S}\vec{F}. \tag{19}$$

This is identical to the collocation formula given in Eq. (6). Conditions specifying precisely when the SDC converges to this limit for linear systems are presented in Section 3.

It is possible to solve this equation directly using, for example, Newton's method [16,17]. However, for $\varphi(t) \in \mathbb{C}^N$ and assuming p interior points are used in each time step, the total number of unknowns in the collocation formula is $M = pN$. Therefore each iteration of Newton's method (or a direct solution if the problem is linear) requires inverting a matrix of size $M \times M$. In contrast, each correction iteration of the SDC method requires solving p linear or nonlinear systems with N unknowns. When the number of iterative corrections is small, SDC methods will be more efficient compared with the direct Newton's method approach, especially when the order p is high.

3. Spectral deferred corrections in matrix form

In the previous section, we show that the SDC method can be considered as an iterative scheme for solving the implicit equation arising from a direct discretization of the Picard integral equation in (19). In this section, we derive an explicit representation of the iteration in matrix form for the linear case which proves that the SDC iterations converge for linear systems and aids in analyzing and accelerating the convergence.

For the present, let $F(t,\varphi(t)) = L\varphi(t) + f(t)$ where L is a constant matrix. Given an approximate solution $\varphi^{[0]}(t)$, the discretized collocation formulation for the error equation in (11) becomes

$$\vec{\delta} - \Delta t \mathbf{S}\mathbf{L}\vec{\delta} = \vec{\varphi}_0 + \Delta t \mathbf{S}\vec{F} - \vec{\varphi}^{[0]},$$

where $\mathbf{L} = I_p \otimes L$ (see Section 2.3). Denoting the right hand side by $\vec{\epsilon}$, the SDC procedure iteratively approximates

$$(\mathbf{I} - \Delta t \mathbf{S}\mathbf{L})\vec{\delta} = \vec{\epsilon} \tag{20}$$

using the low order approximations $\vec{\delta}^{[j]}$ for $j = 1, 2, \ldots$. The goal of this section is to rewrite SDC methods in a matrix form and show that the original SDC technique is equivalent to solving Eq. (20) using a preconditioned Neumann series expansion, i.e., $\vec{\delta} = \sum_{j=1}^{\infty} \vec{\delta}^{[j]}$ where $\vec{\delta}^{[j+1]} = \mathbf{C}\vec{\delta}^{[j]}$ for an explicit matrix $\mathbf{C}$.

3.1. Euler method in matrix form

First consider the forward Euler method in Eq. (14) which is appropriate for non-stiff problems. For the linear correction Eq. (20), a sub-step is given by

$$\delta_{m+1} = \delta_m + \Delta t_m L \delta_m + (\epsilon_{m+1} - \epsilon_m). \tag{21}$$

Summing successive values of δ and using the fact that both the error $\delta(t)$ and the residual $\epsilon(t)$ are zero at t_0, some manipulation gives

$$\delta_{m+1} = \sum_{i=1}^{m} \Delta t_i L \delta_i + \epsilon_{m+1}. \tag{22}$$

Notice that $\sum_{i=1}^{m} \Delta t_i L \delta_i$ is the composite rectangular rule approximation (where the left end point is used) of the integral

$$\int_0^{t_{i+1}} L\delta(s)\,\mathrm{d}s.$$

Therefore, in matrix form, the forward Euler method is equivalent to solving

$$\left(\mathbf{I} - \Delta t \widetilde{\mathbf{S}}\mathbf{L}\right)\vec{\delta}^{[1]} = \vec{\epsilon}, \tag{23}$$

where $\vec{\delta}^{[1]} = [\delta_1, \delta_2, \ldots \delta_p]^{\mathrm{T}}$, $\vec{\epsilon} = [\epsilon_1, \epsilon_2, \ldots, \epsilon_p]^{\mathrm{T}}$, and

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

Notice that $\widetilde{S}$ is a strictly lower triangular approximation of the spectral integration matrix S. Similarly, for the implicit Euler method, the matrix $\widetilde{S}$ takes the form

$$\Delta t\widetilde{S} = \begin{bmatrix} \Delta t_0 & 0 & \cdots & 0 & 0 \\ \Delta t_0 & \Delta t_1 & \cdots & 0 & 0 \\ \Delta t_0 & \Delta t_1 & \cdots & 0 & 0 \\ \cdot & \cdot & \cdots & 0 & 0 \\ \Delta t_0 & \Delta t_1 & \cdots & \Delta t_{p-2} & 0 \\ \Delta t_0 & \Delta t_1 & \cdots & \Delta t_{p-2} & \Delta t_{p-1} \end{bmatrix}. \tag{25}$$

This lower triangular matrix is also an approximation of the spectral integration matrix, with non-zero diagonal entries.

To summarize, each correction in the SDC method may be considered as solving an approximation of the collocation formulation of the correction Eq. (20), where the spectral integration matrix is approximated by a lower triangular matrix. Clearly, the solution given by

$$\vec{\delta}^{[1]} = \left(\mathbf{I} - \Delta t\widetilde{\mathbf{S}}\mathbf{L}\right)^{-1}\vec{\epsilon} \tag{26}$$

is a low order approximation of $\vec{\delta}$ in Eq. (20).

3.2. The Neumann series

Suppose after k corrections, we have a provisional approximation $\vec{\varphi}^{[k]}$, the new residual is then defined as

$$\vec{\epsilon} = \vec{\varphi}_0 + \Delta t\mathbf{S}\mathbf{L}\vec{\varphi}^{[k]} - \vec{\varphi}^{[k]}.$$

Applying Euler method (which is equivalent to Eq. (26)) and denoting the solution by $\vec{\delta}^{[k+1]}$, the relationship between $\vec{\varphi}^{[k+1]}$ and $\vec{\varphi}^{[k]}$ is

$$\begin{aligned} \vec{\varphi}^{[k+1]} &= \vec{\varphi}^{[k]} + \vec{\delta}^{[k+1]} \\ &= \vec{\varphi}^{[k]} + \left(\mathbf{I} - \Delta t\widetilde{\mathbf{S}}\mathbf{L}\right)^{-1}\vec{\epsilon} \\ &= \vec{\varphi}^{[k]} + \left(\mathbf{I} - \Delta t\widetilde{\mathbf{S}}\mathbf{L}\right)^{-1}\left(\vec{\varphi}_0 - (\mathbf{I} - \Delta t\widetilde{\mathbf{S}}\mathbf{L})\vec{\varphi}^{[k]} + \Delta t(\mathbf{S} - \widetilde{\mathbf{S}})\mathbf{L}\vec{\varphi}^{[k]}\right) \\ &= \left(\mathbf{I} - \Delta t\widetilde{\mathbf{S}}\mathbf{L}\right)^{-1}\vec{\varphi}_0 + \mathbf{C}\vec{\varphi}^{[k]}, \end{aligned} \tag{27}$$

where we define

$$\mathbf{C} = \left(\mathbf{I} - \Delta t\widetilde{\mathbf{S}}\mathbf{L}\right)^{-1}\Delta t(\mathbf{S} - \widetilde{\mathbf{S}})\mathbf{L}. \tag{28}$$

It is also straightforward to derive the recursive relationship between $\vec{\delta}^{[k+1]}$ and $\vec{\delta}^{[k]}$. First note that

$$\vec{\varphi}^{[k+1]} = \left(\mathbf{I} - \Delta t\widetilde{\mathbf{S}}\mathbf{L}\right)^{-1}\vec{\varphi}_0 + \mathbf{C}\vec{\varphi}^{[k]}$$

and

$$\vec{\varphi}^{[k]} = \left(\mathbf{I} - \Delta t\widetilde{\mathbf{S}}\mathbf{L}\right)^{-1}\vec{\varphi}_0 + \mathbf{C}\vec{\varphi}^{[k-1]}.$$

Subtracting the two identities yields

$$\vec{\delta}^{[k+1]} = \mathbf{C}\vec{\delta}^{[k]}. \tag{29}$$

Assuming our initial provisional approximation is given by $\vec{\varphi}^{[0]}$, then from the recursive relation (29), the solution after k corrections is given by the Neumann series expansion:

$$\vec{\varphi}^{[k]} = \vec{\varphi}^{[0]} + \sum_{m=1}^{k}\mathbf{C}^{m-1}\vec{\delta}^{[1]}. \tag{30}$$

We can also derive the Neumann series expansion by solving the error Eq. (20). Multiplying both sides by $(\mathbf{I} - \Delta t\widetilde{\mathbf{S}}\mathbf{L})^{-1}$, we have the preconditioned linear system

$$\left(\mathbf{I} - \Delta t\widetilde{\mathbf{S}}\mathbf{L}\right)^{-1}(\mathbf{I} - \Delta t\mathbf{S}\mathbf{L})\vec{\delta} = \left(\mathbf{I} - \Delta t\widetilde{\mathbf{S}}\mathbf{L}\right)^{-1}\vec{\epsilon}. \tag{31}$$

Notice that the right hand side of (31) is $\vec{\delta}^{[1]}$ and the operator on the left is

$$\left(\mathbf{I} - \Delta t\widetilde{\mathbf{S}}\mathbf{L}\right)^{-1}(\mathbf{I} - \Delta t\mathbf{S}\mathbf{L}) = (\mathbf{I} - \mathbf{C}), \tag{32}$$

where $\mathbf{C}$ is defined in Eq. (28). Hence, the preconditioned error equation is given by the linear system

$$(\mathbf{I} - \mathbf{C})\vec{\delta} = \vec{\delta}^{[1]}.$$

As $\widetilde{\mathbf{S}}$ is an approximation of the matrix $\mathbf{S}$, when Δt is small, we expect the norm of $\mathbf{C}$ to be small. If so, the solution to the linear system is given by the Neumann series expansion

$$\vec{\delta} = \vec{\delta}^{[1]} + \mathbf{C}\vec{\delta}^{[1]} + \mathbf{C}^2\vec{\delta}^{[1]} + \cdots. \tag{33}$$

This is clearly equivalent to Eq. (30).

