Available online at www.sciencedirect.com
ScienceDirect
Applied Numerical Mathematics 58 (2008) 845–858
APPLIED NUMERICAL MATHEMATICS
www.elsevier.com/locate/apnum

On the choice of correctors for semi-implicit Picard deferred correction methods ☆

Anita T. Layton

Department of Mathematics, Duke University, NC 27708, USA

Available online 24 March 2007

Abstract

The goal of this study is to assess the implications of the choice of correctors for semi-implicit Picard integral deferred correction (SIPIDC) methods. The SIPIDC methods previously developed compute a high-order approximation by first computing a low-order provisional solution using a semi-implicit method and then using a first-order semi-implicit method to solve a series of correction equations, each of which raises the order of accuracy of the solution by one. In this study, we examine the efficiency of SIPIDC methods that instead use standard second-order semi-implicit methods to solve the correction equations. The accuracy, efficiency, and stability of the resulting methods are compared to previously developed methods, in the context of both nonstiff and stiff problems.

MSC: 65B05

Keywords: Semi-implicit methods; Deferred correction methods; Order reduction

1. Introduction

In applications such as combustion and transport of air pollutants, the dynamics involve processes with widely differing characteristic time scales. As a result, following the method-of-lines approach, with which the model partial differential equations (PDEs) are discretized in space, the resulting system of ordinary differential equations (ODEs) contain both stiff and nonstiff terms. Motivated by the need for an accurate and efficient time integrate of these ODEs, we developed a new class of semi-implicit methods that are based on the Picard integral deferred correction approach [11,8,3,12,9,10]. The semi-implicit Picard integral deferred correction (SIPIDC) methods are a generalization of the explicit and implicit spectral deferred correction (SDC) methods introduced in [4]. SDC methods use a low-order numerical method to compute an approximate solution with arbitrarily high order of accuracy. This is achieved by using the low-order numerical method to solve a series of correction equations, each of which increases the order of accuracy of the approximation. Recent works on the analysis of SDC methods can be found in [5,6].

The original SDC and SIPIDC methods [4,9] use a first-order method both to compute the provisional solution and to approximate the correction equations. In [10], we developed SIPIDC methods that use standard second- and third-order semi-implicit methods to compute the provisional solution, and showed that when BDF methods were

☆ This work was supported by the National Science Foundation, grant DMS-0340654.
E-mail address: alayton@math.duke.edu.

0168-9274/$30.00 © 2007 IMACS. Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.apnum.2007.03.003

used, the overall efficiency of the SIPIDC methods was improved. Thus, a reasonable question to ask is whether the efficiency of SIPIDC methods can be further improved by using a second- or higher-order semi-implicit method to approximate the correction equations. (Forward-backward Euler was used in the correction equation in [10].) Thus, a goal of this study is to investigate whether using second-order methods such as semi-implicit BDF or RK as the corrector[1] in a SIPIDC method will lead to improvement in efficiency. Because the spirit of SDC and PIDC methods is to generate high-order approximations using low-order standard methods, we restrict this investigation to second-order correctors. For the same predictor, SIPIDC methods using a higher-order corrector require fewer iterations of the correction equation to achieve the same overall order of accuracy relative to methods using a first-order corrector; however, it is not clear if the lower computational cost comes at the expense of a more significant loss in accuracy or if using such a higher-order corrector negatively affects the stability of PIDC methods. Thus, we aim to address these questions using analytical and numerical tests.

Another goal of this study is to further address the issue of order reduction exhibited by SIPIDC methods when applied to stiff problems. In a previous study [9], we showed that order reduction is alleviated, to some extent, via an appropriate choice of quadrature nodes. (To be specific, order reduction begins at smaller time steps when uniformly-spaced quadrature modes are used and when function values computed on the left-hand endpoint of the timestep are omitted in approximating the Picard integral associated with the stiff process [9].) In a follow-up study [10], we demonstrated, using both analytic and numerical tests, that the extent and character of order reduction depends critically on the order of the predictor. Specifically, when a kth-order semi-implicit BDF predictor is used, the convergence rate in the region of order reduction is $k-1$, compared to first-order convergence when an semi-implicit RK predictor is used.

In this study, we show that the magnitude of the error in the order reduction regime also depends, in part, on the choice of starting function values in advancing an approximation through a time step. The two choices that we consider are whether at the kth iteration of the correction equation, the starting values are set to (1) the intermediate solution values computed at the kth correction iteration of the previous time step, or (2) to solution values computed at the final iteration of the previous time step. Numerical results suggest that, when applied to a stiff problem, the former yields smaller errors in much of the order reduction regime.

2. Semi-implicit Picard integral deferred correction methods

In this section we present a short description of SIPIDC methods. A detailed derivation of the SIPIDC methods for ODEs and for an advection–diffusion–reaction equation can be found in [11] and [3], respectively. The ODE that we are interested in is given by

$$u'(t) = F_E\big(t, u(t)\big) + F_I\big(t, u(t)\big), \quad t \in [a, b], \tag{1}$$

$$u(a) = u_0, \tag{2}$$

where F_I is assumed to be significantly stiffer than F_E. Thus, SIPIDC methods compute $u(t)$ by integrating F_E explicitly and F_I implicitly.

Without loss of generality, a uniform time step $\Delta t > 0$ is assumed in the numerical discretization. Let $t_n = n\Delta t$, for $n = 0, 1, 2, \ldots$, be the nth time-level. As discussed below, to generate high-order approximations SIPIDC methods require the accurate approximation of a definite integral. To this end, quadrature nodes are chosen for each time interval $[t_n, t_{n+1}]$. Thus, in the integration of the solution from t_n to t_{n+1}, the time interval $[t_n, t_{n+1}]$ is divided into P equally-spaced subintervals $[t_{n,m}, t_{n,m+1}]$, where $t_{n,m} = t_n + m\Delta t_s$, for $m = 0, 1, \ldots, P-1$, where $\Delta t_s \equiv \Delta t/P$. For notational simplicity, the subscript n in $t_{n,m}$ is omitted and $t_{n,m}$ is written as t_m where there is no ambiguity. The interval $[t_m, t_{m+1}]$ is referred to as a substep.

A SIPIDC method proceeds as follow: a typically low-order provisional solution $\tilde{u}$ is computed at the t_m's. Then the corrections $\delta \equiv u - \tilde{u}$ is estimated iteratively, progressively increasing the accuracy of the computed solution.

For an arbitrary function $\psi(t)$, let ψ^k and ψ_m^k denote numerical approximations to $\psi(t)$ and $\psi(t_m)$, respectively, after k iterations. To advance the solution from t_n to t_{n+1}, SIPIDC methods first compute in a predictor step a pro-

[1] We will refer to the method used to compute the provisional solution and to approximate the correction equations in a given PIDC method as the *predictor* and *corrector*, respectively.

visional solution $\tilde{u}(t_m) \equiv u_m^0$, for $m = 0, 1, \ldots, P$, by means of a semi-implicit method The accuracy of $\tilde{u}$ can be improved using an estimate of its error or correction: $u(t) - \tilde{u}(t)$, denoted by $\delta(t)$. To estimate $\delta(t)$, one may iteratively approximate the solution to the correction equation

$$\delta(t) = \int_a^t \big(F_E\big(\tau, \tilde{u}(\tau) + \delta(\tau)\big) - F_E\big(\tau, \tilde{u}(\tau)\big) + F_I\big(\tau, \tilde{u}(\tau) + \delta(\tau)\big) - F_I\big(\tau, \tilde{u}(\tau)\big)\big)\,\mathrm{d}\tau + E\big(t, \tilde{u}(t)\big), \tag{3}$$

where E is the residual function given by

$$E\big(t, \tilde{u}(t)\big) = u_0 + \int_a^t \big(F_E\big(\tau, \tilde{u}(\tau)\big) + F_I\big(\tau, \tilde{u}(\tau)\big)\big)\,\mathrm{d}\tau - \tilde{u}(t). \tag{4}$$

A detailed derivation of (3) is given in [11]. If F_E and F_I are Lipschitz continuous in u, then (3) implies that $\|\delta(t) - E(t, \tilde{u})\| = \mathcal{O}(\Delta t^{s+1})$. Therefore, a $(s+k)$th order approximation for $\delta(t)$ can be computed from a $(s+k)$th order approximation for $E(t, \tilde{u})$ and a kth-order approximation to the integral on the right side of (3).

