SEMI-IMPLICIT KRYLOV DEFERRED CORRECTION METHODS FOR DIFFERENTIAL ALGEBRAIC EQUATIONS

By

Sunyoung Bu

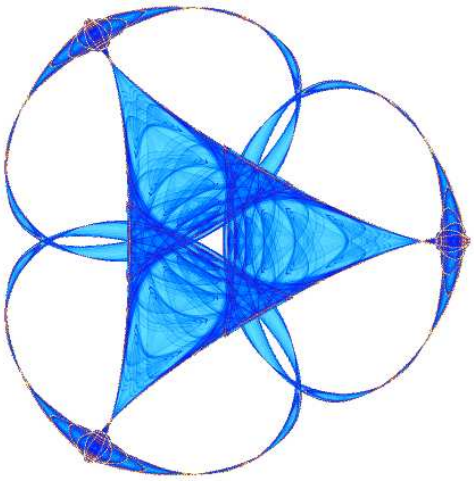
Jingfang Huang

and

Michael L. Minion

IMA Preprint Series # 2275

(August 2009)



INSTITUTE FOR MATHEMATICS AND ITS APPLICATIONS

UNIVERSITY OF MINNESOTA

400 Lind Hall

207 Church Street S.E.

Minneapolis, Minnesota 55455–0436

Phone: 612/624-6066 Fax: 612/626-7370

URL: http://www.ima.umn.edu

SEMI-IMPLICIT KRYLOV DEFERRED CORRECTION METHODS FOR DIFFERENTIAL ALGEBRAIC EQUATIONS*

SUNYOUNG BU, JINGFANG HUANG, AND MICHAEL L. MINION†

Abstract. In the recently developed Krylov deferred correction (KDC) methods for differential algebraic equation initial value problems [31], a Picard-type collocation formulation is preconditioned using low-order time integration schemes based on spectral deferred correction (SDC), and the resulting system is solved efficiently using Newton-Krylov methods. In this paper, we further improve the efficiency of these KDC methods by introducing the semi-implicit KDC (SI-KDC) methods, in which the stiff component of the preconditioner is solved by implicit schemes and the non-stiff parts by explicit methods. Compared with fully implicit KDC (FI-KDC) methods, preliminary analyses show that the convergence of Newton-Krylov iterations in the SI-KDC methods is similar to that in FI-KDC, while for systems with a nonlinear non-stiff component and a linear stiff part, the SI-KDC can greatly reduce the computational cost in each spectral deferred correction iteration for the same accuracy requirement, as only linear solves are required in each SI-KDC iteration. The analyses are validated by preliminary numerical results.

Key words. Differential Algebraic Equations, Krylov Deferred Correction, Semi-Implicit Schemes, Preconditioner

AMS subject classifications. 65B10, 65F10, 65L20, 65L80, 65N35.

1. Introduction. In the last century, many numerical techniques have been developed for the accurate and efficient solution of stiff ordinary and partial differential equation initial value problems with algebraic constraints, including the linear multi-step methods, Runge-Kutta methods, and operator splitting techniques [5, 9, 14, 18, 28, 44, 52, 57, 59, 60]. Their applications include numerical simulations in fluid and solid mechanics, circuits design, electrical power systems, and diffusion-reaction processes in biological and chemical systems. In this paper, we focus on a special class of differential algebraic equations (DAEs) with both stiff and non-stiff components and introduce a semi-implicit technique based on the recently developed Krylov deferred correction (KDC) methods. The method is semi-implicit in the sense that the SDC-based preconditioner is solved using a semi-implicit time-stepping scheme. Krylov deferred correction methods are first studied in [30] in which it is shown that spectral deferred correction (SDC) methods [21] are equivalent to preconditioned Neumann series expansions for linear problems. Instead of the correction in each SDC iteration, KDC method uses the Krylov subspace methods to find the optimal solution and accelerate the convergence. Further, for nonlinear problems, as the spectral deferred correction procedure preconditions the collocation formulation for the DAEs, the Jacobian-Free Newton-Krylov (JFNK) method [36] can be coupled with the KDC technique to avoid the expensive evaluation of the Jacobian matrices[32].

Preliminary analyses for ODEs show that when the Gaussian quadrature nodes are used, the new KDC methods are of arbitrary order, super convergent, A-stable, B-stable, symplectic and symmetric. In [31], the KDC technique was successfully generalized to differential algebraic equations under the Newton-Krylov methods framework. It was shown that the KDC methods can efficiently solve the collocation formulation while the SDC techniques from [21] are divergent for many DAE systems.

In the numerical implementations in [30, 31], explicit low-order time stepping

*The work of Huang and Bu was supported by NSF under grant DMS0411920 and DMS0811130, and part of work was done when Bu was a visiting member of the Institute for Mathematics and Applications (IMA) at the University of Minnesota.

†Department of Mathematics, University of North Carolina, CB # 3250, Phillips Hall, Chapel Hill NC 27599-3250, USA.(emails: agatha,huang,minion@email.unc.edu)

schemes are used in the KDC technique for non-stiff or mildly-stiff problems, while implicit schemes are applied to stiff systems. However, for stiff problems with both stiff and non-stiff components, it is typically advantageous to apply a semi-implicit time integration scheme. In fact, many semi-implicit schemes have already been investigated for ordinary and partial differential equation problems including methods based on Runge-Kutta [7, 11, 12, 17, 35, 47, 55], linear-multistep [2, 6, 23], and SDC [13, 43, 44, 45].

In this paper, we generalize the semi-implicit time integration schemes to the KDC methods for differential algebraic equations, by splitting different components (stiff and non-stiff) and applying different low-order time marching schemes (implicit and explicit) to these components. We study the convergence properties of the resulting semi-implicit KDC (SI-KDC) methods. We show that compared with the fully implicit KDC (FI-KDC) methods, the convergence of the Newton-Krylov methods in SI-KDC are similar. However, as the resulting algebraic equation system in the SI-KDC discretization is simpler than that in FI-KDC, the SI-KDC formulation is in general more efficient to solve. In particular, when a nonlinear non-stiff part is treated explicitly, and fast algorithms can be applied to the linear stiff and algebraic components, optimal performance is expected from SI-KDC as no Newton iteration is required for each preconditioning procedure. The efficiency of the SI-KDC scheme is validated by several examples in this paper. We also demonstrate that unlike for low-index DAE problems, the analysis of the SI-KDC technique becomes more complicated for higher index DAE systems with both algebraic and differential components.

This paper is organized as follows. In Sec. 2, we briefly describe the KDC methods, by introducing the Picard integral collocation formulation, spectral deferred correction (SDC) technique, and Newton-Krylov methods. In Sec. 3, the semi-implicit KDC methods are discussed, and different semi-implicit preconditioning techniques and their convergence properties are analyzed for DAE systems of different index. In Sec. 4, preliminary numerical results are presented to compare different SI-KDC methods with the fully implicit scheme. Finally in Sec. 5, we summarize our results and discuss further applications of the SI-KDC methods.

2. Krylov Deferred Correction Methods. In this section, we discuss the Krylov deferred correction (KDC) technique for general differential algebraic equation (DAE) system

$$F\left(y(t), y'(t), t\right) = 0. \tag{2.1}$$

2.1. Picard Integral Equation and Spectral Integration. In the KDC methods, unlike traditional numerical methods based on the differential form of the equations, we first set $Y(t) = y'(t)$ as the new unknown, and consider the Picard type integral equation

$$F\left(y_0 + \int_0^t Y(\tau)d\tau, Y(\tau), t\right) = 0. \tag{2.2}$$

We refer to Eq. (2.2) as the "yp-formulation" and discuss alternatives to this formulation below.

To discretize the integral equation (2.2) in one time step $[0, \Delta t]$, we linearly map the Gaussian nodes originally defined on $[-1, 1]$ to $[0, \Delta t]$, and denote the p nodes, solution y, and corresponding values of $Y(t)$ at these nodes by $\mathbf{t} = [\mathbf{t_1}, \mathbf{t_2}, \cdots, \mathbf{t_p}]^\mathbf{T}$, $\mathbf{y} = [\mathbf{y_1}, \mathbf{y_2}, \cdots, \mathbf{y_p}]^\mathbf{T}$, and $\mathbf{Y} = [\mathbf{Y_1}, \mathbf{Y_2}, \cdots, \mathbf{Y_p}]^\mathbf{T}$, respectively. Given the discretized $\mathbf{Y}$, a degree $p-1$ interpolating polynomial $P(t) = \sum_{k=0}^{p-1} b_k L_k(t)$ can be constructed to approximate the solution $Y(t)$, where $L_k(t)$ is the Legendre polynomial of degree k, and the coefficients are determined by the integral

$$b_k = (k + \frac{1}{2}) \int_{-1}^{1} L_k(t) f(t) dt$$

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

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

where $\{w_i\}$ are the weights of the Gaussian quadrature. We then approximate $\int_0^{t_m} Y(\tau)d\tau$ using $\int_0^{t_m} P(\tau)d\tau$, and evaluate this degree p polynomial to obtain at $\mathbf{t}$ the approximate function values of $\mathbf{y}$. We refer to this procedure as the spectral integration procedure, and represent the linear mapping from $\mathbf{Y}$ to $\mathbf{y}$ by a matrix $\Delta t S$ where the spectral integration matrix S is independent of the step size Δt. Using the spectral integration matrix, we derive the collocation formulation

$$\mathbf{F}(\mathbf{y_0} + \mathbf{\Delta t S} \otimes \mathbf{Y}, \mathbf{Y}, \mathbf{t}) = \mathbf{0}, \tag{2.3}$$

which will be symbolically denoted as $\mathbf{H}(\mathbf{Y}) = \mathbf{0}$. In the formula, $\mathbf{y}_0 = [y_0, y_0, \cdots, y_0]^T$ is the vector of initial values, and $\otimes$ is the tensor product (i.e. $\Delta t S$ is applied to each component of $\mathbf{Y}$).

Instead of the "yp-formulation", the original SDC method for ODEs in [21] is based on the traditional Picard integral equation or "y-formulation". Methods based on the Picard formulation have also been developed for two point boundary value problems in [25]. The "y-formulation" for ODEs can be generalized for DAE systems of the form

$$\begin{cases} y'(t) = f(y(t), z(t), t), \\ 0 = g(y(t), z(t), t), \end{cases} \tag{2.4}$$

by

$$\begin{cases} y(t) = y_0 + \int_0^t f(y(\tau), z(\tau), \tau)d\tau, \\ 0 = g(y(t), z(t), t). \end{cases} \tag{2.5}$$

However, for an arbitrary DAE system of the form Eq. (2.1), the discretization of the "y-formulation" in the current setting would require a differentiation matrix rather than an integration matrix. Since spectral integration is numerically better conditioned than spectral differention [25, 56], we focus here on the "yp-formulation".