There are two immediate consequences of the Neumann series expansion:

Corollary 3.1. *For linear problems, given a sufficiently small fixed time-step Δt, the correction iteration in the SDC method using either of the first order correction procedures described by Eq. (14) or (15) is convergent.*

Corollary 3.2. *For linear problems, given a sufficiently small fixed time-step Δ t, each iteration of the correction equation in the SDC method using either of the first order correction procedures described by Eq. (14) or (15) increases the formal order of the method by one order of Δt, provided the order is not greater than that of the underlying quadrature rule.*

The proof of both corollaries follows directly from Eq. (33) and the fact that $\mathbf{C}$ in Eq. (28) is $O(\Delta t)$.

4. GMRES accelerated spectral deferred correction methods

In the previous section, we show that for linear problems, an SDC method may be considered as an iteration scheme for solving the collocation formulation (20) using a preconditioned Neumann series expansion. In this section, we show how this fact can be used to accelerate the convergence of the original SDC method.

4.1. Krylov subspace and generalized minimal residual (GMRES) algorithm

Consider the linear system $Ax = b$ with initial guess $x_0 = 0$ and define the Krylov subspace as

$$\mathbf{K}_m(A, b) = \text{span}\{b, Ab, \ldots, A^m b\}.$$

The generalized minimal residual (GMRES) algorithm works by searching for the "best" solution $x_m \in \mathbf{K}_m(A,b)$ that either makes $r_m \perp \mathbf{K}_m$ or minimizes r_m in some L_2 norm where $r_m = b - A\,x_m$. In general, the convergence of the algorithm depends on the eigenvalue distribution of the matrix A. Rather than simply accepting the solution given by the Neumann series in Eq. (33), our strategy here is to use the GMRES method to compute the "best" value of $\vec{\delta}$ in

$$\text{span}\{\vec{\delta}^{[1]}, \mathbf{C}\vec{\delta}^{[1]}, \mathbf{C}^2\vec{\delta}^{[1]}, \ldots, \mathbf{C}^m\vec{\delta}^{[1]}\}. \tag{34}$$

Note that the memory required for the GMRES method increases linearly with the iteration number k, and the number of multiplications scales like $\frac{1}{2}k^2 n$ where n is the number of unknowns and the size of the matrix A is $n \times n$. When k is chosen to be n, a full orthogonalization cycle is implemented and in theory $b - A\,x_n$ should be close to machine precision. Although accurate, this procedure is expensive and requires excessive memory storage. For practical 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 often denoted as GMRES(k_0). Interested readers are referred to the original paper [29] for further discussions.

4.2. GMRES acceleration for linear problems

For linear problems, consider the preconditioned linear system in Eq. (31). The original SDC approximates this equation using a Neumann series expansion in the matrix $\mathbf{C}$ defined in Eq. (28). Since the matrix $\mathbf{C}$ contains a factor of Δt, if Δt is sufficiently small (and hence the expansion is convergent), each additional term in the expansion produces an additional order of accuracy in the approximation. Note however that when the norm of any eigenvalue of $\mathbf{C}$ is greater than 1, the series expansion is divergent. Also, if the norm is smaller but close to 1, the series expansion will still converge, but will do so slowly. The latter case is the cause of order reduction for stiff problems analyzed in Section 6.1.

It is straightforward to apply GMRES or GMRES(k_0) to the linear system in Eq. (31) and to hence find the optimal solution in the Krylov subspace. In the following discussions, this new numerical technique will be referred to as the GMRES-SDC method.

The GMRES algorithm requires a matrix vector product be computed. In the present context, this requires the evaluation of

$$\left(\mathbf{I} - \Delta t\widetilde{\mathbf{S}}\mathbf{L}\right)^{-1}(\mathbf{I} - \Delta t\mathbf{S}\mathbf{L})\vec{x}_0$$

for any given $\vec{x}_0$. However, applying this operator is equivalent to time marching with either the forward or backward Euler method for the correction equation. The full algorithm is as follows:

Pseudo-code: Matrix vector product algorithm

Comment [Suppose input $\vec{x}_0$ is given.]

(1) Calculate $\vec{\epsilon} = (\mathbf{I} - \Delta t\mathbf{S}\mathbf{L})\vec{x}_0$.

(2a) Use the forward Euler method and solve $(\mathbf{I} - \Delta t\widetilde{\mathbf{S}}\mathbf{L})\vec{y} = \vec{\epsilon}$ where $\Delta t\widetilde{\mathbf{S}}$ is defined in Eq. (24).

(2b) Use the backward Euler method and solve $(\mathbf{I} - \Delta t\widetilde{\mathbf{S}}\mathbf{L})\vec{y} = \vec{\epsilon}$ where $\Delta t\widetilde{\mathbf{S}}$ is defined in Eq. (25).

(3) Output $\vec{y}$.

In this algorithm, the first step is equivalent to evaluating the residual function, and the second step is equivalent to time-stepping the correction equation. Therefore, the amount of work for each matrix vector product in the GMRES-SDC methods is the same as one correction in the original SDC method. However, depending on k_0, GMRES-SDC requires additional work to search the optimal solution in the Krylov subspace. Notice that *no additional function evaluations* are required in this searching process, and so we are expecting minimal efficiency loss due to the use of GMRES. Additional storage is necessary, however, and this could prove to be prohibitive when applying the method to PDEs.

4.3. Nonlinear problems

The GMRES-SDC methods can be applied to nonlinear problems as well. This requires the coupling of Newton iterations with the GMRES-SDC technique. In our current implementation, we use a "linearly implicit" formulation as described in [9]. In this formulation, notice that for small $\delta(t)$, the error Eq. (11) can be approximated by

$$\delta(t) = \int_a^t J_{\varphi^{[0]}}(s, \varphi^{[0]})\delta(s)\,\mathrm{d}s + \epsilon(t) + \mathrm{O}(||\delta||^2), \tag{35}$$

where $J_{\varphi^{[0]}}$ is the Jacobian matrix of the function $F(t,\varphi^{[0]})$ defined as

$$J_{\varphi^{[0]}}(t, \varphi^{[0]}) = \frac{\partial F(t, \varphi^{[0]})}{\partial \varphi}.$$

Discretizing Eq. (35) yields the linear system

$$(I - \Delta t \mathbf{SJ})\vec{\delta} = \vec{\epsilon}, \tag{36}$$

where $\mathbf{J}$ is the tensor form of $J_{\vec{\varphi}^{[0]}}$ which represents the Jacobian matrix at each Gaussian node. Since this equation is of the same type as Eq. (26), it can be solved using the GMRES-SDC methods for linear problems discussed in the previous section. The Jacobian matrix $\mathbf{J}$ is updated after the linear problem is solved to a prescribed precision tol_G, as described by the following:

Pseudo-code: Nonlinear GMRES-SDC method

Comment [Compute initial approximation]
Use the Euler method to compute an approximate solution $\vec{\varphi}^{[0]}$.
Comment [Compute successive corrections.]
while residual $\|\vec{\epsilon}\| > tol$ **do**
(1) Compute the Jacobian matrix $J_{\vec{\varphi}^{[0]}}$.
(2) Use GMRES-SDC for linear problems to solve Eq. (36) to tolerance tol_G.
(3) Update the approximate solution $\vec{\varphi}^{[0]} = \vec{\varphi}^{[0]} + \vec{\delta}$.
end do

Note that an alternative to the above linear implicit algorithm which couples Newton method iterations with GMRES-SDC is to implement the method under the "inexact Newton Methods" framework [22]. This alternative is currently being pursued in the more general case of differential algebraic equations. Results along this direction will be reported in the future.

5. Stability and accuracy analysis

Consider the model problem

$$\begin{aligned} &\varphi'(t) = \lambda \cdot \varphi(t), \qquad t \in [0, 1], \\ &\varphi(0) = 1, \end{aligned} \tag{37}$$

following the terminology in [9], the amplification factor, $Am(\lambda)$, for $\lambda \in \mathbb{C}$ is defined by the formula

$$Am(\lambda) = \widetilde{\varphi}(1), \tag{38}$$

where $\widetilde{\varphi}(1)$ is the numerical solution at $t = 1$ using $\Delta t = 1$. If, for a given value of λ,

$$|Am(\lambda)| \leqslant 1, \tag{39}$$

then the numerical method is said to be stable for that value of λ. When a numerical method is applied to the model problem, the *stability region* is defined to be the subset of the complex plane consisting of all λ such that the amplification factor defined in Eq. (38) satisfies $|Am(\lambda)| \leqslant 1$.

