Semi-implicit Krylov Deferred Correction Methods for Ordinary Differential Equations

Sunyoung Bu
University of North Carolina
Department of Mathematics
CB # 3250, Chapel Hill
USA
agatha@email.unc.edu

Jingfang Huang
University of North Carolina
Department of Mathematics
CB # 3250, Chapel Hill
USA
huang@email.unc.edu

Michael L.Minion
University of North Carolina
Department of Mathematics
CB # 3250, Chapel Hill
USA
minion@email.unc.edu

Abstract: In the recently developed Krylov deferred correction (KDC) methods for ordinary differential equation initial value problems [11], 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 a Newton-Krylov method. Existing analyses show that these KDC methods are super convergent, A-stable, B-stable, symplectic, and symmetric. In this paper, we investigate the efficiency of semi-implicit KDC (SI-KDC) methods for problems which can be decomposed into stiff and non-stiff components. Preliminary analysis and numerical results show that SI-KDC methods display very similar convergence of Newton-Krylov iterations compared with fully-implicit (FI-KDC) methods but can significantly reduce the computational cost in each SDC iteration for the same accuracy requirement for certain problems.

Key–Words: Ordinary Differential Equation, Krylov Deferred Correction, Semi-Implicit Schemes, Preconditioner

1 Introduction

Many numerical techniques have been developed for the accurate and efficient solution of stiff ordinary differential equation (ODE) initial value problems, including linear multi-step methods, Runge-Kutta methods, and operator splitting techniques [1, 5, 8, 10]. In this paper, we focus on the recently developed Krylov deferred correction (KDC) technique for ODE initial value problems [11] which was motivated by the desire to accelerate the convergence of the iterative spectral deferred correction (SDC) technique [6]. For linear systems, it is shown in [11] that the iterates in the original SDC method are equivalent to a Neumann series expansion for a particular preconditioned collocation formulation. Instead of simply accepting this Neumann series expansion solution, the KDC method searches for the optimal solution in the corresponding Krylov subspace of SDC iterates. Numerical experiments show that the KDC method can accelerate the convergence and eliminate the order reduction phenomena observed in the SDC approach [16, 17]. For general nonlinear systems, one can either adapt a linearly-implicit approach or use the SDC iterations as preconditioners for the collocation formulation and apply a Jacobian-free Newton-Krylov method [13, 15] directly to the resulting preconditioned system. Preliminary analyses show that when Gaussian quadrature nodes are used, the KDC method is an efficient method to solve the Gauss collocation (Gauss Runge-Kutta) formulation which is super convergent, A-stable, B-stable, symplectic, and symmetric. In the numerical implementations in [11, 12], explicit low-order time integrators are used for non-stiff or mildly stiff problems, while implicit schemes are applied to very stiff systems.

However, for problems with both stiff and non-stiff components, it is usually advantageous to apply a semi-implicit time integration scheme. Semi-implicit versions of classical schemes such as the “Implicit-Explicit” (IMEX) linear multistep methods or additive Runge-Kutta methods have already been developed for ordinary and partial differential equations [2, 3, 14]. Similar semi-implicit SDC methods have also been developed [4, 17]. In this paper, we apply semi-implicit techniques to the KDC methods by applying different preconditioners to different components in the SDC iterations. Our numerical results presented here show that using a semi-implicit precondioner has a negligible effect on the rate of convergence of the SDC iterates compared to a fully-implicit KDC method. On the other hand, the SI-KDC method can have significantly smaller computational cost per iterate.

We organize this paper as follows. In Sec. 2, we briefly describe the KDC methods. In Sec. 3, the semi-implicit KDC methods are discussed and their convergence properties are analyzed. In Sec. 4, preliminary numerical results are presented to compare the SI-KDC methods with FI-KDC, and finally in Sec. 5, we summarize our results and discuss applications and further improvements.