It is shown in [32] that KDC methods for ODEs converge (i.e. $\mathbf{H}(\mathbf{Y}) = \mathbf{0}$) to the same solution as those generated by Gaussian Implicit Runge-Kutta methods. For ODEs, Gaussian nodes based discretization has excellent properties, in particular, we cite the following theorem (mostly from [27]):

THEOREM 2.1. *For ODE problems, the Gauss Runge-Kutta formulation using p Gaussian nodes is order 2p(super convergence), A-stable, B-stable, symplectic (structure preserving), and symmetric (time-reversible). In particular, for fixed time stepsize Δt, the discretization error decreases exponentially when the number of nodes p increases.*

It is also possible to formulate the integration matrix S using Radau or Lobatto type quadrature nodes instead of Gaussian nodes and calculate the Legendre polynomial coefficients accordingly. The Radau Ia quadrature nodes use the left end point (i.e. $t_1 = 0$), the Radau IIa nodes use the right end point (i.e. $t_p = \Delta t$), 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 allow the fast Fourier transform (FFT) to be used for acceleration (FFT is more efficient than existing fast Legendre transforms). Detailed analytical and numerical comparisons of different polynomials and nodes will be reported later.

For a discussion of the choice of nodes for the spectral deferred correction methods for ODEs, the readers are referred to [40].

For DAEs, it is pointed out in [28] that the Gaussian collocation formulation encounters "order reduction". When p Gaussian nodes are applied to an index one DAE system, numerical order for the algebraic component is only p, while for Radau IIa nodes, the order is $2p-1$. We therefore focus on the Radau IIa nodes in our numerical implementations for higher-index DAEs in this paper. Interested readers are referred to [28] (Table 2.3, p18) for further details on the convergence of collocation formulations for different index DAE problems.

2.2. Error Equation and Spectral Deferred Corrections. Notice that for scalar equations, Eq. (2.3) is typically nonlinear with p unknowns as compared to the 1-unknown equation encountered when using backward Euler (or BDF) methods. For N dimensional vector DAEs, the number of unknowns becomes pN as compared to N in BDF methods. Therefore direct application of Newton's method utilizing Gauss elimination for the required linear solves would require $O((pN)^3)$ operations for the collocation formulation with p points, while $O(N^3)$ operations for BDF methods. For this reason, although superior in accuracy and optimal in step size, high order collocation methods for Eq. (2.3) are rarely used in numerical simulations.

There have been several research efforts in designing numerical time integration schemes that are both high order and efficient for ODEs and DAEs. In particular, in the deferred and defect correction methods first proposed by Pereyra and Zadunaisky [49, 59, 60], higher-order accurate solutions of initial value ODEs are built by iteratively approximating an equation for the error or defect to increase the accuracy of a provisional solution. More recently, Dutt et al. [21] 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. In the following, we discuss how the error equation and spectral deferred correction techniques can be generalized to DAEs.

Assume a provisional solution $\tilde{\mathbf{Y}} = [\tilde{\mathbf{Y}}_1, \tilde{\mathbf{Y}}_2, \cdots, \tilde{\mathbf{Y}}_p]^T$ is obtained at the Gaussian type nodes $\mathbf{t}$ using a low-order method or other approximation schemes and denote the corresponding interpolating polynomial approximation to the solution as $\tilde{Y}(t)$, one can define an equation for the error $\delta(t) = Y(t) - \tilde{Y}(t)$ by

$$F\left(y_0 + \int_0^t \left(\tilde{Y}(\tau) + \delta(\tau)\right) d\tau, \tilde{Y}(t) + \delta(t), t\right) = 0. \tag{2.6}$$

Note that Eq. (2.6) gives the identity

$$F\left(y_0 + \int_0^{t_{m+1}} \tilde{Y}(\tau) d\tau + \left(\int_0^{t_m} + \int_{t_m}^{t_{m+1}}\right) \delta(\tau) d\tau, \tilde{Y}(t_{m+1}) + \delta(t_{m+1}), t_{m+1}\right) = 0. \tag{2.7}$$

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

$$F\left(y_0 + [\Delta t S \otimes \tilde{\mathbf{Y}}]_{m+1} + \sum_{l=1}^{m+1} \Delta t_l \tilde{\delta}_{l-1}, \tilde{Y}_{m+1} + \tilde{\delta}_{m+1}, t_{m+1}\right) = 0 \tag{2.8}$$

where $\Delta t_{l+1} = t_{l+1} - t_l$ and t_0 and δ_0 are set to 0. Note that this update formula is in general implicit even though an "explicit" time-marching scheme is used. Similarly, a time-marching scheme based on backward Euler method is given by

$$F\left(y_0 + [\Delta t S \otimes \tilde{\mathbf{Y}}]_{m+1} + \sum_{l=1}^{m+1} \Delta t_l \tilde{\delta}_l, \tilde{Y}_{m+1} + \tilde{\delta}_{m+1}, t_{m+1}\right) = 0. \tag{2.9}$$

These two methods differ only in the way the time integral of $\delta(t)$ is approximated. Eq. (2.8) is equivalent to the rectangle rule using the left endpoint while Eq. (2.9) is the rectangle rule using the right endpoint.

In the SDC methods, the low-order solution $\tilde{\boldsymbol{\delta}}$ is added to the provisional solution $\tilde{\mathbf{Y}}$ in order to form a better approximation, and this iteration continues for a prescribed number of times or until a prescribed error tolerance is achieved. It has been shown that for ODE problems, the accuracy of $\tilde{\mathbf{Y}}$ will increase after each iteration and $\tilde{\mathbf{Y}}$ converges to the solution of the collocation equation for sufficiently small time step Δt. Unfortunately, for general DAE problems of higher-index, it is demonstrated numerically that this SDC iteration procedure is divergent for many DAE systems [31]. It is shown in [30] that for linear systems of ODEs, the spectral deferred correction technique is equivalent to a preconditioned Neumann series expansion, where the preconditioner is the low-order deferred correction procedure. Writing the preconditioned system as

$$(I - C)x = b,$$

one can prove that for ODE problems with sufficient small Δt, all the eigenvalues of C are located inside the unit disc on the complex plane and the Neumann series

$$x = b + Cb + C^2 b + \cdots$$

is convergent. However for DAE problems, there may be eigenvalues whose magnitude is greater than 1 independent of the step size, and hence the SDC procedure becomes divergent. This drawback can be removed by accelerating the convergence using Newton-Krylov methods.

2.3. Newton-Krylov Method and Preconditioners. The Newton-Krylov methods are designed for solving nonlinear algebraic equations of the form $M(x) = 0$ with N equations and unknowns. Assume an initial approximate solution x_0 is known, Newton's method is 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

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

derived using the Krylov subspace methods such as the GMRES, BiCGStab, and TFQMR methods [10, 53, 36] (as J_M is in general non-symmetric). In the formula, $b = M(x_n)$, and $J_M(x_n)$ is the Jacobian matrix of $M(x)$ at x_n. The iterations in Newton's method and the Krylov subspace methods can then be intertwined by reducing the residual of the linear equation by a prescribed factor, and then restart the Newton iterations. The resulting methods are usually called the Newton-Krylov methods, and interested readers are referred to [33, 34, 36] for detailed discussions.

Notice that when

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

where most eigenvalues of C are clustered close to 0, because of the rapid decay of most eigenmodes in $C^q b$, the numerical rank of the Krylov subspace

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

is low and the Newton-Krylov iterations converge rapidly. This is true even for cases when there are a few eigenvalues located outside the unit circle (which causes the

divergence of the SDC methods for DAEs) or inside but close to the unit circle (the order reduction of the SDC methods for stiff ODE problems). In general, an efficient numerical implementation of a Newton-Krylov method depends on: (a) a formulation of the problem $M(x) = 0$ such that J_M is close to the identity matrix $\pm I$, and (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} [19]. Dense integral operators have also been used as preconditioners (see e.g. [37]), which are efficiently applied to an arbitrary vector using fast convolution algorithms such as the fast multipole method [26]. For general DAE system, notice that the low-order time stepping methods in Eqs. (2.8-2.9) can be written in matrix form as

$$\mathbf{F}(\mathbf{y_0} + \mathbf{\Delta t S} \otimes \tilde{\mathbf{Y}} + \mathbf{\Delta t \tilde{S}} \otimes \tilde{\boldsymbol{\delta}}, \tilde{\mathbf{Y}} + \tilde{\boldsymbol{\delta}}, \mathbf{t}) = \mathbf{0}, \tag{2.10}$$

where $\Delta t \tilde{S}$ is the lower triangular representation of the rectangle rule approximation of the spectral integration operator $\Delta t S$. Specifically, for Eq. (2.8)

$$\Delta t \tilde{S}_E = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ \Delta t_1 & 0 & \cdots & 0 & 0 \\ \Delta t_1 & \Delta t_2 & \cdots & 0 & 0 \\ \cdot & \cdot & \cdots & 0 & 0 \\ \Delta t_1 & \Delta t_2 & \cdots & \Delta t_{p-1} & 0 \end{bmatrix} \tag{2.11}$$

and for Eq. (2.9)

$$\Delta t \tilde{S}_I = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 0 & \Delta t_1 & \cdots & 0 & 0 \\ \cdot & \cdot & \cdots & 0 & 0 \\ 0 & \Delta t_1 & \cdots & \Delta t_{p-2} & 0 \\ 0 & \Delta t_1 & \cdots & \Delta t_{p-2} & \Delta t_{p-1} \end{bmatrix}. \tag{2.12}$$

Eq. (2.10) can be considered as an "implicit" function $\tilde{\boldsymbol{\delta}} = \tilde{\mathbf{H}}(\tilde{\mathbf{Y}})$ where the provisional solution $\tilde{\mathbf{Y}}$ is the input variable and the output is $\tilde{\boldsymbol{\delta}}$. It can be seen that the solution of the collocation formulation $\mathbf{H}(\mathbf{Y}) = \mathbf{0}$ also satisfies $\tilde{\mathbf{H}} = \mathbf{0}$. However in [31], it was shown that because the lower order method solves a "nearby" problem, the Jacobian of $\tilde{\mathbf{H}}$ is closer to identity than that of $\mathbf{H}$, and $\tilde{\mathbf{H}} = \mathbf{0}$ is better conditioned. Specifically, applying the implicit function theorem, the Jacobian matrix $J_{\tilde{H}}$ of $\tilde{\mathbf{H}}$ is given by

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

When $\frac{\partial \mathbf{F}}{\partial \mathbf{Y}}$ is non-singular, since $\tilde{S}$ is an approximation of S, when Δt is small, $J_{\tilde{\mathbf{H}}}$ is close to $-I$. For comparison, the Jacobian matrix of $\mathbf{H} = \mathbf{0}$ is given by

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

In regards to point (b), when a forward difference approximation technique is adapted as in most Jacobian-free Newton-Krylov solvers, for any vector v, we can

approximate $J_{\tilde{\mathbf{H}}}(x)v$ by

$$D_h\tilde{\mathbf{H}}(x:v) = \left(\tilde{\mathbf{H}}(x+hv) - \tilde{\mathbf{H}}(x)\right)/h$$

for some properly chosen parameter h (h may be complex). Clearly, computing the function $\tilde{\mathbf{H}}$ in this formulation is simply a deferred correction iteration described succinctly in Eq. (2.10). This difference approximation technique as well as the choice of h have been carefully studied previously and the readers are referred to [34] for details.

