Applied Numerical Mathematics 59 (2009) 1629–1643

Contents lists available at ScienceDirect

Applied Numerical Mathematics

www.elsevier.com/locate/apnum

On the efficiency of spectral deferred correction methods for time-dependent partial differential equations

Anita T. Layton

Department of Mathematics, Duke University, NC 27708, USA

ARTICLE INFO

Article history:
Received 15 February 2008
Received in revised form 10 November 2008
Accepted 11 November 2008
Available online 14 November 2008

MSC:
65B05
65M20

Keywords:
Multiscale problem
Operator splitting
Stiff equations
Semi-implicit methods
Picard integral

ABSTRACT

Many physical and biological systems involve the interactions of two or more processes with widely-differing characteristic time scales. Previously, high-order semi-implicit and multi-implicit formulations of the spectral deferred correction methods (denoted by SISDC and MISDC methods, respectively) have been proposed for solving partial differential equations arising in such model systems. These methods compute a temporally high-order approximation by means of a first-order numerical method, which solves a series of correction equations to increase the temporal order of accuracy of the approximation. MISDC methods also allow several fast-evolving processes to be handled implicitly but independently, allowing for different time steps for each process while avoiding the splitting errors present in traditional operator-splitting methods. In this study, we propose MISDC methods that use second- and third-order integration and splitting methods in the prediction steps, and we assess the efficiency of SISDC and MISDC methods that are based on those moderate-order integration methods. Numerical results indicate that SISDC methods using third-order prediction steps are the most efficient, but the efficiency of SISDC methods using first-order steps improves, particularly in higher spatial dimensions, when combined with a "ladder approach" that uses a less refined spatial discretization during the initial SDC iterations. Among the MISDC methods studied, the one with a third-order prediction step is the most efficient for a mildly-stiff problem, but the method with a first-order prediction step has the least splitting error and thus the highest efficiency for a stiff problem. Furthermore, a MISDC method using a second-order prediction step with Strang splitting generates approximations with large splitting errors, compared with methods that use a different operator-splitting approach that orders the integration of processes according to their relative stiffness.

1. Introduction

The dynamics of many physical and biological systems (e.g., combustion and transport of air pollutants) involve several processes with differing characteristic time scales. When the time scales of the physical processes vary widely, efficient solution of the partial differential equations (PDEs) that describe the dynamics may require specialized numerical methods. Thus, in previous studies, high-order semi-implicit spectral deferred correction (SISDC) [7,12,14–16] and multi-implicit spectral deferred correction (MISDC) [4,13] methods were proposed for computing solutions of such systems. SISDC and MISDC methods are generalizations of the explicit and implicit spectral deferred correction (SDC) methods introduced in [6] and further analyzed in [8,11]. SDC methods use a low-order numerical method to compute a high-order approximation. 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.

E-mail address: alayton@math.duke.edu.

0168-9274/$30.00
doi:10.1016/j.apnum.2008.11.004

To integrate time-dependent PDEs using the SISDC or MISDC methods, one first discretizes the PDEs in space using the method of lines (MOL) approach, which yields in a large coupled system of ordinary differential equations (ODEs). For systems involving processes with large disparity in time scales, the ODE system is stiff. By integrating non-stiff terms explicitly, SISDC methods are more efficient than fully-implicit methods, which solve implicit equations that couple every term in the system, some of which may be nonlinear. By incorporating the operator-splitting approach, MISDC methods provide an even more affordable numerical solution, particularly for systems with multiple fast-moving processes: by decoupling processes and integrating them sequentially, MISDC methods generate implicit equations that are even easier to solve than in the semi-implicit approach. Moreover, different time steps may be used for different processes. Unlike traditional operator-splitting approaches such as Strang splitting [17], the temporal order of accuracy of MISDC methods can be arbitrarily high because both the integration and splitting errors are eliminated during the deferred correction process. Numerical results in [4] demonstrate that one may improve the efficiency of a MISDC method by selectively reducing only the time step for the fastest-moving process, and that MISDC methods compare favorably with semi-implicit Runge–Kutta methods in terms of efficiency [10].