2 Krylov Deferred Correction Methods

In this section, we discuss the KDC technique for a general ODE system

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

Here we follow the derivation of KDC methods for differential algebraic equations (DAEs) in [12] by defining $Y(t) = y'(t)$ as the new unknown and considering the equivalent Picard type integral equation

$$Y(\tau) = F(t, y_0 + \int_0^t Y(\tau) d\tau). \tag{2}$$

A variant of the KDC method based on the more traditional Picard integral equation as in [11] is also possible.

To march one step from $t = 0$ to $t = \Delta t$, we choose substeps corresponding to the p Gaussian nodes in $[0, \Delta t]$. Note that once the discretized solutions $\mathbf{Y} = [\mathbf{Y_1}, \mathbf{Y_2}, \cdots, \mathbf{Y_p}]^\mathbf{T}$ at the Gaussian nodes are obtained, we can derive a degree $p-1$ Legendre polynomial $P(t)$ which interpolates $\mathbf{Y}$ at the nodes. A continuous approximate solution to $y(t)$ in Eq. (1) can then be obtained by integrating $P(t)$ analytically, and the discretized solution $\mathbf{y}$ is obtained by evaluating the resulting degree p polynomial at the same set of nodes. Using the notation from [6], we call the linear mapping $\mathbf{y} = \mathbf{\Delta t S Y}$ the spectral integration operator, where $\mathbf{S}$ denotes the component-wise tensor product of the spectral integration matrix. To derive equations for $\mathbf{Y}$, we apply the collocation formulation which requires the interpolating polynomial $P(t)$ to satisfy Eq. (2) at the Gaussian nodes, i.e.,

$$\mathbf{Y} = \mathbf{F}(\mathbf{t}, \mathbf{y_0} + \mathbf{\Delta t S Y}), \tag{3}$$

where $\mathbf{y}_0 = [y_0, y_0, \cdots, y_0]^T$ is the vector of initial values. We symbolically denote Eq. (3) as $\mathbf{H}(\mathbf{Y}) = \mathbf{0}$. It has been shown that Eq. (3) is equivalent to a p-stage Runge-Kutta method, and is often referred to as the Gauss Runge-Kutta (GRK) method [9]. The GRK formulations have excellent accuracy and stability properties as summarized by the following theorem from [9].

Theorem 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.*

However, as solutions at different times (nodes) are coupled in Eq. (3), the direct solution for nonlinear systems is in general numerically inefficient especially for large p. As far as we know, the largest p used in existing implementations is approximately $p = 5$ (order 10). On the other hand, the discretized solution at a specific time in most existing low-order time integrators only couples with the solutions at previous times, hence the numerical algorithm is more efficient due to the reduced problem size. To take advantage of the excellent accuracy and stability properties of the GRK formulation and the efficiency of the low-order time integrators, we introduce the Krylov deferred correction approach for solving Eq. (3).

Assuming a provisional solution $\tilde{\mathbf{Y}} = [\tilde{Y_1}, \tilde{Y_2}, \cdots, \tilde{Y_p}]^T$ is obtained at the Gaussian nodes using a low-order time integrator and denoting the corresponding interpolating polynomial approximation to the solution as $\tilde{Y}(t)$, we define an equation for the error $\delta(t) = Y(t) - \tilde{Y}(t)$ by

$$\tilde{Y}(t) + \delta(t) = F(t, y_0 + \int_0^t (\tilde{Y}(\tau) + \delta(\tau)) d\tau). \tag{4}$$

Separating the provisional solution and the error in Eq. (4) at node t_{m+1}, i.e.

