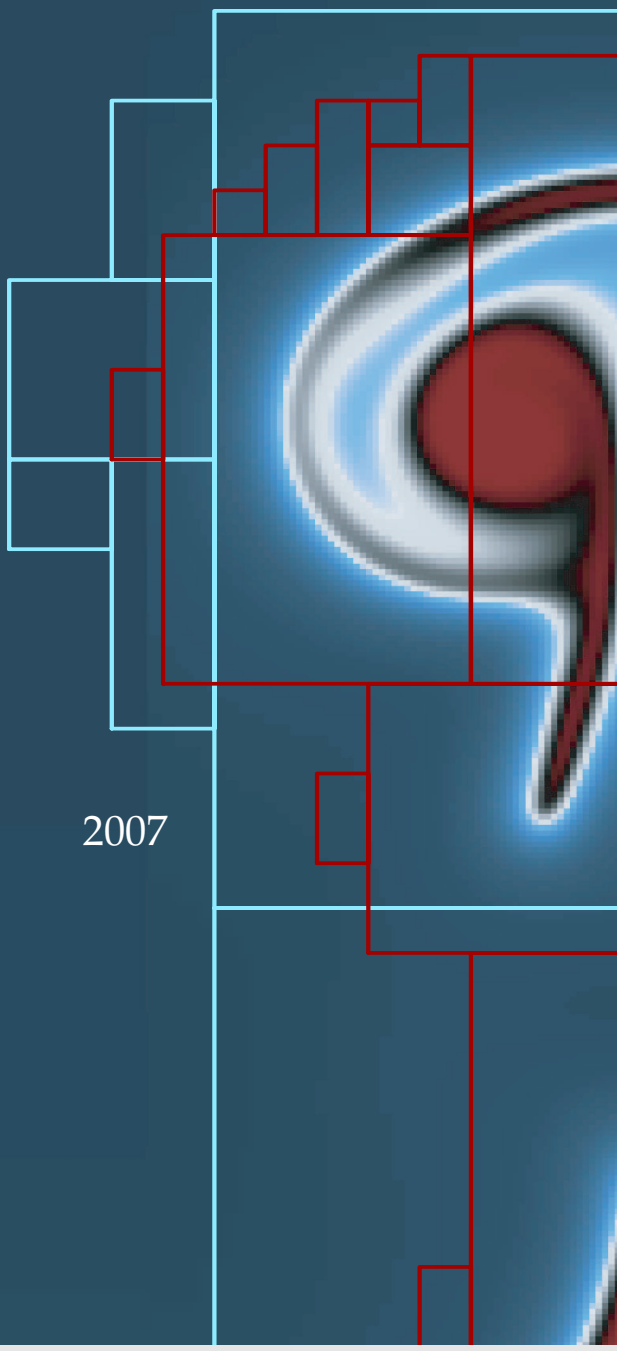
Communications in Applied Mathematics and Computational Science

Volume 2 No. 1 2007



IMPLICATIONS OF THE CHOICE OF PREDICTORS FOR SEMI-IMPLICIT PICARD INTEGRAL DEFERRED CORRECTION METHODS

ANITA T. LAYTON AND MICHAEL L. MINION

mathematical sciences publishers

COMM. APP. MATH. AND COMP. SCI.
Vol. 2, No. 1, 2007

IMPLICATIONS OF THE CHOICE OF PREDICTORS FOR SEMI-IMPLICIT PICARD INTEGRAL DEFERRED CORRECTION METHODS

ANITA T. LAYTON AND MICHAEL L. MINION

High-order semi-implicit Picard integral deferred correction (SIPIDC) methods have previously been proposed for the time-integration of partial differential equations with two or more disparate time scales. The SIPIDC methods studied to date compute a high-order approximation by first computing a provisional solution with a first-order semi-implicit method and then using a similar semi-implicit method to solve a series of correction equations, each of which raises the order of accuracy of the solution by one. This study assesses the efficiency of SIPIDC methods that instead use standard semi-implicit methods with orders two through four to compute the provisional solution. Numerical results indicate that using a method with more than first-order accuracy in the computation of the provisional solution increases the efficiency of SIPIDC methods in some cases. First-order PIDC corrections can improve the efficiency of semi-implicit integration methods based on backward difference formulae (BDF) or Runge–Kutta methods while maintaining desirable stability properties. Finally, the phenomenon of order reduction, which may be encountered in the integration of stiff problems, can be partially alleviated by the use of BDF methods in the computation of the provisional solution.

1. Introduction

The dynamics of many physical and biological systems of interest today involve processes with two or more characteristic time scales. When the time scales of the physical processes vary widely, efficient time-marching of the partial differential equations (PDEs) that describe the dynamics may require specialized numerical methods, particularly when one wishes to accurately resolve processes at each time

MSC2000: primary 65B05; secondary 65L20.
Keywords: semi-implicit methods, deferred correction methods, order reduction.
A. T. Layton was supported in part by the National Science Foundation, grant DMS-0340654. M. L. Minion was supported in part under contract DE-AC03-76SF00098 by the Director, Department of Energy (DOE) Office of Science; Office of Advanced Scientific Computing Research; Office of Mathematics, Information, and Computational Sciences; Applied Mathematics Sciences Program; and by the Alexander von Humboldt Foundation.

scale. For example, following the method-of-lines approach, when the PDEs are discretized in space, the resulting system of coupled ordinary differential equations (ODEs) typically contains both stiff and nonstiff terms. When the stiffness of one of these terms corresponds to eigenvalues with a large negative real part (for example, from the discretization of a diffusive term), an implicit treatment of this term can allow a much larger stable time step (without significantly sacrificing accuracy) than an explicit treatment. Hence, the use of semi-implicit methods for such systems, that is, methods that treat only the stiff terms implicitly, can result in a considerable improvement in efficiency compared to fully implicit methods, particularly when other nonstiff terms in the equations are computationally expensive to treat implicitly. Provided that a sufficiently high level of accuracy is desired, and/or the temporal interval is sufficiently long, high-order methods for ODEs are more efficient than low-order methods in that less computational cost is required by high-order methods to achieve a given, sufficiently stringent error tolerance. Hence the construction of stable and efficient higher-order semi-implicit methods for ODEs is desirable.

Indeed, semi-implicit (also known as implicit-explicit or IMEX) versions of popular time-integration methods such as Runge–Kutta (RK), linear multistep, or backward difference formulae (BDF) methods have been developed to efficiently integrate ODEs with both nonstiff and stiff components. Semi-implicit RK methods have been proposed and tested by a number of authors [Ascher et al. 1997; Kennedy and Carpenter 2003; Pareschi and Russo 2001; Shen and Zhong 1996; Calvo et al. 2001]; however, owing in part to the complexity of deriving such schemes, only semi-implicit RK methods with order up to five have so far been developed. Similarly, several papers have analyzed the stability and accuracy of semi-implicit methods derived from linear multistep methods [Akrivis et al. 1999; Ascher et al. 1995; Frank et al. 1997; in't Hout 2002]. In this case, stable schemes up to order six are easily constructed, although higher-order versions have the disadvantages that they require multiple starting values, require care when used with variable time stepping schemes, and, as further discussed in Section 3, have less satisfactory stability characteristics.

In a series of studies [Minion 2003; Minion 2004; Layton and Minion 2005], we developed and analyzed a new class of semi-implicit methods for integrating ODEs that arise from a method-of-lines discretization of PDEs involving time-scale disparity. The methods are based on a semi-implicit Picard integral deferred correction (SIPIDC) approach, which is a generalization of the explicit and implicit spectral deferred correction (SDC) methods introduced in [Dutt et al. 2000]. SDC methods use a low-order numerical method to compute an approximate solution with an 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.

The SDC methods introduced in [Dutt et al. 2000] and the SIPIDC methods described in [Layton and Minion 2005], as well as most of the SISDC methods described in [Minion 2003], use a first-order method both to compute the provisional solution and to approximate the correction equations. It has previously been demonstrated that higher-order versions of these methods are more efficient than lower-order methods, and that the stability properties of the methods with very high order remain similar to those with lower order [Dutt et al. 2000; Minion 2003]. A reasonable question to ask is whether the efficiency of SIPIDC methods can be improved by using a semi-implicit method with higher than first-order accuracy to compute the provisional solution. (We will refer to the standard method used to compute the provisional solution in a particular PIDC method as the *predictor*.) Hence we wish to investigate whether using a semi-implicit BDF or RK method as the predictor in a PIDC method improves the overall efficiency of the PIDC method. PIDC methods using a predictor with higher than first-order accuracy require fewer iterations of the correction equation to achieve the same overall order of accuracy relative to methods using a first-order predictor. However, it is not immediately clear if the lower computational cost comes at the expense of a loss in accuracy, or if using such a predictor negatively affects the stability of PIDC methods. Another relevant question addressed here is whether performing a series of SIPIDC corrections on a solution generated from a semi-implicit BDF or RK method results in a SIPIDC method with greater numerical efficiency than that afforded by simply using the base methods alone. The primary goal of this paper is to address these questions using the linear stability analysis in Section 3 and numerical tests in Section 4.

A further issue addressed here concerns order reduction for stiff problems, something observed in connection with both SIPIDC methods and semi-implicit RK methods [Minion 2003; Layton and Minion 2005; Kennedy and Carpenter 2003]. In [Layton and Minion 2005] the effect of the choice of quadrature nodes on the extent and character of order reduction of SIPIDC methods on stiff problems is investigated. Both analytical and numerical results in [Layton and Minion 2005] show that, for a sufficiently stiff problem, SIPIDC methods using a first-order forward-backward Euler predictor exhibit a reduction of order for a range of time steps, and the extent and character of the reduction depends on the choice of quadrature rule used in the method. The results presented in Section 4.3 show that the extent and character of order reduction also depend critically on the predictor. Specifically, when a kth-order IMEX BDF predictor is used with uniform quadrature nodes, the convergence rate in the region of order reduction is $k-1$, compared to a reduction to first-order when an IMEX RK predictor (regardless of order) is used.

2. SIPIDC methods

Below is 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 [Minion 2003] and [Bourlioux et al. 2003]. The target ODE takes the form

$$\begin{aligned} u'(t) &= F_E\big(t, u(t)\big) + F_I\big(t, u(t)\big), \qquad t \in [a, b] \\ u(a) &= u_a, \end{aligned} \tag{1}$$

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 $\Delta t = (b-a)/N_T$, for some positive integer N_T, is assumed in the numerical discretization. Let $t_n = n\Delta t$, for $n = 0, 1, 2, \ldots, N_T$, be the n-th time-level. 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 subintervals by choosing points $t_{n,m}$ for $m = 0, 1, \ldots, P$ such that $t_n = t_{n,0} < t_{n,1} < \cdots < t_{n,m} < \cdots < t_{n,P} \leq t_{n+1}$. For notational simplicity, the subscript n in $t_{n,m}$ is omitted and $t_{n,m}$ is written as t_m. Let $\Delta t_m \equiv t_{m+1} - t_m$; the interval $[t_m, t_{m+1}]$ is referred to as a *substep*.

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}, a SIPIDC method first computes a provisional solution $\tilde{u}(t_m) \equiv u_m^0$, for $m = 0, 1, \ldots, P$, by means of a semi-implicit method that we refer to as the predictor. Presumably any method could be chosen to compute the provisional solution, and the main point of this paper is to investigate the relative performance of SIPIDC methods using different predictors.

Given a provisional solution $\tilde{u}(t)$, the accuracy of that solution can be improved using an estimate of its error (or correction): $u(t) - \tilde{u}(t)$, denoted by $\delta(t)$. Using the Picard integral form of the solution to Equation (1), one can express $\delta(t)$ as the solution to the integral equation

$$\delta(t) = \int_a^t \big(F_E(\tau, \tilde{u}+\delta) - F_E(\tau, \tilde{u}) + F_I(\tau, \tilde{u}+\delta) - F_I(\tau, \tilde{u})\big)\, d\tau + E(t, \tilde{u}), \tag{2}$$

where E is the residual function given by

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

We have suppressed the time dependence of $\tilde{u}(t)$ and $\delta(t)$ in the integrands to avoid clutter. A detailed derivation of Equation (2) is given in [Minion 2003].