The most interesting stability diagrams are generated by the GMRES-SDC schemes based on the forward (*explicit*) Euler method. In Fig. 1, we show the stability regions for the restarted GMRES(k_0) using 4 Radau

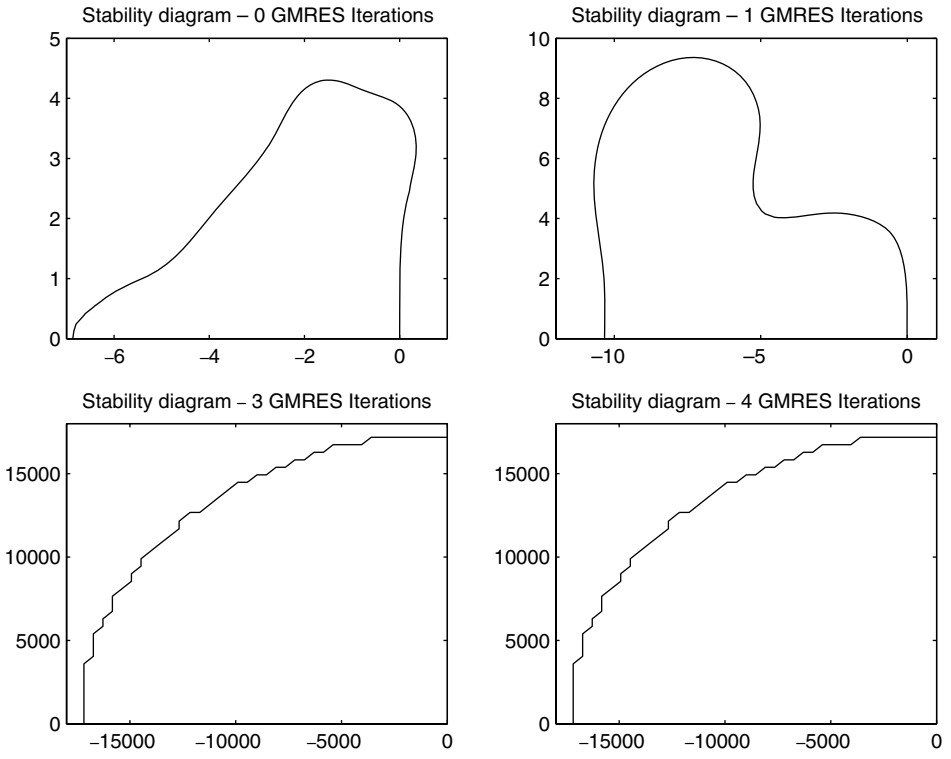

Fig. 1. Stability region of GMRES(k_0), 4 Radau IIa nodes.

IIa nodes. For $k_0 = 0$, this gives the original SDC, and when $k_0 = 4$, GMRES(k_0) is equivalent to the full GMRES which solves the collocation formulation. It can be seen that the stability region of the GMRES-SDC method is much larger than that of the original SDC method. This is not surprising if one considers the preconditioned system (31): Even though the explicit Euler method is a bad preconditioner for λ with large negative real part, the GMRES procedure can still converge to the collocation solution as long as the preconditioning process does not produce numerical overflow. This suggests the possibility of using explicit GMRES-SDC methods for mildly stiff problems. However, we want to mention that when more substeps are used, the explicit Euler based preconditioner is more likely to encounter overflow problems. Hence the stability region will be much smaller. This can be seen in Fig. 2 where 10 Radau IIa nodes are used.

For *implicit* GMRES-SDC methods (using the backward Euler scheme) where the preconditioner is well conditioned, when the full GMRES is performed, the stability regions can be considered the same as those of the corresponding collocation method. A-stability of these methods can be proven in some cases (all collocation methods using the Gaussian points are A-stable), and appears to be true for many others based on numerical results [2]. Our numerical results also show that all the implicit GMRES-SDC methods using Radau IIa nodes we tested (up to machine precision) are A-stable. Further stability and convergence analysis for the GMRES-SDC methods are still being pursued, including the B-stability and B-convergence.

For the original SDC methods, recently, Hagstrom and Zhou showed that when p Gauss nodes are used, after $2p$ corrections, the order of the method is $2p$ [13]. This result can be generalized to the GMRES-SDC methods which solve the collocation formulation as shown by the following theorem. Notice that when GMRES is applied, at most p corrections are necessary for linear scalar problems, compared with $2p$ in [13].

Theorem 5.1. *Using p Gauss nodes, the collocation method which solves Eq. (19) has order 2p.*

Proof. The proof follows closely that of Theorem 1.5 in [14]. Notice that the collocation solution to Eq. (19) at t_{n+1} is derived by spectral integration, which is equivalent to evaluating at t_{n+1} the degree p polynomial $P(t)$ obtained by integrating the degree $p-1$ interpolating polynomial $L^p(\vec{F}, \tau)$ where $\vec{F} = [F(t_1, \varphi_1), \ldots F(t_p, \varphi_p)]^{\mathrm{T}}$, i.e.

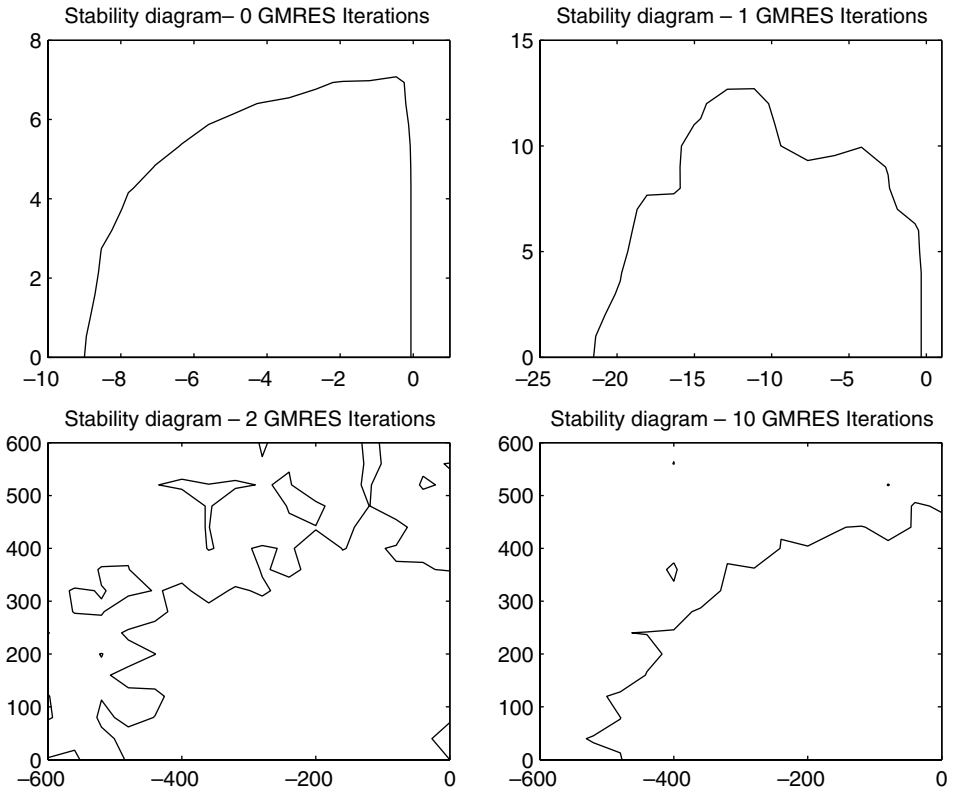

Fig. 2. Stability region of GMRES(k_0), 10 Radau IIa nodes.

$$P(t) = \varphi_0 + \int_{t_0}^{t} L^p(\vec{F}, \tau)\,\mathrm{d}\tau.$$

For this polynomial $P(t)$ it is straightforward to show:

(1) $P(t_0) = \varphi_0$.
(2) $P'(t_i) = F(t_i,\varphi_i)$ at all Gauss nodes (by the definition of $P(t)$).
(3) $P(t_i) = \varphi_i$ for $i = 1,\ldots,p$ (from the collocation formulation).

Therefore, for $t_n < t < t_{n+1}$, the polynomial $P(t)$ satisfies

$$P'(t) = F(t, P(t)) + \sigma(t),$$

where $\sigma(t) = P'(t) - F(t, P(t))$ and satisfies $\sigma(t_i) = 0$ at the Gauss nodes. The error $P(t) - \varphi(t)$ then satisfies

$$P'(t) - \varphi'(t) = F(t, P(t)) - F(t, \varphi(t)) + \sigma(t).$$

Constructing the Taylor expansion of $F(t, P(t))$ at $F(t,\varphi(t))$ yields

$$P'(t) - \varphi'(t) = \frac{\partial F}{\partial \varphi}(t, \varphi(t))(P(t) - \varphi(t)) + \sigma(t) + \mathrm{O}(\|P(t) - \varphi(t)\|^2).$$

As $P(t_0) - \varphi(t_0) = 0$, the solution to this equation is given by the variation of constants formula (see [15])

$$P(t_{n+1}) - \varphi(t_{n+1}) = \int_{t_0}^{t_{n+1}} R(t_{n+1}, \tau)(\sigma(\tau) + \mathrm{O}(\|P(\tau) - \varphi(\tau)\|^2))\,\mathrm{d}\tau,$$

where $R(t,\tau)$ is the Green's function of the corresponding homogeneous differential equation and is smooth for $\tau < t$. Applying the theorem that the local truncation error $P(t) - \varphi(t)$ is at least $O(\Delta t^{p+1})$ (see e.g., p. 29 in [14]), we can neglect the term $O(\|P(t) - \varphi(t)\|^2)$ which is at least $O(\Delta t^{2p+2})$ and derive

$$P(t_{n+1}) - \varphi(t_{n+1}) = \int_{t_0}^{t_{n+1}} R(t_{n+1},\tau)\sigma(\tau)\,d\tau + O(\Delta t^{2p+2}).$$

Since Gauss quadrature is applied to the integral and $\sigma(t_i) = 0$ at the Gauss nodes, the collocation solution $P(t)$ has the same order as the underlying quadrature formula. When Gauss–Legendre nodes are used, the order of the local truncation error is therefore $2p + 1$. The same proof can be applied to show that when Radau IIa nodes are used, the local truncation error is $O(\Delta t^{2p})$. □

6. Numerical experiments

In this section, we show some preliminary numerical results for both linear and nonlinear problems. Depending on the stiffness of the problem, we present results for both the explicit and implicit GMRES-SDC methods.