We now present a semi-implicit discretization of (3) using forward-backward Euler method, which increases the order of accuracy of u_{m+1}^0 by one. In Section 2.1, we will derive second-order discretization of (3). For an arbitrary function $\psi(t)$, let ψ_m^k denote a numerical approximation to $\psi(t_m)$ after k deferred correction iterations. Furthermore, for an arbitrary operator $F(t, u(t))$, let the numerical approximation $F(t_m, u_m^k)$ be written as $F(u_m^k)$. Using this notation, the semi-implicit discretized form of (3) at the kth iteration is given by

$$\delta_{m+1}^k = \delta_m^k + \Delta t_s\big(F_E\big(u_m^{k+1}\big) - F_E\big(u_m^k\big) + F_I\big(u_{m+1}^{k+1}\big) - F_I\big(u_{m+1}^k\big)\big) + E_{m+1}\big(u^k\big) - E_m\big(u^k\big). \tag{5}$$

Using the definition of the residual function E (4), one obtains a direct update equation that can be used to improve the accuracy of u^k. Let $\mathcal{Q}_m^{m+1}(F)$ be a numerical quadrature approximation to $\int_{t_m}^{t_{m+1}} F(\tau)\,\mathrm{d}\tau$, i.e.,

$$\mathcal{Q}_m^{m+1}(F) = \Delta t_s \sum_l q_l^m F_l. \tag{6}$$

Then at the kth iteration, one solves the following equation

$$\begin{aligned} u_m^{k+1} = {} & u_{m-1}^{k+1} + \Delta t_s\big(F_E\big(\tau, u_m^{k+1}\big) - F_E\big(\tau, u_m^k\big) + F_I\big(\tau, u_{m+1}^{k+1}\big) - F_I\big(\tau, u_{m+1}^k\big)\big) \\ & + \mathcal{Q}_n^{m+1}\big(F_E\big(u^k\big) + F_I\big(u^k\big)\big). \end{aligned} \tag{7}$$

The quadrature $\mathcal{Q}$ should have at least the same order of accuracy as the updated approximation u^{k+1}. As in [3,9–11], the quadrature $\mathcal{Q}_m^{m+1}$ is computed as the integral of an interpolating polynomial over the subinterval $[t_m, t_{m+1}]$.

In a previous study [9] we showed that an $L(\alpha)$-stable SIPIDC method can be constructed by omitting function values associated with the stiff component at t_n in the quadrature rule. We also showed that the accuracy of the quadrature associated with the explicit piece is improved, with no loss in stability, by including the left-hand endpoint in the nonstiff quadrature rule. This choice of quadrature rules, which we referred to as *LR* (for "left-right") in [9], is adopted in this study. To compute a Kth-order approximation, the quadrature $\mathcal{Q}$ should also have Kth-order accuracy. If $P+1$ nodes (or P substeps) are used in the interval $[t_n, t_{n+1}]$, uniform integration nodes yield order P accuracy for the integral $\mathcal{Q}_m^{m+1}$ over the subinterval $[t_m, t_{m+1}]$. Thus, to construct an Kth-order SIPIDC method that uses the LR quadrature rule (which excludes the left-hand endpoints in the stiff quadrature rule), $K+1$ nodes or K substeps are required.

In previous studies, SDC and SIPIDC methods were based on Euler methods [4,9]. In [10], we investigated the stability and efficiency of SIPIDC method that use standard second- and third-order methods in the prediction step. The methods studied include implicit–explicit (IMEX) backward difference formula (BDF) [2], IMEX Runge–Kutta (RK) methods [1,7], and multistep methods. The multistep methods were shown to exhibit instability when applied to sufficiently stiff problems. Thus, in this study, we limit our choice of predictors and correctors to forward-backward Euler, IMEX BDF, and IMEX RK methods.

2.1. High-order corrections

The accuracy of u^k can be increased by two, instead of one, by replacing the forward-backward Euler method used in the discretization of the correction equation (5) by a second-order method. We consider two second-order implementations of the correction steps: one based on the L-stable IMEX RK2 [1], and one based on IMEX BDF2 [2].

We first consider the second-order corrector based on the IMEX RK2 [1]. After discretizing the correction equation (3) using RK2, one obtains the update equations given by:

RK2-based:

$$\phi^{(1)}_{m+c_1} = u^k_m + c_1\Delta t_s\big(F_E\big(u^{k+1}_m\big) - F_E\big(u^k_m\big) + F_I\big(\phi^{(1)}_{m+c_1}\big) - F_I\big(u^k_{m+c_1}\big)\big) + \mathcal{Q}^{m+c_1}_m\big(F_E\big(u^k\big) + F_I\big(u^k\big)\big), \tag{8}$$

$$\begin{aligned}\phi^{(2)}_{m+1} &= u^k_m + \Delta t_s\big(c_2\big(F_E\big(u^{k+1}_m\big) - F_E\big(u^k_m\big)\big) + (1-c_2)\big(F_E\big(\phi^{(1)}_{m+c_1}\big) - F_E\big(u^k_{m+c_1}\big)\big)\\ &\quad + (1-c_1)\big(F_I\big(\phi^{(1)}_{m+c_1}\big) - F_I\big(u^k_{m+c_1}\big)\big) + c_1\big(F_E\big(\phi^{(2)}_{m+1}\big) - F_E\big(u^k_{m+1}\big)\big)\big)\\ &\quad + \mathcal{Q}^{m+1}_m\big(F_E\big(u^k\big) + F_I\big(u^k\big)\big),\end{aligned} \tag{9}$$

$$\begin{aligned}u^{k+1}_{m+1} &= u^{k+1}_m + \Delta t_s\big((1-c_1)\big(F_E\big(\phi^{(1)}_{m+c_1}\big) - F_E\big(u^k_{m+c_1}\big) + F_I\big(\phi^{(1)}_{m+c_1}\big) - F_I\big(u^k_{m+c_1}\big)\big)\\ &\quad + c_1\big(F_E\big(\phi^{(2)}_{m+1}\big) - F_E\big(u^k_{m+1}\big) + F_I\big(\phi^{(2)}_{m+1}\big) - F_I\big(u^k_{m+1}\big)\big)\big) + \mathcal{Q}^{m+1}_m\big(F_E\big(u^k\big) + F_I\big(u^k\big)\big),\end{aligned} \tag{10}$$

where $c_1 = 1 - \sqrt{2}/2$, $c_2 = -2\sqrt{2}/3$.

Note that quadrature rule in (8) approximates the integral $\int (F_E + F_I)\,\mathrm{d}\tau$ from t_m to $t_m + c_1\Delta t_s$ rather than the entire substep $[t_m, t_{m+1}]$. Because the correction equation (7) is derived from integral equations (3) and (4), and is not an ODE, strictly speaking (8)–(10) do not correspond to a discretization using IMEX RK2 owing to the integral term $\mathcal{Q}$. It is for this reason that (8)–(10) is labeled as "RK2-*based*" rather than simply "RK2."