In SIPIDC methods, a semi-implicit discretization of Equation (2) is used to iteratively increase the order of accuracy of the provisional solution, that is,

$$u_m^{k+1} = u_m^k + \delta_m^k.$$

Specifically, a forward-backward Euler method for computing δ_m^k is given by

$$\delta_{m+1}^k = \delta_m^k + \Delta t_m\big(F_E(u_m^{k+1}) - F_E(u_m^k) + F_I(u_{m+1}^{k+1}) - F_I(u_{m+1}^k)\big) + E_{m+1}(u^k) - E_m(u^k), \quad (3)$$

where the terms $E_m(u^k)$ are approximated with numerical quadrature. Let

$$\mathbb{Q}_m^{m+1}(F)$$

be a Pth-order numerical quadrature approximation to $\int_{t_m}^{t_{m+1}} F(\tau)d\tau$, that is,

$$\mathbb{Q}_m^{m+1}(F) = \Delta t_m \sum_{l=0}^{p} q_l^m F_l = \int_{t_m}^{t_{m+1}} F(\tau)d\tau + \mathcal{O}(\Delta t^{P+1}). \quad (4)$$

By adding u^k to both sides of (3), one obtains a direct update equation that can be used to improve the accuracy of u^k:

$$u_{m+1}^{k+1} = u_m^{k+1} + \Delta t_m\big(F_E(u_m^{k+1}) - F_E(u_m^k) + F_I(u_{m+1}^{k+1}) - F_I(u_{m+1}^k)\big) + \mathbb{Q}_m^{m+1}\big(F_E(u^k) + F_I(u^k)\big). \quad (5)$$

Equation (5) is solved at the k-th iteration, referred to as a deferred correction iteration; see [Minion 2003] for details. The quadrature $\mathbb{Q}$ should have at least the same order of accuracy as the updated approximation u^{k+1}. As in [Bourlioux et al. 2003; Minion 2003], the quadrature $\mathbb{Q}_m^{m+1}$ is computed as the integral of an interpolating polynomial over the subinterval $[t_m, t_{m+1}]$ (see further discussion below).

In the SDC methods presented in [Dutt et al. 2000], the points t_m are chosen to be the Gaussian quadrature nodes of the interval $[t_n, t_{n+1}]$, and the solution is only integrated at these nodes on the interior of the interval. In [Bourlioux et al. 2003; Minion 2003] the points t_m are chosen to be Gauss–Lobatto quadrature nodes, which are more convenient in that the solution is directly computed at both endpoints of the time step interval. However, because Gauss–Lobatto nodes are not evenly spaced for orders of accuracy >2, predictors with higher than first order are less convenient to implement (nonetheless, it can be done; see [Minion 2003]). This is particularly true if the SIPIDC method is used for the temporal integration of PDEs in which block structured adaptive mesh refinement is used (see [Berger and Oliger 1984]). In this instance (currently a topic of research by the authors), the colocation of coarse and fine grid data requires the use of uniform substeps. Other examples in which it is either convenient or necessary to choose substeps that are uniformly spaced have been discussed in [Layton and Minion 2005]. For these reasons, SIPIDC methods presented here use uniform nodes such that $\Delta t_1 = \cdots = \Delta t_m \cdots \equiv \Delta t_s$.

The form of the quadrature rule also has a significant effect on the stability and accuracy of the SIPIDC method for stiff equations. The methods in [Layton and Minion 2005] actually use two separate quadrature rules for the two terms in $\mathbb{Q}_m^{m+1}\big(F_E(u^k) + F_I(u^k)\big)$ in Equation (4). It is shown in [Layton and Minion 2005] that, when function values at the left endpoint t_n are omitted in the quadrature rule associated with the stiff component (that is, $q_0^m = 0$ for all m), the resulting SIPIDC method is $L(\alpha)$-stable. Also, by including the left endpoint in the nonstiff quadrature rule, the accuracy of the quadrature associated with the explicit piece is improved. This choice of quadrature rules, denoted *LR* in [Layton and Minion 2005], is adopted in this study. To construct a method with Kth-order accuracy, the quadrature $\mathbb{Q}$ should also have at least 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 $\mathbb{Q}_m^{m+1}$ over the subinterval $[t_m, t_{m+1}]$. Thus, to construct a Kth-order SIPIDC method with uniform nodes that uses the LR quadrature rule (which excludes the left endpoints in the stiff quadrature rule), $K+1$ nodes or K substeps are required.

2.1. ***Moderate-order predictors.*** The SDC methods presented in [Bourlioux et al. 2003; Dutt et al. 2000] are based on forward-backward Euler methods; that is, the prediction and correction steps are first order. Because higher-order methods are generally more efficient than lower-order methods, we investigate SIPIDC methods that are based on second- through fourth-order semi-implicit methods in the prediction step. We refer to these predictors as *moderate-order* predictors. The methods that we use for computing the provisional solution are based on Euler methods, IMEX BDF [Ascher et al. 1995], IMEX RK methods [Ascher et al. 1997; Kennedy and Carpenter 2003], and classical Adams-type multistep methods. These methods, chosen either for their popularity or known stability, are described below.

IMEX BDF methods. BDF methods are a class of linear multistep methods specifically developed for the solution of stiff ODEs. Hence it is natural when constructing semi-implicit generalizations of linear multistep methods to base the treatment of the stiff piece of the equation on BDF methods. IMEX BDF methods have been previously studied [Ascher et al. 1995; Akrivis et al. 1998; in't Hout 2002]. Here we use second- through fourth-order semi-implicit BDF methods from [Ascher et al. 1995] (denoted BDF2, BDF3, and BDF4) in the provisional step. Forward-backward Euler methods, which can be considered as either a first-order IMEX BDF or IMEX RK method, are included here. For brevity, IMEX BDF will be referred to simply as BDF. The specific formulae are

$$\text{Euler: } u_{m+1}^0 = u_m^0 + \Delta t_s\big(F_E(u_m^0) + F_I(u_{m+1}^0)\big), \tag{6}$$

$$\text{BDF2: } \tfrac{3}{2}u^0_{m+1} = 2u^0_m - \tfrac{1}{2}u^0_{m-1} + \Delta t_s\big(2F_E(u^0_m) - F_E(u^0_{m-1}) + F_I(u^0_{m+1})\big), \quad (7)$$

$$\text{BDF3: } \tfrac{11}{6}u^0_{m+1} = 3u^0_m - \tfrac{3}{2}u^0_{m-1} + \tfrac{1}{3}u^0_{m-2} + \Delta t_s\big(3F_E(u^0_m) - 3F_E(u^0_{m-1}) + F_E(u^0_{m-2}) + F_I(u^0_{m+1})\big), \quad (8)$$

$$\text{BDF4: } \tfrac{25}{12}u^0_{m+1} = 4u^0_m - 3u^0_{m-1} + \tfrac{4}{3}u^0_{m-2} - \tfrac{1}{4}u^0_{m-3} + \Delta t_s\big(4F_E(u^0_m) - 6F_E(u^0_{m-1}) + 4F_E(u^0_{m-2}) - F_E(u^0_{m-3}) + F_I(u^0_{m+1})\big). \quad (9)$$

IMEX RK methods. There are several different implementations of second-order IMEX RK methods. The one used in this study is a two-stage L-stable RK2 method described by Ascher et al [Ascher et al. 1997]. This particular implementation of IMEX RK2 is chosen owing to its L-stability, even though it requires two stages and is thus more costly than some alternative implementations (for example, the IMEX midpoint [Ascher et al. 1997]). The L-stable IMEX RK2 method, which we refer to as RK2 for brevity, generates a provisional solution as follows:

$$\begin{aligned}
\text{RK2: } \phi^{(1)}_{m+c_1} &= u^0_m + c_1\Delta t_s\big(F_E(u^0_m) + F_I(\phi^{(1)}_{m+c_1})\big),\\
\phi^{(2)}_{m+1} &= u^0_m + \Delta t_s\Big(c_2F_E(u^0_m) + (1-c_2)F_E(\phi^{(1)}_{m+c_1})\\
&\qquad + (1-c_1)F_I(\phi^{(1)}_{m+c_1}) + c_1F_I(\phi^{(2)}_{m+1})\Big),\\
u^0_{m+1} &= u^0_m + \Delta t_s\Big((1-c_1)\big(F_E(\phi^{(1)}_{m+c_1}) + F_I(\phi^{(1)}_{m+c_1})\big)\\
&\qquad + c_1\big(F_E(\phi^{(2)}_{m+1}) + F_I(\phi^{(2)}_{m+1})\big)\Big),
\end{aligned}$$

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

The third- and fourth-order IMEX RK methods used here are based on the Additive RK methods developed by Kennedy and Carpenter [Kennedy and Carpenter 2003], specifically, the ARK3(2)4L[2]SA-ERK and ARK4(3)6L[2]SA methods. These methods, which we refer to simply as ARK3 and ARK4, involve three and five stages, respectively, and we refer interested reader to [Kennedy and Carpenter 2003] for the relevant coefficients. There is a fifth-order ARK method introduced in [Kennedy and Carpenter 2003], but Kennedy and Carpenter determine that it is not competitive with the fourth-order methods, and hence we do not study it here. We know of no IMEX RK methods of order greater than five in the literature (although it is possible to construct such methods).

Multistep methods. We also investigate the well known second- and third-order multistep methods: Crank–Nicolson/Adam–Bashforth (CNAB) and Adam–Bashforth/Adam–Moulton (ABAM) methods. When these methods are used, the provisional solution is given by:

$$\begin{aligned}
\text{CNAB: } u^0_{m+1} = u^0_m + \Delta t_s \Big(&\tfrac{3}{2} F_E(u^0_m) \\
&- \tfrac{1}{2} F_E(u^0_{m-1}) + \tfrac{1}{2} F_I(u^0_{m+1}) + \tfrac{1}{2} F_I(u^0_m) \Big), \\
\text{ABAM: } u^0_{m+1} = u^0_m + \tfrac{\Delta t_s}{12} \Big(&23 F_E(u^0_m) - 16 F_E(u^0_{m-1}) + 5 F_E(u^0_{m-2}) \\
&+ 5 F_I(u^0_{m+1}) + 8 F_I(u^0_m) - F_I(u^0_{m-1}) \Big). \qquad (10)
\end{aligned}$$

Both BDF and multistep methods require function values from multiple previous time-levels, values that are not available at the initial substeps of the first time step $[t_0, t_1]$. To generate these starting values for a Kth-order SIPIDC method, initial conditions at t_0 are advanced to t_1 using one time step (or K substeps) of the Kth-order SIPIDC method that uses the forward-backward Euler method in the prediction step. The substep values from this first step are then used as starting values for the subsequent time steps.

We use the notation SIPIDCK[$\mathrm{P}_{\mathrm{name}}$] to denote a Kth-order SIPIDC method using $\mathrm{P}_{\mathrm{name}}$ as the predictor, where $\mathrm{P}_{\mathrm{name}}$ is one of the methods described above. The forward-backward Euler method in (3) is used in the correction steps, hence, if a pth-order predictor is used to construct a Kth-order SIPIDC method, then the correction equation must be iterated $K - p$ times.

3. Linear stability analysis

One of the motivations for the development of high-order SIPIDC methods is that stable methods with very high order of accuracy can be easily constructed. This is in contrast to BDF methods, the stability of which degrades significantly as the order increases, and to IMEX RK methods, where no methods of order greater than five are known. When using either of these methods as predictors in a SIPIDC method, it is important to understand the effect these predictors have on the stability of the overall method.

Hence, the linear stability of SIPIDC methods using BDF or RK predictors is studied in this section. Traditionally, the stability of single-step implicit or explicit methods is studied by considering the model problem

$$\begin{aligned}
u'(t) &= \lambda u(t), \\
u(0) &= 1, \qquad (11)
\end{aligned}$$

for some complex constant λ. By applying a numerical method to this problem, one can derive an amplification factor $\rho(\lambda\Delta t)$, such that

$$u_{n+1} = \rho(\lambda\Delta t)u_n,$$

where u_n is the numerical solution at the nth time step.