2.4. Krylov Deferred Correction Methods. The results in [30] show that the KDC method for DAEs converges more efficiently (to the Gauss Runge-Kutta solution) using a low-order preconditioning iteration compared with a direct solution of the coupled collocation formulation. In addition, the introduction of the Newton-Krylov methods eliminates the divergence of the standard SDC for higher-index DAEs and order reduction for ODE problems. The KDC method consists of two components in its numerical implementation: a Newton-Krylov method that can be applied directly to solve the preconditioned collocation formulation $\tilde{\mathbf{H}}(\tilde{\mathbf{Y}}) = \mathbf{0}$; and the "function evaluation" required for the Newton-Krylov method, which is simply one deferred correction iteration for the given provisional solution. It is the efficiency of this "function evaluation" with which we are concerned in this paper, in particular the use of semi-implicit time-marching methods to reduce the computational cost of the time-marching scheme in the deferred correction iteration.

3. Semi-implicit Preconditioning Techniques for Stiff DAEs. We first define the stiffness of a DAE system $F(y(t), y'(t), t) = 0$ with initial conditions $y(t_0) = y_0$ and $y'(t_0) = y'_0$, by studying the corresponding linearized equation

$$F(y_0, y'_0, t_0) + \frac{\partial F}{\partial y}(y - y_0) + \frac{\partial F}{\partial y'}(y' - y'_0) = By' - Ay + C = 0 \tag{3.1}$$

where $B = \frac{\partial F}{\partial y'}$, $A = -\frac{\partial F}{\partial y}$, and all other quantities are collected in C. Applying the single value decomposition to get $B = UDV^T$ where U and V are unitary matrices and D is a singular diagonal matrix with diagonal entries $\{d_i\}$, one can split Eq. (3.1) into differential part ($d_i \neq 0$) and algebraic component ($d_i = 0$). We call the DAE system $F(y(t), y'(t), t) = 0$ stiff if the differential part is stiff, which can be measured by studying the eigenvalues of $D^{-1}_{nonzero}U^TA$ where $D_{nonzero}$ represents the non-zero submatrix of D.

3.1. Semi-implicit KDC Technique. Consider a DAE system which can be split into two parts

$$F(y(t), y'(t), t) = F_E(y(t), y'(t), t) + F_I(y(t), y'(t), t) = 0 \tag{3.2}$$

where F_E represents the non-stiff component and F_I the stiff component. To derive a semi-implicit discretization of this equation, we introduce $Y(t) = y'(t)$ as the new unknown to get

$$F_E\left(y_0 + \int Y(\tau)d\tau, Y(t), t\right) + F_I\left(y_0 + \int Y(\tau)d\tau, Y(t), t\right) = 0. \tag{3.3}$$

This Picard type integral equation can be directly discretized using the spectral integration matrix S to yield the collocation formulation

$$\mathbf{F_E}(\mathbf{y_0} + \triangle\mathbf{t}\mathbf{S} \otimes \tilde{\mathbf{Y}}, \tilde{\mathbf{Y}}, \mathbf{t}) + \mathbf{F_I}(\mathbf{y_0} + \triangle\mathbf{t}\mathbf{S} \otimes \tilde{\mathbf{Y}}, \tilde{\mathbf{Y}}, \mathbf{t}) = \mathbf{0} \tag{3.4}$$

where $\tilde{\mathbf{Y}} = [\tilde{\mathbf{Y}}_1, \tilde{\mathbf{Y}}_2, ..., \tilde{\mathbf{Y}}_p]^T$ is the desired solution which approximates $Y(t) = y'(t)$ at the quadrature nodes. We further define the error as $\delta(t) = Y(t) - \tilde{Y}(t)$ where $\tilde{Y}$ is a provisional solution to the DAE system. Eq. (3.4) can then be rewritten in the error equation form as

$$F_E\left(y_0 + \int (\tilde{Y}(\tau) + \delta(\tau))d\tau, \tilde{Y} + \delta, t\right) + F_I\left(y_0 + \int (\tilde{Y}(\tau) + \delta(\tau))d\tau, \tilde{Y} + \delta, t\right) = 0. \tag{3.5}$$

To improve the provisional solution $\tilde{Y}(t)$, low-order methods can be applied to derive an approximation of the error denoted by $\tilde{\delta}$. When the explicit Euler method ($\tilde{S}_E$ in Eq. (2.11)) is applied to the non-stiff part and the backward Euler method ($\tilde{S}_I$ in Eq. (2.12)) to the stiff one, the low-order method can be rewritten in the matrix form as

$$\mathbf{F_E}(\mathbf{y_0} + \triangle \mathbf{t}\mathbf{S} \otimes \tilde{\mathbf{Y}} + \triangle \mathbf{t}\tilde{\mathbf{S}}_\mathbf{E} \otimes \tilde{\boldsymbol{\delta}}, \tilde{\mathbf{Y}} + \tilde{\boldsymbol{\delta}}, \mathbf{t}) + \mathbf{F_I}(\mathbf{y_0} + \triangle \mathbf{t}\mathbf{S} \otimes \tilde{\mathbf{Y}} + \triangle \mathbf{t}\tilde{\mathbf{S}}_\mathbf{I} \otimes \tilde{\boldsymbol{\delta}}, \tilde{\mathbf{Y}} + \tilde{\boldsymbol{\delta}}, \mathbf{t}) = \mathbf{0}.$$

This equation gives the preconditioned "implicit" function $\tilde{\boldsymbol{\delta}} = \tilde{\mathbf{H}}_{SI}(\tilde{\mathbf{Y}})$, and the application of the Newton-Krylov methods is then straightforward. This technique is referred to as the semi-implicit KDC (SI-KDC) technique. As discussed in sec. 2.3, the Jacobian matrix of $\tilde{\mathbf{H}}_{SI}$ is obtained by

$$J_{\tilde{\mathbf{H}}_{SI}} = -\left(\frac{\partial \mathbf{F}}{\partial \mathbf{Y}} + \frac{\partial \mathbf{F_E}}{\partial \mathbf{y}}\Delta t\tilde{S}_E + \frac{\partial \mathbf{F_I}}{\partial \mathbf{y}}\Delta t\tilde{S}_I\right)^{-1}\left(\frac{\partial \mathbf{F}}{\partial \mathbf{Y}} + \frac{\partial \mathbf{F}}{\partial \mathbf{y}}\Delta t S\right)$$

$$= -I + \left(\frac{\partial \mathbf{F}}{\partial \mathbf{Y}} + \frac{\partial \mathbf{F_E}}{\partial \mathbf{y}}\Delta t\tilde{S}_E + \frac{\partial \mathbf{F_I}}{\partial \mathbf{y}}\Delta t\tilde{S}_I\right)^{-1}\left(\frac{\partial \mathbf{F_E}}{\partial \mathbf{y}}\Delta t(\tilde{S}_E - S) + \frac{\partial \mathbf{F_I}}{\partial \mathbf{y}}\Delta t(\tilde{S}_I - S)\right).$$

which is closer to $-I$ compared with the original collocation formulation, since $\tilde{S}_E$ and $\tilde{S}_I$ are approximations of S, and Δt is small.

As the semi-implicit KDC discretization scheme converges to the solution of the collocation formulation in Eq. (3.4), its accuracy is not significantly different from results derived using other preconditioning techniques. It will, however, change the condition number of the original system and different preconditioning techniques (choices of F_E and F_I) usually result in very different convergence properties in the Newton-Krylov methods. We found that an optimal preconditioning strategy is in general problem dependent. In the following, using an index one DAE system and an index two system as examples, we show different formulations and semi-implicit preconditioning strategies, and analyze the impact on the convergence of the Newton-Krylov iterations.

3.2. Index One DAE System. As an illustrative example, we first focus on a specific linearized index one stiff DAE system

$$\begin{cases} x_t = a_{11}x + a_{12}y + a_{13}z + F(t), \\ y_t = a_{21}x + a_{22}y + a_{23}z + G(t), \\ 0 = a_{31}x + a_{32}y + a_{33}z + H(t), \end{cases} \tag{3.6}$$

with $a_{33} \neq 0$. We assume all constants a_{ij} are $O(1)$ except for a_{22} which is a large negative number, i.e., the term with coefficient a_{22} represents the stiff component and all others terms are non-stiff. In this system, we refer to the unknowns x and y as the differential variables and z the algebraic variable as z' never appears in the system. For the convenience of notations, we assume $F(t) = G(t) = H(t) = 0$.

As discussed in Sec. 2, we apply the "yp-formulation" to the differential variables instead of the traditional "y-formulation". For the algebraic variable z, there are several ways that this can be done, and we examine three possibilities here.

We first focus on a scheme based on applying the "yp-formulation" to z, and the corresponding error equation system of Eq. (3.6) becomes

$$\begin{bmatrix} \tilde{\mathbf{X}} + \tilde{\boldsymbol{\delta}}_{\mathbf{1}} \\ \tilde{\mathbf{Y}} + \tilde{\boldsymbol{\delta}}_{\mathbf{2}} \\ \mathbf{0} \end{bmatrix} = A \begin{bmatrix} \mathbf{x_0} + \mathbf{\Delta tS}\tilde{\mathbf{X}} + \mathbf{\Delta tS}\tilde{\boldsymbol{\delta}}_{\mathbf{1}} \\ \mathbf{y_0} + \mathbf{\Delta tS}\tilde{\mathbf{Y}} + \mathbf{\Delta tS}\tilde{\boldsymbol{\delta}}_{\mathbf{2}} \\ \mathbf{z_0} + \mathbf{\Delta tS}\tilde{\mathbf{Z}} + \mathbf{\Delta tS}\tilde{\boldsymbol{\delta}}_{\mathbf{3}} \end{bmatrix} \tag{3.7}$$

where

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}. \tag{3.8}$$

The implicit function $\tilde{\delta}$ can then be explicitly written as

$$\begin{bmatrix} I - \Delta t\tilde{S}a_{11} & -\Delta t\tilde{S}a_{12} & -\Delta t\tilde{S}a_{13} \\ -\Delta t\tilde{S}a_{21} & I - \Delta t\tilde{S}a_{22} & -\Delta t\tilde{S}a_{23} \\ -\Delta t\tilde{S}a_{31} & -\Delta t\tilde{S}a_{32} & -\Delta t\tilde{S}a_{33} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{\delta}}_1 \\ \tilde{\boldsymbol{\delta}}_2 \\ \tilde{\boldsymbol{\delta}}_3 \end{bmatrix} = A \begin{bmatrix} \mathbf{x_0} + \mathbf{\Delta tS}\tilde{\mathbf{X}} \\ \mathbf{y_0} + \mathbf{\Delta tS}\tilde{\mathbf{Y}} \\ \mathbf{z_0} + \mathbf{\Delta tS}\tilde{\mathbf{Z}} \end{bmatrix} - \begin{bmatrix} \tilde{\mathbf{X}} \\ \tilde{\mathbf{Y}} \\ 0 \end{bmatrix} \tag{3.9}$$

where I is an identity matrix, $\tilde{S}$ is either $\tilde{S}_I$ or $\tilde{S}_E$, representing different preconditioning schemes for different terms. Clearly, $\tilde{S}_I$ should be applied to the stiff term with coefficient a_{22}. We further assume that we want the provisional solution to remain on the manifold due to the algebraic equation constraint, by applying $\tilde{S}_I$ to $\{a_{31}, a_{32}, a_{33}\}$ terms. The explicit low-order scheme $\tilde{S}_E$ can then be applied to all remaining terms. Notice that to march from t_j to t_{j+1} in this specific semi-implicit low-order time stepping procedure, the unknowns are decoupled, therefore the evaluation of the implicit function $\tilde{\mathbf{H}}_{SI}$ is less expensive than evaluating $\tilde{\mathbf{H}}_{FI}$ in the FI-KDC scheme where $\tilde{S}_I$ is applied to all terms in the system.