To discretize the update equation arising from (3) using BDF2, we rewrite the equation as an ODE

$$\begin{aligned}&u'(t) + q'(t) = F_E\big(t, u^{k+1}\big) - F_E\big(t, u^k\big) + F_I\big(t, u^{k+1}\big) - F_I\big(t, u^k\big),\\ &u(t_m) = u^{k+1}_m\end{aligned} \tag{11}$$

where

$$q(t) \equiv \int_{t_m}^{t} \big(F_E\big(\tau, u^k\big) + F_I\big(\tau, u^k\big)\big)\,\mathrm{d}\tau. \tag{12}$$

Integrating (11) using BDF2, one obtains

$$\begin{aligned}\frac{3}{2}u^{k+1}_{m+1} &= 2u^{k+1}_m - \frac{1}{2}u^{k+1}_{m-1} + \Delta t_s\big(2F_E\big(u^{k+1}_m\big) - F_E\big(u^{k+1}_{m-1}\big) - 2F_E\big(u^k_m\big) + F_E\big(u^k_{m-1}\big)\\ &\quad + F_I\big(u^{k+1}_{m+1}\big) - F_I\big(u^k_{m+1}\big)\big) - \frac{3}{2}q_{m+1} + 2q_m - \frac{1}{2}q_{m-1}.\end{aligned} \tag{13}$$

Next, we approximate the q terms using the numerical quadrature (6), which yields

BDF2-based:

$$\begin{aligned}u^{k+1}_{m+1} &= 2u^{k+1}_m - \frac{1}{2}u^{k+1}_{m-1} + \Delta t_s\big(2F_E\big(u^{k+1}_m\big) - F_E\big(u^{k+1}_{m-1}\big) - 2F_E\big(u^k_m\big) + F_E\big(u^k_{m-1}\big) + F_I\big(u^{k+1}_{m+1}\big) - F_I\big(u^k_{m+1}\big)\big)\\ &\quad + \frac{3}{2}\mathcal{Q}^{m+1}_m\big(F_E\big(u^k\big) + F_I\big(u^k\big)\big) - \frac{1}{2}\mathcal{Q}^{m}_{m-1}\big(F_E\big(u^k\big) + F_I\big(u^k\big)\big).\end{aligned} \tag{14}$$

We have previously shown that the choice of quadrature nodes has a significant impact on the characteristic of order reduction of SIPIDC methods when applied to stiff problems [9]. In this study, we aim to show that the choice of the starting value(s) for each time-step also plays a role. Consider advancing a solution u from t_n to t_{n+1}, using a SIPIDC method that requires K deferred correction iterations and P substeps. In our previous work, the starting value(s) are set to be those value(s) computed at the last deferred correction iteration of the previous time step. That is, to compute u^0_1 during the predictor step, the value $u^K(t_{n-1,P})$ is used as the starting value of a single-step method (e.g., Euler

or RK), and the values $u^K(t_{n-1,P}), \ldots, u^K(t_{n-1,P-(m-1)})$ are used as the starting values of a m-step method (e.g., an mth-order BDF). This choice of starting values will be referred to as "fixed starting values".

Fixed starting values were used in previously-applied SIPIDC methods [9,10]. Consider, for example, a Kth-order SIPIDC method that uses IMEX BDF2 as the predictor. To advance the provisional solution through the first substep $[t_{n,0}, t_{n,1}]$, one solves

$$\frac{3}{2}u^0_{n,1} = 2u^K_{n-1,P} - \frac{1}{2}u^K_{n-1,P-1} + \Delta t_s\big(2F_E\big(u^K_{n-1,P}\big) - F_E\big(u^K_{n-1,P-1}\big) + F_I\big(u^0_{n,1}\big)\big). \tag{15}$$

Note that in (15) we set the two starting values $u^0_{n,0} = u^K_{n-1,P}$ and $u^0_{n,-1} = u^K_{n-1,P-1}$. The forward-backward Euler method was used in the correction steps of all previously applied SIPIDC methods [9,10]. At the kth iteration, the following equation is solved to integrate the correction equation over the first substep $[t_{n,0}, t_{n,1}]$:

$$\begin{aligned} u^{k+1}_{n,1} &= u^K_{n-1,P} + \Delta t_s\big(F_E\big(u^K_{n-1,P}\big) - F_E\big(u^K_{n-1,P}\big) + F_I\big(u^{k+1}_{n,1}\big) - F_I\big(u^k_{n,1}\big)\big) + \mathcal{Q}^{m+1}_n\big(F_E\big(u^k\big) + F_I\big(u^k\big)\big) \\ &= u^K_{n-1,P} + \Delta t_s\big(F_I\big(u^{k+1}_{n,1}\big) - F_I\big(u^k_{n,1}\big)\big) + \mathcal{Q}^{m+1}_n\big(F_E\big(u^k\big) + F_I\big(u^k\big)\big). \end{aligned} \tag{16}$$

The starting value $u^{k+1}_{n,0}$ is set in (16) to $u^K_{n-1,P}$.

The BDF2-based corrector considered in this study is a two-step method. When a multi-step corrector is used, care must be taken in computing u^{k+1}_1 at the first substep during a deferred correction iteration. The derivation of the correction equation (7) is based on the integration of the correction term $\delta^{k+1} \equiv u^{k+1} - u^k$ [11]. The correction term δ^{k+1} is continuous within the interval $[t_n, t_{n+1}]$, but may be discontinuous at t_n, where δ^{k+1} vanishes. For the first substep $m = 0$, $t_{m-1} \notin [t_n, t_{n+1}]$, and the lack of smoothness means that for $m = 0$, a second-order correction step only improves the accuracy of $\tilde{u}$ by one order instead of two. In other words, using final values computed in the previous time step introduces unsmoothness into solutions computed using a BDF2-based corrector limits the increase in order of accuracy during the correction steps to one. To maintain the expected second-order accuracy of the corrector, the m starting values for an m-step method are set to different values at each deferred correction iteration: at the $(k-1)$th iteration, we use values computed at the corresponding kth iteration in the previous time step, $u^k(t_{n-1,P}), u^k(t_{n-1,P-1})$. This choice of starting values will be referred to as "variable starting values."

Variable starting values must be used in conjunction with a multi-step corrector, or a multi-step predictor that is followed by a second-order corrector. For a single-step corrector, or a predictor that is followed by a first-order corrector, one could use either fixed or variable starting values. The choice of fixed or variable starting values may have an impact on the accuracy of the computed solution. This issue will be investigated below using analysis and numerical tests.

We use the notation SIPIDCK[$\mathrm{P}_{\mathrm{name}}$, $\mathrm{C}_{\mathrm{name}}$] to denote a Kth-order SIPIDC method using numerical methods $\mathrm{P}_{\mathrm{name}}$ and $\mathrm{C}_{\mathrm{name}}$ as the predictor and corrector, respectively. If a pth-order predictor and a qth-order corrector is used to construct a Kth-order SIPIDC method, then the correction equation is solved in $(K-p)/q$ iterations.

3. Numerical examples

Two numerical examples are used to study the stability and accuracy of SIPIDC methods. The first example is the van der Pol's equation, which is a popular nonlinear test problem for methods for stiff ODEs. The equation prescribes the motion of a particle $x(t)$ by

$$x''(t) + \mu\big(1 - x(t)^2\big)x'(t) + x(t) = 0. \tag{17}$$

After applying the transformation $y_1(t) = x(t)$, $y_2(t) = \mu x'(t)$, and $t = t/\mu$ one obtains the system

$$y_1(t)' = y_2(t), \tag{18}$$

$$y_2(t)' = \frac{1}{\epsilon}\big(-y_1(t) + \big(1 - y_1(t)^2\big)y_2(t)\big) \tag{19}$$

where $\epsilon = 1/\mu^2$. As ϵ approaches zero, these equations become increasingly stiff. Initial conditions are shown in Table 1. Eqs. (18) and (19) are integrated for $t \in [0, 0.5]$. In the integration of (18) and (19), the first equation is treated explicitly, whereas the second equation is treated implicitly. Because an exact solution is not known for this problem, errors are computed using a reference solution computed using a 7th-order implicit PIDC method and a

Table 1
Initial conditions for van der Pol's equation

ϵ	$y_1(0)$	$y_2(0)$
10^{-3}	2	−0.66654321
10^{-4}	2	−0.666654321
10^{-5}	2	−0.6666654321
10^{-6}	2	−0.66666654321
10^{-7}	2	−0.666666654321

very small time step, chosen so that the solutions computed using that PIDC method and using the ARK4(3)6L[2]SA method [7] converge to the same solution.

The second example is the *cosine test*, which consists of the ODE

$$y(t)' = -2\pi \sin(2\pi t) - \frac{1}{\epsilon}\big(y - \cos(2\pi t)\big), \tag{20}$$

$$y(0) = 0. \tag{21}$$

for $t \in [0, 10]$. The cosine test is a special case of the Prothero–Robinson model equation [13]. The exact solution of (20) and (21) is $y(t) = \cos(2\pi t)$, and as $\epsilon \to 0$ this equation becomes increasingly stiff. In this implementation, SIPIDC methods are used to integrate (20) and (21) for $t \in [0, 10]$ by treating the term $-2\pi \sin(2\pi t)$ explicitly and the term $-(y - \cos(2\pi t))/\epsilon$ implicitly. The cosine test is chosen because its simplicity allows an explicit examination of dominant error terms in SIPIDC methods (see Appendix A).