Most of the early formulations of SDC, SISDC, and MISDC methods use forward and backward Euler methods to compute the provisional solutions and to solve the correction equations [4,6]. Provided that a sufficiently high level of accuracy is desired, and/or the temporal interval is sufficiently long (owing to accumulation of numerical truncation errors), high-order methods are more efficient than low-order methods. Thus, SISDC methods were developed based on moderate-order integration methods [7,12,15]. Numerical results in [15] suggest that using moderate-order BDF methods in the prediction step gives rise to SISDC methods that are more efficient, compared to SISDC methods that use Euler methods, because the former require fewer correction steps to attain a given (high) order of accuracy. Moreover, the use of moderate-order predictor changes the extent and characteristics of order reduction of the resulting SISDC methods when applied to stiff problems [15]. However, the benefits of using second-order integration methods in the correction step are less clear [12]. Results in [12,15] were obtained using ODE problems. Because the development of SDC methods was motivated, in part, by multiscale PDE problems, in this study, we propose MISDC methods that use second- and third-order integration and splitting methods in the prediction steps; in particular, Strang splitting is used to compute a temporally second-order provisional solution of a MISDC method, and a novel operator-splitting approach is developed to compute a temporally third-order provisional solution. We assess the accuracy and efficiency of SISDC and MISDC methods that are based on moderate-order integration methods using PDE problems.

The computational cost of an SISDC or MISDC method may also be reduced by means of a ladder approach, which takes advantage of the lower order of accuracy of the approximations computed by the initial SDC iterations and allows larger temporal or spatial errors in those initial iterations. Another goal of this study is to examine the effectiveness of different ladder approaches using PDE examples. Because the accuracy and efficiency of SISDC and MISDC methods have been compared favorably to existing integration methods (e.g., implicit–explicit (IMEX) Runge–Kutta methods) [4,16], in this study we focus on the comparison of efficiency among SISDC and MISDC methods.

2. SISDC methods

SISDC methods are suitable for solving ODEs and PDEs that involve two processes of differing time scales; see [16] for a detailed derivation of the SISDC methods. Below we give a brief review in the context of a PDE problem. Let $u(x,t)$ be a function that satisfies the PDE

$$u_t = f_E + f_I, \tag{1}$$

where f_E and f_I are functions of x, t, u, and the spatial derivatives of u. The time scale of the process associated with f_I is assumed to be substantially shorter than f_E. To complete the specification of the problem, boundary conditions and initial conditions must be given.

The solution to (1) is approximated by means of the MOL approach. To this end, one first discretizes (1) spatially to obtain a system of ODEs

$$u'(t) = F_E\big(t,u(t)\big) + F_I\big(t,u(t)\big), \tag{2}$$

$$u(a) = u_0, \tag{3}$$

for $t \in [a,b]$. The terms F_E and F_I are obtained from the spatial discretization of f_E and f_I, respectively. Because F_I is assumed to be much stiffer than F_E, SISDC methods treat 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 SISDC 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.

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 $F(u_m^k)$. Let

$\mathcal{E}_m^{m+1}(F)$ and $\mathcal{I}_m^{m+1}(F)$ be explicit and implicit, respectively, numerical integration approximations to $\int_{t_m}^{t_{m+1}} F(\tau)\,d\tau$. Typically, $\mathcal{E}$ and $\mathcal{I}$ are obtained using low- or moderate-order methods, e.g., forward and backward Euler methods as in [4,14]. To advance the solution from t_n to t_{n+1}, SISDC methods first compute in a prediction step a provisional solution $\tilde{u}(t_m) \equiv u_m^0$, for $m = 0, 1, \ldots, P$, by means of an $(s+1)$-step semi-implicit method:

$$u_{m+1}^0 = \sum_{j=0}^{s} b_j u_{m-j}^0 + \mathcal{E}_m^{m+1}\big(F_E\big(u^0\big)\big) + \mathcal{I}_m^{m+1}\big(F_I\big(u^0\big)\big). \tag{4}$$

We then seek to improve the accuracy of $\tilde{u}$ by iteratively approximating the correction $\delta(t) \equiv u(t) - \tilde{u}(t)$ given by

$$\begin{aligned}\delta(t) = \int_a^t &\big(F_E\big(\tau, \tilde{u}(\tau) + \delta(\tau)\big) - F_E\big(\tau, \tilde{u}(\tau)\big) + F_I\big(\tau, \tilde{u}(\tau) + \delta(\tau)\big)\\ &- F_I\big(\tau, \tilde{u}(\tau)\big)\big)\,d\tau + E\big(t, \tilde{u}(t)\big),\end{aligned} \tag{5}$$

where E is the residual function given by

$$E\big(t, \tilde{u}(t)\big) = u_0 + \int_a^t F_E\big(\tau, \tilde{u}(\tau)\big) + F_I\big(\tau, \tilde{u}(\tau)\big)\,d\tau - \tilde{u}(t). \tag{6}$$

See [16] for a detailed derivation of (5).