6.1. The cosine problem

For the first numerical example, define $p(t) = \cos(t)$ and consider

$$\varphi'(t) = p'(t) - \frac{1}{\varepsilon}(\varphi(t) - p(t)), \quad t \in [0, t_{\text{final}}],$$
$$\varphi(0) = p(0).$$

The exact solution is clearly $\varphi(t) = p(t)$. Notice that when ε is small, this problem is stiff, however, the solution itself is smooth and independent of ε.

For the first example, we set $\varepsilon = 0.02$ and $\Delta t = 1$. For each time step, we use 12 Radau IIa nodes. For the time-stepping we use the explicit Euler method in Eq. (14). In Table 1, we show the numerical error after one step ($\Delta t = t_{\text{final}} = 1$) for different GMRES($k_0$). For $k_0 = 0$, the method is the original SDC. Also, for each step, we fix the number of explicit Euler corrections to 12. The total number of function evaluations is therefore fixed to 12×12.

These results are consistent with the stability analysis in Section 5. Clearly, the GMRES-SDC methods give better numerical results even though the original SDC method is unstable. Also, for the restarted GMRES(k_0), keeping more data in memory (larger k_0) reduces the error. The full orthogonalization process ($k_0 = 12$, the same as the number of unknowns) returns converged numerical results but loses a few digits in accuracy due to the fact that the forward Euler predictor is actually unstable here (see also Fig. 3).

Next, consider the case $\varepsilon = 10^{-6}$. As the problem is very stiff, the implicit GMRES-SDC method is used. Note that for this example, the original SDC method is stable. As in the explicit examples, the results shown in Table 2 demonstrate that increasing k_0 reduces the error and residual (defined as $b - Ax$ when solving $Ax = b$) and that both go to machine precision with the full GMRES.

Note that in both the explicit and implicit examples, the error first decays slowly as a function of k_0, and then suddenly decreases to close to machine precision once k_0 is the same as the number of nodes p. The convergence of the GMRES procedure depends in general on the distribution of the eigenvalues of the matrix being considered. In the present context, the eigenvalues of the matrix $\mathbf{C}$ defined in Eq. (28) are of interest, and these in turn depend on the matrix $S - \widetilde{S}$. The eigenvalues of $\mathbf{C}$ for both the explicit and implicit cases

Table 1
Errors versus k_0 for the cosine problem for the explicit GMRES-SDC method

k_0	0	1	2	3	4	6	12
Error	4.2e + 57	4.6e − 1	3.8e − 3	2.1e − 3	9.2e − 4	1.7e − 4	3.6e − 13

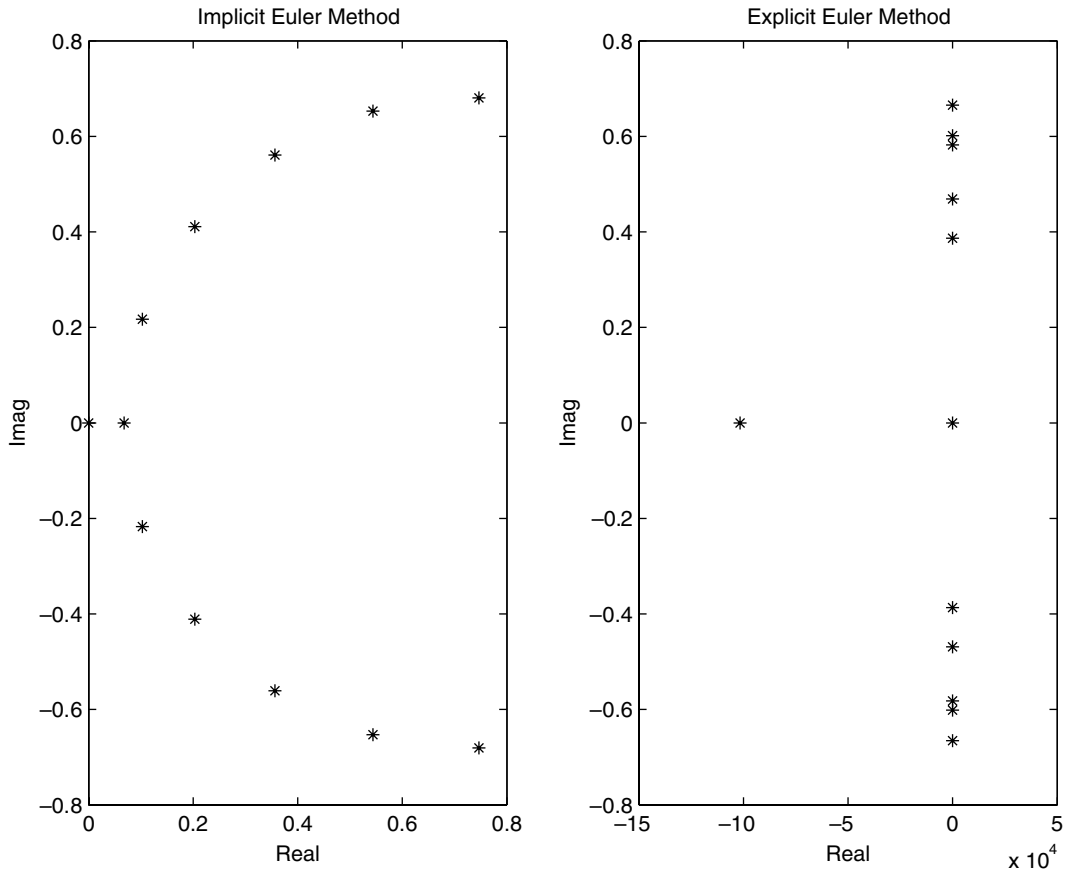

Fig. 3. Eigenvalue distribution of **C** for both the implicit and explicit method. Note that the axis in the right panel are scaled by 10^4.

Table 2
Errors and residuals versus k_0 for the cosine problem for the implicit GMRES-SDC method

k_0	0	1	2	3	4	6	12
Error	1.4e − 4	3.6e − 4	1.6e − 4	5.5e − 5	3.3e − 5	1.2e − 5	4.4e − 16
Residual	2.3e − 4	2.9e − 4	1.6e − 4	8.1e − 5	4.8e − 5	1.6e − 5	2.8e − 16

above are shown in Fig. 3. The eigenvalues in the implicit case are smaller by about four orders of magnitude than in the explicit case, but in neither case are the eigenvalues clustered about a single point.

6.1.1. Order reduction

In [24], when the original SDC method is applied to stiff problems and the number of corrections for each step is fixed, the effective order of accuracy is reduced for values of the time step size in a certain range. This type of order reduction is also present in many popular types of Runge–Kutta methods [7,8,30]. The implication of order reduction is that, although the methods are stable for larger time steps, one must use a very small time step, or increase the number of SDC corrections for the method to converge with full order. However, with the GMRES-SDC methods, when full orthogonalization is used, order reduction is no longer observed. In Fig. 4, convergence results are presented for both the original SDC and the new implicit GMRES-SDC methods for different ε and step size selections. In the calculation, 10 Radau IIa nodes are used, and 10 iterations are performed. The order reduction phenomenon can be easily observed when ε is small (curves on the left). The plots also indicate the benefit in terms of computational cost the GMRES acceleration provides. For example, when $\varepsilon = 10^{-5}$, in order to have 13 digits of accuracy, the original SDC requires a step size of

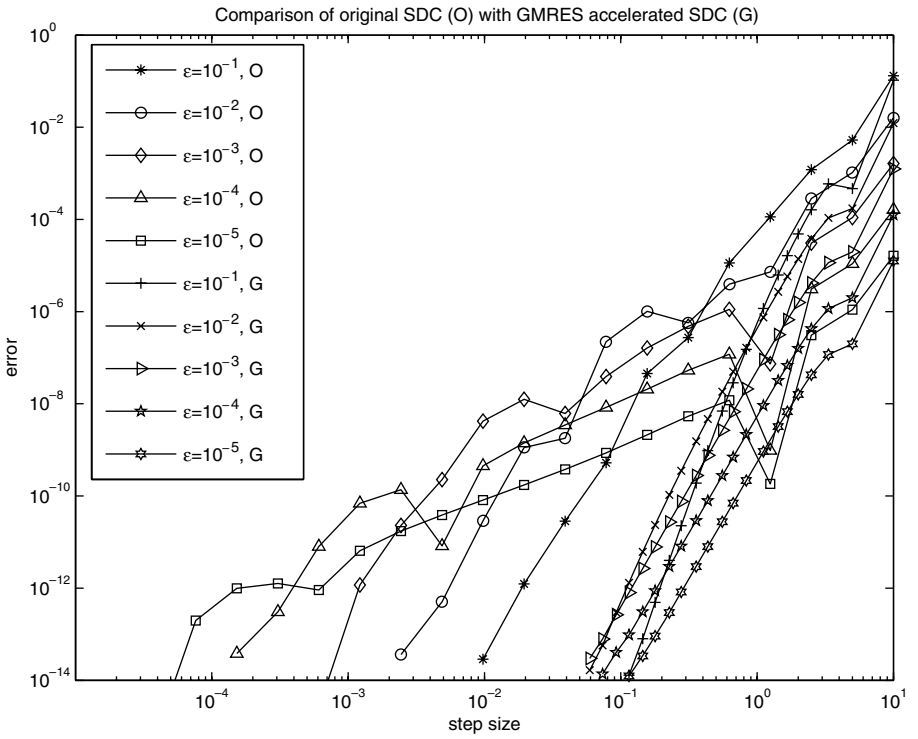

Fig. 4. Order reduction: the original SDC and the GMRES-SDC.

approximately 10^{-5}. For the GMRES-SDC method with full orthogonalization, the necessary step size is approximately 0.1, or 4 orders of magnitude greater.