Comparing the Jacobian matrix of the resulting semi-implicit KDC scheme

$$\left(E - \Delta t\tilde{S}_E \otimes \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & 0 & a_{23} \\ 0 & 0 & 0 \end{bmatrix} - \Delta t\tilde{S}_I \otimes \begin{bmatrix} 0 & 0 & 0 \\ 0 & a_{22} & 0 \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \right)^{-1} (\Delta tS \otimes A - E)$$

with that from the fully-implicit KDC approach

$$\left(E - \Delta t\tilde{S}_I \otimes \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \right)^{-1} (\Delta tS \otimes A - E), \quad \text{where} \quad E = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

it can be seen that the eigenvalues of the SI-KDC Jacobian matrix are similarly distributed to those from FI-KDC for sufficiently small Δt, as $\Delta t\tilde{S}_I a_{22}$ is the dominant part in both matrices. Therefore the convergence properties of the Jacobian-Free Newton-Krylov methods are similar for both SI-KDC and FI-KDC methods.

Note that applying $\tilde{S}_I$ to more terms in Eq. (3.9) will generate schemes with similar convergence properties. However the evaluation of the implicit functions may become more expensive as the unknowns may no longer decouple and a larger system has to be solved. Also, it is possible to modify the requirement that the provisional solution always satisfy the algebraic equation, e.g., we can apply $\tilde{S}_E$ to $\{a_{31}, a_{32}\}$ terms, however this will significantly change the eigenvalue distribution compared with the FI-KDC scheme. In Sec. 4, eigenvalue distributions are numerically computed for different preconditioning techniques.

In our second formulation, instead of applying the "yp-formulation" to the algebraic variable z, we use z directly to avoid the spectral integration for efficiency considerations as in

$$\begin{bmatrix} \tilde{\mathbf{X}} + \tilde{\boldsymbol{\delta}}_{\mathbf{1}} \\ \tilde{\mathbf{Y}} + \tilde{\boldsymbol{\delta}}_{\mathbf{2}} \\ \mathbf{0} \end{bmatrix} = A \begin{bmatrix} \mathbf{x_0} + \boldsymbol{\Delta}\mathbf{tS}\tilde{\mathbf{X}} + \boldsymbol{\Delta}\mathbf{tS}\delta_1 \\ \mathbf{y_0} + \boldsymbol{\Delta}\mathbf{tS}\tilde{\mathbf{Y}} + \boldsymbol{\Delta}\mathbf{tS}\delta_2 \\ \tilde{z} + \delta_3 \end{bmatrix}. \tag{3.10}$$

It can be shown that this formulation is in fact very similar to the first formulation if we replace the explicit $\mathbf{z_0} + \boldsymbol{\Delta}\mathbf{tS}\tilde{\mathbf{Z}} + \boldsymbol{\Delta}\mathbf{t}\tilde{\mathbf{S}}_{\mathbf{E}}\tilde{\boldsymbol{\delta}}_{\mathbf{3}}$ by $\tilde{z}^j + \tilde{\delta}_3^j$ and the implicit $\mathbf{z_0} + \boldsymbol{\Delta}\mathbf{tS}\tilde{\mathbf{Z}} + \boldsymbol{\Delta}\mathbf{t}\tilde{\mathbf{S}}_{\mathbf{I}}\tilde{\boldsymbol{\delta}}_{\mathbf{3}}$ by $\tilde{z}^{j+1} + \tilde{\delta}_3^{j+1}$ when marching from t_j to t_{j+1}. We therefore skip the detailed discussions for this formulation.

In our third formulation, notice that it is unnecessary to apply the error equation to the algebraic variable z in the second formulation, we can therefore work on the "simplified" error equation system

$$\begin{bmatrix} \tilde{\mathbf{X}} + \delta_1 \\ \tilde{\mathbf{Y}} + \delta_2 \\ \mathbf{0} \end{bmatrix} = A \begin{bmatrix} \mathbf{x_0} + \boldsymbol{\Delta}\mathbf{tS}\tilde{\mathbf{X}} + \boldsymbol{\Delta}\mathbf{tS}\delta_1 \\ \mathbf{y_0} + \boldsymbol{\Delta}\mathbf{tS}\tilde{\mathbf{Y}} + \boldsymbol{\Delta}\mathbf{tS}\delta_2 \\ \mathbf{z} \end{bmatrix}. \tag{3.11}$$

An immediate advantage of this formulation is that given the provisional solutions $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{Y}}$, we can use a semi-implicit scheme to derive low-order solutions of $\tilde{\boldsymbol{\delta}}_1$, $\tilde{\boldsymbol{\delta}}_2$ and $\mathbf{z}$ at each node point, and define a reduced size implicit function $[\tilde{\boldsymbol{\delta}}_1, \tilde{\boldsymbol{\delta}}_2] = \tilde{\mathbf{H}}_{RS}(\tilde{\mathbf{X}}, \tilde{\mathbf{Y}})$. Due to the reduce system size, the Newton-Krylov method becomes more efficient while requiring less memory. For our specific index one system, detailed algebraic manipulation of the implicit function $\tilde{\mathbf{H}}_{RS}$ returns the explicit form of the Jacobian matrix

$$\begin{bmatrix} I - \Delta t\tilde{S}a_{11} + \Delta t\tilde{S}\frac{a_{13}a_{31}}{a_{33}} & -\Delta t\tilde{S}a_{12} + \Delta t\tilde{S}\frac{a_{13}a_{32}}{a_{33}} \\ -\Delta t\tilde{S}a_{21} + \Delta t\tilde{S}\frac{a_{23}a_{31}}{a_{33}} & I - \Delta t\tilde{S}a_{22} + \Delta t\tilde{S}\frac{a_{23}a_{32}}{a_{33}} \end{bmatrix}^{-1} \left(\Delta tS \otimes \begin{bmatrix} a_{11} - \frac{a_{13}a_{31}}{a_{33}} & a_{12} - \frac{a_{13}a_{32}}{a_{33}} \\ a_{21} - \frac{a_{23}a_{31}}{a_{33}} & a_{22} - \frac{a_{23}a_{32}}{a_{33}} \end{bmatrix} - \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \right).$$

In this formulation, $\tilde{S}_I$ is associated with a_{22} term. The requirement that the provisional solution satisfies the algebraic equation constraint at all nodes is equivalent to applying $\tilde{S}_I$ to terms associated with coefficient factors a_{13} and a_{23}, and $\tilde{S}_E$ can be applied to all remaining terms. For large stiff systems, when proper time step size Δt is used, it can be shown that the Jacobian matrix of $\tilde{\mathbf{H}}_{RS}$ is very close to the identity matrix except for a few bad eigenvalues due to the stiff components. Finally, similar to the first formulation, it is not necessary to enforce the requirement that the provisional solution always stays in the manifold described by the algebraic equation, and under appropriate conditions, $\tilde{S}_E$ can be applied to terms with coefficient factors a_{13} and a_{23}, e.g., when the coefficients $\frac{a_{13}a_{31}}{a_{33}}$, $\frac{a_{13}a_{32}}{a_{33}}$, $\frac{a_{23}a_{31}}{a_{33}}$, and $\frac{a_{23}a_{32}}{a_{33}}$ are $O(1)$.

In summary, it can be seen that the choice of splitting of the DAE into explicit and implicit pieces can profoundly affect the efficiency and expected convergence of the Newton-Krylov iterates in SI-KDC methods even for index 1 problems. Therefore the choice of splitting must be carefully considered and will depend on the problem at hand.

3.3. Index Two DAE System. Now consider the case of index 2 problems. We demonstrate here that the tasking of finding proper semi-implicit preconditioners becomes even more involved for higher-index DAE systems. In this section, focusing

on the formulation where the "yp-formulation" is applied to both differential and algebraic variables, we again consider a simple linear index two DAE system

$$\begin{cases} x_t = a_{11}x + a_{12}y + a_{13}z, \\ y_t = a_{21}x + a_{22}y + a_{23}z, \\ 0 = a_{31}x + a_{32}y \end{cases} \tag{3.12}$$

with stiff component a_{22}, and study the convergence and stability properties of different preconditioning techniques.

Assume a provisional solution is available, the error equation form of this index two system is given by

$$\begin{bmatrix} \tilde{\mathbf{X}} + \delta_{\mathbf{1}} \\ \tilde{\mathbf{Y}} + \delta_{\mathbf{2}} \\ \mathbf{0} \end{bmatrix} = A \begin{bmatrix} \mathbf{x_0} + \mathbf{\Delta tS}\tilde{\mathbf{X}} + \mathbf{\Delta tS}\delta_{\mathbf{1}} \\ \mathbf{y_0} + \mathbf{\Delta tS}\tilde{\mathbf{Y}} + \mathbf{\Delta tS}\delta_{\mathbf{2}} \\ \mathbf{z_0} + \mathbf{\Delta tS}\tilde{\mathbf{Z}} + \mathbf{\Delta tS}\delta_{\mathbf{3}} \end{bmatrix} \tag{3.13}$$

where

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & 0 \end{bmatrix}. \tag{3.14}$$

The implicit function $\tilde{\delta} = \tilde{\mathbf{H}}(\tilde{X}, \tilde{Y}, \tilde{Z})$ can then be derived by solving the system

$$\begin{bmatrix} I - \Delta t\tilde{S}a_{11} & -\Delta t\tilde{S}a_{12} & -\Delta t\tilde{S}a_{13} \\ -\Delta t\tilde{S}a_{21} & I - \Delta t\tilde{S}a_{22} & -\Delta t\tilde{S}a_{23} \\ -\Delta t\tilde{S}a31 & -\Delta t\tilde{S}a_{32} & 0 \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{\delta}}_1 \\ \tilde{\boldsymbol{\delta}}_2 \\ \tilde{\boldsymbol{\delta}}_3 \end{bmatrix} = A \begin{bmatrix} \mathbf{x_0} + \mathbf{\Delta tS}\tilde{\mathbf{X}} \\ \mathbf{y_0} + \mathbf{\Delta tS}\tilde{\mathbf{Y}} \\ \mathbf{z_0} + \mathbf{\Delta tS}\tilde{\mathbf{Z}} \end{bmatrix} - \begin{bmatrix} \tilde{\mathbf{X}} \\ \tilde{\mathbf{Y}} \\ \mathbf{0} \end{bmatrix}.$$

where $\tilde{S}$ is either $\tilde{S}_I$ or $\tilde{S}_E$ representing the low-order approximation of the integration operator. Clearly, we have to apply $\tilde{S}_I$ to the stiff component with coefficient a_{22}. If we want to enforce the condition that the provisional solution stays in the manifold defined by the algebraic equation, $\tilde{S}_I$ should be applied to both a_{31} and a_{32} terms. Also, we can apply $\tilde{S}_E$ to a_{11}, a_{12}, and a_{21} terms. In the following, we discuss different strategies for a_{13} and a_{23} terms, corresponding to terms related with the algebraic variable z in the system.