To use δ to improve the accuracy of $\tilde{u}$, one obtains an update equation by adding $\tilde{u}$ to both sides of (5). Let $\mathcal{Q}_m^{m+1}(F)$ be a numerical quadrature approximation to $\int_{t_m}^{t_{m+1}} F(\tau)\,d\tau$. Then at the kth iteration, one solves the following equation

$$\begin{aligned}u_{m+1}^{k+1} = &\sum_{j=0}^{s} b_j u_{m-j}^{k+1} + \mathcal{E}_m^{m+1}\big(F_E\big(u^{k+1}\big) - F_E\big(u^k\big)\big)\\ &+ \mathcal{I}_m^{m+1}\big(F_I\big(u^{k+1}\big) - F_I\big(u^k\big)\big) + \mathcal{Q}_m^{m+1}\big(F_E\big(u^k\big) + F_I\big(u^k\big)\big).\end{aligned} \tag{7}$$

The quadrature $\mathcal{Q}$ should have at least the same order of accuracy as the updated approximation u^{k+1}. As in [4,16], 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 some implementations, the integration methods used to obtain $\mathcal{E}$ and $\mathcal{I}$ in (4) may differ from those used in the correction equation (7), and different methods may also be used for each iteration of (7). For instance, one may compute the provisional solution using a third-order IMEX method, and then solve the correction equation using forward and backward Euler methods. Nonetheless, for notational simplicity, the same symbols $\mathcal{E}$ and $\mathcal{I}$ are used in (4) and (7).

2.1. Test problems

The accuracy and efficiency of SISDC methods using integration methods in the prediction and correction steps are assessed using PDE examples. Specifically, we consider the Burgers equation and the Allen–Cahn equation:

Burgers equation

$$u_t = -uu_x + \epsilon u_{xx}, \tag{8}$$

$$u(x, 0) = \frac{1}{2} + \frac{1}{2}\sin(2\pi x), \tag{9}$$

with periodic boundary conditions, for $x \in [0, 1]$ and $t \in [0, 0.25]$. The diffusion coefficient ϵ is set to 0.01.

Allen–Cahn equation

$$u_t = \epsilon u_{xx} + u - u^3, \tag{10}$$

$$u(x, 0) = \frac{1}{2} - \frac{1}{2}\tanh(20x - 10), \tag{11}$$

for $x \in [0, 1]$ and $t \in [0, 0.25]$, with Dirichlet boundary conditions $u(-\infty, t) = 1$ and $u(\infty, t) = 0$. Strictly speaking, the boundary conditions hold only at $x = \pm\infty$ for finite t. Nevertheless, for the time and space intervals considered in this example, u is sufficiently close to 1 and 0 at the left and right boundary points. The diffusion coefficient ϵ is set to 0.01.

These test problems were chosen because each involves two processes of significantly different time scales: the slow processes arise from the advection and diffusion terms, for the Burgers equation and for the Allen–Cahn equation, respectively, whereas the fast processes arise from the diffusion and reaction terms, for the Burgers equation and for the Allen–Cahn equation, respectively.

In both examples, the equations are approximated using the MOL approach: first, the equations are discretized in space using sixth-order centered differencing; the resulting ODEs are then integrated in time using third-, fourth-, and fifth-order

SISDC methods. In the Burgers equation, the advection term $-uu_x$ is integrated explicitly and the diffusion term ϵu_{xx} implicitly; whereas in the Allen–Cahn equation, the diffusion term ϵu_{xx} is integrated explicitly and the reaction term $u-u^3$ implicitly. The nonlinear equations associated with the reaction term in the Allen–Cahn equation are solved using Newton's method.