6.2. The linear multiple mode problem

As mentioned above, the convergence of a GMRES-SDC method will depend on the eigenvalues of the matrix **C** in Eq. (28), which depend also on the eigenvalues of the linear operator **L**. Hence in our second set of tests, we study an ODE system similar to the cosine problem in which we can specify the distribution of the eigenvalues. When GMRES is applied to the original SDC, it is usually expensive to use the full orthogonalization process since it would require $k_0 = pN$ iterations for a system of N ODEs using p nodes. This increases both the memory required and the amount of work performed. Therefore a natural question is, given some information on the distribution of the eigenvalues, can we determine the "optimal" number k_0? The following numerical experiments are intended to provide some basic guidelines.

The problem studied in this example is

$$\bar{y}'(t) = \bar{p}'(t) - B(\bar{y}(t) - \bar{p}(t)),$$
$$\bar{y}(0) = \bar{p}(0),$$

where $\bar{y}(t)$ and $\bar{p}(t)$ are vectors of dimension N. The exact solution is again $\bar{y}(t) = \bar{p}(t)$. The matrix B is constructed by

$$B = U^{\mathrm{T}} \Lambda U,$$

where U is a randomly generated orthogonal matrix, and Λ is a diagonal matrix whose diagonal entries $\{\lambda_i\}_{i=1}^{N}$ are all positive. For $\bar{p}(t)$, we choose the ith component as $\cos(t + \alpha_i)$ with phase parameter $\alpha_i = 2\pi i/N$.

In our first experiment, we set the dimension of the system to 10, and use 10 Radau IIa nodes in the simulation. We use $\Delta t = 0.1$ and study one time step (i.e., $t_{\mathrm{final}} = \Delta t$). In the left of Fig. 5, we set $\lambda_1 = 10^7$, and all other λ_i to 1. It can be seen that when $k_0 \approx 12$, the residual converges to machine precision in about 25 iterations. Notice

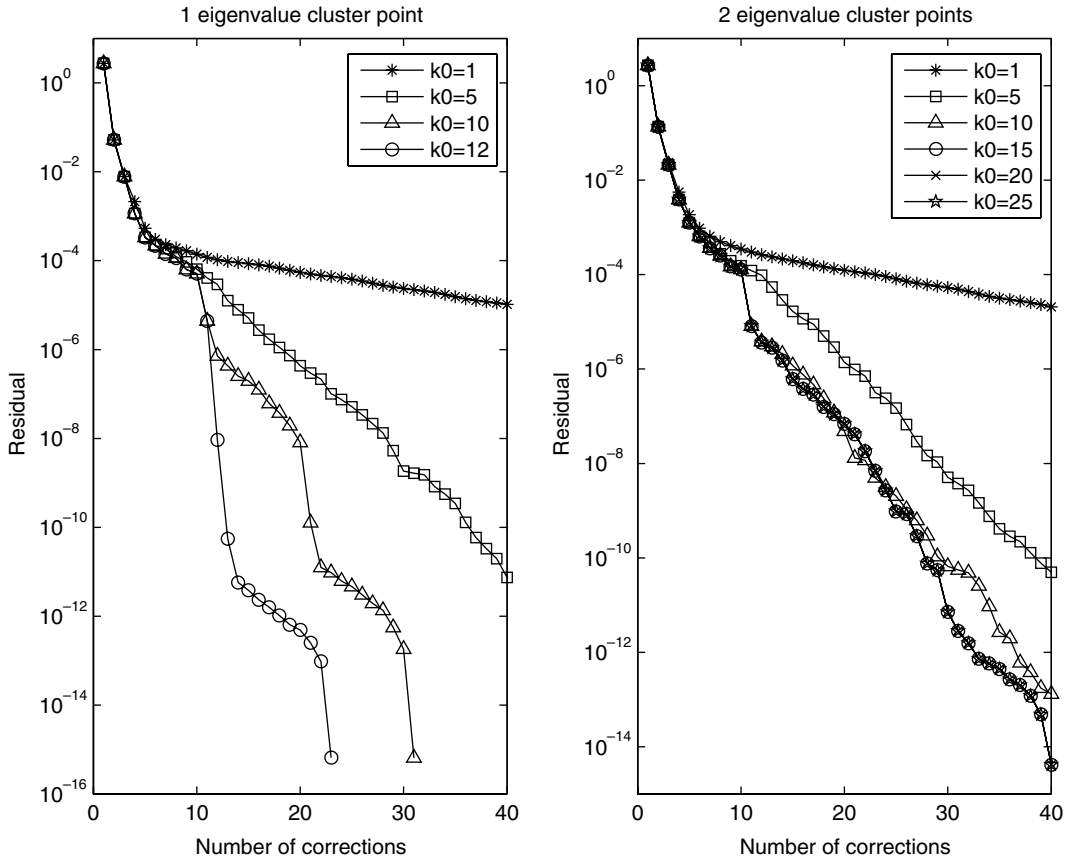

Fig. 5. Comparison of errors for different k_0.

that 10 Radau IIa nodes resolve the solution to 14 digits, therefore the residual is equivalent to the error (up to a constant factor). In the right panel, results are shown for the case when two eigenvalues are 10^7, two are 10^4, and the rest are 1. In this case, more iterations are required to reduce the residual to machine precision, and it requires a slightly higher k_0 of approximately 15 to yield the best convergence results (i.e. results for $k_0 = 15$ are almost identical to those using larger k_0). However, these values of k_0 are much smaller compared with the full GMRES which requires $k_0 = 100$. Since the original SDC method would require ten iterations of the correction equation, in this case, there is a factor of 4 increase in the number of iterations for GMRES-SDC method, however, this results in a reduction in the error of approximately 10 orders of magnitude.

In our second experiment, we consider the case where $N = 100$ and the $\log_{10}$ of the eigenvalues are uniformly distributed on $[0, 7]$. For 10 Radau IIa nodes, numerical results for different k_0 are shown in Fig. 6. In this example, convergence profiles for $k_0 > 10$ are very similar. Notice that the full GMRES requires $k_0 = 1000$, hence only a small fraction of the full method is required for machine precision. At present, the optimal strategy for picking k_0 for a given problem is not completely understood, although these experiments suggest that a successful strategy must depend on the time step, the size of the system, the distribution of the eigenvalues, and of course any memory restrictions based on the problem size. The authors are currently investigating strategies for choosing k_0 in the broader context of step size selection.

6.3. The Van der Pol oscillator

In our third example, we consider the nonlinear ODE initial value problem which describes the behavior of vacuum tube circuits. It was proposed by B. Van der Pol in the 1920's, and is often referred to as the Van der Pol oscillator. As a first order ODE system, the problem takes the form

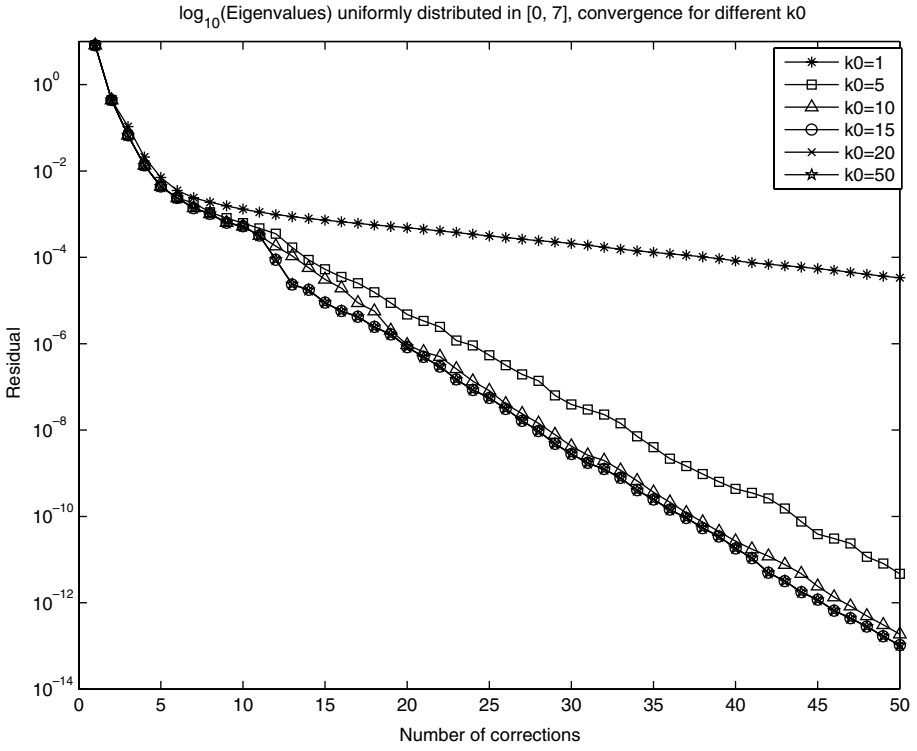

Fig. 6. Comparison of errors for different k_0.

$$\begin{cases} y_1'(t) &= y_2(t), \\ y_2'(t) &= (1 - y_1^2(t)y_2(t) - y_1(t))/\varepsilon, \end{cases} \tag{40}$$

where the initial values are given by $[y(0),y'(0)] = [2,-0.6666654321121172]$. This is a stiff system when ε is small. For this nonlinear problem, we use the "linear implicit" GMRES-SDC methods discussed in Section 4, and choose the following strategy in the implementation: GMRES(k_0) is applied to the linearized system until the residual $b-Ax$ is reduced by a factor of tol_G; once this is done, we update the Jacobian matrix and restart GMRES(k_0).