Our first observation is that unlike in the first formulation for index one DAE systems, $\tilde{S}_E$ can no longer be applied to both a_{13} and a_{23} terms, as doing so generates an over-determined linear system when marching from t_j to t_{j+1}. Three remaining possibilities are to apply $\tilde{S}_I$ to both terms (SIKDC-II); or $\tilde{S}_E$ to a_{13} and $\tilde{S}_I$ to a_{23} (SIKDC-EI); or $\tilde{S}_I$ to a_{13} and $\tilde{S}_E$ to a_{23} (SIKDC-IE).

Applying the implicit function theorem, we can derive the Jacobian matrix for each implicit function $\tilde{\mathbf{H}}$ in the KDC framework. The Jacobian matrix J_{II} for SIKDC-II is given by

$$\left(E - \Delta t\tilde{S}_E \otimes \begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \Delta t\tilde{S}_I \otimes \begin{bmatrix} 0 & 0 & a_{13} \\ 0 & a_{22} & a_{23} \\ a_{31} & a_{32} & 0 \end{bmatrix} \right)^{-1} (\Delta tS \otimes A - E),$$

J_{EI} is

$$\left(E - \Delta t\tilde{S}_E \otimes \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \Delta t\tilde{S}_I \otimes \begin{bmatrix} 0 & 0 & 0 \\ 0 & a_{22} & a_{23} \\ a_{31} & a_{32} & 0 \end{bmatrix} \right)^{-1} (\Delta tS \otimes A - E),$$

and J_{IE} is

$$\left(E - \Delta t \tilde{S}_E \otimes \begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & 0 & a_{23} \\ 0 & 0 & 0 \end{bmatrix} - \Delta t \tilde{S}_I \otimes \begin{bmatrix} 0 & 0 & a_{13} \\ 0 & a_{22} & 0 \\ a_{31} & a_{32} & 0 \end{bmatrix} \right)^{-1} (\Delta t S \otimes A - E).$$

Similarly, repeating this procedure for the FI-KDC scheme, we get the Jacobian matrix J_{FI}

$$\left(E - \Delta t \tilde{S}_I \otimes \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & 0 \end{bmatrix} \right)^{-1} (\Delta t S \otimes A - E).$$

It is possible to understand the properties of the four different preconditioning techniques by studying the condition numbers of simple 3×3 matrices representing the linear system to be solved during each step when marching from t_j to t_{j+1}. In the following, assume the stiff component $I - \Delta t \tilde{S}_I a_{22}$ in the matrix

$$\begin{bmatrix} I - \Delta t \tilde{S} a_{11} & -\Delta t \tilde{S} a_{12} & -\Delta t \tilde{S} a_{13} \\ -\Delta t \tilde{S} a_{21} & I - \Delta t \tilde{S} a_{22} & -\Delta t \tilde{S} a_{23} \\ -\Delta t \tilde{S} a_{31} & -\Delta t \tilde{S} a_{32} & 0 \end{bmatrix} \tag{3.15}$$

is about order λ, and the magnitude of other terms is either order ϵ when $\Delta t \tilde{S}_I$ is applied, or 0 when an explicit time stepping scheme is used. The matrices for SIKDC-II, SIKDC-EI, SIKDC-IE, and FIKDC are

$$\begin{bmatrix} 1 & 0 & \epsilon \\ 0 & \lambda & \epsilon \\ \epsilon & \epsilon & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & \lambda & \epsilon \\ \epsilon & \epsilon & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 & \epsilon \\ 0 & \lambda & 0 \\ \epsilon & \epsilon & 0 \end{bmatrix}, \quad \text{and} \quad \begin{bmatrix} 1 \pm \epsilon & \epsilon & \epsilon \\ \epsilon & \lambda & \epsilon \\ \epsilon & \epsilon & 0 \end{bmatrix},$$

respectively. For $\lambda = 10^3$ and $\epsilon = 10^{-3}$, the condition number of the Jacobian matrix corresponding to these matrices are $9.99 \cdot 10^8$, $1.00 \cdot 10^{12}$, $1.00 \cdot 10^9$ and $9.99 \cdot 10^8$, and the corresponding eigenvalues of SIKDC-II, SIKDC-IE, and FIKDC are almost identical. We therefore conclude that the convergence and stability properties of SIKDC-II, SIKDC-IE, and FIKDC are similar, while SIKDC-EI is not a proper preconditioner as it is more ill-conditioned. Our numerical experiments also reveal that the numbers of iterations in the Newton-Krylov methods for both SIKDC-II and SIKDC-IE are approximately the same as that of FI-KDC. However as the unknowns can be decoupled in the SIKDC-IE formulation (when marching from t_j to t_{j+1}, we first solve the second equation for $\tilde{\delta}_2$ at t_{j+1}, then the third equation for $\tilde{\delta}_1$, and finally the first equation for $\tilde{\delta}_3$), SIKDC-IE is therefore the most efficient preconditioning approach.

Finally in this section, note that a good semi-implicit preconditioning technique should reduce the amount of work required for evaluating the corresponding implicit function $\tilde{\mathbf{H}}$ without significantly changing the convergence properties of the Newton-Krylov methods. This is possible for many stiff DAE systems, especially for those with nonlinear non-stiff components and linear stiff parts. However, finding the optimal semi-implicit preconditioner is usually problem dependent and requires better understanding of the underlying properties of the system. In order to fully exploit the efficiency of the new KDC methods, optimized strategies have to be developed for the selection of adaptive step-size, order of the method, proper Newton-Krylov methods, as well as several different parameters. As the discussion here indicates, optimizing the performance of KDC methods for a particular class of problems is an open research problem.

4. Preliminary Numerical Results. In this section, we present several numerical examples to illustrate the performance of SI-KDC methods and validate the analyses presented in previous section.

4.1. Linear Index One DAE System. In the first example, we consider an index one DAE system

$$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}' = \begin{bmatrix} 2 & 0 & -1 & 1 \\ 0 & -10^4 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 - \exp(t) \\ y_3 \\ y_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \exp(t) \\ 0 \\ 0 \end{bmatrix}. \tag{4.1}$$

Notice that the equations can be decomposed into three parts: the first and third differential equations are non-stiff, so an explicit time stepping scheme can be applied; the second equation contains a stiff part due to the coefficient -10^4, so an implicit scheme is used for this stiff component; as the fourth equation is an algebraic equation, we apply implicit schemes to keep the provisional solution in the manifold defined by this algebraic equation. This is the first approach discussed in Sec. 3.2. This semi-implicit scheme can be represented in the matrix form as

$$\begin{bmatrix} 2 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -10^4 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

where the first matrix is corresponding to an explicit part F_E and the second to an implicit one F_I. We use Δt= 0.2 and 5 Radau IIa nodes. The eigenvalue distribution of the matrix $J_{SI}+I$ derived from Eq. (3.6) is compared with that of $J_{FI}+I$ from the FI-KDC method in Fig. 4.1. It can be seen that the results from the SI-KDC approach are almost identical to those from FI-KDC. As the convergence of the Newton-Krylov schemes is determined by the eigenvalue distributions, our numerical experiments also show that the convergence rate of the SI-KDC is almost identical to that from FI-KDC. However, to march from t_j to t_{j+1}, as the unknowns are decoupled in the semi-implicit preconditioning approach, the system size is reduced, which makes the SI-KDC method more efficient.

One common concern for the KDC type methods is the storage and number of operations required by the Krylov subspace methods. When GMRES is used and the number of iterations increases, the required memory and number of multiplications grow linearly and quadratically, respectively. Therefore the GMRES procedure becomes very expensive and requires excessive memory storage for large problems and large numbers of iterations. It is for these reasons that in practice, alternative methods are often preferred, such as the restarted GMRES, Bi-conjugate gradients stabilized (BiCGStab), and transpose free quasi minimal residual (TFQMR) methods. In Fig. 4.2, we compare the convergence of GMRES with BiCGStab and TFQMR. In the experiments, we use 5 Radau points and march with step-size $\Delta t = 0.1$ for $t \in [1, 1.1]$. Our numerical results show very similar convergence rates for these methods, however for large number of Newton-Krylov iterations, the required memory is bounded and the number of multiplications only grows linearly for BiCGStab and TFQMR based methods.

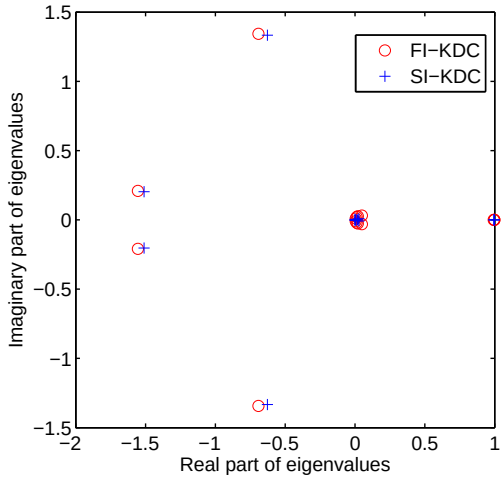


FIG. 4.1. *Comparing the eigenvalue distributions of SI-KDC with FI-KDC for the linear index 1 example.*

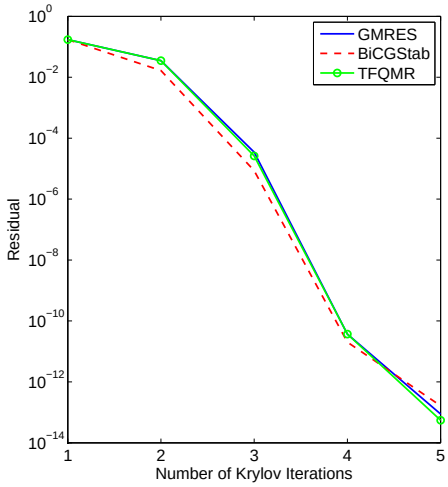


FIG. 4.2. *Comparing different Krylov subspace methods.*

4.2. Nonlinear Index One DAE System. In our second example, we consider a nonlinear DAE problem

$$\begin{bmatrix} y_1 - \cos(t) \\ y_2 - \sin(t) \\ 0 \end{bmatrix}' = \left(\begin{bmatrix} 0 & 0 & 0 \\ 0 & -10^6 & 0 \\ 0 & 0 & 0 \end{bmatrix} + U \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} U^T \right) \begin{bmatrix} (y_1 - \cos(t))y_2 \\ y_2 - \sin(t) \\ y_3 - t \end{bmatrix} \tag{4.2}$$

where U is an orthogonal matrix. For this system, the non-stiff component is nonlinear and the stiff part linear, we therefore apply the explicit $\tilde{S}_E$ to the non-stiff component and the implicit $\tilde{S}_I$ to the stiff part. Notice that only one linear solve is required in

the SI-KDC scheme, in the following, we compare the convergence behavior and CPU time of the SI-KDC scheme with those from FI-KDC where implicit time stepping schemes are applied to all terms in the system. In Fig. 4.3, we compute the solution from $t_0 = 0.2$ to $t_{final} = 0.25$ with step-size $\Delta t = 0.05$, using 5 Radau IIa points, and examine the residual after each Newton-GMRES iteration for both SI-KDC and FI-KDC methods. It can be seen that the convergence behavior in SI-KDC scheme is very similar to that in FI-KDC.