When studying the linear stability of semi-implicit methods, one must specify how the standard model problem Equation (11) is decomposed into explicit and implicit parts. Numerous choices of the splitting have appeared in the literature [Frank et al. 1997; Ascher et al. 1995; Pareschi and Russo 2001; Zhong 1996; Pareschi and Russo 2005]. The most general approach is to decompose the problem into explicit and implicit terms by

$$\begin{aligned} u'(t) &= \lambda_E u + \lambda_I u, \\ u(0) &= 1, \end{aligned}$$

where λ_E and λ_I are complex constants [Frank et al. 1997; Pareschi and Russo 2001; Liotta et al. 2000; Pareschi and Russo 2005; Zhong 1996]. Then additional constraints are made to define a stability region which depends only on a single complex number. For example in [Frank et al. 1997] the stability region is defined as the set of λ_I such that the method is stable for all λ_E in the stability region of the explicit method. This approach is also used in [Layton and Minion 2005] but is not used in the following comparisons, since by this definition a method could have a very large stability region despite a severe restriction on the step size due to the properties of the explicit method. Instead, the procedure used in [Ascher et al. 1995; Ascher et al. 1997; Minion 2003] is followed, wherein the imaginary part of the right side of Equation (11) is associated with the nonstiff process and treated explicitly, while the real part is associated with the stiff process and treated implicitly. It should be noted that, regardless of the choice of splitting, the scalar stability analysis only carries over to linear systems when the matrices which define the explicit and implicit terms are simultaneously diagonalizable.

SIPIDC methods using a p-step method in the prediction step advance $u(t_n)$ to $u(t_{n+1})$ using p starting values $u_{n-1,P}(\equiv u_n)$, $u_{n-1,P-1}, \ldots, u_{n-1,P-p+1}$, where P denotes the number of substeps. Let $\vec{u}_n$ denote the vector $(u_{n-1,0}, u_{n-1,1}, \ldots, u_{n-1,P})$. Then the procedure for advancing u_n to u_{n+1} can be written in matrix form:

$$M(\lambda\Delta t)\vec{u}_n = \vec{u}_{n+1},$$

where $M \in \Re^{P\times P}$ and depends on the product $\lambda\Delta t$. To define the stability region for this method, set $\Delta t = 1$ and denote by $\rho(\lambda)$, the maximum magnitude of the eigenvalues $M(\lambda)$. The stability region is then the set of λ such that $\rho(\lambda) \leq 1$. For SIPIDC methods with single-step predictors, this definition reduces to the usual

definition of the amplification factor of a method. In the following, the stability regions for SIPIDC methods with multistep predictors are numerically computed by setting $\vec{u}_n$ to be e_j for $j = 1, \ldots, P$, where the i-th entry of e_j is given by

$$(e_j)_i = \begin{cases} 0, & i \neq j, \\ 1, & i = j. \end{cases}$$

For each λ, the resulting P vectors $\vec{u}_{n+1}$ form the P columns of $M(\lambda)$. MATLAB is used to compute the maximum of the magnitude of eigenvalues of $M(\lambda)$ at a regular array of points in the complex plane. The condition number of the eigenvalues are also monitored to check for degenerate eigenvalues with magnitude near 1, but none were found. The standard definition of $A(\alpha)$-stability [Widlund 1967] is easily extended to the semi-implicit case by defining a method to be $A(\alpha)$-stable for some $\alpha > 0$, if the defined stability region contains the region $\lambda = re^{i\theta}$, for all $\theta \in [\pi - \alpha, \pi + \alpha]$.

It is well known that the size of the stability region for implicit BDF methods decreases as the order increases; indeed, BDF methods with order above six are not acceptable [Gear 1971]. However, the stability properties of IMEX versions of these methods are not as well known and hence are investigated here. The numerically computed stability diagrams for IMEX BDF methods of orders 2, 3, 4, 6, and 7 are displayed in Figure 1. In this and all other figures in this section, the axes are scaled cubically to show both detail near the origin and the general

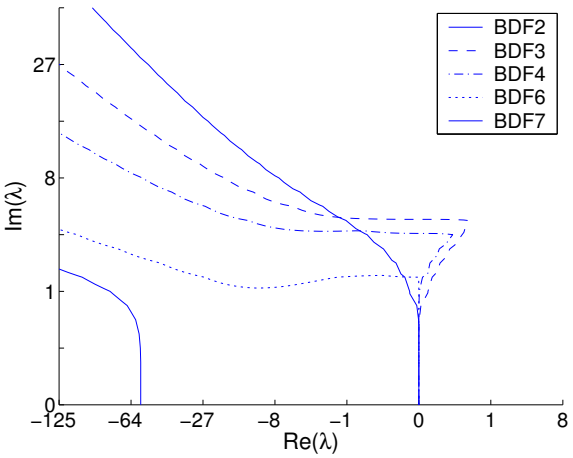


Figure 1. Stability diagrams for second-, third-, fourth-, sixth-, and seventh-order IMEX BDF. Stability regions for IMEX BDF decrease significantly as the order increases, and BDF7 is not stable near the origin.

shape of the stability region in the left half of the complex plane. Figure 1 shows that, as with fully implicit methods, the size of the stability regions of IMEX BDF methods decreases significantly as the order of the method increases, and that the seventh-order method is not stable near the origin. In particular, each method is $A(\alpha)$-stable with α decreasing with increasing order. The stability of certain IMEX RK methods of orders up to three has been studied previously (e.g. [Ascher et al. 1997]). For completeness, we include a plot of the stability regions of the ARK methods of orders 3 and 4 used in this study in Figure 2, which demonstrates that both methods are $A(\alpha)$-stable with similar stability regions. As noted previously, we do not consider fifth-order ARK as it has been deemed to be not competitive [Kennedy and Carpenter 2003], and we are not aware of sixth- or higher-order IMEX RK methods.

Extending the standard definition of L-stability [Ehle 1969] requires care since

$$\lim_{\Re(\lambda)\to-\infty} \rho(\lambda) \tag{12}$$

will in general depend on how the limit is taken. Here we define a method to be $L(\alpha)$-stable if it is $A(\alpha)$-stable and the limit in Equation (12) is zero whenever the imaginary part of λ is fixed in the limit, that is,

$$\lim_{\substack{\Re(\lambda)\to-\infty \\ \Im(\lambda)\equiv c}} \rho(\lambda) = 0, \tag{13}$$

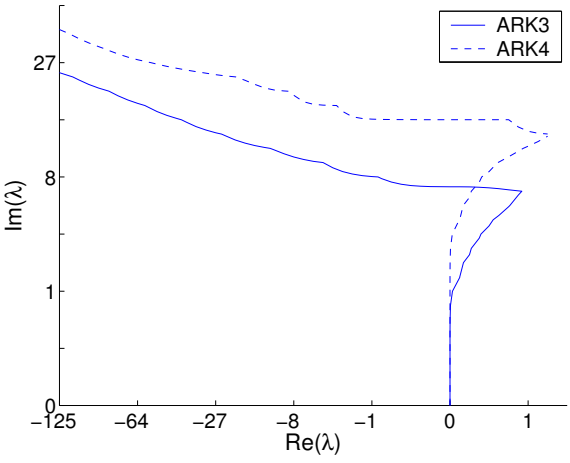


Figure 2. Stability diagrams for third- and fourth-order ARK methods.

for all $c \in \Re$. This, for example, would be the relevant infinitely diffusive stability limit of an approximation to an advection-diffusion equation based on finite differences and the method of lines.

It is shown in [Layton and Minion 2005] that $A(\alpha)$-stable SIPIDC methods can be constructed using forward-backward Euler methods, and that those methods using LR quadrature rules are also $L(\alpha)$-stable. Given an SIPIDC method for which the corrector is based on the forward-backward Euler method and for which the quadrature rule for the implicit piece does not include the left endpoint, one can show that if the predictor satisfies Equation (13), then the overall scheme will also (see [Layton and Minion 2005, Theorems 3.1–3.3].) Hence, since the IMEX BDF and RK methods that are used as predictors in this paper are $L(\alpha)$-stable, $A(\alpha)$-stability for the SIPIDC methods in this paper implies $L(\alpha)$-stability. As an example, stability regions for the SIPIDC6[ARK3] method are shown in Figure 3. In this figure, stability curves corresponding to $\rho(\lambda) = 0.001$ 0.01, 0.1, and 1 are shown to demonstrate that the method is $L(\alpha)$-stable.

An $L(\alpha)$-stable method can also be constructed using BDF3 in the prediction step (not shown). Note also that the stability region of the SIPIDC6[ARK3] method corresponding to $\rho = 1$ in Figure 3 is significantly larger than the stability region of IMEX BDF6 (see Figure 1). However, the SIPIDC6[ARK3] method is also computationally more expensive than IMEX BDF6, owing to the deferred correction iterations and the multiple stages. Thus, it is not immediately clear that for a given

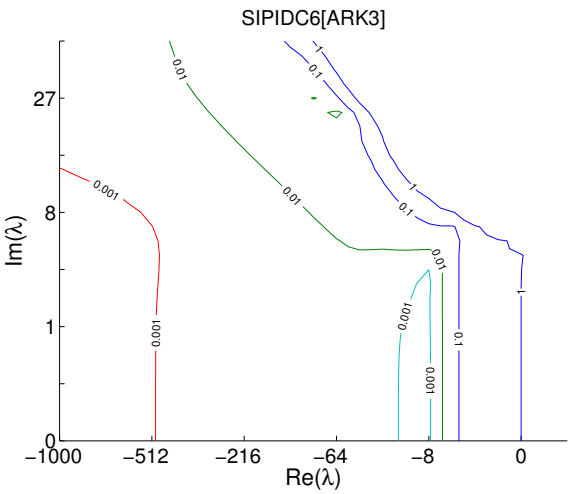


Figure 3. Contours of the amplification factor $\rho(\lambda)$ for the SIPIDC6 method using ARK3 in the provisional step. SIPIDC6[BDF3] is $L(\alpha)$-stable.

computational cost, the SIPIDC[ARK3] has a larger stability region than IMEX BDF6. This issue is further investigated below using *scaled* stability diagrams.

We will now use three different examples to demonstrate the main point of this section, namely that higher-order SIPIDC methods using moderate-order predictors have similar stability regions as the predictors. A corollary to this is that combining moderate-order IMEX BDF or ARK methods with SIPIDC corrections results in a higher-order method with better stability characteristics than the corresponding higher-order IMEX BDF or RK methods. We will demonstrate these points with three separate comparisons:

(1) a comparison of SIPIDC methods of a fixed order using different types of predictors of the same order (that is, IMEX BDF, RK, or multistep);

(2) a comparison of SIPIDC methods of a fixed order using one specific type of predictor with differing orders;

(3) a comparison of SIPIDC methods of differing order using one specific type of predictor with fixed order.

In the first example, we consider the effect of applying SIPIDC corrections on the stability region of different third-order predictors. To this end, we obtain stability diagrams (contour curves of $|\rho| = 1$) for IMEX BDF3, ARK3, and ABAM. These stability diagrams, shown in Figure 4A, indicate that BDF3 and ARK3 are $A(\alpha)$-stable, whereas ABAM is not. SIPIDC methods using the above three methods as predictors (not shown) exhibit similar stability properties as the predictors, that is, SIPIDC6[BDF3] and SIPIDC6[ARK3] are $A(\alpha)$-stable, but SIPIDC6[ABAM] is not.

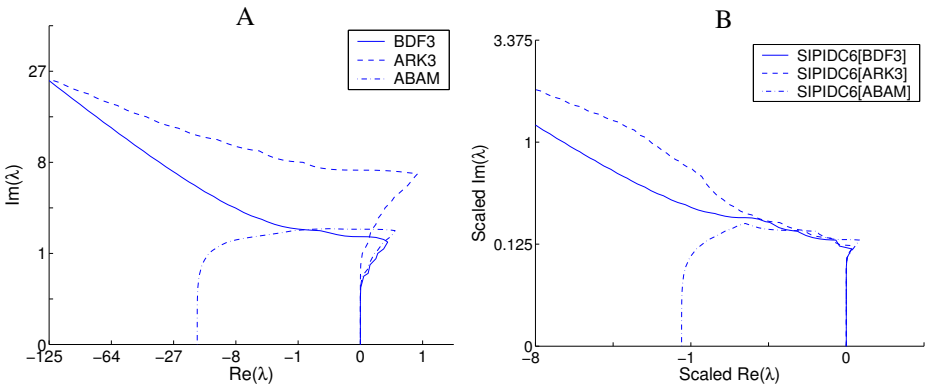


Figure 4. A: stability diagrams for three third-order ODE methods: IMEX BDF3, ARK3, and ABAM. B: scaled stability diagrams for SIPIDC6 methods using the methods in panel A as predictors.