$$\begin{aligned} &\tilde{Y}(t_{m+1}) + \delta(t_{m+1}) = F(t_{m+1}, y_0 + \\ &\textstyle\int_0^{t_{m+1}} \tilde{Y}(\tau) d\tau + (\int_0^{t_m} + \int_{t_m}^{t_{m+1}}) \delta(\tau) d\tau), \end{aligned} \tag{5}$$

allows a lower-order method to be applied to approximate $\delta(t)$ (denoted by $\tilde{\boldsymbol{\delta}}$). For the explicit (forward) Euler method, $\tilde{\boldsymbol{\delta}} = [\tilde{\delta}_1, \tilde{\delta}_2, \cdots, \tilde{\delta}_p]^T$ is obtained by solving the "decoupled" equation

$$\begin{aligned} &\tilde{Y}_{m+1} + \tilde{\delta}_{m+1} = F(t_{m+1}, y_0 + \\ &(\Delta t \mathbf{S} \tilde{\mathbf{Y}})_{\mathbf{m+1}} + \textstyle\sum_{\mathbf{l=1}}^{\mathbf{m+1}} \mathbf{\Delta t_l} \tilde{\delta}_{\mathbf{l-1}}), \end{aligned} \tag{6}$$

where $\Delta t_{l+1} = t_{l+1} - t_l$ and t_0 and δ_0 are set to 0. The matrix form of Eq. (6) can be written as

$$\tilde{\mathbf{Y}} + \tilde{\boldsymbol{\delta}} = \mathbf{F}(\mathbf{t}, \mathbf{y_0} + \mathbf{\Delta t S} \otimes \tilde{\mathbf{Y}} + \mathbf{\Delta t} \tilde{\mathbf{S}} \tilde{\boldsymbol{\delta}}) \tag{7}$$

where $\Delta t \tilde{S}$ is a lower triangular matrix representing the rectangle rule approximation of the spectral integration operator. Matrix forms for other low-order schemes can be obtained similarly.

In the SDC methods, the approximation $\tilde{\boldsymbol{\delta}}$ of the error is added to $\tilde{\mathbf{Y}}$ to obtain an improved provisional solution, and a new error equation is then derived and solved by efficient low-order schemes. This procedure is repeated until the error approximation is smaller than a prescribed tolerance or a maximum number of iterations is reached. In [11], it was shown that the SDC approach is equivalent to a Neumann series expansion solution of the GRK formulation preconditioned by the low-order time integrators. For stiff systems, as there may exist a few “bad” eigenvalues, order reduction in the original SDC method is often observed unless extremely small time stepsizes are chosen.

Instead of simply accepting the Neumann series solution, in the KDC methods, we consider Eq. (7) 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 zero of $\tilde{\mathbf{H}}$ also solves the original collocation formulation $\mathbf{H}(\mathbf{Y}) = \mathbf{0}$. Moreover, it was shown in [12] that because the low-order method solves a “nearby” problem, the Jacobian of $\tilde{\mathbf{H}}$ is closer to identity than that of $\mathbf{H}$, therefore $\tilde{\mathbf{H}} = \mathbf{0}$ is better conditioned and can be solved efficiently by the Newton-Krylov methods where the optimal solution is sought for in the Krylov subspace instead of simple iterative refinements. As Krylov subspace methods are used for each linearized equation which is preconditioned by the SDC techniques, we refer to this new scheme as the “Krylov deferred correction” method.

There are two “Newton” iterations involved in the KDC methods: In the “outer” Newton-Krylov iterations for the preconditioned system, a “Jacobian-free” approach is in general applied where the matrix vector multiplication is approximated by a “forward difference” approach and each function evaluation is simply one SDC iteration to derive $\tilde{\boldsymbol{\delta}}$; Inside each SDC iteration, to march from t_m to t_{m+1} using a low-order scheme, a Newton type approach (e.g., the simplified Newton’s method) is commonly applied if the resulting discretized equation is nonlinear. This will be referred to as the “inner” Newton iteration.

As KDC methods are simply efficient ways to solve the GRK formulation, their accuracy and stability properties are therefore identical to those of the GRK formulation when the solution converges. The efficiency (rate of convergence) of the KDC methods depends on the choice of low-order preconditioners, the efficiency of the “outer” Newton-Krylov and “inner” Newton type methods, and the properties of the considered ODE system. It has been shown in [11, 12] that the KDC methods allow “optimal” stepsizes for prescribed accuracy requirements, and can effectively eliminate the order reduction phenomena observed in many existing time integration schemes. In general, for stiff ODE systems, numerical experiments show that the KDC methods are more efficient than the SDC schemes for the same accuracy requirement, and are competitive alternatives for existing state-of-art solvers especially for long-time simulations with high accuracy requirements.