As our first experiment, we set $\varepsilon = 10^{-3}$, and use $\Delta t = 0.001$. We apply the *explicit* GMRES-SDC method, and in Fig. 7, we show how the residual decays (as the analytical solution is not readily available) for different k_0 when $tol_G = 0.01$ (left) and $tol_G = 0.1$ (right). Here, the residual is defined as the error $\|b-Ax\|$ for the linearized system in Eq. (36). It can be seen that the GMRES-SDC method converges quickly to the solution of the collocation formulation. This is consistent with the stability analysis in Section 5 and the linear cosine test problems in Section 6.1. Notice that for this problem, the original SDC method is divergent (not shown on plot), and GMRES(1) converges very slowly.

Because of the nonlinearity of the problem, the convergence behavior of the GMRES-SDC method also depends on tol_G. The two panels in Fig. 7 compare convergence for $tol_G = 0.01$ and $tol_G = 0.1$. In the left panel, it appears that using $k_0 = 10$ is sufficient for achieving the best convergence results since the convergence for $k_0 = 15$ is nearly identical. In the right panel, convergence for $k_0 = 5$ is the same as for $k_0 = 10$ and $k_0 = 15$, although the overall number of iterations required to achieve a specified error tolerance increases slightly compared to $tol_G = 0.01$. Determining the "optimal" choice of tol_G or an adaptive strategy for choosing tol_G is an open issue.

Next we apply the implicit GMRES-SDC method to the very stiff case with $\varepsilon = 10^{-8}$ and $\Delta t = 0.5$. The results are shown in Fig. 8 for different choices of k_0 and $tol_G = 0.1$. In all cases, the GMRES-SDC methods converge more rapidly to the collocation solution than the original SDC method.

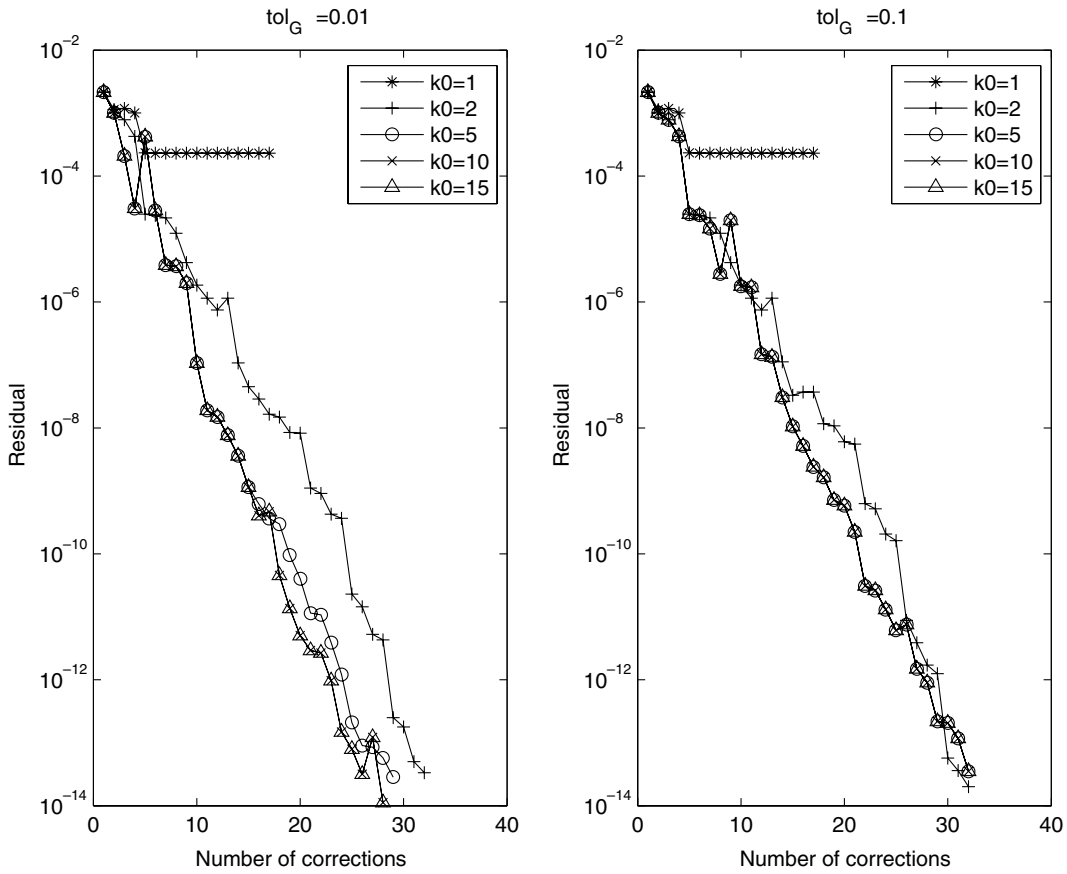

Fig. 7. Comparison of different k_0 for the Van der Pol problem, explicit method.

It is possible to apply a different numerical method for the time marching of the correction equation. In the above examples, either the explicit or implicit Euler method is used for the correction equation. It is reasonable to expect that the use of a higher-order numerical method for the time marching of the correction equation would result in a method which requires fewer iterations of the correction equation to converge to a specified tolerance. In the linear case, this is equivalent to choosing a different preconditioner for the Neumann series expansion. We investigate this idea by repeating the above numerical example using the trapezoid rule. The results are compared with those from the implicit Euler method in Fig. 9 for $k_0 = 1, 2$ and 10. From this figure, we can see that using larger k_0 again improves the numerical convergence in both cases. However, when $k_0 = 10$, the trapezoid rule results are not significantly better than those computed with the first-order method. Hence, at least for this limited experiment, using a higher-order marching method does not seem to have a significant effect on the convergence when the GMRES acceleration procedure is used (see also [19]).

6.4. The nonlinear multi-mode problem

In this example, we study a nonlinear generalization of the multi-mode example in Section 6.2. The problem is given by a system of N nonlinear equations

$$\begin{cases} y_i'(t) &= p_i'(t) - \lambda_i y_{i+1}(t)(y_i(t) - p_i(t)), \quad 1 < i < N-1, \\ y_N'(t) &= p_N'(t) - \lambda_i (y_i(t) - p_i(t)), \quad i = N. \end{cases}$$

The analytical solution is again $\bar{y}(t) = \bar{p}(t)$ where the ith component of $\bar{p}(t)$ is given by $p_i(t) = 2 + \cos(t + \alpha_i)$ with phase parameter $\alpha_i = 2\pi i/N$. In our first experiment, we set $N = 7$ and the eigenvalues are chosen as

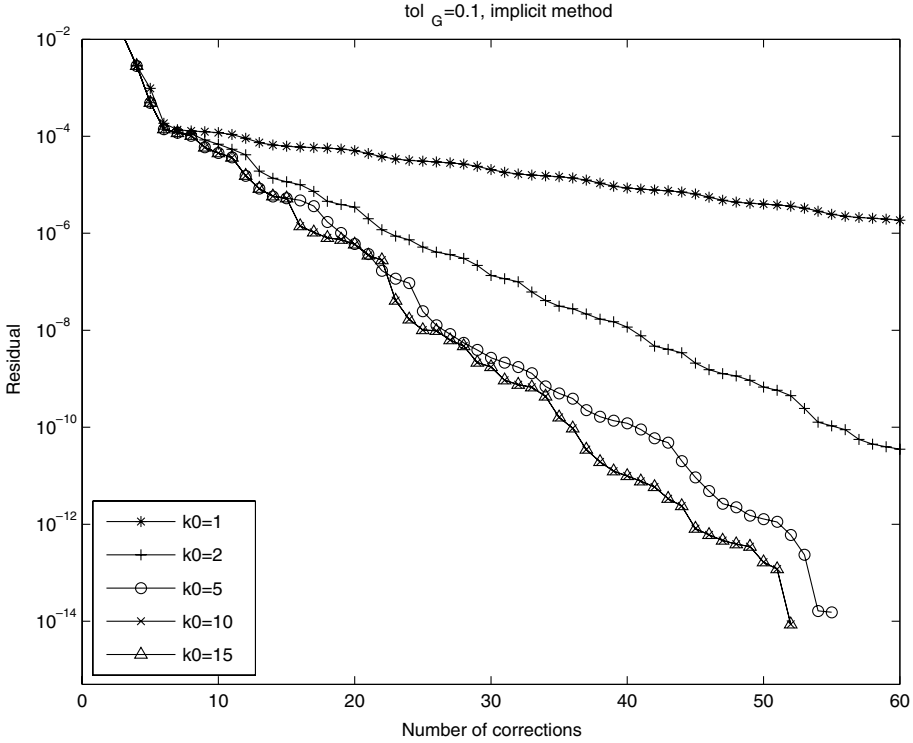

Fig. 8. Comparison of different k_0 for the very stiff Van der Pol problem with implicit method.