Although the ARK3 method has the largest stability region of the three predictors above (see Figure 4A), ARK3 is also more computationally expensive owing to the multiple stages required. To take into account the additional computational costs, we show *scaled* stability diagrams for SIPIDC6 methods using the three predictors in Figure 4B. By assuming that the solution of the implicit part of the system is much more expensive than the explicit part (even though in the model problem (11), the solution of the implicit piece is a simple scalar division), the computational costs of SIPIDC methods are measured in terms of the numbers of implicit solves. To obtain the scaled stability diagrams, Re(λ) and Im(λ) are divided by the number of implicit function evaluations. These results show that even with computational costs taken into account, SIPIDC6[ARK3] still has the largest stability region.

We now present the second example to examine the effect on the stability of the overall SIPIDC method for a given type of predictor of differing orders. To this end, we compare the stability of SIPIDC6 methods implemented using first-order forward-backward Euler, IMEX BDF2, and IMEX BDF3 in the prediction step, and using forward-backward Euler methods in the correction steps. Figure 5 shows the scaled stability diagrams for the three SIPIDC6 methods, with the stability diagram for BDF6 included for comparison. (Note that the BDF6 method is applied to compute the solution at each substep of the SIPIDC methods as is done with the other BDF predictors, hence the stability region for BDF6 is scaled by a factor of 6 compared to Figure 1.) The unscaled stability diagrams for SIPIDC6[Euler], SIPIDC6[BDF2], SIPIDC6[BDF3], and BDF6 are qualitatively similar to those for the predictors (Figure 1); however, when computational costs are taken into account, the relative size of the stability diagrams change. The scaled stability region

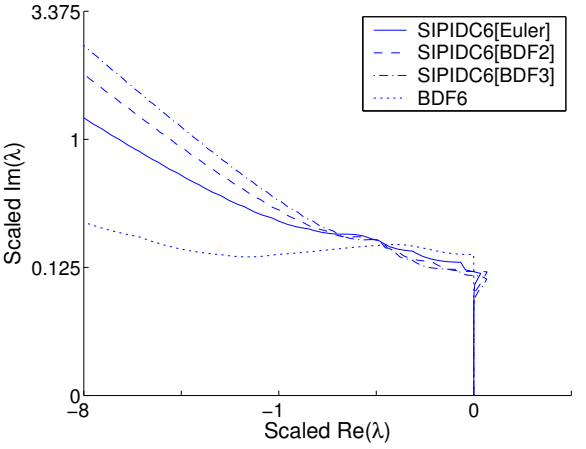


Figure 5. Scaled stability diagrams for SIPIDC6[Euler], SIPIDC6[BDF2], SIPIDC6[BDF3], and IMEX BDF6.

associated with the SIPIDC6[BDF3] is the largest, followed by SIPIDC6[BDF2], and by SIPIDC6[Euler]. Also noteworthy is that the stability regions of all three SIPIDC6 methods are substantially larger than that of the BDF6 method, even when the stability diagrams are scaled by the computational costs. This suggests that applying PIDC steps to a provisional solution computed by a BDF method generates an approximation with accuracy comparable to that computed by a high-order BDF method, without a decrease in the size of the stability region associated with the BDF6 scheme.

Finally, we consider the stability regions of SIPIDC schemes of varying order using the BDF3 scheme as a predictor. Figure 6 shows the scaled stability regions for SIPIDCk[BDF3] schemes for k ranging from 4 to 7, as well as that of the BDF3 method for comparison. Each method is $A(\alpha)$-stable with roughly the same α. Comparing Figure 6 with Figure 1 further demonstrates that higher-order SIPIDC methods do not suffer from a reduction in the size of the stability region as do the BDF methods. Note in particular that the stability region for SIPIDC7[BDF3] method is not significantly smaller than that of the moderate-order methods.

The accuracy and stability of SIPIDC methods using predictors of differing types and orders will be further assessed in Section 4 using more complex problems.

4. Numerical examples

In this section, numerical examples are used to further assess 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

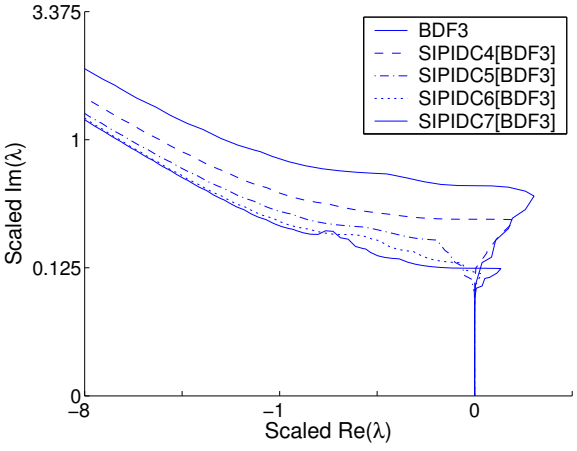


Figure 6. Scaled stability diagrams for IMEX BDF3 and SIPIDCk[BDF3] methods with $k = 4$ through 7.

prescribes the motion of a particle $x(t)$ governed by

$$x''(t) + \mu(1 - x(t)^2)x'(t) + x(t) = 0.$$

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

$$y_1(t)' = y_2(t), \tag{14}$$

$$y_2(t)' = \tfrac{1}{\epsilon}\big(-y_1(t) + (1 - y_1(t)^2)y_2(t)\big), \tag{15}$$

where $\epsilon = 1/\mu^2$. As ϵ approaches zero, these equations become increasingly stiff. In the integration of (14) and (15), the first equation is treated explicitly, whereas the second equation is treated implicitly. Equations (14) and (15) are integrated for $t \in [0, 0.5]$ with the equilibrium initial conditions shown in Table 1. Because an exact solution is not known for this problem, errors are computed from a reference solution obtained using a 7th-order implicit PIDC[Euler] method and a very small time step, chosen so that the solutions computed with the PIDC method and the ARK4(3)6L[2]SA method in [Kennedy and Carpenter 2003] agree to 14 digits.

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

Table 1. Initial conditions for van der Pol's equation.

The second example is a linear system of four equations given by

$$y'(t) = Ay + By \tag{16}$$

where $B \in \Re^{4\times4}$ contains at least one eigenvalue with a large negative real part that scales as $1/\epsilon$, and $A \in \Re^{4\times4}$ has eigenvalues close to the origin. A and B are given

by

$$A = \begin{pmatrix} 0 & c_1 & 0 & 0 \\ -c_1 & 2a_1 & 0 & 0 \\ 0 & 0 & 0 & c_2 \\ 0 & 0 & -c_2 & 2a_2 \end{pmatrix}, \qquad D = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -10 & 0 & 0 \\ 0 & 0 & -10^2 & 0 \\ 0 & 0 & 0 & 1/\epsilon \end{pmatrix},$$

$$S = \begin{pmatrix} 1 & 0.5 & 0 & 0 \\ 0 & 1 & 0.5 & 0 \\ 0 & 0 & 1 & 0.5 \\ 0.5 & 0 & 0 & 1 \end{pmatrix}, \qquad B = SDS^{-1}$$

where $a_1 = -0.2$, $b_1 = 5$, $a_2 = -0.4$, $b_2 = 12$, and $c_k = \sqrt{a_k + b_k}$ for $k = 1$ and 2. If ϵ is chosen carefully, then the sum $A + B$ contains one complex eigenvalue pair with small negative real part, and two negative real eigenvalues, one with magnitude of $\sim 1/\epsilon$. A and B do not commute, so the eigenvalues of $A + B$ do not correspond to the sum of eigenvalues of A and B. Equation (16) is integrated for $t \in [0.4, 2.4]$. The initial conditions are chosen to be the sum of the two normalized eigenvectors corresponding to the complex eigenvalues, so that transients are eliminated from the solution. We refer to this example as the *linear system test*.

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

$$\begin{aligned} y(t)' &= -2\pi \sin(2\pi t) - \tfrac{1}{\epsilon}\big(y - \cos(2\pi t)\big), \\ y(0) &= 0, \end{aligned}$$

for $t \in [0, 10]$. The exact solution of is $y(t) = \cos(2\pi t)$, and as $\epsilon \to 0$, this equation becomes increasingly stiff. In this implementation, SIPIDC methods treat the term $-2\pi \sin(2\pi t)$ explicitly and the term $-(y - \cos(2\pi t))/\epsilon$ implicitly. A slightly more general problem was studied in [Prothero and Robinson 1974] and is considered here in Appendix A, since its simplicity allows an explicit examination of dominant error terms.

Because SIPIDC methods can be used to integrate ODEs arising from a method-of-lines discretization of PDEs, we include here a PDE example: the *Kuramoto–Silvashinsky (KS) equation*, which is used in [Akrivis and Smyrlis 2004] to study the accuracy of IMEX BDF methods. The inhomogeneous KS equation is given by

$$\begin{aligned} u_t + uu_x + u_{xx} + \nu u_{xxxx} &= f(x, t), \\ u(x, 0) &= g(x), \end{aligned} \tag{17}$$

for $x \in [0, 2\pi]$ and $t \in [0, T]$ and periodic boundary conditions $u(x + 2\pi, t) = u(x, t)$. As in [Akrivis and Smyrlis 2004], the functions $f(x, t)$ and $g(x)$ are constructed so that the exact solution is $u(x, t) = \sin(x + t)$; T and μ are taken to be 1 and 0.5, respectively. Equation (17) is first discretized in space using a pseudo-spectral

method as in [Akrivis and Smyrlis 2004], and then integrated in time using SIPIDC methods.

Note that the use of periodic boundary conditions for this problem avoids the issue of how to correctly impose boundary conditions for the provisional solutions in SIPIDC methods applied to PDEs with time-dependent boundary conditions. It is now well established that a naive imposition of the exact boundary conditions for PDEs within a RK method often results in a reduction of order of accuracy in the solution [Sanz-Serna et al. 1987; Carpenter et al. 1995], and a similar problem exists for PIDC methods. Strategies for addressing this problem have been proposed for RK methods for certain classes of problems (see [Abarbanel et al. 1996; Pathria 1997; Calvo and Palencia 2002; Alonso-Mallo 2002b; Alonso-Mallo 2002a; Portero et al. 2004]). Results in this direction for PIDC methods will be reported in future works.

All calculations reported below were performed using MATLAB programs. For brevity, we report results of only one or two examples for each study. Unless otherwise stated, qualitatively similar results were also obtained using other examples. For the ODE problems, 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. For the KS equation, the error reported is the discrete L_2 norm of the error at the final time.

4.1. ***Efficiency improvement due to deferred corrections.*** We first assess the effect on the accuracy and stability of solutions computed by IMEX BDF and ARK methods after SIPIDC correction steps have been applied to those solutions. To this end, we compare the efficiency of IMEX BDF and ARK methods with SIPIDC methods that use these BDF and ARK methods in the prediction step. The SIPIDC methods use the first-order Euler method in the correction steps to improve the accuracy of the intermediate approximations. As noted previously, the solution of the implicit part of the ODEs is assumed to be much more expensive than the explicit part. For simplicity, we further assume that the implicit solves in all methods have similar computational costs. With these assumptions, we measure computational costs in terms of the numbers of implicit solves. Recall that starting values required for IMEX BDF and multistep methods are generated using a SIPIDC[Euler] method to advance the initial solution to t_1. The computational cost associated with this initial step is included in the total cost.

A comparison among SIPIDC5, SIPIDC7, BDF, and ARK methods using the van der Pol and cosine problems is shown in Figure 7. The comparison is obtained for the nonstiff case, with the stiffness parameter ϵ set to 10^{-1}. We first compare IMEX BDF methods with SIPIDC methods that use BDF as predictor. The left panels of Figure 7 show log-log plots of solution error versus the number of implicit function evaluations obtained using SIPIDC5[BDFk], SIPIDC7[BDFk], and BDFk methods,

for $k = 3$ and 4. The errors shown in Figure 7 for the van der Pol problem are for y_2; results for y_1 are similar. For a sufficiently high accuracy requirement, the method with the highest order, i.e., the SIPIDC7[BDFk] method, is the most efficient; and both SIPIDC5[BDFk] and SIPIDC7[BDFk] methods are more efficient than IMEX BDF3 and BDF4.