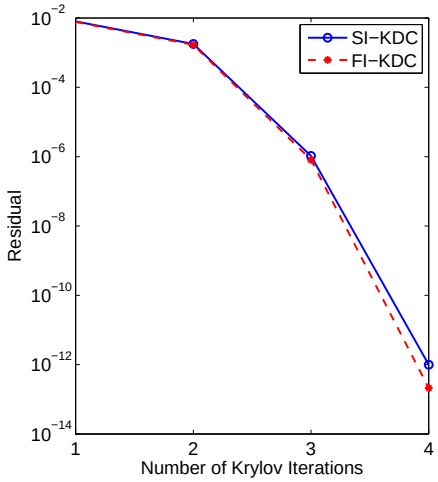


FIG. 4.3. *Comparing the convergence of the SI-KDC and FI-KDC methods for the nonlinear index 1 problem.*

To compare the CPU time and the number of function evaluations, we use different number of nodes (3, 5, 8, 12 and 20) and march from $t = 0$ to $t_{final} = 10.0$ with step-size $\Delta t = 1.0$. In Fig. 4.4, we plot the error as functions of (left) the CPU time and (right) number of function evaluations (each access to Eq. (4.2) is defined as one function call) for each method. Each data point represents the result for a specific number of nodes. Clearly, for the same accuracy requirement, the SI-KDC scheme is much faster than FI-KDC, since only one linear solve is required when marching from t_j to t_{j+1} for the SI-KDC method, while a nonlinear equation system has to be solved in the FI-KDC approach. In Fig. 4.5, we plot how the error changes as a function of the number of function evaluations using different time step sizes (Δt= 0.25, 0.5, 1.0, and 2.0) and number of node points (3, 4, and 5) for both SI-KDC and FI-KDC when marching from $t_0 = 0$ to $t_{final} = 10.0$. Each data point represents the number of function evaluations and accuracy for a specific set of Δt and number of nodes. We can see that for the same accuracy requirement, the SI-KDC scheme is much faster than FI-KDC; and for both methods, higher order schemes (more nodes) are more efficient than lower order ones for higher-accuracy requirements.

4.3. Electrical Power System. The differential algebraic equations have been widely used in the study and engineering of the bulk transfer of electrical powers. Typical electrical power system networks include a large number of dynamic and static components such as generators, exciters, governors, loads, transformers, and other power electronic devices, where the dynamics and constraints for each individual component are often modeled by a system of DAEs. As the power systems exhibit a wide-range of time varying dynamics that may span several orders of magnitude, their efficient numerical simulations are often considered challenging. In this section,

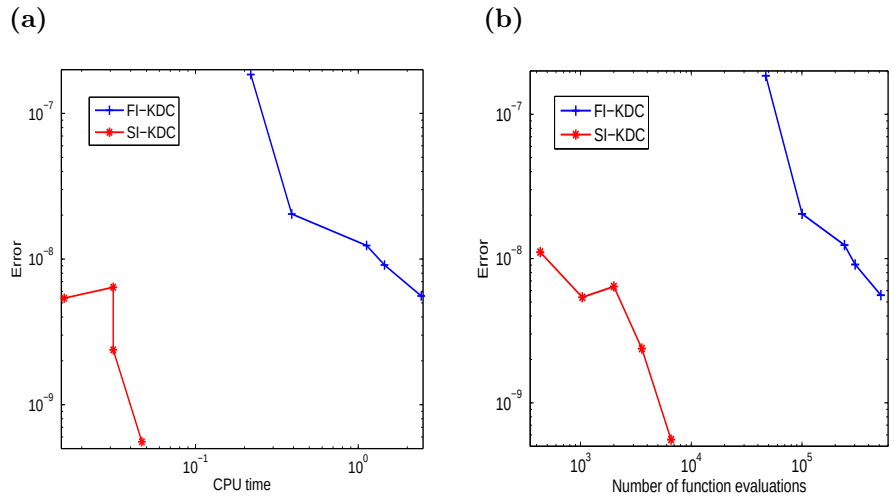


FIG. 4.4. *The error as functions of CPU time (left) and number of function evaluations (right).*

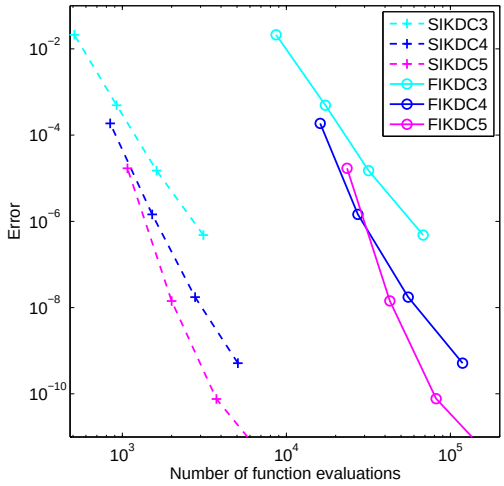


FIG. 4.5. *Comparing SI-KDC with FI-KDC for different step sizes and number of nodes.*

to evaluate the performance of the SI-KDC approach, we consider a simple power stabilizer system which has 9 buses and 3 generators with constant power loads. Each generator has 2 states, so the number of differential states and algebraic equations are 6 and 18, respectively. This system can be described by the following index 1 DAE system:

$$\delta' = \Omega_b(\omega - 1),$$

$$\omega' = (P_m - P_e - D(\omega - 1))/M,$$

$$V_i \sum (V_j(G_{ij}\cos(\theta_i - \theta_j) + B_{ij}\sin(\theta_i - \theta_j))) + P_{gi} - P_{di} = 0,$$

$$V_i \sum (V_j(G_{ij}\sin(\theta_i - \theta_j) - B_{ij}\cos(\theta_i - \theta_j))) + Q_{gi} - Q_{di} = 0$$

where the differential variables δ and ω are the internal state vectors (generators and loads), and V_i and θ_i are the algebraic variables for voltage magnitude and phase. In the system, D is a coefficient representing a damping factor, and when D has a large magnitude, the system becomes stiff and the term $-D(\omega - 1)$ is linear stiff and other terms are nonlinear nonstiff; M is an inertia constant; P_m denotes the mechanical power; $P_e = f(V_i, \theta_i)$ is the electrical power; $P_g = f(\delta_i, \theta_i)$ and $Q_g = f(\delta_i, \theta_i)$ represent respectively the active and reactive power injected in the network by generators; P_d and Q_d are respectively the active and reactive power absorbed from the network by loads; and G_{ij} and B_{ij} are respectively a real and an imaginary part of an admittance matrix to represent the current status between load i and load j. Also, the first two sets of differential equations model the dynamics of the generators and loads, and the remaining algebraic equations represent the fast power balance dynamics on the sparsely connected distribution network of power lines and buses. To study the dynamics of the system, we assume an one-phase fault on a line between bus 2 and bus 7 occurs at $t = 1$, and clears out at $t = 2$. When the fault occurs, the shunt admittances of the network are modified and the admittance matrix is recomputed. We neglect further details of this model and techniques to split other systems to stiff and nonstiff in general, and refer interested readers to [4, 46, 58].

In the simulation, we require that the provisional solution stays on the manifold defined by the algebraic constraints. However we apply explicit schemes to both nonstiff components and algebraic variables in the differential equations. In Fig. 4.6, we first show how the accuracy of the SI-KDC method depends on the number of Radau IIa nodes and different time step sizes, by plotting the accuracy as a function of the number of total nonlinear solves (one nonlinear solve is required to marching from t_j to t_{j+1}). It can be seen that (a) for a fixed number of nodes, smaller time step sizes

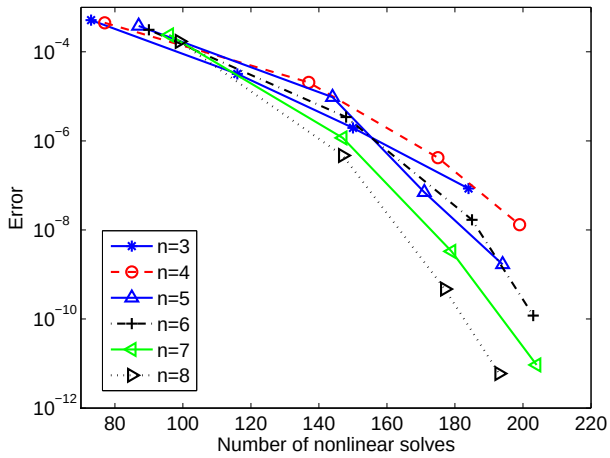


FIG. 4.6. *Accuracy of SI-KDC method vs. number of nonlinear solves for different number of nodes.*

(more nonlinear solves) are required for higher-accuracy requirements, and (b) higher order methods (more nodes) are in general more efficient than lower order ones for higher-accuracy requirements.

In Fig 4.7, using 6 Radau IIa nodes for each time step from $t = 0$ to $t = 1.8$, to study the convergence properties of the Newton-Krylov methods in the SI-KDC approach, we show the residual after each low-order SDC iteration (one $\tilde{\mathbf{H}}_{SI}$ evaluation) (left plot), and how the accuracy depends on the number of total nonlinear

solves required to march from t_j to t_{j+1} (right plot). These results are compared with those from the FI-KDC approach. Clearly, the FI-KDC approach is optimal in stability and has (slightly) better convergence rates in the Newton-Krylov iterations, however the residual after each $\tilde{\mathbf{H}}$ evaluation (SDC iteration) in the SI-KDC method decays in a very similar way as in FI-KDC. As explicit schemes are applied to the non-stiff components and algebraic variables in the differential equations, e.g., the size of nonlinear system from SI-KDC scheme is smaller than that from FI-KDC method, therefore the SI-KDC preconditioning technique is more efficient than FI-KDC for this specific application.

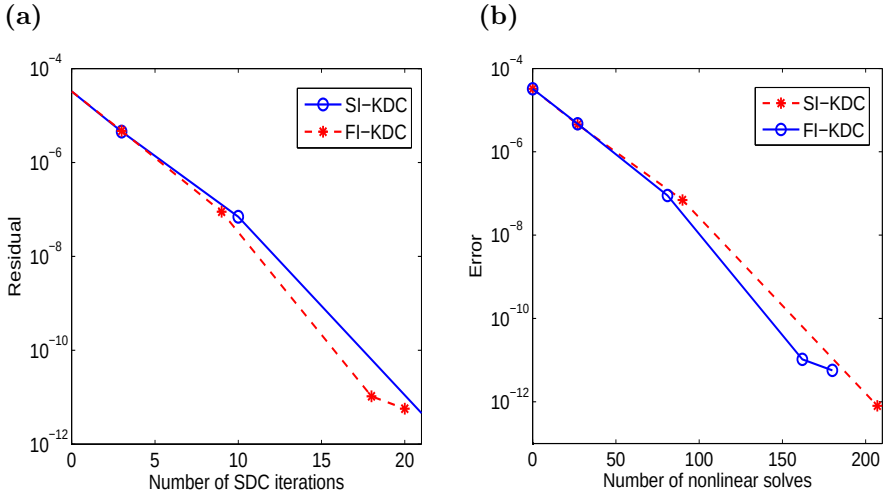


FIG. 4.7. *(left) Residual after each SDC iterations, and (right) accuracy vs. number of nonlinear solves.*

There exist many numerical simulation tools and methods for power systems [1, 38, 58], including the techniques based on splitting the DAE systems to differential and algebraic parts and solve them separately using ODE solvers for the differential parts and a Newton-type method (e.g. Newton-Raphson) for algebraic components. In the following, we compare the performance of our SI-KDC approach with a matlab based package called "PSAT", a power system solver based on the Newton-Raphson methods and trapezoidal rules [42]. In Fig. 4.8, we examine the accuracy of the SI-KDC approach for different time step sizes and number of nodes, and compare the results with those from PSAT. In the figure, each curve represents the results for a fixed time step size, and each point on the curve represents the different number of nodes used in the simulation, ranging from 3 to 10. Clearly, for a fixed step-size, more nodes generate higher-accuracy results, and higher order methods are more efficient for a prescribed accuracy requirement. Also, compared with PSAT, the SI-KDC requires much less nonlinear solves for the same accuracy requirement. In our simulation, fixed step sizes are used for both SI-KDC and PSAT.