3 Semi-Implicit Preconditioning Techniques

Consider a general nonlinear ODE system which can be split into two parts

$$y'(t) = F_E(t, y(t)) + F_I(t, y(t)), \tag{8}$$

where F_E represents the non-stiff component and F_I the stiff part. When explicit low-order schemes are used to precondition Eq. (8) in the KDC methods, due to the stability region constraints, extremely small time stepsizes have to be used to avoid the overflows in the solution process. The resulting explicit KDC (Ex-KDC) is therefore inefficient even though no inner Newton iterations are required. The fully-implicit KDC (FI-KDC) approach based on implicit low-order preconditioners, on the other hand, allows much larger stepsizes, however, inner Newton iterations must be performed in each SDC iteration.

For stiff ODE systems with a linear or semi-linear stiff component, it is possible to improve the algorithm efficiency by introducing the semi-implicit KDC (SI-KDC) technique, in which the non-stiff component is discretized using an explicit scheme, while an implicit method is applied to the linear stiff part. Using the error equation and spectral integration matrix as discussed in Sec. 2, the SI-KDC technique can be presented in the matrix form as

$$\begin{aligned} \tilde{\boldsymbol{\delta}}(t) = {} & F_E(t, y_0 + \Delta t \mathbf{S}\tilde{\mathbf{Y}} + \mathbf{\Delta t}\tilde{\mathbf{S}}_{\mathbf{E}}\tilde{\boldsymbol{\delta}}) \\ & + F_I(t, y_0 + \Delta t \mathbf{S}\tilde{\mathbf{Y}} + \mathbf{\Delta t}\tilde{\mathbf{S}}_{\mathbf{I}}\tilde{\boldsymbol{\delta}}) - \tilde{\mathbf{Y}}(\mathbf{t}), \end{aligned} \tag{9}$$

where $\tilde{\mathbf{S}}_{\mathbf{E}}$ represents strictly lower triangular integration matrix for an explicit scheme, and $\tilde{\mathbf{S}}_{\mathbf{I}}$ is the lower triangular matrix for an implicit method. The implicit function $\tilde{\mathbf{H}}$ is then similarly defined as in Sec. 2, and a Newton-Krylov method is applied for its efficient solution. Notice that when the nonlinear component is non-stiff, using $\tilde{\mathbf{S}}_{\mathbf{E}}$ in SI-KDC won’t introduce instability as in a fully explicit method even when the chosen stepsize is comparable to that in FI-KDC. Also, as the converged solution in SI-KDC solves the collocation formulation, its accuracy is the same as those derived using other preconditioning techniques (it will, however, slightly change the condition number of the original system). Therefore, we conclude that the SI-KDC methods have similar accuracy and stability properties as the FI-KDC schemes.

To understand the efficiency of the SI-KDC methods, notice that when the stiff component is linear, unlike in the FI-KDC methods, no inner Newton iteration is required in a Newton-Krylov solution in the SI-KDC. Therefore, we focus on the eigenvalue distributions of the Jacobian matrix for the implicit function $\tilde{\mathbf{H}}$ in SI-KDC, which approximately measures the number of outer Newton-Krylov iterations. In the following, using the linearized stiff ODE system

$$\begin{cases} y_t = \lambda_1 A_1 y + \lambda_2 A_2 y + F(t), \\ y(0) = y_0 \end{cases} \tag{10}$$

as an example, we explicitly derive the Jacobian matrix and compare it with that from the FI-KDC scheme. In the formula, we assume λ_1 is a large negative number, λ_2 is $O(1)$, and all eigenvalues of A_1 and A_2 are $O(1)$ and positive, i.e., $\lambda_1 A_1 y$ represents the stiff component and $\lambda_2 A_2 y$ the non-stiff part.