The comparisons between SIPIDC5[ARKk], SIPIDC7[ARKk], and ARKk, for $k = 3$ and 4, are similar. The results shown in the right-hand panels of Figure 7 indicate that, for a sufficiently high accuracy requirement, SIPIDC7[ARKk] is the most efficient, and that both SIPIDC methods are more efficient than the moderate-order ARK methods. Similar results (not shown) were also obtained for SIPIDC methods of order > 4, using a third- or fourth-order predictor and at least one correction step.

4.2. ***Comparison of predictors.*** In the next set of tests, we compare the efficiency of SIPIDC methods using different predictors. We first consider predictors of

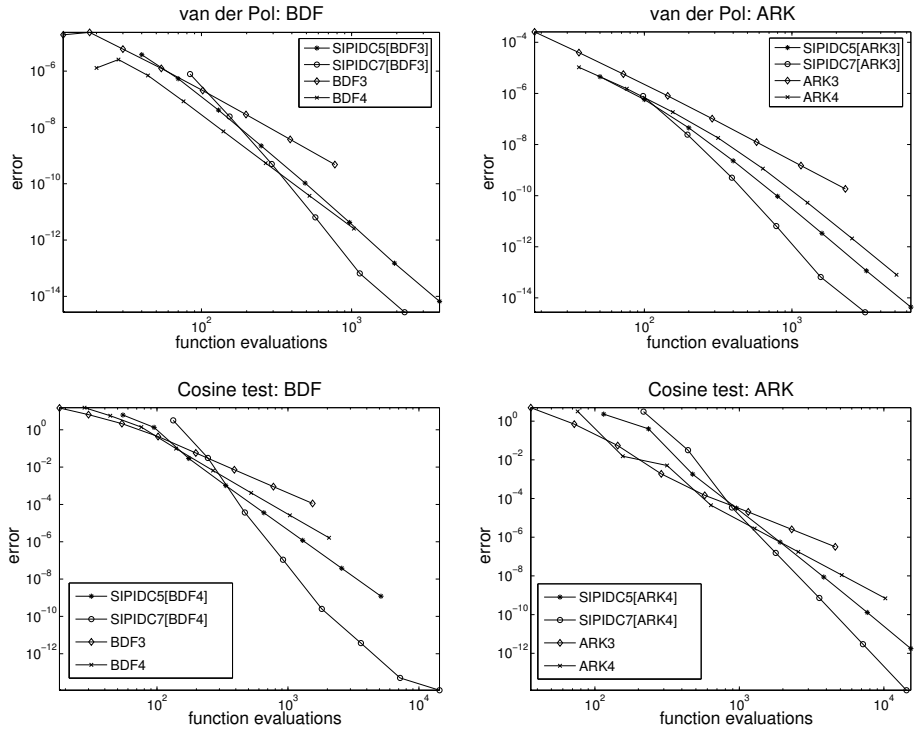


Figure 7. Efficiency comparison for SIPIDC5, SIPIDC7, IMEX BDF, and ARK methods using the van der Pol and cosine tests with $\epsilon = 10^{-1}$. Results show that correction steps improve efficiency of overall methods.

the same order but differing types (for example, BDF2 versus RK2). Results are obtained for the van der Pol problem with both nonstiff and stiff parameters. In the first set of experiments, we compare SIPIDC methods using second-order methods in the prediction step. Figure 8A1 and Figure 8A2 compare the efficiency of SIPIDC6 methods using three different second-order predictors—BDF2, RK2, and CNAB. Error curves obtained for IMEX BDF4 and ARK4 are also included for comparison. The stiffness parameters ϵ are 10^{-1} and 10^{-4} for results in Figure 8A1 and Figure 8A2, respectively. For the nonstiff problem (panel A1), the SIPIDC6[BDF2] and SIPIDC6[CNAB] methods, which have similar accuracy and the same computational costs, are the most efficient for sufficiently high accuracy requirement. (The two error curves approximately overlap.) These methods are more efficient than the SIPIDC6[RK2] method because of the lower computational costs in their prediction

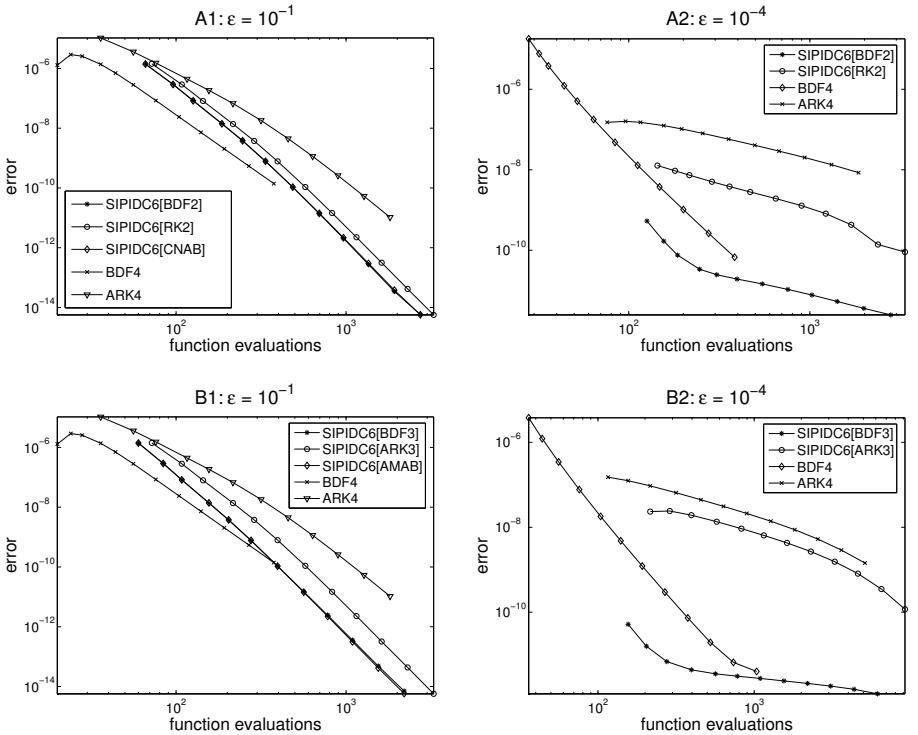


Figure 8. Efficiency comparison among SIPIDC6 methods using various predictors. Results for van der Pol problem. Errors in y_2 shown. A1, B1: $\epsilon = 10^{-1}$. A2, B2: $\epsilon = 10^{-4}$. A1, A2: SIPIDC6 methods with second-order predictors. B1, B2: SIPIDC6 methods with third-order predictors. Results for IMEX BDF4 and ARK4 are also included for comparison.

step. The IMEX BDF4 and ARK4 are less efficient, as expected, at sufficiently high accuracy requirement.

For the stiff case, the results shown in Figure 8A2 are markedly different. First, in this case SIPIDC6[CNAB] is unstable for sufficiently large Δt, owing to its lack of L-stability, and thus its error curve is not shown. Secondly, although approximations computed by BDF4 converge at fourth order, the SIPIDC6[RK2] and ARK4 methods appear to be converging at approximately a first-order rate in the range of Δt shown. Finally, the SIPIDC6[BDF2] method exhibits two regions of convergence: an approximately first-order convergence region at sufficiently small Δt and a higher-order region at larger Δt (although the latter region is too small for the order of convergence to be determined).

The results of the above tests are now presented using third-order predictors—IMEX BDF3, ARK3, and AMAB. The nonstiff results ($\epsilon = 10^{-1}$) are shown in

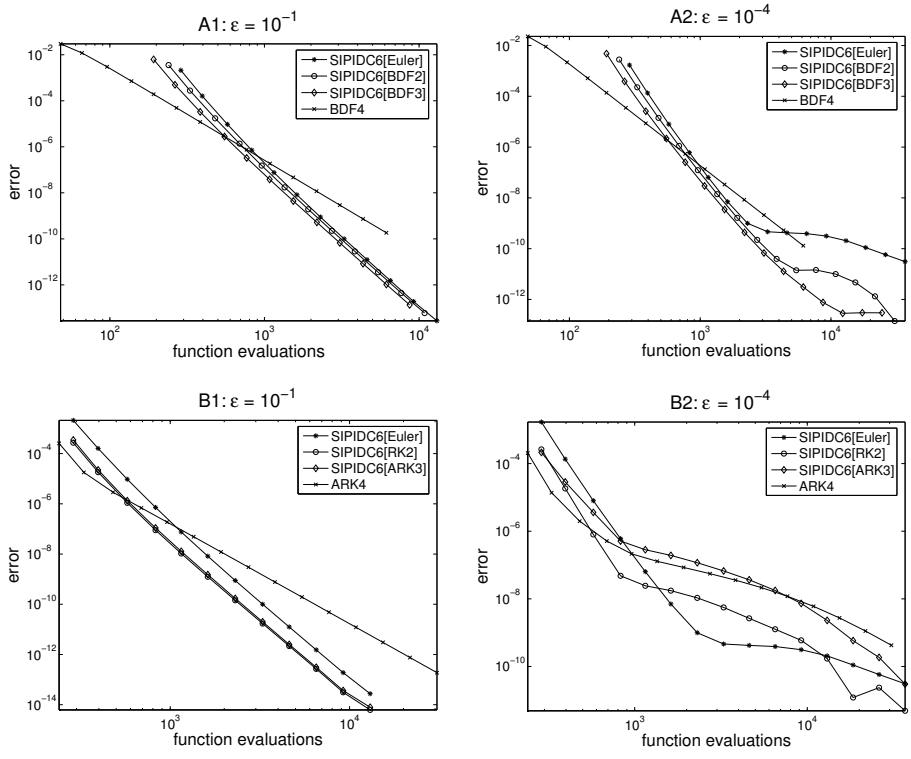


Figure 9. Efficiency comparison among SIPIDC6 methods using various predictors. Results for linear system problem. A1, B1: $\epsilon = 10^{-1}$. A2, B2: $\epsilon = 10^{-4}$. A1, A2: SIPIDC6 methods with Euler, BDF2, and BDF3 predictors, and BDF4. B1, B2: SIPIDC6 methods with Euler, RK2, and ARK3 predictors, and ARK4.

Figure 8B1 and are very similar to those for the second-order predictors in Figure 8B2. The SIPIDC6[BDF3] and SIPIDC6[AMAB] methods are more efficient than SIPIDC6[ARK3] as well as the BDF4 and ARK4 methods for a sufficiently high accuracy requirement.

For the stiff problem ($\epsilon = 10^{-4}$) the observed results shown in panel B2 are again different from the nonstiff results, although they are similar to the results for second-order methods in the stiff case shown in panel A2. The SIPIDC6[AMAB] method, with a predictor that is not L-stable, is unstable like the SIPIDC6[CNAB] above and is not shown. Both ARK4 and SIPIDC6[ARK3] appear to be converging at approximately a first-order rate in the range of Δt shown, while the BDF4 method converges at the proper order. Also, the solutions computed by SIPIDC6[BDF3], show two different convergence regimes, although the limits of machine precision make it difficult to determine the respective rates. The order reduction behavior of SIPIDC methods with BDF and RK predictors will be further investigated in Section 4.3.

We will now compare predictors of the same type but differing orders. For a SIPIDCK method that is based on a first-order method, K implicit solves are required (one for each of the K substeps) for the provisional step and for each of the $K-1$ correction steps. Thus, a total of K^2 implicit solves are required. On the other hand, an SIPIDCK method that uses a first-order corrector but a pth-order predictor requiring s implicit solves per substep will require $K-p$ correction iterations and thus a total of $(K-p+s)K$ implicit solves per time step. Hence, assuming the implicit solves require similar computational costs for all methods, the resulting SIPIDC methods have the same order but a smaller computational cost if $p > s$ (e.g BDF methods where $s = 1$). However, regardless of whether $p > s$, it is not clear that increasing the order of the predictor results in a more efficient method in terms of error per function evaluation.

The linear system test is used to assess the extent to which the efficiency of a SIPIDC method is improved by using IMEX BDF and RK methods in the prediction step, first for a nonstiff problem with $\epsilon = 1$. For BDF methods, Figure 9A1 compares the efficiency of SIPIDC6 methods using BDF predictors of order one (Euler) through three. The error curve for BDF4 is included for comparison. In this case SIPIDC6[BDF3] requires the fewest correction steps and it is indeed the most efficient, albeit by a slight amount. Next we compare the efficiency of SIPIDC6 methods using IMEX RK-type predictors of differing orders (see Figure 9B1). For this comparison, the three SIPIDC6 methods have similar computational costs, but SIPIDC6[ARK3] and SIPIDC[RK2] appear more efficient than SIPIDC6[Euler]. Similar results were also obtained for SIPIDC methods of other overall orders.