4.4. Linear Index Two DAE Systems. Finally in this section, to numerically validate the analyses in Sec. 3.3, we study the SI-KDC techniques for two different index two DAE systems. We first consider the system (see [5])

$$\begin{cases} x_1' = (\alpha - \frac{1}{2-t})x_1 + (2-t)\alpha z + \frac{3-t}{2-t}\exp^t = f_1(x_1, z), \\ x_2' = \frac{1-\alpha}{t-2}x_1 - 10^4 x_2 + (\alpha - 1)z + (10^4+1)\exp^t = f_2(x_1, x_2, z), \\ 0 = (t+2)x_1 + (t^2-4)x_2 - (t^2+t-2)\exp^t = g(x_1, x_2) \end{cases}$$

with $\alpha \sim O(1)$. Several semi-implicit approaches can be applied as discussed in Sec. 3.3. The SIKDC-II approach applies implicit schemes to the algebraic variable

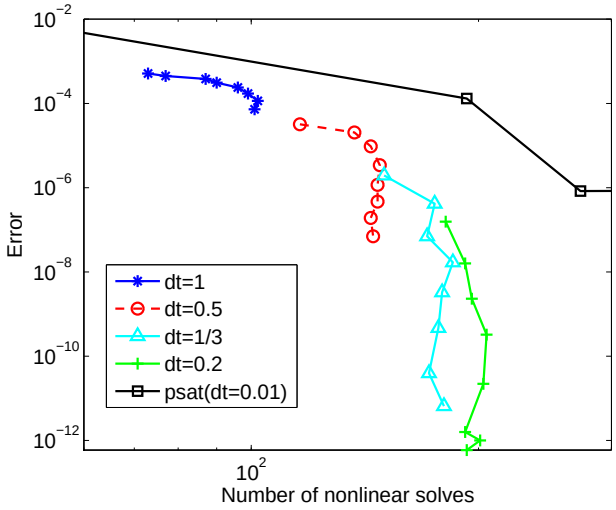


FIG. 4.8. *Comparing the accuracy and efficiency of SI-KDC with PSAT.*

z in both differential equations, and the resulting low-order stepping scheme can be succinctly represented as

$$\begin{cases} X_1^{j+1} = f_1(x_1^j, z^{j+1}), \\ X_2^{j+1} = f_2(x_1^j, x_2^{j+1}, z^{j+1}), \\ 0 = g(x_1^{j+1}, x_2^{j+1}) \end{cases} \tag{4.3}$$

where the superscript j represents the node point t_j, and the original equation is used instead of the error equation form to simplify the notations. The SIKDC-IE formulation applies an implicit scheme to z in the first equation, and an explicit method to z in the second equation as in

$$\begin{cases} X_1^{j+1} = f_1(x_1^j, z^{j+1}), \\ X_2^{j+1} = f_2(x_1^j, x_2^{j+1}, z^j), \\ 0 = g(x_1^{j+1}, x_2^{j+1}). \end{cases} \tag{4.4}$$

Similarly, the SIKDC-EI formulation is given by

$$\begin{cases} X_1^{j+1} = f_1(x_1^j, z^j), \\ X_2^{j+1} = f_2(x_1^j, x_2^{j+1}, z^{j+1}), \\ 0 = g(x_1^{j+1}, x_2^{j+1}), \end{cases} \tag{4.5}$$

and the FI-KDC scheme uses the discretization

$$\begin{cases} X_1^{j+1} = f_1(x_1^{j+1}, z^{j+1}), \\ X_2^{j+1} = f_2(x_1^{j+1}, x_2^{j+1}, z^{j+1}), \\ 0 = g(x_1^{j+1}, x_2^{j+1}). \end{cases} \tag{4.6}$$

As we discussed in Sec. 3.3, the SIKDC-EI preconditioning technique is ill-conditioned, which is validated by the eigenvalue distribution of the matrix $J_{EI} + I$ plotted in the left of Fig. 4.9, in comparison with those from $J_{IE} + I$. In the right plot of Fig. 4.9, we compare the eigenvalue distributions of the SIKDC-IE approach with the fully

implicit approach in Eq. (4.6), it can be seen that the eigenvalues are very similarly distributed, therefore the convergence properties of the SIKDC-IE approach is similar to those of the FI-KDC method. In Table. 4.1, we show the condition number of the Jacobian matrix for different low-order stepping schemes and different number of nodes. Not surprisingly, the condition number of the SIKDC-EI matrix is huge and increases very rapidly as the number of nodes increases.

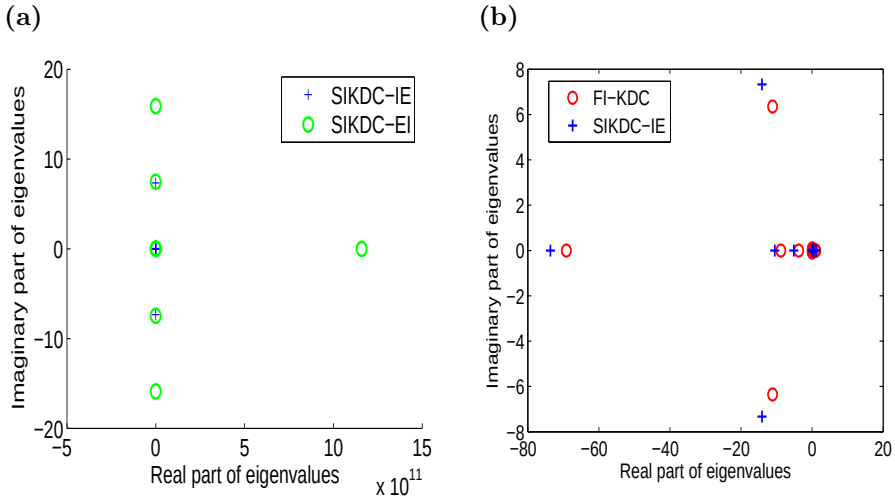


FIG. 4.9. *Comparing the eigenvalue distributions for (left) SIKDC-IE and SIKDC-EI, and (right) SIKDC-IE and FI-KDC.*

	SIKDC-II	SIKDC-IE	SIKDC-EI	FIKDC
n=3	1.0961e+10	1.0895e+10	3.7478e+17	9.2462e+09
n=4	1.0488e+10	1.0372e+10	3.6052e+19	9.4638e+09
n=5	1.0406e+10	1.0229e+10	2.7762e+21	9.7303e+09
n=8	1.0691e+10	1.0248e+10	1.1832e+27	1.0405e+10
n=10	1.1015e+10	1.0323e+10	8.7312e+30	1.0821e+10
n=15	1.2053e+10	1.0491e+10	2.2292e+38	1.1951e+10
n=20	1.3376e+10	1.0603e+10	4.8088e+43	1.3307e+10

TABLE 4.1
The condition number of Eq. (3.15) for different number of nodes and preconditioners.

Note that for special systems, SIKDC-EI can be stable. Consider the index two system

$$\begin{cases} x_1' = x_1 = f_1(x_1), \\ x_2' = 2x_1 - 10^5 x_2 + z + (10^5 + 1)\exp(t) = f_2(x_1, x_2, z), \\ 0 = x_1 + x_2 = g(x_1, x_2) \end{cases}$$

where the algebraic variable z does not appear in the first equation. For this problem, the eigenvalue distribution of the matrix $J_{EI} + I$ is almost identical to that of the FI-KDC as shown in Fig. 4.10, and the SIKDC-EI approach becomes stable. In our numerical simulation, we use 7 Radau nodes for each time step, and march from $t = 0.2$ to $t = 1.2$ using step-size $\Delta t = 0.1$. As the system is linear, no Newton iteration is required and we use the GMRES method to solve the preconditioned system. In Fig. 4.11, we compare the residual after each GMRES step for both the

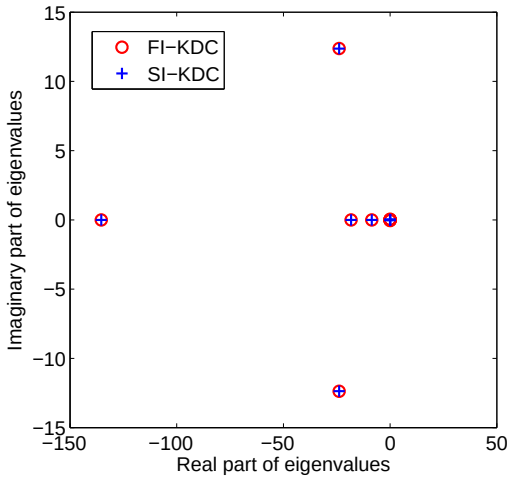


FIG. 4.10. *Comparing the SI-KDC and the FI-KDC for index 2 linear DAE.*

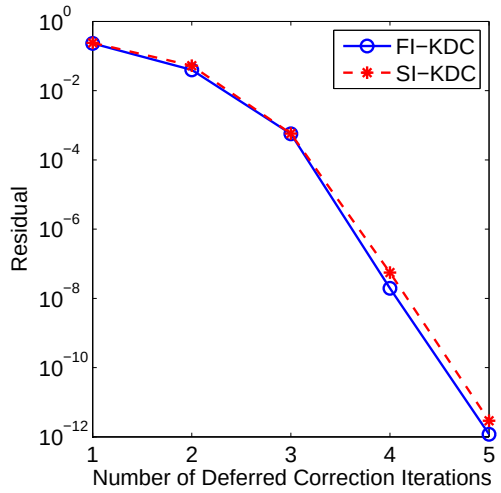


FIG. 4.11. *Comparing the SI-KDC and the FI-KDC for index 2 linear DAE.*

SIKDC-EI and FI-KDC methods for one time step. Again, the convergence of the SIKDC-EI approach is very similar to that of the FI-KDC.

We have also studied other higher-index DAE systems and our analysis and numerical experiments show that designing optimal semi-implicit schemes for stiff DAE systems are highly problem dependent, and requires detailed study of the linearized system.

5. Conclusion. In this paper, semi-implicit KDC (SI-KDC) techniques are introduced for stiff DAE systems and compared with fully-implicit KDC (FI-KDC) methods. The SI-KDC technique treats the non-stiff components in the SDC preconditioner using explicit methods and solves the stiff parts using implicit schemes. Our analysis and numerical experiments show that when proper semi-implicit schemes are

used, the eigenvalues from the Jacobian matrix of the preconditioned implicit function $\tilde{\mathbf{H}}_{SI}$ are similarly distributed as those from the FI-KDC method. However, when marching from t_j to t_{j+1}, the SI-KDC preconditioner only requires the solution of a simplified system, compared with the generally fully nonlinear system in the FI-KDC discretization. The SI-KDC methods are therefore numerically less expensive than the FI-KDC methods for the same accuracy requirements. However, our analysis also shows that unlike the ODE cases, the existence of algebraic equations and algebraic variables makes the design of optimal semi-implicit schemes a challenging task for higher-index DAE systems, and requires detailed analysis and understanding of the underlying system. It is therefore unrealistic to develop general purpose SI-KDC numerical solvers for higher-index DAEs with optimal efficiency.