For the linearized equation, the discretized error equation takes the form

$$\begin{aligned} \tilde{\mathbf{Y}} + \tilde{\boldsymbol{\delta}} = \lambda_1 A_1(\mathbf{y}_0 + \triangle t\mathbf{S}\tilde{\mathbf{Y}} + \triangle \mathbf{t}\tilde{\mathbf{S}}_{\mathbf{I}}\tilde{\boldsymbol{\delta}}) \\ +\lambda_2 A_2(\mathbf{y}_0 + \triangle t\mathbf{S}\tilde{\mathbf{Y}} + \triangle \mathbf{t}\tilde{\mathbf{S}}_{\mathbf{E}}\tilde{\boldsymbol{\delta}}) + \mathbf{F}. \end{aligned} \tag{11}$$

The implicit function $\tilde{\delta} = \tilde{H_{SI}}(\tilde{Y})$ can then be explicitly written as

$$\begin{aligned} \tilde{\delta} = \tilde{H_{SI}}(\tilde{Y}) = (I - \lambda_1 A_1 \triangle t\tilde{\mathbf{S}}_{\mathbf{I}} - \boldsymbol{\lambda_2 \mathbf{A}_2} \triangle \mathbf{t}\tilde{\mathbf{S}}_{\mathbf{E}})^{\mathbf{-1}} \\ ((\lambda_1 A_1 + \lambda_2 A_2)(y_0 + \triangle tS\tilde{Y}) + \mathbf{F} - \tilde{\mathbf{Y}})), \end{aligned} \tag{12}$$

and the Jacobian matrix $J_{\tilde{H_{SI}}} = \frac{\partial\tilde{\delta}}{\partial\tilde{Y}}$ of this explicit form of $\tilde{H_{SI}}$ is

$$\begin{aligned} J_{\tilde{H_{SI}}} = (I - \lambda_1 A_1 \triangle t\tilde{\mathbf{S}}_{\mathbf{I}} - \boldsymbol{\lambda_2 \mathbf{A}_2} \triangle \mathbf{t}\tilde{\mathbf{S}}_{\mathbf{E}})^{\mathbf{-1}} \\ ((\lambda_1 A_1 + \lambda_2 A_2)\triangle t\mathbf{S} - \mathbf{I}) . \end{aligned} \tag{13}$$

Similarly, repeating this process for the FI-KDC method, we get

$$\begin{aligned} J_{\tilde{H_{FI}}} = (I - \lambda_1 A_1 \triangle t\tilde{\mathbf{S}}_{\mathbf{I}} - \boldsymbol{\lambda_2 \mathbf{A}_2} \triangle \mathbf{t}\tilde{\mathbf{S}}_{\mathbf{I}})^{\mathbf{-1}} \\ ((\lambda_1 A_1 + \lambda_2 A_2)\triangle t\mathbf{S} - \mathbf{I}) . \end{aligned} \tag{14}$$

Notice that as λ_1 is a large negative number, the dominating term in both Jacobian matrices is $\lambda_1 A_1 \triangle t\tilde{\mathbf{S}}_{\mathbf{I}}$. The eigenvalues of the Jacobian matrix from SI-KDC are therefore similarly distributed as those from FI-KDC. In Fig. 1, we set $\lambda_1 = -10^4$ and $\lambda_2 = 1$, and show the almost identical eigenvalue distributions for Jacobian matrices from SI-KDC and FI-KDC. We therefore conclude that the number of outer SI-KDC Newton-Krylov iterations is approximately the same as that in FI-KDC. Since no inner Newton iterations are required in SI-KDC, the SI-KDC technique is therefore more efficient compared with FI-KDC.