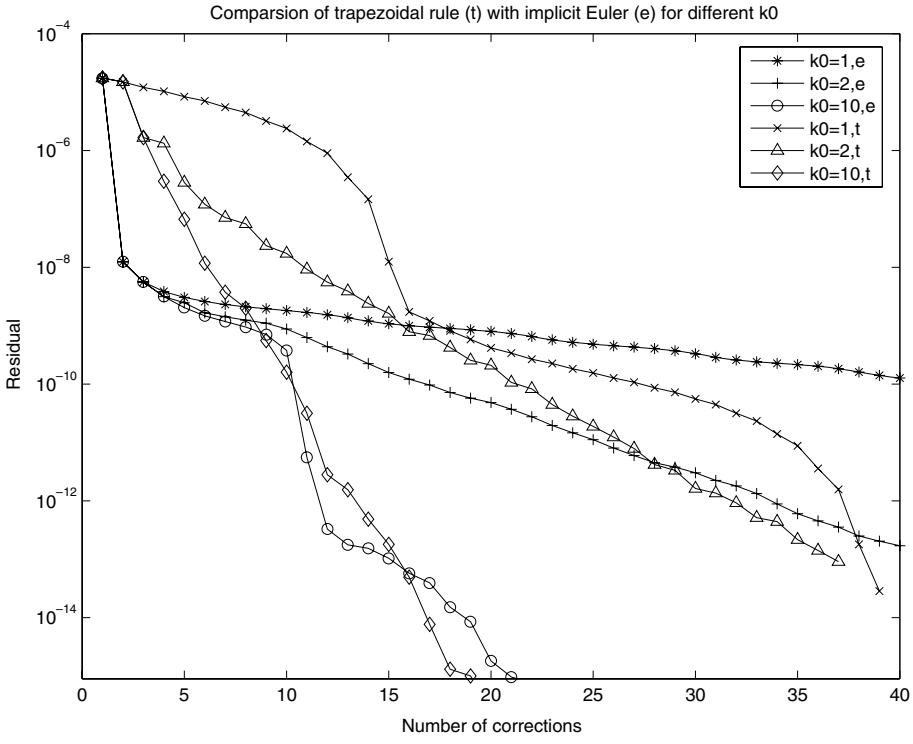

Fig. 9. Comparison using trapezoidal rule versus the implicit Euler method.

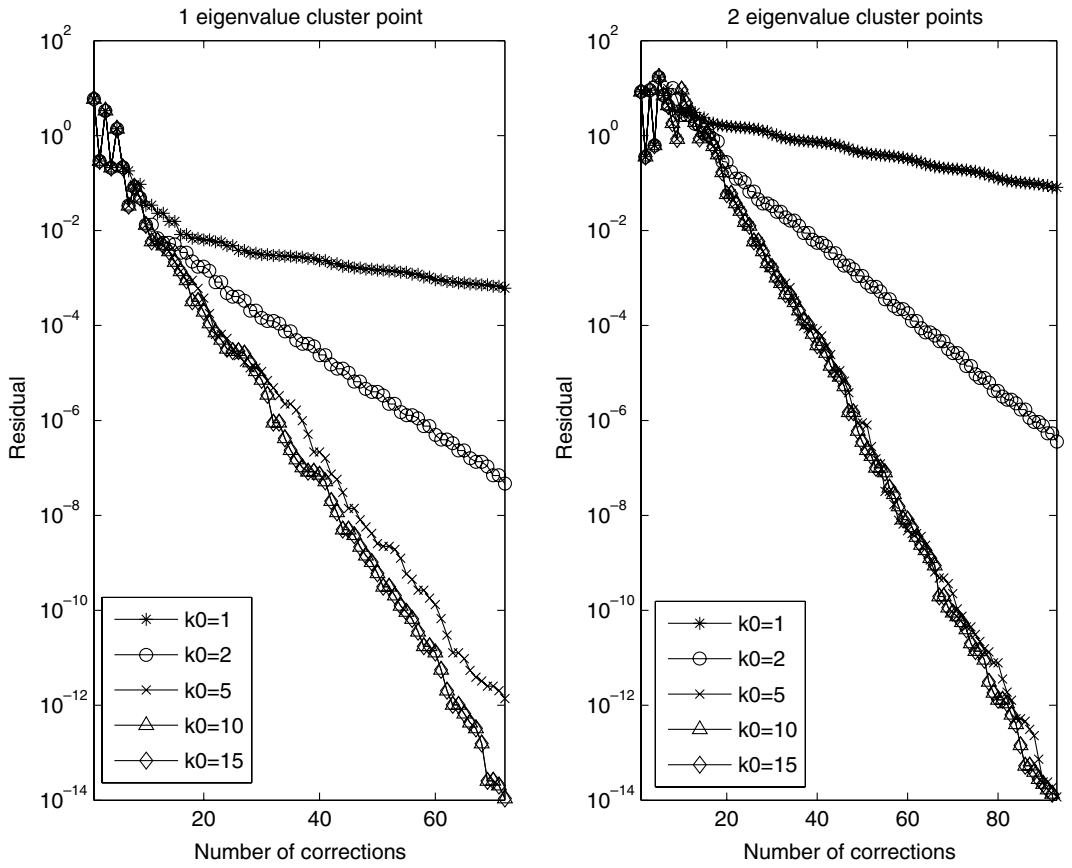

Fig. 10. Nonlinear multi-mode convergence.

$[10^8,10^8,1,1,1,1,1]$. In the left of Fig. 10, as in the first linear multi-mode test, we show how the residual decays in one time step for different k_0 where $tol_G = 10^{-1}$. In the simulation, we use the implicit GMRES-SDC method with $p = 10$ Radau IIa nodes and Δt is chosen to be 0.3. It can be seen that the linear implicit GMRES-SDC greatly improves the convergence of the SDC procedure. Also, when $k_0 > p$, the convergence of the method is very satisfactory. In our second experiment, we choose the eigenvalues as $[10^8, 10^8, 10^5, 10^5, 1, 1, 1]$ so that there are two eigenvalue cluster points away from 1 as in the second linear multi-mode example. The right panel of Fig. 10 shows that somewhat more corrections are required for convergence in this case; however, the minimum k_0 required for reasonable performance does not increase over the case with only one cluster. In both cases, $k_0 = 5$ now gives convergence behavior very similar to using larger k_0.

6.5. The ring modulator problem

In our last example, we consider a stiff nonlinear ODE system of 15 equations. The problem originates from electrical circuit analysis. Specifically, it describes the behavior of the ring modulator [1], and takes the form

$$\frac{\mathrm{d}\vec{y}}{\mathrm{d}t} = \vec{f}(t,\vec{y}), \quad \vec{y} = \vec{y}_0,$$

with

$$\vec{y} \in \mathbb{R}^{15}, \quad 0 \leqslant t \leqslant 10^{-5}.$$

In this equation, the function $\vec{f}$ is defined by

$$\vec{f}(t,\vec{y}) = \begin{pmatrix} C^{-1}(y_8 - 0.5y_{10} + 0.5y_{11} + y_{14} - R^{-1}y_1) \\ C^{-1}(y_9 - 0.5y_{12} + 0.5y_{13} + y_{15} - R^{-1}y_2) \\ C_s^{-1}(y_{10} - q(U_{D1}) + q(U_{D4})) \\ -C_s^{-1}(y_{11} - q(U_{D2}) + q(U_{D3})) \\ C_s^{-1}(y_{12} + q(U_{D1}) - q(U_{D3})) \\ -C_s^{-1}(y_{13} + q(U_{D2}) - q(U_{D4})) \\ C_p^{-1}(-R_p^{-1}y_7 + q(U_{D1}) + q(U_{D2}) - q(U_{D3}) - q(U_{D4})) \\ -L_h^{-1}y_1 \\ -L_h^{-1}y_2 \\ L_{s2}^{-1}(0.5y_1 - y_3 - R_{g2}y_{10}) \\ -L_{s3}^{-1}(0.5y_1 - y_4 + R_{g3}y_{11}) \\ L_{s2}^{-1}(0.5y_2 - y_5 - R_{g2}y_{12}) \\ -L_{s3}^{-1}(0.5y_2 - y_6 + R_{g3}y_{13}) \\ L_{s1}^{-1}(-y_1 + U_{\text{in1}}(t) - (R_i + R_{g4})y_{14}) \\ L_{s1}^{-1}(-y_2 - (R_c + R_{g1})y_{15}) \end{pmatrix}.$$

The auxiliary functions $U_{D1}, U_{D2}, U_{D3}, U_{D4}, q, U_{\text{in1}}$ and U_{in2} are given by

$$\begin{aligned} U_{D1} &= y_3 - y_5 - y_7 - U_{\text{in2}}(t), \\ U_{D2} &= -y_4 + y_6 - y_7 - U_{\text{in2}}(t), \\ U_{D3} &= y_4 + y_5 - y_7 + U_{\text{in2}}(t), \\ U_{D4} &= -y_3 - y_6 + y_7 + U_{\text{in2}}(t), \\ q(U) &= \gamma(\mathrm{e}^{\delta U} - 1), \\ U_{\text{in1}}(t) &= 0.5\sin(2000\pi t), \\ U_{\text{in2}}(t) &= 2\sin(2000\pi t). \end{aligned}$$

The values of the parameters are

$C = 1.6 \cdot 10^{-8}$	$R = 25000$
$C_s = 2 \times 10^{-12}$	$R_i = 50$
$C_p = 10^{-8}$	$R_p = 50$
$L_h = 4.45$	$L_c = 600$
$L_{s1} = 0.002$	$R_{g1} = 36.3$
$L_{s2} = 5 \times 10^{-4}$	$R_{g2} = 17.3$
$L_{s3} = 5 \times 10^{-4}$	$R_{g3} = 17.3$
$\gamma = 40.67286402 \times 10^{-9}$	$\delta = 17.7493332$

and the initial value $\vec{y}_0$ is given by

$$\vec{y}_0 = \vec{0}.$$

In the simulation, we use the implicit GMRES-SDC method with $p = 7$ Radau IIa nodes. We set $tol_G = 0.1$, $k_0 = p + 1$, and $t_{\text{final}} = 10^{-5}$. Our *uniform step* GMRES-SDC method is then compared with available *adaptive* ODE packages described in [1] and the results are shown in Table 3. In the table, the parameters *r*tol, *atol* and h_0 for each method are chosen experimentally to produce a numerical solution with at least 9 significant digits, which has the fewest possible number of function evaluations.