The comparison is repeated for a stiff problem ($\epsilon = 10^{-4}$) in Figures 9A2 and 9B2. Results in Figure 9A2 show the advantage of using a moderate-order BDF method

in the provisional step. The error curve for SIPIDC6[Euler] shows three regions of convergence: for sufficiently large Δt (fewer than 2×10^3 function evaluations) and for sufficiently small Δt (more than 10^4 function evaluations, where convergence begins to increase), convergence is approximately sixth order; however, order reduction is observed for middle range Δt, where the curve is flat (i.e., zeroth-order convergence). Unlike SIPIDC6[Euler], the error curve corresponding to the order reduction region for SIPIDC6[BDF2] is less flat, although the region is too small for a reasonable estimate of the order of accuracy. Order reduction is not observed for BDF4, which is consistent with the analysis for fully implicit BDF methods (see [Hairer and Wanner 1991] Chapter V). Finally, the behavior of the SIPIDC6[BDF3] is difficult to determine due to machine precision. The extent of order reduction of SIPIDC methods using BDF predictors of differing orders is further investigated in Section 4.3

The stiff test is repeated for SIPIDC6[Euler], SIPIDC6[RK2], SIPIDC6[ARK3], and ARK4 and the results are shown in Figure 9B2. Three regions of convergence were obtained for each of these methods. As noted previously, the order reduction region for SIPIDC6[Euler] error curve is flat. In contrast, the order reduction region for the error curves associated with SIPIDC6[RK2], SIPIDC6[ARK3], and ARK4 appears to be first order. For sufficiently small Δt, the methods will again exhibit full order accuracy (in the absence of precision errors).

4.3. ***Order reduction.*** Numerical results in [Layton and Minion 2005] show that the characteristics of order reduction of SIPIDC methods depends critically on the choice of quadrature nodes: when uniform nodes are used and when the left endpoint is not used in the quadrature rule associated with the implicit piece (recall that such quadrature nodes are referred to as "LR" [Layton and Minion 2005]), an order reduction to $\mathbb{O}(\epsilon^2)$ is observed, compared to $\mathbb{O}(\epsilon\Delta t)$ for SIPIDC methods using nonuniform nodes (for example, Gauss quadrature nodes) or those including the left endpoint in the quadrature rules. SIPIDC methods studied in [Layton and Minion 2005] use Euler in both the provisional and correction steps. Below we examine convergence behavior and order reduction for stiff problems of SIPIDC methods using moderate-order methods in the provisional step, using first the van der Pol's problem and then the cosine problem.

To investigate the dependence of order reduction on the choice of predictor, we computed solutions for the van der Pol equation for increasing stiffness (for $\epsilon = 10^{-k}$, $k = 1, 3, 4, 5, 6$) by means of SIPIDC5 methods using different predictors (Euler, BDF2, BDF3, and ARK3). Log-log plots of errors for y_2 versus implicit function evaluations are shown in Figure 10. For sufficiently stiff parameters ($\epsilon < 10^{-3}$), the convergence rate drops to the zeroth order for SIPIDC5[Euler], to the first order for SIPIDC5[BDF2] and SIPIDC5[ARK3], and to the second

order for SIPIDC5[BDF3] in the order-reduction regime. We see that for the methods SIPIDC5[Euler], SIPIDC5[BDF2], and SIPIDC5[BDF3], the magnitude of the error in the regions of reduced convergence scales approximate as ϵ^2, and for SIPIDC5[ARK3] it scales approximately as ϵ. It is noteworthy that the order reduction results for SIPIDC5[ARK3] are similar to the SIPIDC[Euler] method using nonuniform points or using the left endpoint in the quadrature rules [Layton and Minion 2005]. Similar results are also shown for the cosine problem for increasingly stiff values of ϵ in Figure 11, shown as log-log plots of errors versus time step size Δt.

The above results for SIPIDC5[Euler] are consistent with those reported in [Layton and Minion 2005] for SIPIDC6[Euler] and SIPIDC7[Euler] using LR uniform nodes. The error formula derived in [Layton and Minion 2005] shows that the dominant error term for these methods, after one correction step, is $\mathbb{O}(\epsilon^2)$; thus, the region of reduced convergence is flat with magnitude that scales as ϵ^2.

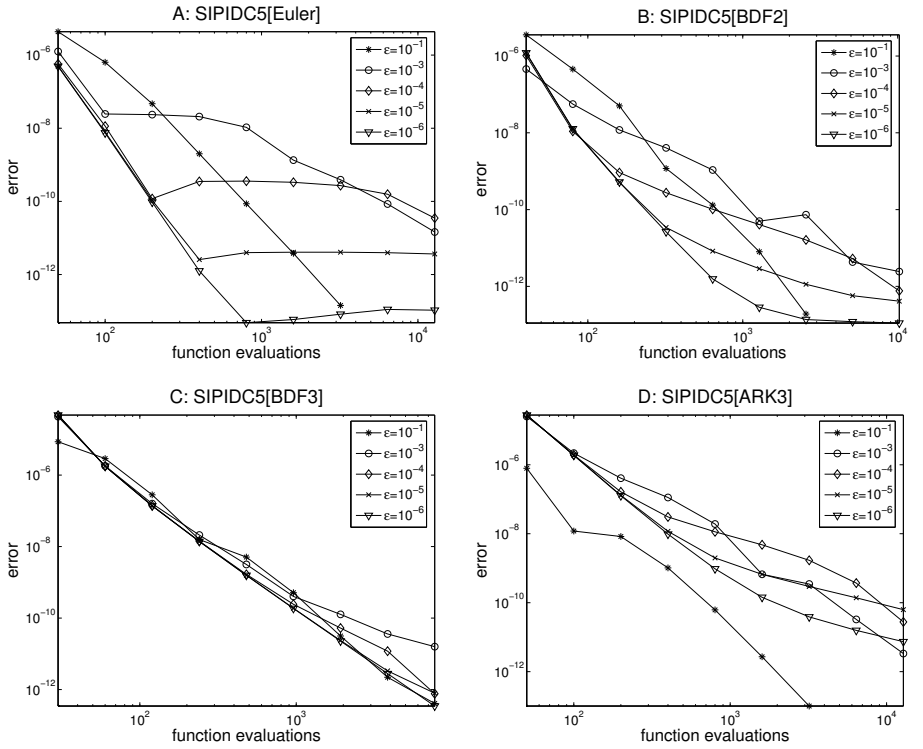


Figure 10. Error curves obtained for the van der Pol problem with a range of ϵ values, computed using the SIPIDC5 methods with differing predictors. The region of order reduction shows zeroth-order convergence in A, first-order in B and D, second-order in C.

Analogous error formulae are derived in the Appendix for SIPIDC[BDF2] and SIPIDC[BDF3]; these error formulae show that for SIPIDC[BDF2], the dominant error term is $\mathbb{O}(\epsilon^2\Delta t)$, and for SIPIDC[BDF3], it is $\mathbb{O}(\epsilon^2\Delta t^2)$, thereby explaining the shape of the error curves shown in Figure 10 and Figure 11.

4.4. ***A ladder approach.*** The approximations computed by the provisional step and by the initial correction steps have lower orders of accuracy than the final solution. A "ladder" approach makes use of this fact to reduce the computational cost of a SIPIDC method without compromising the overall order of the solution. This is achieved by allowing larger temporal or spatial errors in the initial PIDC iterations. One such ladder approach was implemented in [Minion 2003]. To obtain a Kth-order solution, the quadrature $\mathbb{Q}$ in (5) must be approximated to Kth order. When LR uniformly-spaced nodes are used, $K+1$ nodes or K substeps are required. In [Minion 2003], based on the observation that the kth correction equation

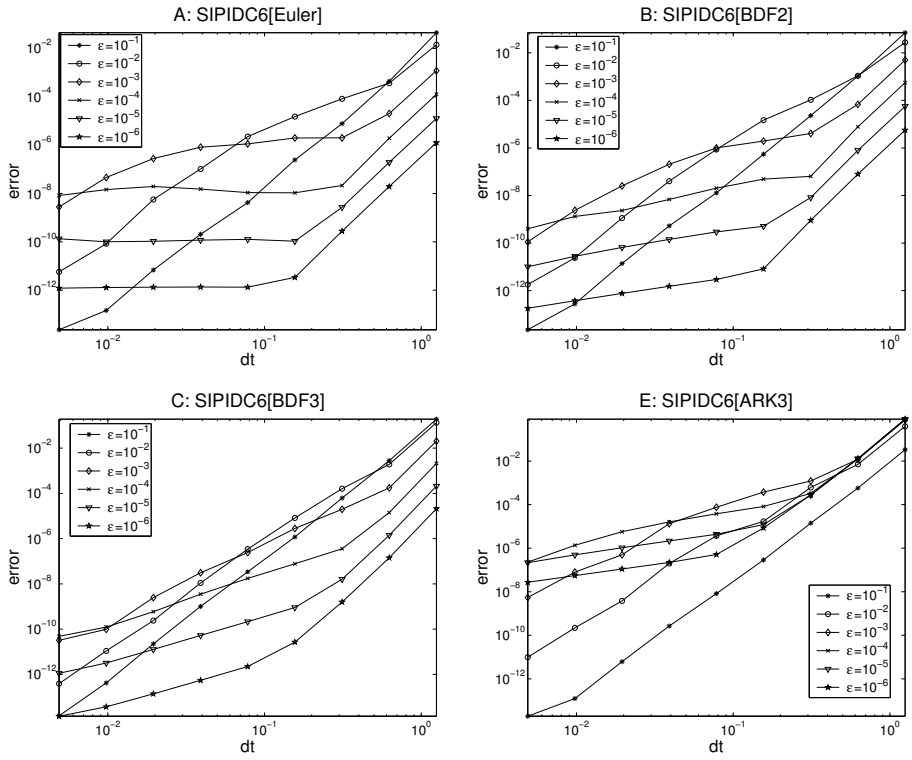


Figure 11. Error curves obtained for the cosine problem with a range of ϵ values, computed using the SIPIDC methods with differing predictors. The region of order reduction shows $\mathbb{O}(\epsilon^2)$ in A, $\mathbb{O}(\epsilon^2\Delta t)$ in B, $\mathbb{O}(\epsilon^2\Delta t)$ C, and $\mathbb{O}(\epsilon\Delta t)$ in D.

computes a globally $\mathbb{O}(\Delta t^{k+1})$ approximation (Euler is used in the predictor in [Minion 2003]), the number of substeps used to compute the solution during the initial PIDC iterations was reduced, i.e., fewer substeps were used when k is small. Although no significant improvement in efficiency was noted in [Minion 2003] when this approach was applied to a linear problem, the nonlinear KS equation (17) is used here to re-examine the effects of ladder approach, with a new focus on SIPIDC methods using BDF2 and BDF3 as predictors.

We compare the efficiency among SIPIDC6 methods using Euler, BDF2, and BDF3 methods in the provisional step, and with or without incorporating the ladder approach. Results are shown in Figure 12 . Consistent with results described previously, SIPIDC6 methods using moderate-order predictors are more efficient. Also, although the ladder approach reduces computational cost, it also increases error. These two competing effects result in a negligible improvement in efficiency. Qualitatively similar results were also obtained for SIPIDC methods of other orders.

5. Discussion

We have presented alternative implementations of SIPIDC methods for the temporal integration of ODEs with both nonstiff and stiff components corresponding to eigenvalues with large negative real part. In these implementations, various types of second- through fourth-order integration methods are used in the prediction step. The stability and efficiency of these SIPIDC methods are assessed and compared to traditional IMEX methods. High-order SIPIDC methods are proposed as alternatives

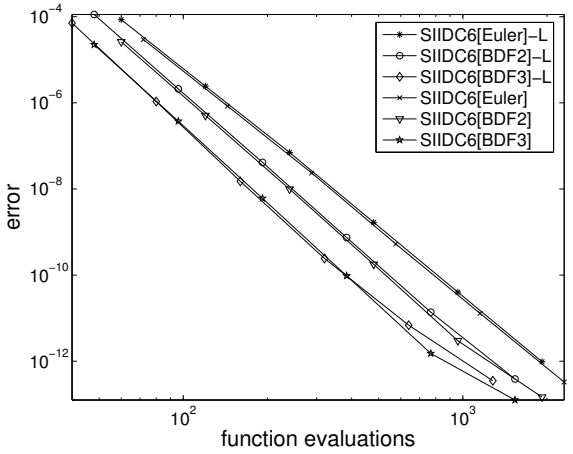


Figure 12. Error curves obtained for the KS equation using the SIPIDC6 methods. '-L' denotes methods using ladder approach.

to IMEX BDF or IMEX RK methods, which exhibit instability at high orders. In contrast, our stability analysis shows attractive stability properties for high-order SIPIDC methods using a moderate-order IMEX BDF or IMEX RK method in the prediction step.