Note that the SI-KDC methods can be generalized to many partial differential equation systems with algebraic constraints. In particular, we are currently working on the SI-KDC techniques for the Navier-Stokes equations and a two-scale partial differential equation model for water treatment processes. Results along these directions will be reported in the future.

REFERENCES

[1] V. Ajjarapu, *Computational Techniques for Voltage Stability Assessment and Control*, Springer, 2007.

[2] G. Akrivis, M. Crouzeix, and C. Makridakis, *Implicit-explicit multistep methods for quasi-linear parabolic equations*, Numer. Math., 82:521–541, 1999.

[3] B. K. Alpert, and V. Rokhlin, *A fast algorithm for the evaluation of Legendre expansions*, SIAM J. on Sci. and Stat. Computing, 12,158-179, 1991.

[4] P. M. Anderson, *Analysis of Faulted Power Systems*, Wiley-IEEE Press, 1995.

[5] U. M. Ascher, and L. R. Petzold, *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*, SIAM, Philadelphia, 1998.

[6] U. M. Ascher, S. J. Ruuth, and B. Wetton, *Implicit-Explicit Methods For Time-Dependent PDEs*, SIAM J. Numer. Anal, 32, 797–823, 1997.

[7] U. M. Ascher, S. J. Ruuth, and R. J. Spiteri, *Implicit-Explicit Runge-Kutta Methods for Time-Dependent Partial Differential Equations* Appl. Numer. Math, 25, 151–167, 1997.

[8] K. Atkinson, *An Introduction to Advanced Numerical Analysis*, 2nd edition, John Wiley, 1989.

[9] W. Auzinger, H. Hofstatter, W. Kreuzer, and E. Weinmuller, *Modified defect correction algorithms for ODEs. Part I: General theory*, Numer. Algorithms, 36: 135-156, 2004.

[10] R. Barrett, et al., *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, 2nd Edition, SIAM, Philadelphia, 1994.

[11] S. Boscarino, *Erroro analysis of IMEX Runge-Kutta methods derived from differential-algebraic systems*, SIAM J. Numer. Anal., Vol. 45, No. 4, pp. 1600–1621, 2007.

[12] S. Boscarino, *On an accurate third order implicit-explicit Runge-Kutta method for stiff problems*, Appl. Numer. Math, Vol. 59, pp. 1515–1528, 2009.

[13] A. Bourlioux, A.T. Layton, and M.L. Minion, *High-Order Multi-implicit spectral deferred correction methods for problems of reactive flow*, J.Comput. Phys., 189, 351–376, 2003.

[14] K. E. Brenan, S. L. Campbell, and L. R. Petzold, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, SIAM, Philadelphia, 1995.

[15] , S. Bu, J. Huang, and M.M. Minion, *Semi-implicit Krylov Deferred Correction Methods for Ordinary Differential Equations*, Proceedings of the American Conference on Applied Mathematics, Houston, May, 2009.

[16] M. P. Calvo, and C. Palencia, *Avoiding the order eduction of Runge-Kutta methods for linear initial boundary value problems*, Math. Comput., 71, 1529-1543, 2002.

[17] M. P. Calvo, J. de Frutos, and J. Novo, *Linearly implicit Runge-Kutta methods for advection-reaction-diffusion equations*, Appl. Numer. Math., 37:535–549, 2001.

[18] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral Methods in Fluid Dynamics*, Springer-Verlag, 1988.

[19] K. Chen, A. Iserles, and P. G. Ciarlet (Editors), *Matrix Preconditioning Techniques and Applications*, Cambridge University Press, 2005.

[20] K. Dekker, and J. G. Verwer, *Stability of Runge-Kutta methods for stiff nonlinear differential equations*. CWI Monographs. North-Holland, 1984.

[21] A. Dutt, L. Greengard, and V. Rokhlin, *Spectral deferred correction methods for ordinary*

differential equations, BIT, 40(2) 241-266, 2000.

[22] A. Dutt, M. Gu, and V. Rokhlin, *Fast Algorithms for Polynomial Interpolation, Integration, and Differentiation*, SIAM J. on Num. Anal., 33(5), 1689-1711, 1996.

[23] J. Frank, W. Hundsdorfer and J. G. Verwer, *Stability of Implicit-Explicit Linear Multistep Methods*, Applied Numerical Mathematics: Transactions of IMACS, Vol.25,2–3, 1997

[24] D. Gottlieb, and S. S. Orszag, *Numerical Analysis of Spectral Methods*, SIAM, Philadelphia, 1977.

[25] L. Greengard, *Spectral Integration and Two-Point Boundary Value Problems*, SIAM J. Num. Anal. 28, 1071-1080 1991.

[26] L. Greengard, and V. Rokhlin, *A Fast Algorithm for Particle Simulations*, J. Comput. Phys., 73, 325–348, 1987.

[27] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*, Springer-Verlag, 2002.

[28] E. Hairer, C. Lubich, and M. Roche, *The Numerical Solution of Differential-Algebraic Systems by Runge-Kutta Methods*, Springer-Verlag, 1989.

[29] E. Hairer, and G. Wanner, *Solving Ordinary Differential Equations II*, Springer, 1996.

[30] J. Huang, J. Jia, and M. Minion, *Accelerating the Convergence of Spectral Deferred Correction Methods*, J. of Comp. Physics, 214(2), 633–656 , 2006.

[31] J. Huang, J. Jia, M. Minion, *Arbitrary Order Krylov Deferred Correction Methods for Differential Algebraic Equations*, Comput. Phys., 221,(2), 739–760, 2007.

[32] J. Jia, J. Hunag, *Krylov deferred correction accelerated method of lines transpose for parabolic problems*, J. Comput Phys, 227(3),1739-1753, 2008.

[33] C. T. Kelly, *Iterative Methods for Linear and Nonlinear Equations*, SIAM, 1995.

[34] C. T. Kelly, *Solving Nonlinear Equations with Newton's Method*, SIAM, 2003.

[35] C. A. Kennedy and M. H. Carpenter, *Additive Runge-Kutta schemes for convection-diffusion-reaction equations.*, Appl. Numer. Math., 44:139–181, 2003.

[36] D.A. Knoll, D.E. Keyes, *Jacobian-free Newton Krylov methods: A survey of approaches and applications*, J. Comput. Phys. 193 (2004) 357-397.

[37] P. Kolm, S. Jiang, and V. Rokhlin, *Quadruple and octuple layer potentials in two dimensions. I. Analytical apparatus*, Appl. Comput. Harmon. Anal. 14, no. 1, 2003.

[38] A. Kurita, H. Okubo, K. Oki, S. Agematsu, D. B. Klapper, N. W. Miller, W. W. Price , J. J. Jr.Sanchez-Gasca, K. A. Wirgau, T. D. Younkins , *Multiple time-scale power system dynamic simulation* , Power Systems, IEEE Transactions on On page(s): 216- 223, Vol. 8, Issue: 1, Feb. 1993.

[39] A. T. Layton, and M. L. Minion, *Conservative Multi-Implicit Spectral Deferred Correction Methods for Reacting Gas Dynamics*, J. Comput. Phys, 194(2), 697-714, 2004.

[40] A. T. Layton, and M. L. Minion, *Implications of the choice of quadrature nodes for Picard integral deferred corrections methods for ordinary differential equations*, BIT, 45(2), 341-373, 2005.

[41] A. T. Layton and M. L. Minion, *Implications of the Choice of Predictors for Semi-Implicit Picard Integral Deferred Correction Methods*, Comm.App. Math. and Comp. Sci., 2(1),1–34, 2007.

[42] F. Milano, *An Open Source Power System Analysis Toolbox*, Power Systems, IEEE Transactions on, Vol. 20, No.3, 2005.

[43] M. L. Minion, *Higher-order Semi-implicit Projection Methods in Numerical Simulations of Incompressible Flows*, Papers from the workshop held in Half Moon Bay, CA, June 19-21, 2001. also LLNL Technical Report UCRL-JC-145295.

[44] M. L. Minion, *Semi-implicit spectral deferred correction methods for ordinary differential equations*, Comm. Math. Sci., 1:471–500, 2003.

[45] M. L. Minion, *Semi-Implicit Projection Methods for Incompressible Flow based on Spectral Deferred Corrections*, Appl. Numer. Math., 48 (3-4), 369-387, 2004.

[46] N. Mohan, *First Course on Power Systems*, MNPERE, 2006.

[47] L. Pareschi and G. Russo, *Implicit-Explicit Runge-Kutta schemes for stiff systems of differential equations*, volume 3, pages 269–287. Nova Science, 2000.

[48] J. O. Pessanha and A. A. Paz, *Testing a differential-algebraic equation solver in Long-term Voltage Stability Simulation*, Mathematical Problems in Engineering, 2006.

[49] V. Pereyra, *Iterated Deferred Correction for Nonlinear Boundary Value Problems*, Numer. Math. 11, 111–125 1968.

[50] L. R. Petzold, *A Description of DASSL: A Differential-Algebraic System Solver*, SAND82-8637, Sandia National Lab, 1982.

[51] A. Rangan, *Adaptive Solvers for Partial Differential and Differential-Algebraic Equations*, Ph.D. Thesis, University of California at Berkeley , 2003.

[52] A. Rangan, *Deferred Correction Methods for Low Index Differential Algebraic Equations*, BIT, Vol.43, No.1,1-18, 2003.

[53] Y. Saad, and M. H. Schultz. *GMRES: a generalized minimal residual algorithm for solving non-symmetric linear systems*, SIAM J. Sci. Stat. Comp., 7:856–869, 1986.

[54] J. M. Sanz-Serna, J. G. Verwer, and W. H. Hundsdorfer, *Convergence and Order Reduction of Runge-Kutta Schemes Applied to Evolutionary Problems in Partial Differential Equations*, Numer. Math., 50, 405-418, 1986.

[55] J. W. Shen and X. Zhong, *Semi-implicit Runge-Kutta schemes for the non-autonomous differential equations in reactive flow computations*, Proceedings of the 27th AIAA Fluid Dynamics Conference, AIAA, June 1996.

[56] L. N. Trefethen, and M. R. Trummer, *An instability phenomenon in spectral methods*, SIAM J. Numer. Anal, 24, 1008-1023, 1987

[57] P. K. Vijalapura, J. Strain, and S. Govindjee, *Fractional step methods for index-1 differential-algebraic equations*, J. of Comp. Phys., 203(1), 305-320, 2005.

[58] D. Yong, V. Ajjarapu, *A Decoupled Time-Domain Simulation Method via Invariant Subspace Partition for Power System Analysis*, Power Systems, IEEE Transactions on On page(s): 11- 18, Volume: 21, Issue: 1, Feb. 2006

[59] 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. Proceedings of International Astronomical Union, Symposium 25, 1964.

[60] P. E. Zadunaisky, *On the Estimation of Errors Propagated in the Numerical Integration of Ordinary Differential Equations*, Numer. Math. 27, 21–40 1976.