Our results suggest that the new GMRES-SDC method is a very competitive alternative to existing ODE solvers. However, in order to perform more extensive (and convincing) tests, an automatic step-size selection

Table 3
Performance comparison of different solvers

	G-SDC	DASSL	GAMD	MEBDFI	PSIDE	RADAU	VODE
Rtol	1e − 8	1e − 12	1e − 10	1e − 9	1e − 10	1e − 9	1e − 11
Atol	*	1e − 12	1e − 10	1e − 11	1e − 11	1e − 10	1e − 14
h_0	2.5e − 6	*	1e − 10	1e − 10	*	1e − 10	*
Rerr	3.0e − 9	1.1e − 9	3.1e − 9	2.9e − 9	2.1e − 9	2.1e − 9	1.3e − 9
F	1134	2104	4057	2284	3417	2172	2961
Steps	4	1591	76	669	154	47	2277

*, not needed; rtol, relative tolerance; atol, absolute tolerance; h_0, initial step-size; rerr, maximum relative error; F, number of function evaluations; steps, number of steps taken.

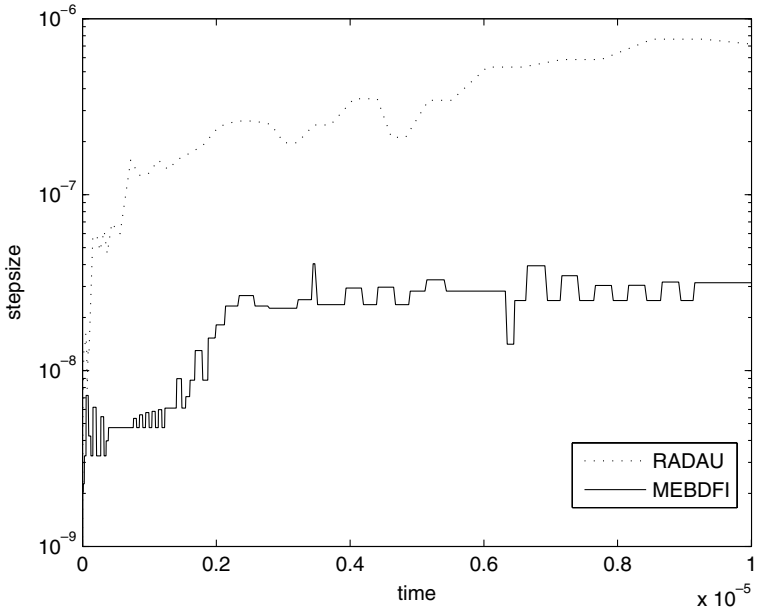

Fig. 11. Step-sizes selected by RADAU and MEBDFI.

strategy is required for the GMRES-SDC method. In Fig. 11, we show the step-sizes used by the adaptive solvers MEBDFI and RADAU. Currently, we are studying strategies for step-size selections along with strategies for computing better initial provisional solutions, for adaptively choosing the parameters k_0 and tol_G, and for adaptively varying the order of the SDC method. Progress will be reported in the future.

Finally for this section, we want to mention that we have also studied several other problems from the Test Set for IVP solvers [1]. In all cases, the convergence of the original SDC methods is greatly improved by the GMRES-SDC procedure.

7. Conclusions

In this paper, a matrix based analysis of the original SDC method for linear problems shows that the iterated corrections are equivalent to a preconditioned Neumann series expansion. By introducing the Krylov subspace based GMRES method, we show how the convergence of the original SDC can be accelerated. Preliminary analytical and numerical results show that the stability and accuracy of this new class of methods are greatly improved for both linear and nonlinear problems compared to the original method.

In order to fully explore the efficiency of the new accelerated SDC methods, a direct comparison with existing methods on standard test problems needs to be carried out. This requires that a variable time step selection algorithm be implemented with the GMRES-SDC method. The authors have been investigating strategies for

varying both the time step and the order of the method to optimally achieve a desired error tolerance. The problem of developing a robust algorithm is further complicated by the fact that the performance of the GMRES acceleration depends on the size of the time step, the size of the system, the stiffness of the equation, and on the restart parameter k_0. We have also investigated more effective methods of coupling the GMRES process with Newton's method for the implicit version on nonlinear problems. Results along these lines will be presented in the future.

One case where it is clear that the GMRES acceleration is advantageous is for ODE initial value problems where the stiffness is caused by only a few eigenvalues with large negative real parts. For this case, the reduction to first order accuracy for a range of time step size which is observed for the original SDC methods is effectively eliminated in the tests presented in Section 6.1. The analysis here clearly shows that order reduction is equivalent to the slow decay of the Neumann series expansion derived in Section 3.

Several other extensions of the accelerated method are also being pursued. One advantageous feature of SDC methods in general is that semi- and multi-implicit versions of the method for problems with more than one stiff time scale have been developed [6,26]. Acceleration of these methods is also being pursued. It should be noted that for PDE applications (for which the semi- and multi-implicit methods were developed), the size of the linear system that GMRES is being applied to in the current methods may be very large, and hence memory restrictions may require that a small restart parameter k_0 be used. More numerical tests need to be conducted in such cases to determine the benefits of the GMRES acceleration.

Finally, a generalization of the GMRES-SDC method has been applied to differential algebraic equations. Initial numerical results are very promising, and a paper reporting on these results is in preparation [21].

Acknowledgements

The work of J.H. and J.J. was supported in part by the NSF under Grants DMS0411920 and DMS0327896. M.M. 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.

References

[1] http://pitagora.dm.uniba.it/~testset/.
[2] R. Barrio, On the A-stability of Runge–Kutta collocation methods based on orthogonal polynomials, SIAM J. Numer. Anal. 36 (4) (1999) 1291–1303.
[3] K.E. Brenan, S.L. Campbell, L.R. Petzold, Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations, SIAM, Philadelphia, 1995.
[4] J. Butcher, The Numerical Analysis of Ordinary Differential Equations: Runge–Kutta and General Linear Methods, Wiley, 1987.
[5] K. Böhmer, H.J. Stetter (Eds.), Defect Correction Methods, Theory and Applications, Springer-Verlag, Wien-New York, 1984.
[6] 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.
[7] 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.
[8] K. Dekker, J.G. Verwer, in: Stability of Runge–Kutta Methods for Stiff Nonlinear Differential EquationsCWI Monographs, North-Holland, 1984.
[9] A. Dutt, L. Greengard, V. Rokhlin, Spectral deferred correction methods for ordinary differential equations, BIT 40 (2) (2000) 241–266.
[10] C.W. Gear, Numerical Initial Value Problems in Ordinary Differential Equations, Prentice-Hall, New Jersey, 1971.
[11] D. Gottlieb, S.A. Orszag, Numerical Analysis of Spectral Methods, SIAM, Philadelphia, 1977.
[12] L. Greengard, Spectral integration and two-point boundary value problems, SIAM J. Numer. Anal. 28 (1991) 1071–1080.
[13] T. Hagstrom, R. Zhou, On the spectral deferred correction of splitting methods for initial value problems, CAMCos (submitted).
[14] E. Hairer, C. Lubich, G. Wanner, Geometric Numerical Integration: Structur-Preserving Algorithms for Ordinary Differential Equaitons, Springer-Verlag, Berlin, 2002.
[15] E. Hairer, S.P. Norsett, G. Wanner, Solving Ordinary Differential Equations I, Non-Stiff Problems, Springer-Verlag, Berlin, 1993.
[16] E. Hairer, G. Wanner, Solving Ordinary Differential Equations II, Springer-Verlag, Berlin, 1996.
[17] E. Hairer, C. Lubich, M. Roche, The Numerical Solution of Differential-Algebraic Systems by Runge–Kutta Methods, Springer-Verlag, Berlin, 1989.
[18] A.C. Hansen, J. Strain, On the order of deferred correction, BIT (submitted).

[19] A.C. Hansen, J. Strain, Convergence theory for spectral deferred correction, (in press).
[20] D.J. Higham, L.N. Trefethen, Stiffness of ODEs BIT 33 (1993) 285–303.
[21] J. Huang, J. Jia, M.L. Minion, Arbitrary order Krylov deferred correction methods for differential algebraic equations, Prog. Appl. Math. Preprint Series, University of North Carolina, PAMPS 2005-01.
[22] C.T. Kelly, Solving Nonlinear Equations with Newton's Method, SIAM, 2003.
[23] J.D. Lambert, Numerical Methods for Ordinary Differential Equations, Wiley, Berlin, 1991.
[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 (2005) 341–373.
[25] B. Lindberg, Error estimation and iterative improvement for discretization algorithms, BIT 20 (1980) 486–500.
[26] M.L. Minion, Semi-implicit spectral deferred correction methods for ordinary differential equations, CMS 1 (2003) 471–500.
[27] M.L. Minion, Semi-implicit projection methods for incompressible flow based on spectral deferred corrections, APNUM 48 (3–4) (2004) 369–387.
[28] V. Pereyra, Iterated deferred correction for nonlinear boundary value problems, Numer. Math. 11 (1968) 111–125.
[29] Y. Saad, M.H. Schultz, GMRES: a generalized minimal residual algorithm for solving non-symmetric linear systems, SIAM J. Sci. Stat. Comp. 7 (1986) 856–869.
[30] 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.
[31] R.D. Skeel, A theoretical framework for proving accuracy results for deferred correction, SIAM J. Numer. Anal. 19 (1981) 171–196.
[32] J. Stoer, R. Bulirsch, Introduction to Numerical Analysis, Springer, Berlin, 1992.
[33] L.N. Trefethen, M.R. Trummer, An instability phenomenon in spectral methods, SIAM J. Numer. Anal. 24 (1987) 9.
[34] 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, vol. 25, 1964.
[35] P.E. Zadunaisky, On the estimation of errors propagated in the numerical integration of ordinary differential equations, Numer. Math. 27 (1976) 21–40.