Another goal of this study is to determine whether SIPIDC methods that use moderate-order predictors are more efficient. Numerical results suggest that using moderate-order IMEX BDF methods in the prediction step gives rise to SIPIDC methods that are more efficient and also stable for stiff problems. In contrast, moderate-order predictors based on IMEX RK do not significantly improve efficiency because of the multiple implicit solves required at the stages, and predictors based on multistep methods such as AMAB result in overall methods that are unstable when applied to stiff problems.

Although we only consider SIPIDC methods that use the forward-backward Euler method to solve the correction equations, moderate-order integration methods can no doubt be used in the correction steps. For example, the correction Equation (2) may be discretized by means of a second-order method, for example, CNAB, IMEX RK2, or IMEX BDF2. Such methods require fewer iterations of the correction equation to achieve the same overall order of accuracy relative to methods based on first-order methods, but each iteration of the correction equation may be more expensive. The behavior of various moderate-order correctors is the focus of an on-going project.

IMEX BDF predictors also change the extent and characteristics of order reduction of the SIPIDC methods when applied to stiff problems. The convergence rate in the region of order reduction is $k-1$ for a kth-order BDF predictor, with errors of $\mathbb{O}(\epsilon^2)$ magnitude, where ϵ is the stiffness parameter such that as $\epsilon \to 0$ the problem becomes increasingly stiff. In contrast, IMEX RK predictors give rise to first-order convergence in the region of order reduction, with $\mathbb{O}(\epsilon)$ errors. Thus, for stiff problems SIPIDC methods with IMEX BDF predictors likely generate solutions with higher accuracy than SIPIDC methods using non-BDF predictors.

A uniform time-step has been assumed throughout this work. However, SIPIDC methods are suitable candidates for adaptive time-marching: the correction term can be used to dynamically determine the appropriate time step size to meet certain accuracy requirements. However, when a BDF or multistep predictor is used, where solution values at k previous substeps are needed to advance the solution, care must be taken in computing the provisional solution at the first $k-1$ substeps, because the substep size may not be equal in $[t_{n-1}, t_n]$ and $[t_n, t_{n+1}]$. In this case, variable-step form of the methods can be used for the first $k-1$ iterations, where the coefficients in (7)–(9) and in (10) depend on the relative substep sizes.

The ultimate target applications for PIDC methods are PDEs with multiple stiff terms, such as the advection-diffusion-reaction equations. Indeed, in earlier studies

[Bourlioux et al. 2003; Layton and Minion 2004], we have proposed the multi-implicit PIDC (MIPIDC) methods (formerly MISDC methods) which decouple the stiff processes and integrate them separately, possibly using differing time steps. The MIPIDC methods developed so far are based on the forward/backward Euler methods and a first-order splitting. A project that develops and analyzes the performance of MIPIDC methods based on moderate-order IMEX BDF methods and on a moderate-order splitting is underway. It should be noted that no analysis of semi- and multi-implicit PIDC methods applied to PDEs with stiffness characterized by rapidly oscillatory modes (that is, corresponding to eigenvalues with large imaginary parts) has yet been attempted.

A ladder approach, which uses fewer substeps in the initial PIDC iterations, fails to significantly improve the efficiency of SIPIDC methods. Because of the extra effort involved in its implementation, the value of this ladder approach is not obvious. Alternatively, when integrating a PDE, one may use a less refined spatial grid during the initial PIDC iterations. This spatial ladder approach is likely to be particularly effective in higher spatial dimensions and warrants attention.

Appendix

In this Appendix we develop an analytical formulation for the truncation error for SIPIDC methods applied to the simple stiff equation analyzed in [Prothero and Robinson 1974].

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

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

Here ϵ is the stiffness parameter where the equation becomes more stiff as $\epsilon \to 0$. We integrate (A.1) 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 first consider a provisional solution computed using the BDF2 method given by (7). 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 $e_m^0 = y_m^0 - p_m$, one step of BDF2 applied to Equation (A.1) yields

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

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

$$\frac{1}{\frac{3}{2} + \frac{\Delta t_m}{\epsilon}} = \frac{\epsilon}{\Delta t_m}\left(1 - \frac{3}{2}\frac{\epsilon}{\Delta t_m} + \left(\frac{3}{2}\frac{\epsilon}{\Delta t_m}\right)^2 - \dots\right). \tag{A.3}$$

Substituting (A.3) into (A.2) yields

$$y^0_{m+1} = p_{m+1} + \Big(2p'_m - p'_{m-1} + \frac{2y^0_m - \frac{1}{2}y^0_{m-1} - \frac{3}{2}p_{m+1}}{\Delta t_m}\Big) \times \left(\epsilon - \frac{3}{2}\frac{\epsilon^2}{\Delta t_m} + \left(\frac{3}{2}\frac{\epsilon}{\Delta t_m}\right)^2\epsilon - \dots\right) \quad \text{(A.4)}$$

To simplify (A.4), we make use of the following relations derived using Taylor's expansion

$$\begin{aligned} 2p'_m - p'_{m-1} &= p'_{m+1} - \Delta t_m^2 p^{(3)}_{m+1} + \Delta t_m^3 p^{(4)}_{m+1} + \mathbb{O}(\Delta t_m^4), \\ -\frac{3}{2\Delta t_m}p_{m+1} &= -p'_{m+1} - \frac{1}{\Delta t_m}\left(2p_m - \tfrac{1}{2}p_{m-1}\right) + \frac{\Delta t_m^2}{3}p^{(3)}_{m+1} \\ &\quad + \tfrac{3}{4}\Delta t_m^3 p^{(4)}_{m+1} + \mathbb{O}(\Delta t_m^4). \quad \text{(A.5)} \end{aligned}$$

From (A.5), one obtains

$$\begin{aligned} 2p'_m - p'_{m-1} + \frac{2y_m - \frac{1}{2}y_{m-1} - \frac{3}{2}p_{m+1}}{\Delta t_m} &= \frac{2e_m - \frac{1}{2}e_{m-1}}{\Delta t_m} \\ &\quad - \tfrac{2}{3}\Delta t_m^2 p^{(3)}_{m+1} + \tfrac{3}{4}\Delta t_m^2 p^{(4)}_{m+1} \quad \text{(A.6)} \end{aligned}$$

Thus,

$$y_{m+1} = p_{m+1} + \Big(\frac{2e_m - \frac{1}{2}e_{m-1}}{\Delta t_m} - \tfrac{2}{3}\Delta t_m^2 p^{(3)}_{m+1} + \tfrac{3}{4}\Delta t_m^3 p^{(4)}_{m+1}\Big) \times \left(\epsilon - \tfrac{3}{2}\frac{\epsilon^2}{\Delta t_m} + \left(\tfrac{3}{2}\frac{\epsilon}{\Delta t_m}\right)^2\epsilon - \dots\right). \quad \text{(A.7)}$$

Substituting (A.6) into (A.4) and making use of the definition of $e^0_m \equiv p_m - y^0_m$,

$$\begin{aligned} e^0_{m+1} &= \frac{2e^0_m - \frac{1}{2}e^0_{m-1}}{\Delta t_m}\left(\epsilon - \frac{3}{2}\frac{\epsilon^2}{\Delta t_m} + \Big(\frac{3}{2}\frac{\epsilon}{\Delta t_m}\Big)\epsilon\right) \\ &\quad -\tfrac{2}{3}\Delta t_m^2 p^{(3)}_{m+1}\left(\epsilon - \frac{3}{2}\frac{\epsilon^2}{\Delta t_m} + \Big(\frac{3}{2}\frac{\epsilon}{\Delta t_m}\Big)\epsilon\right) + \mathbb{O}(\epsilon\Delta t^3) + \mathbb{O}(\epsilon^2\Delta t^2) + \mathbb{O}(\epsilon^3). \quad \text{(A.8)} \end{aligned}$$

Now consider the correction equation given a provisional solution y^0_m. Note that $f(y^0_m, t_m) = p'_m - \frac{1}{\epsilon}e^0_m$. The direct form of the correction equation using

forward-backward Euler for (A.1) is

$$y^1_{m+1} = y^1_m + \Delta t_m\Big(p'_m - p'_m - \frac{1}{\epsilon}(y^1_{m+1} - p_{m+1}) + \frac{1}{\epsilon}(y^0_{m+1} + p_{m+1})\Big) + \mathcal{Q}_m^{m+1}(y^0)$$
$$= y^1_m + \Delta t_m\Big(-\frac{1}{\epsilon}(y^1_{m+1} - y^0_{m+1})\Big) + \mathcal{Q}_m^{m+1}(y^0).$$

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

$$y^1_{m+1} = \frac{y^1_m + \frac{\Delta t_m}{\epsilon} y^0_{m+1} + \mathcal{Q}_m^{m+1}\big(p'(t) - \frac{1}{\epsilon}e^0(t)\big)}{1 + \frac{\Delta t_m}{\epsilon}}. \tag{A.9}$$

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 Equation (4) defines

$$\mathcal{Q}_m^{m+1}\left(p'(t) - \tfrac{1}{\epsilon}e^0(t)\right) = \Delta t_m \sum_{l=0}^{p} q_m^l \left(p'_l - \tfrac{1}{\epsilon}e^0_l\right).$$

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

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

Substituting this expression into Equation (A.9) gives

$$y_{m+1} = \frac{y^1_m + p_{m+1} - p_m + \frac{\Delta t_m}{\epsilon}\Big(y^0_{m+1} - \sum_{l=0}^{p} q_m^l e^0_l\Big) + O(\Delta t^q)}{1 + \frac{\Delta t_m}{\epsilon}}.$$

Applying the expansion (A.3), one obtains

$$y^1_{m+1} = \frac{\epsilon}{\Delta t_m}\Big(y^1_m + p_{m+1} - p_m + \frac{\Delta t_m}{\epsilon}\Big(y^0_{m+1} - \sum_{l=0}^{p} q_m^l e^0_l\Big) + O(\Delta t^q)\Big)$$
$$-\Big(\frac{\epsilon}{\Delta t_m}\Big)^2\Big(y^1_m + p_{m+1} - p_m + \frac{\Delta t_m}{\epsilon}\Big(y^0_{m+1} - \sum_{l=0}^{p} q_m^l e^0_l\Big) + O(\Delta t^q)\Big)$$
$$+\Big(\frac{\epsilon}{\Delta t_m}\Big)^3\Big(y^1_m + p_{m+1} - p_m + \frac{\Delta t_m}{\epsilon}\Big(y^0_{m+1} - \sum_{l=0}^{p} q_m^l e^0_l\Big) + O(\Delta t^q)\Big)\ldots,$$

hence

$$y_{m+1}^1 = y_{m+1}^0\left(1 - \frac{\epsilon}{\Delta t_m} + \left(\frac{\epsilon}{\Delta t_m}\right)^2 \dots\right) + \left(y_m^1 - p_m + p_{m+1}\right)\left(\frac{\epsilon}{\Delta t_m} - \left(\frac{\epsilon}{\Delta t_m}\right)^2 + \left(\frac{\epsilon}{\Delta t_m}\right)^3 \dots\right) - \left(\sum_{l=0}^{p} q_m^l e_l^0\right)\left(1 - \frac{\epsilon}{\Delta t_m} + \left(\frac{\epsilon}{\Delta t_m}\right)^2 \dots\right) + O(\epsilon \Delta t^{q-1}) + O(\epsilon^2 \Delta t^{q-2}) + O(\epsilon^3 \Delta t^{q-3}) \dots .$$