All calculations reported below were performed using MATLAB programs. The error reported is the discrete L_2-norm of the error in time of the computed solution $y(t_n)$ at each time step.

3.1. Efficiency comparison

We first assess any potential efficiency improvement introduced by the use of a second-order corrector. This investigation is motivated by the observation that a higher-order corrector requires fewer deferred correction iterations to achieve the same overall order of accuracy, and by the numerical results in a previous study [10] which indicate that the use of a higher-order BDF method in the provisional step improves the overall efficiency of SIPIDC methods. Recall that a Kth-order SIPIDC method constructed using a Pth-order predictor requires $K - P$ deferred correction iterations if forward-backward Euler is used in the correction steps, whereas half as many iterations are needed if a second-order corrector is used. Thus, in the next set of tests we investigate the effects on overall computational costs, accuracy, and efficiency when IMEX BDF2 and IMEX RK2 are used in the correction steps. The comparison is made in the context of nonstiff problems.

In Fig. 1A we compare the efficiency among two SIPIDC6 methods, both using BDF2 predictor, while one uses Euler and the other uses BDF2 in the correction steps; two SIPIDC5 methods, both using BDF3 predictor, and again one uses Euler and the other uses BDF2 correctors; and BDF4. Results were obtained using the nonstiff cosine test with $\epsilon = 0.1$. In the estimation of computational costs, the solution of the implicit part of the system is assumed to be much more expensive than the explicit part. This assumption can be justified by the observation that in practice, the implicit part of a system is usually nonlinear, and to compute its solution requires the application of an iterative nonlinear solver. For simplicity, we further assume that the implicit solves in all methods have similar computational costs. With these assumptions, the computational costs of the above SIPIDC methods are compared in terms of the numbers of implicit solves. Results in Fig. 1A show that, for a sufficiently high accuracy requirement, methods with the highest order, i.e., the SIPIDC6 methods, is the most efficient; and both SIPIDC5 and SIPIDC6 methods are more efficient than BDF4.

To assess the effect of BDF2 corrector on overall efficiency compared to a first-order corrector, we first consider the two SIPIDC6 methods, where one uses an Euler corrector and the other uses a BDF corrector. As argued previously, using BDF2 corrector reduces the total number of iterations and thus lowers the overall computational costs. Specifically, SIPIDC6[BDF2,Euler] needs 4 deferred correction iterations, whereas SIPIDC6[BDF2,BDF2] needs only 2. Unfortunately, the computed solution for the cosine test using the BDF2 corrector, although of the same order in both cases, is also less accurate than with the Euler corrector. These two competing factors yield little gain (in fact,

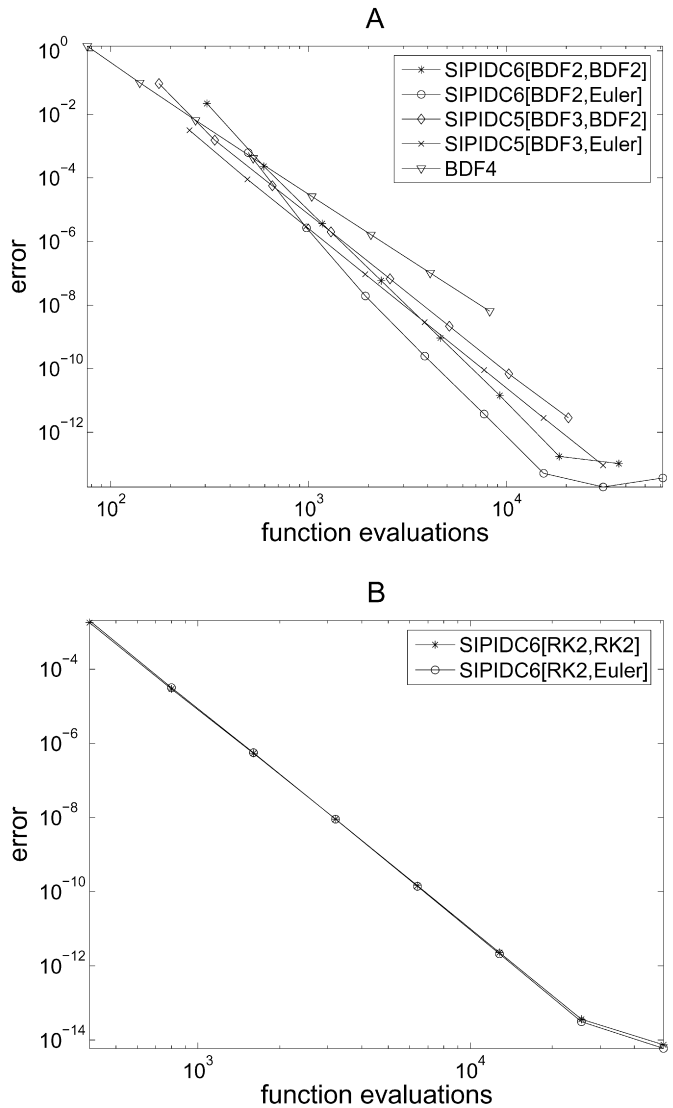

Fig. 1. Efficiency comparison among SIPIDC methods using differing predictors and correctors, obtained using the cosine test. A, Comparison between BDF4 and SIPIDC methods demonstrates the superior efficiency of higher-order methods at a sufficiently high accuracy requirement. Nonetheless, comparison among SIPIDC6 methods and among SIPIDC5 methods suggests that BDF2 correctors reduce computational cost but increase errors, resulting in little gain in overall efficiency. B, SIPIDC methods using RK2 instead of Euler corrector need fewer correction steps, but the two stages involved increased computational costs. The net effect is the almost indistinguishable error and costs for SIPIDC6 methods using RK2 and Euler correctors.

some loss) in overall efficiency for SIPIDC6[BDF2,BDF2]. When comparing the two error curves corresponding to SIPIDC5[BDF3,BDF2] and SIPIDC5[BDF3,Euler], one also observes that the increase in error when BDF2 corrector is used is sufficiently large that its use again results in a decrease in overall efficiency. Test results obtained for the nonstiff van der Pol problem ($\epsilon = 0.1$) also fail to identify any efficiency improvement introduced by a BDF2-based corrector.

In the next set of tests, we compare the efficiency of RK2 and Euler correctors. Fig. 1B shows efficiency results for SIPIDC6[RK2,RK2] and SIPIDC6[RK2,Euler]. Replacing the Euler corrector by RK2 reduces the number of deferred correction iterations by 2. However, each RK2 corrector, which has two stages, also requires two function evaluations whereas Euler requires only one. Consequently, RK2 corrector reduces the number of correction steps but not the number of function evaluations. Results in Fig. 1B show that accuracy and efficiency of these two SIPIDC6 methods

are practically indistinguishable. Similar results (not shown) were also obtained for SIPIDC methods of other orders, and for the van der Pol problem.

3.2. Order reduction

In a previous study [9], we show that the characteristics of order reduction of SIPIDC methods depend critically on the choice of quadrature nodes: when uniform nodes are used and when the left-hand endpoint is not used in the quadrature rule associated with the implicit piece, order reduction begins at smaller Δt's, compared to SIPIDC methods using nonuniform nodes (e.g., Gauss quadrature nodes) or those including the left-hand endpoint in the quadrature rules. Results in another study [10] show that BDF predictors also change the characteristics of order reduction of SIPIDC methods on stiff problems. The convergence rate in the region of order reduction is $k-1$ for a kth-order BDF predictor, with errors of $\mathcal{O}(\epsilon^2)$ magnitude, where ϵ is the stiffness parameter such that as $\epsilon \to 0$ the problem becomes increasingly stiff.

Fixed starting values and Euler correctors were used in all previous implementations of SIPIDC methods. Below we examine convergence behavior of SIPIDC methods that use second-order methods in the correction steps and use variable starting values, when applied to stiff problems.