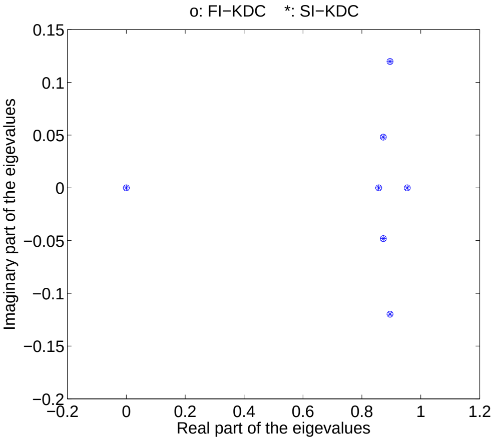

Figure 1: The eigenvalue distributions of $(J_{\tilde{H_{SI}}} + I)$ and $(J_{\tilde{H_{FI}}} + I)$ are almost identical.

4 Preliminary Numerical Results

In this section, we present two numerical examples to illustrate the performance of the SI-KDC methods.

4.1 Nonlinear ODE Example

First, we study a stiff nonlinear multi-mode ODE problem from [11] consisting of the N coupled equations

$$\begin{aligned} y_i'(t) = p_i'(t) - \lambda_i y_{i+1}(t)(y_i(t) - p_i(t)),\ i \leq N-1 \\ y_N'(t) = p_N'(t) - \lambda_i(y_i(t) - p_i(t)),\ i = N. \end{aligned} \tag{15}$$

The analytical solution is $y_i(t) = p_i(t)$ where $p_i(t) = 2 + cos(t+\alpha_i)$ and the phase parameter $\alpha_i = 2\pi i/N$. We set $N = 7$ and choose λ_i as $[1, 1, 1, 1, 1, 1, 10^7]$. These equations can be split into two groups: the first six equations are nonlinear and non-stiff, and the last equation is linear and stiff.

In the calculation, we march from t_0 = 0 to t_{final} = 3, and use 8 Gaussian nodes in each time step with Δt = 0.5. We apply the SI-KDC method with forward Euler for the non-stiff component and backward Euler for stiff part, and compare results with those from FI-KDC. In Fig. 2, we compare the accuracy and convergence. It can be seen that the number of outer Newton-Krylov iterations for the SI-KDC is comparable to that in FI-KDC for the same accuracy requirement. However, in the SI-KDC scheme, no inner Newton iterations are required, as compared with 10 inner Newton

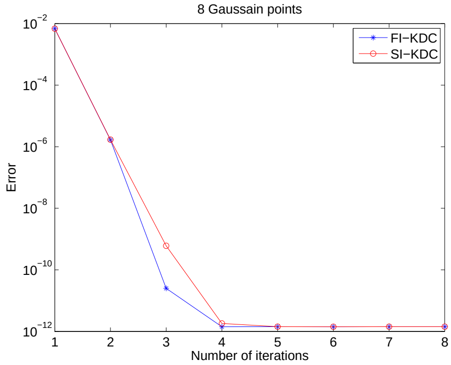

Figure 2: Comparing the convergence of SI-KDC and FI-KDC.

iterations required in the FI-KDC approach. The SI-KDC is therefore more efficient for the same accuracy requirement.

4.2 Van der Pol Problem

In our second example we consider the Van der Pol oscillator which after rescaling gives

$$y_1' = y_2, \tag{16}$$

$$y_2' = (-y_1 + (1 - y_1^2)y_2)/\epsilon. \tag{17}$$

This is a popular test problem for nonlinear stiff ODE solvers. In this problem, as ϵ approaches zero, the second equation becomes increasingly stiff. Notice that when the first equation is treated explicitly to update y_1, the second equation becomes linear with respect to y_2. Therefore, only linear equations appear in the low-order time marching scheme when an semi-implicit approach is applied.

In the experiment, we set $\epsilon = 10^{-6}$ and use 8 Gaussian points for each time step. We march from $t = 0$ to $t = 0.05$ using $\Delta t = 0.0125$. Our numerical experiments show that the number of outer Newton-Krylov iterations in the SI-KDC is comparable to that in FI-KDC for the same accuracy requirement and parameter settings. In the following, focusing on the restarted GMRES based Newton-Krylov method, we compare the convergence of the SI-KDC and FI-KDC methods. When a full GMRES orthogonalization scheme is used, as both the memory and number of operations grow rapidly when the number of iterations increases, a common practice is to use the restarted GMRES so the size of the Krylov subspace is bounded by a restarting value k_0. In general, large k_0 means better convergence properties of the Newton-Krylov method, at the cost of additional memory allocation and extra arithmetic operations.