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

$$e_{m+1}^1 = e_m^1\left(\frac{\epsilon}{\Delta t_m} - \left(\frac{\epsilon}{\Delta t_m}\right)^2 + \left(\frac{\epsilon}{\Delta t_m}\right)^3 \dots\right) + \left(e_{m+1}^0 - \sum_{l=0}^{p} q_m^l e_l^0\right)\left(1 - \frac{\epsilon}{\Delta t_m} + \left(\frac{\epsilon}{\Delta t_m}\right)^2 \dots\right) + O(\epsilon \Delta t^{q-1}) + O(\epsilon^2 \Delta t^{q-2}) + O(\epsilon^3 \Delta t^{q-3}) + \dots . \quad \text{(A.10)}$$

Consider now the first time step of a SIPIDC method for Equation (A.1). Assume that the error at the beginning of the time step is given by e_0^0. The dominant error terms in the provisional solution Equation (A.8) are

$$e_{m+1}^0 = -\frac{2}{3}\Delta t_m^2 p_{m+1}^{(3)}\left(\epsilon - \frac{2}{3}\frac{\epsilon^2}{\Delta t_m}\right) + z_m, \quad \text{(A.11)}$$

where

$$z_m = \begin{cases} \frac{\epsilon}{\Delta t_m}\left(2e_0^0 - \frac{1}{2}e_{-1}^0\right), & m = 1, \\ -\frac{4}{3}p_1^{(3)}\Delta t_m \epsilon^2 - \frac{\epsilon}{2\Delta t_m}e_0^0, & m = 2, \\ \left(\frac{4}{3}p_m^{(3)} - \frac{1}{3}p_{m-1}^{(3)}\right)\Delta t_m \epsilon^2, & m > 2. \end{cases}$$

In deriving the above expression, we made use of the assumption of uniform substep, i.e., $\dots = t_{m-2} = t_{m-1} = t_m = \dots$. Likewise, the dominant pieces of the correction equation error (A.10) comes from the term

$$e_{m+1}^1 = e_{m+1}^0 - \sum_{l=0}^{p} q_m^l e_l^0 + e_m^1 \frac{\epsilon}{\Delta t_m}. \quad \text{(A.12)}$$

Substituting the dominant provisional error (A.11) into the dominant correction error (A.12) gives

$$e^1_{m+1} = -\frac{2}{3}\Delta t_m^2 p^{(3)}_{m+1}\Big(\epsilon - \frac{2}{3}\frac{\epsilon^2}{\Delta t_m}\Big) + z_m - \sum_{l=1}^{p} q_m^l \left(-\frac{2}{3}\Delta t_m^2 p_l^{(3)}\Big(\epsilon - \frac{2}{3}\frac{\epsilon^2}{\Delta t_m}\Big) + z_l\right). \quad \text{(A.13)}$$

The summation term can be rewritten via a Taylor series expansion

$$\sum_{l=1}^{p} q_m^l \left(-\frac{2}{3}\Delta t_m^2 p_l^{(3)}\Big(\epsilon - \frac{2}{3}\frac{\epsilon^2}{\Delta t_m}\Big) + z_l\right) = -\frac{2}{3}\Delta t_m^2 p^{(3)}_{m+1}\Big(\epsilon - \frac{2}{3}\frac{\epsilon^2}{\Delta t_m}\Big) + \mathbb{O}(\epsilon\Delta t_m^3 + \epsilon^2\Delta t_m^2) + \sum_{l=1}^{p} q_m^l z_l. \quad \text{(A.14)}$$

Substituting (A.14) into (A.13) and simplifying gives

$$e^1_{m+1} = z_m - \sum_{l=1}^{p} q_m^l z_l + \mathbb{O}(\Delta t^q + \epsilon\Delta t^3 + \epsilon^2\Delta t^2) \quad \text{(A.15)}$$

owing to the mismatch between z_m for $m = 1$ and 2 and for $m > 2$, $z_m - \sum q_m^l z_l = \mathbb{O}(\epsilon^2\Delta t_m)$. Thus, $e^1_{m+1} = \mathbb{O}(\Delta t^q + \epsilon^2\Delta t)$.

Following similar procedures, an error formula for the correction step of a SIPIDC method using uniform quadrature nodes and BDF3 in the predictor step can be shown to be $\mathbb{O}(\Delta t^p + \epsilon^2\Delta t^2)$.

References

[Abarbanel et al. 1996] S. Abarbanel, D. Gottlieb, and M. H. Carpenter, "On the removal of boundary errors caused by Runge–Kutta integration of nonlinear partial differential equations", *SIAM J. Sci. Comput.* **17** (1996), 777–782.

[Akrivis and Smyrlis 2004] G. Akrivis and Y.-S. Smyrlis, "Implicit-explicit BDF methods for the Kuramoto–Silvashinsky equation", *Appl. Numer. Math.* **51** (2004), 151–169.

[Akrivis et al. 1998] G. Akrivis, M. Crouzeix, and C. Makridakis, "Implicit-explicit multistep finite element methods for nonlinear parabolic equations", *Math. Comp.* **67** (1998), 457–477.

[Akrivis et al. 1999] G. Akrivis, M. Crouzeix, and C. Makridakis, "Implicit-explicit multistep methods for quasilinear parabolic equations", *Numer. Math.* **82** (1999), 521–541.

[Alonso-Mallo 2002a] B. Alonso-Mallo, I. and Cano, "Spectral/Rosenbrock discretizations without order reduction for linear parabolic problems", *Appl. Numer. Math.* **41** (2002), 247–268.

[Alonso-Mallo 2002b] I. Alonso-Mallo, "Runge–Kutta methods without order reduction for intial boundary value problems", *Numer. Math.* **91** (2002), 577–603.

[Ascher et al. 1995] U. M. Ascher, S. J. Ruuth, and B. T. R. Wetton, "Implicit-explicit methods for time-dependent partial differential equations", *SIAM J. Numer. Anal.* **32**:3 (1995), 797–823. MR 96j:65076

[Ascher et al. 1997] 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** (1997), 151–167.

[Berger and Oliger 1984] M. J. Berger and J. Oliger, "Adaptive mesh refinement for hyperbolic partial differential equations", *J. Comput. Phys.* **53**:3 (1984), 484–512. MR 85h:65211

[Bourlioux et al. 2003] A. Bourlioux, A. T. Layton, and M. L. Minion, "Higher-order multi-implicit spectral deferred correction methods for problems of reacting flow", *J. Comput. Phys.* 189 (2003), 351–376.

[Calvo and Palencia 2002] M. P. Calvo and C. Palencia, "Avoiding the order reduction of Runge–Kutta methods for linear initial boundary value problems", *Math. Comp.* **71**:240 (2002), 1529–1543. MR 2003h:65091

[Calvo et al. 2001] M. P. Calvo, J. de Frutos, and J. Novo, "Linearly implicit Runge-Kutta methods for advection-reaction-diffusion equations", *Appl. Numer. Math.* **37**:4 (2001), 535–549. MR 2002e:65114

[Carpenter et al. 1995] M. H. Carpenter, D. Gottlieb, S. Abarbanel, and W. S. Don, "The theoretical accuracy of Runge–Kutta time discretizations for the initial-boundary value problem: a study of the boundary error", *SIAM J. Sci. Comput.* **16**:6 (1995), 1241–1252. MR 96h:65088

[Dutt et al. 2000] A. Dutt, L. Greengard, and V. Rokhlin, "Spectral deferred correction methods for ordinary differential equations", *BIT* **40**:2 (2000), 241–266. MR 2001e:65104

[Ehle 1969] B. L. Ehle, *On Padé approximations to the exponential function and A-stable methods for the numerical solution of initial value problems*, Dept. Applied Analysis and Computer Sci., Research Rpt. CSRR 2010, University of Waterloo, 1969.

[Frank et al. 1997] J. Frank, W. Hundsdorfer, and J. G. Verwer, "On the stability of implicit-explicit linear multistep methods", *Appl. Numer. Math.* **25**:2-3 (1997), 193–205. Special issue on time integration (Amsterdam, 1996). MR 98m:65126

[Gear 1971] C. W. Gear, *Numerical initial value problems in ordinary differential equations*, Prentice-Hall, Englewood Cliffs, N.J., 1971. MR 47 #4447

[Hairer and Wanner 1991] E. Hairer and G. Wanner, *Solving ordinary differential equations. II*, Springer Series in Computational Mathematics **14**, Springer, Berlin, 1991. Stiff and differential-algebraic problems. MR 92a:65016

[in't Hout 2002] K. J. in't Hout, "On the contractivity of implicit-explcit linear multistep methods", *Appl. Numer. Math.* **42** (2002), 201–212.

[Kennedy and Carpenter 2003] C. A. Kennedy and M. H. Carpenter, "Additive Runge–Kutta schemes for convection-diffusion-reaction equations", *Appl. Numer. Math.* **44**:1-2 (2003), 139–181. MR 2003m:65111

[Layton and Minion 2004] A. T. Layton and M. L. Minion, "Conservative multi-implicit spectral deferred correction methods for reacting gas dynamics", *J. Comput. Phys.* **194**:2 (2004), 697–715. MR 2004k:76089

[Layton and Minion 2005] 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 (2005), 341–373. MR 2006h:65087

[Liotta et al. 2000] S. F. Liotta, V. Romano, and G. Russo, "Central schemes for balance laws of relaxation type", *SIAM J. Numer. Anal.* **38**:4 (2000), 1337–1356. MR 2001j:65128

[Minion 2003] M. L. Minion, "Semi-implicit spectral deferred correction methods for ordinary differential equations", *Commun. Math. Sci.* **1**:3 (2003), 471–500. MR 2005f:65085

[Minion 2004] M. L. Minion, "Semi-implicit projection methods for incompressible flow based on spectral deferred corrections", *Appl. Numer. Math.* **48**:3-4 (2004), 369–387. Workshop on Innovative Time Integrators for PDEs. MR MR2056924

[Pareschi and Russo 2001] L. Pareschi and G. Russo, "Implicit-explicit Runge–Kutta schemes for stiff systems of differential equations", pp. 269–288 in *Recent trends in numerical analysis*, Adv. Theory Comput. Math. **3**, Nova Sci. Publ., Huntington, NY, 2001. MR 2005a:65065

[Pareschi and Russo 2005] L. Pareschi and G. Russo, "Implicit-explicit Runge–Kutta schemes and applications to hyperbolic systems with relaxation", *J. Sci. Comput.* **25**:1-2 (2005), 129–155. MR MR2231946

[Pathria 1997] D. Pathria, "The correct formulation of intermediate boundary conditions for Runge–Kutta time integration of initial-boundary value problems", *SIAM J. Sci. Comput.* **18**:5 (1997), 1255–1266. MR 98d:65100

[Portero et al. 2004] L. Portero, J. C. Jorge, and B. Bujanda, "Avoiding order reduction of fractional step Runge–Kutta discretizations for linear time dependent coefficient parabolic problems", *Appl. Numer. Math.* **48**:3-4 (2004), 409–424. Workshop on Innovative Time Integrators for PDEs. MR 2005d:65153

[Prothero and Robinson 1974] A. Prothero and 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. MR 48 #10125

[Sanz-Serna et al. 1987] 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**:4 (1987), 405–418. MR 88f:65146

[Shen and Zhong 1996] J. W. Shen and X. Zhong, "Semi-implicit Runge–Kutta schemes for the non-autonomous differential equations in reactive flow computations", in *Proceedings of the 27th AIAA Fluid Dynamics Conference* (New Orleans, 1996), Amer. Inst. Aeronautics and Astronautics, June 1996.

[Widlund 1967] O. B. Widlund, "A note on unconditionally stable linear multistep methods", *Nordisk Tidskr. Informations-Behandling* **7** (1967), 65–70. MR 35 #6373

[Zhong 1996] X. Zhong, "Additive semi-implicit Runge–Kutta methods for computing high-speed nonequilibrium reactive flows", *J. Comput. Phys.* **128**:1 (1996), 19–31. MR 97e:80019

Received December 5, 2005.

ANITA T. LAYTON: alayton@math.duke.edu
Department of Mathematics, Duke University, Box 90320, Durham, NC 27708, United States
http://www.math.duke.edu/faculty/alayton

MICHAEL L. MINION: minion@amath.unc.edu
Department of Mathematics, CB 3250 Phillips Hall, University of North Carolina, Chapel Hill, NC 27599, United States
http://amath.unc.edu/Minion