We first investigate the effect of the choice of the starting value $u^k(t_n)$ on the order reduction behavior of SIPIDC methods using a Euler corrector. The difference in order reduction behavior of SIPIDC methods using the two choices of starting values, fixed and variable, are considered. (Recall that at the kth deferred correction iterations of a SIPIDC method that requires a total of K iterations and P substeps, $u^k(t_{n,1})$ is set to be $u^K(t_{n-1,P})$ and $u^k(t_{n-1,P})$ of the previous time-step in the fixed and variable starting values approaches, respectively.) We computed solutions for the cosine problem for increasingly stiff values of ϵ using SIPIDC6[BDF2,Euler] and SIPIDC6[BDF3,Euler] and the two choices of starting values. Error curves are shown in Fig. 2. In panels A and C, we show error versus the number of implicit function evaluations for approximations obtained by means of SIPIDC6[BDF2,Euler] and SIPIDC6[BDF3,Euler] methods using variable starting values; analogous results obtained using fixed starting values are shown in panels B and D. For a first-order Euler corrector, the two choices of starting values yield the same order of convergence outside of the order reduction regime, i.e., for Δt's that are sufficiently large or small. In the order reduction regime, which can be observed for $\epsilon < 10^{-3}$, the SIPIDC6[BDF2,Euler] and SIPIDC6[BDF3,Euler] methods using fixed starting values generated (panels B and D) approximations with errors that scale like $\epsilon^2\Delta t$ and $\epsilon^2\Delta t^2$, a result that is consistent with results in [10]. When variable starting values are used (panels A and C), the same error scaling ($\epsilon^2\Delta t$ and $\epsilon^2\Delta t^2$ for SIPIDC6[BDF2,Euler] and SIPIDC6[BDF3,Euler], respectively) is found at the early portion of the order reduction regime, but the magnitude of the error is smaller than the corresponding values obtained using fixed starting values. As Δt decreases, the convergence rate of the approximations decreases, but error magnitude remains smaller than the corresponding values in panels B and D.

The reduction in error magnitude in the order reduction regime introduced by the use of variable starting values can be attributed to the smoothness maintained in the correction term δ^k. That increased smoothness is achieved via the use of starting values that have the same order of accuracy as the intermediate values within the time step. With a smoother δ^k, the correction equation can be integrated more accurately, although the overall order of accuracy remains the same for the two choices of starting values.

When forward-backward Euler corrector is used, the overall convergence rate of the SIPIDC method in the order reduction regime is determined solely by the predictor and is independent of the number of deferred correction iterations. In the next set of numerical tests, we investigate whether the overall order of accuracy can be improved by applying second-order correctors. Fig. 3 shows convergence results for SIPIDC6 methods using differing predictors and correctors: SIPIDC6[BDF2,BDF2], SIPIDC5[BDF3,BDF2], and SIPIDC6[RK2,RK2]. Results in panel A indicate that SIPIDC6[BDF2,BDF2] yields $\mathcal{O}(\epsilon^2\Delta t)$ approximations in the order reduction regime. A comparison between the error curves for SIPIDC6[BDF2,Euler] in Fig. 2A and for SIPIDC6[BDF2,BDF2] in Fig. 3A reveals only minute differences. Results in panels B show that SIPIDC6[BDF3,BDF2] yields $\mathcal{O}(\epsilon^2\Delta t^2)$ approximations.

To explain the above convergence behavior, we derived in the Appendices in [10] and in the present work error formulae for the approximations computed in the first correction step. For a SIPIDC method using a kth-order BDF predictor and fixed starting value, the low-order terms in the local truncation error have the form

$$e \approx c_1\epsilon^2\Delta t^k + c_2\epsilon\Delta t^{k+1}. \tag{22}$$

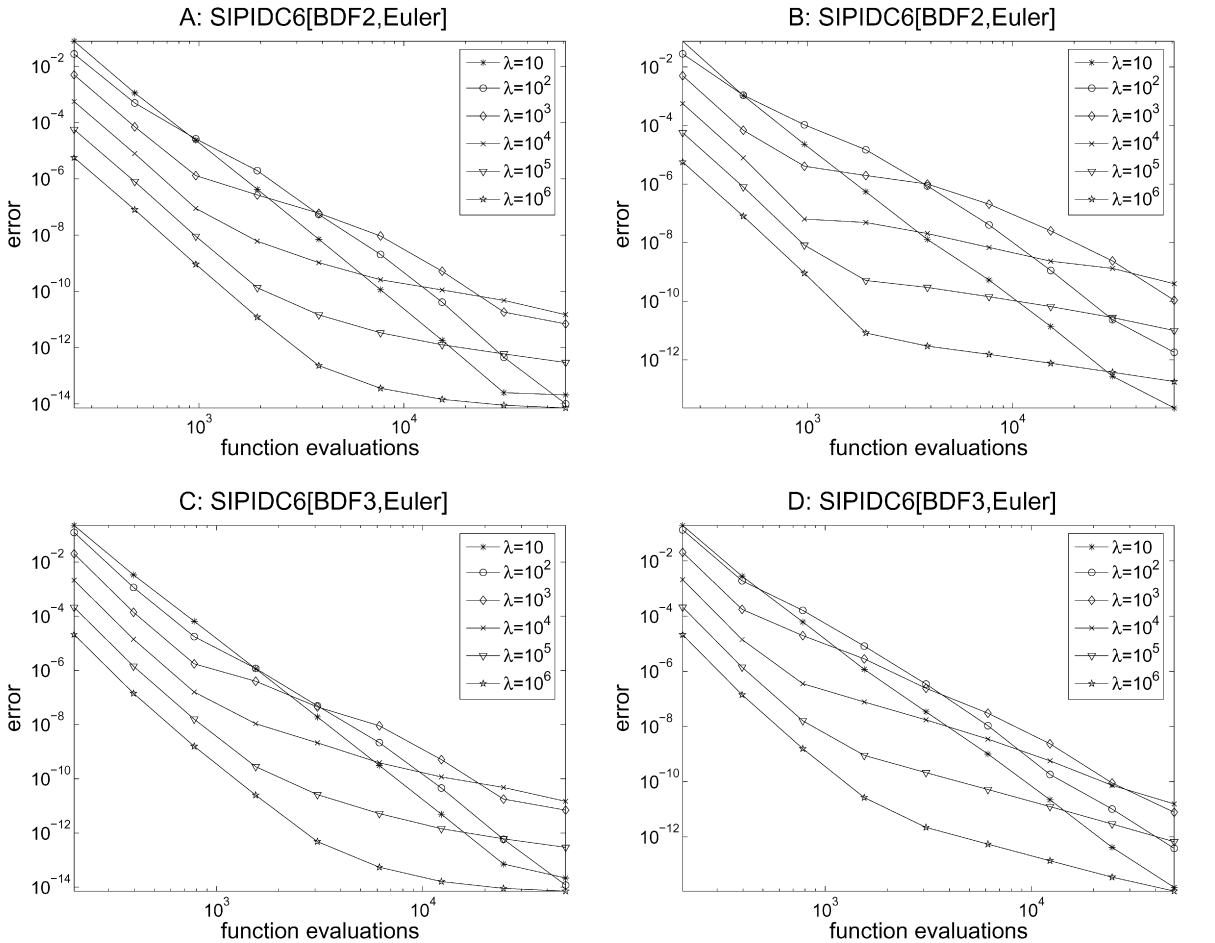

Fig. 2. Error curves obtained for the cosine test, for differing degrees of stiffness, computed using the SIPIDC6[BDF2,Euler] and SIPIDC6[BDF3,Euler] methods. Results in panels A and C were obtained using intermediate starting values, and panels B and D using final starting values. For the time steps used in these simulations, intermediate starting values yielded smaller errors for stiff problems.

The order of the error remains the same whether a Euler corrector is used, or a kth-order BDF corrector is used, and remains the same regardless of the number of correction steps. Also, although the error formulae are derived for a simple equation of the same form as the cosine test, they are consistent with results obtained for the van der Pol problem (not shown). That is, although the error formulae in the Appendices in [10] and in the present work were derived for a linear problem, errors for the nonlinear van der Pol problem also have the form (22).

Numerical results in a previous study [10] show that, when applied to a sufficiently stiff problem, SIPIDC methods using ARK predictors and Euler corrector suffer from order reduction and generate only first-order approximations. Similar results were also obtained for SIPIDC methods using RK-based methods in both the prediction and correction steps: results in Fig. 3C show that SIPIDC6[RK2,RK2] yields $\mathcal{O}(\epsilon\,\Delta t)$ approximations.