In Fig. 3, we show how different choices of k_0 change the properties of the Newton-Krylov iterations, and compare the convergence of the SI-KDC to that of the FI-KDC method. In this example, the residual represents the 2-norm of the residual for the linearized equation. It can be seen that FI-KDC is optimal in stability and has better convergence properties in the outer Newton-Krylov iterations under the same parameter settings. However, the residual after each Newton-Krylov iteration in SI-KDC decays in a very comparable way as in FI-KDC. In each SDC iteration, approximately 10 inner Newton iterations are required in FI-KDC to march from t_m to t_{m+1}, while only one linear solve is needed in SI-KDC, we therefore conclude that the SI-KDC approach is more efficient than FI-KDC for this example.

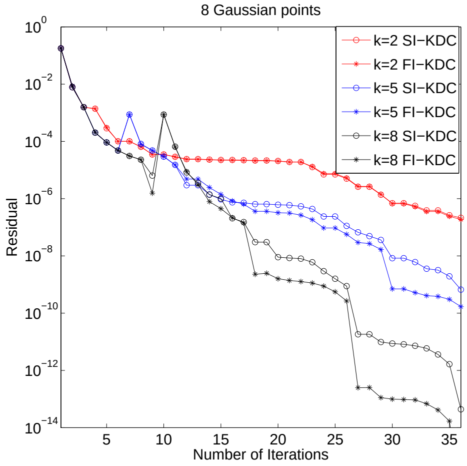

Figure 3: Comparing the convergence of GMRES(k_0) for different k_0 for SI-KDC and FI-KDC

Note that for fixed size k_0 in GMRES, when k_0 is large, unnecessary GMRES iterations will be performed, while much slower convergence is observed when k_0 is too small. Indeed, finding optimal parameters in the Newton-Krylov methods is an active research area. Our experiments indicate that dynamically chosing k_0 may result in optimal Newton-Krylov algorithms which converge super-linearly or even quadratically.

5 Conclusion

In this paper, semi-implicit KDC methods are introduced for stiff ODE systems with nonlinear non-stiff component and linear or semi-linear stiff part. Analyses and numerical experiments show that the SI-KDC methods are more efficient compared with the fully implicit KDC methods as no inner Newton iterations are required, while they have the same accuracy and stability properties as the FI-KDC schemes.

Currently, we are generalizing the SI-KDC ideas to differential algebraic equations (DAEs) (as in [12]) and partial differential equations (PDEs), and developing numerical packages based on the SI-KDC methods for applications in power systems and computational solid and fluid mechanics. Results along these directions will be reported in the future.

Acknowledgments: This work was supported by NSF under grant DMS0411920 and DMS0811130, and part of work was done when Bu was a visiting memeber of the Institute for Mathematics and Applications (IMA) at the University of Minnesota. Their support is thankfully acknowledged.

References:

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

[2] Uri M. Ascher, Steven J. Ruuth, and Raymond J. Spiteri, *Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations*, Appl. Numer. Math., **25** 151-167, 1997.

[3] Uri M. Ascher, Steven J. Ruuth, and Brian T. R. Wetton, *Implicit-explicit methods for time-dependent PDE's.* SIAM J. Numer. Anal., **32**, 797–823, 1995.

[4] 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.

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

[6] A. Dutt, L. Greengard, and V. Rokhlin, *Spectral deferred correction methods for ordinary differential equations*, BIT **40:** (2) 241-266, 2000.

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

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

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

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

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

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

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

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

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

[16] A. Layton, M. Minion, *Implications of the choice of quadrature nodes for Picard integral deferred corrections methods for ordinary differential equations*, BIT Numerical Mathematics, **45** (2), 341–373, 2005.

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

[18] 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.