4. Discussion

The goal of this work is to develop new formulations of SIPIDC methods for the temporal integration of ODEs involving both stiff and nonstiff components. The target SIPIDC methods use various types of first-, second-, and third-order methods in the predictor step and second-order BDF and RK methods in the correction step. We focus on two issues: the overall efficiency of SIPIDC methods using BDF2- and RK2-based correctors, and the impact of the choice of starting values on accuracy.

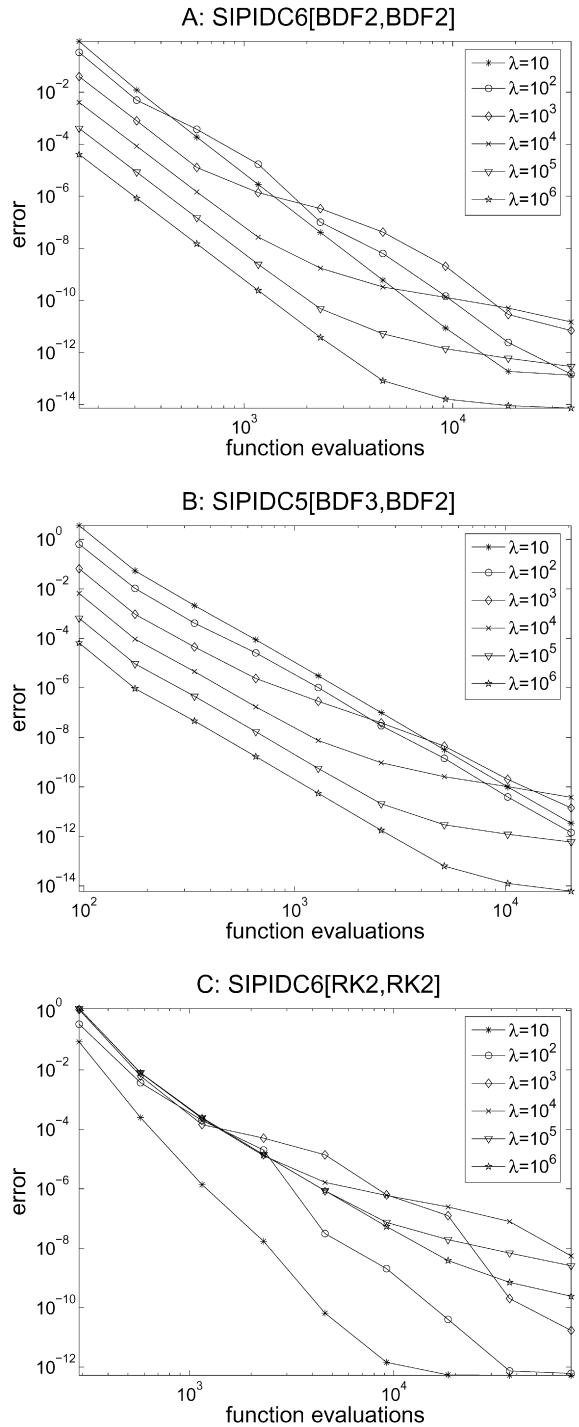

Fig. 3. Error curves obtained for the cosine test, for differing degrees of stiffness, computed using the SIPIDC6[BDF2,BDF2], SIPIDC5[BDF3,BDF2], and SIPIDC6[RK2,RK2] methods. The region of order reduction shows a reduction in the order of convergence in A and B, but shows a first-order reduction (but larger error magnitude) in C.

In terms of efficiency, the benefit that one may gain by using a second-order corrector is not clear. Compared to a first-order corrector, a second-order BDF corrector requires fewer deferred correction iterations to achieve the same overall order of accuracy, and thus lower computational costs. However, our numerical results indicate that owing

to a reduction in the accuracy of the computed solution, the efficiency of SIPIDC methods does not appear to be significantly improved via the use of the BDF2 corrector. In some cases, efficiency actually worsens. Similarly, the RK2-based corrector does not improve efficiency because of the multiple implicit solves required at the stages. It must be noted that the accuracy and efficiency that a given method can achieve is likely problem dependent. Nonetheless, because we have failed to identify, in the numerical examples considered in this study, any efficiency improvement that can be introduced via the use of a second-order corrector, it is not obvious that second-order correctors offer any significant advantage over the simple Euler corrector.

It may seem somewhat paradoxical that BDF2 and BDF3 predictors were shown to improve efficiency, by reducing the number of correction steps without a significant reduction in accuracy [10], whereas the BDF2 corrector fails to achieve an analogous improvement. We believe that the failure of the BDF2 can be attributed to the form of the correction equation (7), which is not an ODE. To obtain an update equation by discretizing (7) using BDF2, one needs to approximate not only the time derivative for the target function u, but also the time derivative for the integral q. The approximation for q', which is not required in the prediction step, may have contributed to the larger overall error when BDF2 corrector is used.

One observation that we made is that the starting values for a multi-step predictor must be carefully chosen if that predictor is to be used in conjunction with a second-order corrector. In our previous work [10], the starting values were set to be the final approximations computed in the previous time step. However, the accuracy of these starting values is of higher order than the approximations computed in the prediction step. The consequence of this mismatch in error is that the smoothness of the provisional solution is reduced, and its accuracy can only be increased by one in the subsequent correction steps, regardless of the order of the corrector used. This issue was irrelevant in [10] where only the Euler corrector was considered. In the present work, in order for the BDF2 and RK2 correctors to yield the expected second-order improvement in solution accuracy, the starting values are set to values computed at the corresponding iteration in the previous time step.

It is noteworthy that when using a multi-step second-order corrector, one could avoid using starting values from the previous time-step by using a single-step second-order method such as the midpoint rule for the first substep only. However, the use of two different methods (mid-point rule in the first substep and BDF2 in the remainder, for example) may render the correction term δ^{k+1} unsmooth at t_m for $m = 1$, owing to the difference in the truncation errors of the two methods. Consequently, although the current correction step improves the accuracy of the solution by two orders, the next correction step can at most be first order, because δ^{k+1} is no longer smooth within $[t_n, t_{n+1}]$. This observation has been verified numerically.

Numerical results suggest that the extent and characteristics of order reduction behavior of SIPIDC methods when applied to stiff problems may depend on the choice of starting values. Indeed, when variable starting values are used, the magnitude of the error in the order reduction regime is smaller compared to the same SIPIDC method using fixed starting values. In contrast, using a second-order corrector appears to have little impact on the accuracy of the approximation. When the BDF2-based corrector is used, SIPIDC methods generate approximations that have similar accuracy of those obtained using the Euler corrector. Also, additional correction steps do not improve or reduce the accuracy.

The principal conclusion that can be drawn from this study is that SIPIDC methods using Euler and BDF2 correctors have comparable efficiency and stability behaviors. That results has implications in partial differential equation (PDE) applications. Indeed, the development of PIDC methods were originally motivated by PDEs arising in mathematical models of physical systems that involve processes of widely-differing characteristic time-scales [3]. The time integration of those multi-scale PDEs can be greatly benefited from operator splitting, which allows the use of different time steps for differing processes. The splitting errors thus introduced can be reduced via the deferred correction procedure [3]. Because in such PDE applications, a higher-order corrector must be used in conjunction with an equally high-order splitting method to achieve the expected accuracy in the correction steps, the simplicity of using the Euler corrector with a simple first-order splitting, as in [3], is particularly appealing—especially given the results in this study.

Appendix A

In this appendix we develop an analytical formulation for the truncation error for SIPIDC methods applied to a simple stiff equation.

Given a smooth function $p(t)$, consider the ODE with exact solution $y(t) = p(t)$ given by

$$\begin{aligned} y' &= p'(t) - \frac{1}{\epsilon}\big(y - p(t)\big), \\ y(0) &= p(0). \end{aligned} \tag{23}$$

Here ϵ is the stiffness parameter where the equation becomes increasingly stiff as $\epsilon \to 0$. We integrate (23) by treating the first term explicitly and the second term implicitly. The following analysis applies to the stiff case where $\epsilon \ll \Delta t$.

We seek to derive a formula for the truncation error of a SIPIDC method using BDF2 as both its predictor and corrector. To that end, we first consider a provisional solution computed using the BDF2 method. Let $p_m \equiv p(t_m)$ and $y_m^k \equiv y^k(t_m)$. Given a previously computed value y_m^0 with error $e_m^0 = p_m - y_m^0$, BDF2 advances the solution of Eq. (23) by one substep given by

$$y_{m+1}^0 = \frac{2y_m^0 - \frac{1}{2}y_{m-1}^0 + \Delta t_s(2p'_m - p'_{m-1} + \frac{1}{\epsilon}p_{m+1})}{3/2 + \Delta t_s/\epsilon}. \tag{24}$$

When $\epsilon < \Delta t_s$, the quantity $1/(3/2 + \Delta t_s/\epsilon)$ can be expanded into the series

$$\frac{1}{3/2 + \Delta t_s/\epsilon} = \frac{\epsilon}{\Delta t_s}\left(1 - \frac{3}{2}\frac{\epsilon}{\Delta t_s} + \left(\frac{3}{2}\frac{\epsilon}{\Delta t_s}\right)^2 - \cdots\right). \tag{25}$$

Substituting (25) into (24) yields

$$y_{m+1}^0 = p_{m+1} + \left(2p'_m - p'_{m-1} + \frac{2y_m^0 - \frac{1}{2}y_{m-1}^0 - \frac{3}{2}p_{m+1}}{\Delta t_s}\right)\left(\epsilon - \frac{3}{2}\frac{\epsilon^2}{\Delta t_s} + \left(\frac{3}{2}\frac{\epsilon}{\Delta t_s}\right)^2\epsilon - \cdots\right). \tag{26}$$

To simplify (26), we make use of the following relations derived using Taylor's expansion:

$$2p'_m - p'_{m-1} = p'_{m+1} - \Delta t_m^2 p_{m+1}^{(3)} + \Delta t_m^3 p_{m+1}^{(4)} + \mathcal{O}\big(\Delta t_m^4\big), \tag{27}$$

$$-\frac{3}{2\Delta t_s}sp_{m+1} = -p'_{m+1} - \frac{1}{\Delta t_s}\left(2p_m - \frac{1}{2}p_{m-1}\right) + \frac{\Delta t_m^2}{3}p_{m+1}^{(3)} + \frac{3}{4}\Delta t_m^3 p_{m+1}^{(4)} + \mathcal{O}\big(\Delta t_s^4\big). \tag{28}$$

From (27) and (28), one obtains that

$$2p'_m - p'_{m-1} + \frac{2y_m - \frac{1}{2}y_{m-1} - \frac{3}{2}p_{m+1}}{\Delta t_s} = \frac{2e_m - \frac{1}{2}e_{m-1}}{\Delta t_s} - \frac{2}{3}\Delta t_m^2 p_{m+1}^{(3)} + \frac{3}{4}\Delta t_m^2 p_{m+1}^{(4)}. \tag{29}$$

Thus,

$$y_{m+1} = p_{m+1} + \left(\frac{2e_m - \frac{1}{2}e_{m-1}}{\Delta t_s} - \frac{2}{3}\Delta t_m^2 p_{m+1}^{(3)} + \frac{3}{4}\Delta t_m^3 p_{m+1}^{(4)}\right)\left(\epsilon - \frac{3}{2}\frac{\epsilon^2}{\Delta t_s} + \left(\frac{3}{2}\frac{\epsilon}{\Delta t_s}\right)^2\epsilon - \cdots\right). \tag{30}$$

Substituting (29) into (26) and making use of the definition of $e_m^0 \equiv p_m - y_m^0$,

$$\begin{aligned} e_{m+1}^0 = {} & \frac{2e_m^0 - \frac{1}{2}e_{m-1}^0}{\Delta t_s}\left(\epsilon - \frac{3}{2}\frac{\epsilon^2}{\Delta t_s} + \left(\frac{3}{2}\frac{\epsilon}{\Delta t_s}\right)^2\epsilon\right) - \frac{2}{3}\Delta t_m^2 p_{m+1}^{(3)}\left(\epsilon - \frac{3}{2}\frac{\epsilon^2}{\Delta t_s} + \left(\frac{3}{2}\frac{\epsilon}{\Delta t_s}\right)^2\epsilon\right) \\ & + \mathcal{O}\big(\epsilon\Delta t^3\big) + \mathcal{O}\big(\epsilon^2\Delta t^2\big) + \mathcal{O}\big(\epsilon^3\big). \end{aligned} \tag{31}$$

Now consider the correction equation given a provisional solution y_m^0. Note that $f(y_m^0, t_m) = p'_m - \frac{1}{\epsilon}e_m^0$. When a BDF2-based corrector is used, the direct form of the correction equation for (23) is given by

$$\begin{aligned} \frac{3}{2}y_{m+1}^1 = {} & 2y_m^1 - \frac{1}{2}y_{m-1}^1 + \Delta t_s\Bigg(2p'_m - p'_{m-1} - 2p'_m + p'_{m-1} \\ & - \frac{1}{\epsilon}\big(y_{m+1}^1 - p_{m+1}\big) - y_{m+1}^0 + p_{m+1}\Bigg) + \frac{3}{2}\mathcal{Q}_m^{m+1}\big(y^0\big) - \frac{1}{2}\mathcal{Q}_{m-1}^m\big(y^0\big) \end{aligned} \tag{32}$$

$$= 2y_m^1 - \frac{1}{2}y_{m-1}^1 + \Delta t_s\left(-\frac{1}{\epsilon}\big(y_{m+1}^1 - y_{m+1}^0\big)\right) + \frac{3}{2}\mathcal{Q}_m^{m+1}\big(y^0\big) - \frac{1}{2}\mathcal{Q}_{m-1}^m\big(y^0\big). \tag{33}$$

Solving for y^1_{m+1} yields

$$y^1_{m+1} = \frac{2y^1_m - \frac{1}{2}y^1_{m-1} + (\Delta t_s/\epsilon)y^0_{m+1} + \frac{3}{2}\mathcal{Q}^{m+1}_m(p'(t) + \frac{1}{\epsilon}e^0(t)) - \frac{1}{2}\mathcal{Q}^{m+1}_m(p'(t) + \frac{1}{\epsilon}e^0(t))}{3/2 + \Delta t_s/\epsilon}. \tag{34}$$

To derive an error formula for y^1_{m+1}, we first consider the last quadrature term in the numerator. The integration rule given by Eq. (6) defines

$$\mathcal{Q}^{m+1}_m\left(p'(t) - \frac{1}{\epsilon}e^0(t)\right) = \Delta t_s \sum_{l=0}^{p} q^l_m\left(p'_l - \frac{1}{\epsilon}e^0_l\right). \tag{35}$$

Since the integration rule is assumed to be $\mathrm{O}(\Delta t^q)$, the first term can be integrated to give

$$\mathcal{Q}^{m+1}_m\left(p'(t) - \frac{1}{\epsilon}\tilde{e}(t)\right) = p_{m+1} - p_m + \mathcal{O}\left(\Delta t^q\right) + \frac{\Delta t_s}{\epsilon}\sum_{l=0}^{p} q^l_m e^0_l. \tag{36}$$

Similarly,

$$\mathcal{Q}^{m}_{m-1}\left(p'(t) - \frac{1}{\epsilon}\tilde{e}(t)\right) = p_m - p_{m-1} + \mathcal{O}\left(\Delta t^q\right) + \frac{\Delta t_{m-1}}{\epsilon}\sum_{l=0}^{p} q^l_{m-1} e^0_l. \tag{37}$$

Substituting expressions (36) and (37) into Eq. (34) gives

$$y^1_{m+1} = \frac{2y^1_m - \frac{1}{2}y^1_{m-1} + \frac{3}{2}p_{m+1} - 2p_m + \frac{1}{2}p_{m-1} + (\Delta t_s/\epsilon)(y^0_{m+1} + \sum_{l=0}^{p}\bar{q}^l_m e^0_l) + \mathrm{O}(\Delta t^q)}{3/2 + \Delta t_s/\epsilon}, \tag{38}$$

where $\bar{q}^l_m \equiv \frac{3}{2}q^l_m - \frac{1}{2}q^l_{m-1}$.

Applying the expansion (25), one obtains that

$$\begin{aligned} y^1_{m+1} = {} & y^0_{m+1}\left(1 - \frac{3}{2}\frac{\epsilon}{\Delta t_s} + \left(\frac{3}{2}\frac{\epsilon}{\Delta t_s}\right)^2 \cdots\right) \\ & + \left(2y^1_m - \frac{1}{2}y^1_{m-1} + \frac{3}{2}p_{m+1} - 2p_m + \frac{1}{2}p_{m-1}\right)\left(\frac{\epsilon}{\Delta t_s} - \frac{3}{2}\left(\frac{\epsilon}{\Delta t_s}\right)^2 + \left(\frac{3}{2}\right)^2\left(\frac{\epsilon}{\Delta t_s}\right)^3 \cdots\right) \\ & + \left(\sum_{l=0}^{p}\bar{q}^l_m e^0_l\right)\left(1 - \frac{3}{2}\frac{\epsilon}{\Delta t_s} + \left(\frac{3}{2}\frac{\epsilon}{\Delta t_s}\right)^2 \cdots\right) + \mathrm{O}\left(\epsilon\Delta t^{q-1}\right) + \mathrm{O}\left(\epsilon^2\Delta t^{q-2}\right) + \mathrm{O}\left(\epsilon^3\Delta t^{q-3}\right)\cdots. \end{aligned} \tag{39}$$

Finally, define the error in the updated solution $e^1_m = y^1_m - p_m$. Then subtracting p_{m+1} from both sides of the equation and manipulating yields

$$\begin{aligned} e^1_{m+1} = {} & \left(2e^1_m - \frac{1}{2}e^1_{m-1} + \mathcal{O}\left(\Delta t^2\right)\right)\left(\frac{\epsilon}{\Delta t_s} - \frac{3}{2}\left(\frac{\epsilon}{\Delta t_s}\right)^2 + \left(\frac{3}{2}\right)^2\left(\frac{\epsilon}{\Delta t_s}\right)^3 \cdots\right) \\ & \times \left(e^0_{m+1} - \sum_{l=0}^{p}\bar{q}^l_m e^0_l\right)\left(1 - \frac{3}{2}\frac{\epsilon}{\Delta t_s} + \left(\frac{3}{2}\frac{\epsilon}{\Delta t_s}\right)^2 \cdots\right) \\ & + \mathrm{O}\left(\epsilon\Delta t^{q-1}\right) + \mathrm{O}\left(\epsilon^2\Delta t^{q-2}\right) + \mathrm{O}\left(\epsilon^3\Delta t^{q-3}\right) + \cdots. \end{aligned} \tag{40}$$

Consider now the first time step of a SIPIDC method for Eq. (23). Ignoring in (31) terms with second or higher powers of ϵ, the dominant error terms in the provisional solution (i.e., terms with lowest power of ϵ) are found to be

$$\begin{aligned} e^0_{m+1} &= -\frac{2}{3}\epsilon\Delta t^2_m p^{(3)}_{m+1} + \frac{\epsilon}{\Delta t_s}\left(2e^0_m - \frac{1}{2}e^0_{m-1}\right) \\ &= -\frac{2}{3}\epsilon\Delta t^2_m p^{(3)}_{m+1} - \frac{2}{3}\epsilon^2\Delta t_s\left(2p^{(3)}_m - \frac{1}{2}p^{(3)}_{m-1}\right). \end{aligned} \tag{41}$$

By assumption $\epsilon/\Delta t_s \ll 1$, thus the second term on the right side of (41) is much smaller than the first term. Likewise, the dominant pieces of the correction equation error (40) comes from the term

$$e^1_{m+1} = e^0_{m+1} - \sum_{l=0}^{p} \bar{q}^l_m e^0_l + \frac{\epsilon}{\Delta t_s}\left(2e^1_m - \frac{1}{2}e^1_{m-1}\right). \tag{42}$$

Substituting the dominant provisional error (41) into the dominant correction error (42) gives

$$\begin{aligned} e^1_{m+1} = {} & -\frac{2}{3}\epsilon\Delta t^2_m p^{(3)}_{m+1} - \frac{2}{3}\epsilon^2\Delta t_s\left(2p^{(3)}_m - \frac{1}{2}p^{(3)}_{m-1}\right) \\ & - \sum_{l=1}^{p}\bar{q}^l_m\left(-\frac{2}{3}\epsilon\Delta t^2_m p^{(3)}_l - \frac{2}{3}\epsilon^2\Delta t_s\left(2p^{(3)}_{l-1} - \frac{1}{2}p^{(3)}_{l-2}\right)\right) + \frac{\epsilon}{\Delta t_s}\left(2e^1_m - \frac{1}{2}e^1_{m-1}\right). \end{aligned} \tag{43}$$

The summation term can be further expanded in Taylor series

$$\begin{aligned} & -\sum_{l=1}^{p}\bar{q}^l_m\left(-\frac{2}{3}\epsilon\Delta t^2_m p^{(3)}_l - \frac{2}{3}\epsilon^2\Delta t_s\left(2p^{(3)}_{l-1} - \frac{1}{2}p^{(3)}_{l-2}\right)\right) \\ & \quad = \sum_{l=1}^{p}\bar{q}^l_m\left(\frac{2}{3}\epsilon\Delta t^2_m p^{(3)}_{m+1} + \mathcal{O}\left(\epsilon\Delta t^3\right) + \frac{2}{3}\epsilon^2\Delta t_s\left(2p^{(3)}_m - \frac{1}{2}p^{(3)}_{m-1}\right) + \mathcal{O}\left(\epsilon^2\Delta t^2\right)\right). \end{aligned} \tag{44}$$

Substituting (44) into (43) and simplifying gives

$$e^1_{m+1} = \frac{\epsilon}{\Delta t_s}\left(2e^1_m - \frac{1}{2}e^1_{m-1}\right) + \mathcal{O}\left(\epsilon^2\Delta t^2 + \epsilon\Delta t^3 + \Delta t^q\right). \tag{45}$$

This error formula suggests that in the order reduction region, the error scales like $\mathcal{O}(\epsilon^2\Delta t^2 + \epsilon\Delta t^3)$, for sufficiently large Δt. As Δt decreases, while still satisfying $\epsilon \ll \Delta t$, the factor $\epsilon/\Delta t_s$ in the first term becomes increasingly important, and the convergence rate decreases.

References

[1] U.M. Ascher, S.J. Ruuth, R.J. Spiteri, Implicit-explicit Runge–Kutta methods for time-dependent partial differential equations, Appl. Numer. Math. 25 (1997) 151–167.

[2] U.M. Ascher, S.J. Ruuth, B.T.R. Wetton, Implicit-explicit methods for time-dependent partial differential equations, SIAM J. Numer. Anal. 32 (3) (1995) 797–823.

[3] A. Bourlioux, A.T. Layton, M.L. Minion, Higher-order multi-implicit spectral deferred correction methods for problems of reacting flow, J. Comput. Phys. 189 (2003) 651–675.

[4] A. Dutt, L. Greengard, V. Rokhlin, Spectral deferred correction methods for ordinary differential equations, BIT 40 (2000) 241–266.

[5] T. Hagstrom, R. Zhou, On the spectral deferred correction of splitting-method for initial value problems, Comm. Appl. Math. Comput. Sci. 1 (1) (2006) 169–206.

[6] A.C. Hansen, J. Strain, Convergence theory for spectral deferred correction, Numer. Math. (2007), submitted for publication.

[7] C.A. Kennedy, M.H. Carpenter, Additive Runge–Kutta schemes for convection–diffusion–reaction equations, Appl. Numer. Math. 44 (1–2) (2003) 139–181.

[8] A.T. Layton, M.L. Minion, Conservative multi-implicit spectral deferred methods for reacting gas dynamics, J. Comput. Phys. 194 (2) (2004) 697–715.

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

[10] A.T. Layton, M.L. Minion, Implications of the choice of predictors for semi-implicit Picard integral deferred correction methods for ordinary differential equations, Comm. Appl. Math. Comput. Sci. 2 (1) (2007) 1–34.

[11] M.L. Minion, Semi-implicit spectral deferred correction methods for ordinary differential equations, Comm. Math. Sci. 1 (3) (2003) 471–500.

[12] M.L. Minion, Semi-implicit projection methods for incompressible flow based on spectral deferred corrections, Appl. Numer. Math. 48 (3–4) (2004) 369–387.

[13] A. Prothero, A. Robinson, On the stability and accuracy of one step methods for solving stiff systems of ordinary differential equations, Math. Comp. 28 (1974) 145–162.