2.2. Moderate-order predictions and corrections

Let SISDCK[n_pPn_cC] denote a Kth-order SISDC method having an n_pth-order prediction step and an n_cth-order correction step ($n_p+n_c \leqslant K$). Results in our previous study [15] indicate that moderate-order IMEX BDF, owing to their efficiency and to their stability and accuracy when applied to stiff ODEs (desirable properties that have been revealed in other studies [1,3,9]), are better methods to be used in the prediction step, compared to IMEX Runge–Kutta and other multi-step methods. Thus, below we consider three implementations of the prediction steps: one based on forward and backward Euler methods, one on a second-order IMEX BDF, and one on a third-order IMEX BDF. Specifically, these methods compute provisional solutions as follows:

$$\text{Euler:}\quad u^0_{m+1} = u^0_m + \Delta t_m\big(F_E(u^0_m) + F_I(u^0_{m+1})\big), \tag{12}$$

$$\text{BDF2:}\quad \frac{3}{2}u^0_{m+1} = 2u^0_m - u^0_{m-1} + \Delta t_m\big(2F_E(u^0_m) - F_E(u^0_{m-1}) + F_I(u^0_{m+1})\big), \tag{13}$$

$$\text{BDF3:}\quad \frac{11}{6}u^0_{m+1} = 3u^0_m - \frac{3}{2}u^0_{m-1} + \frac{1}{3}u^0_{m-2} + \Delta t_m\big(3F_E(u^0_m) - 3F_E(u^0_{m-1}) + F_E(u^0_{m-2}) + F_I(u^0_{m+1})\big). \tag{14}$$

Two implementations of the correction steps are considered: one based on forward and backward Euler methods, and the other on IMEX BDF2. The discretized correction equations are given by [12]:

Euler:

$$u^{k+1}_{m+1} = u^{k+1}_m + \Delta t_m\big(F_E(u^{k+1}_m) - F_E(u^k_m) + F_I(u^{k+1}_{m+1}) - F_I(u^k_{m+1})\big) + \mathcal{Q}^{m+1}_m\big(F_E(u^k) + F_I(u^k)\big), \tag{15}$$

BDF2:

$$\begin{aligned} u^{k+1}_{m+1} = {} & 2u^{k+1}_m - \frac{1}{2}u^{k+1}_{m-1} + \Delta t_s\big(2F_E(u^{k+1}_m) - F_E(u^{k+1}_{m-1}) - 2F_E(u^k_m) \\ & + F_E(u^k_{m-1}) + F_I(u^{k+1}_{m+1}) - F_I(u^k_{m+1})\big) + \frac{3}{2}\mathcal{Q}^{m+1}_m\big(F_E(u^k) + F_I(u^k)\big) \\ & - \frac{1}{2}\mathcal{Q}^m_{m-1}\big(F_E(u^k) + F_I(u^k)\big). \end{aligned} \tag{16}$$

2.3. Numerical results: Moderate-order predictors improve overall efficiency

Using the Burgers equation and Allen–Cahn equation, we compare the efficiency of SISDC methods using prediction and correction steps of differing orders. We compare SISDCK[1P1C], SISDCK[2P1C], SISDCK[3P1C], and SISDCK[2P2C], where $K=4$ or 5. To compute a fifth-order approximation, the SISDC5[2P2C] first computes a second-order provisional solution, then solves a second-order correction equation, followed by a first-order correction step.

Before reviewing the numerical results, we first estimate the computational costs of the SISDC methods under consideration. To that end, we assume that the solution of the implicit part of the system is much more expensive than the explicit part. For simplicity, we further assume that the implicit solves in all SISDC methods have similar computational costs. With these assumptions, the computational costs of the above SISDC methods can be compared in terms of the numbers of implicit solves. For the SISDCK[1P1C] method, $K-1$ solves are required (one for each of the $K-1$ substeps) for the provisional step and for each of the $K-1$ correction steps. Thus, a total of $K(K-1)$ implicit solves are required. For the SISDCK[2P1C] and SISDCK[3P1C] methods, there are $K-2$ and $K-3$ correction steps, respectively; thus, these methods require $(K-1)^2$ and $(K-1)(K-2)$ implicit solves, respectively. There are 1 and 2 correction steps in the SISDC4[2P2C] and SISDC5[2P2C] methods, respectively, which implies that these methods require 6 and 12 (or $(K-1)(K-2)$) implicit solves. The above analysis shows that, if the implicit solves require similar computational costs for all methods, then the SISDCK[3P1C] and SISDCK[2P2C] methods are likely the most efficient.

In the numerical tests, a spatial grid of $N=512$ subintervals was used when solving the Burgers equation, whereas a less refined spatial grid $N=64$ was used for the Allen–Cahn equation. For Burgers equation, time steps were set to $\Delta t = 8,4,2,1\times\Delta x$ for fourth-order SISDC methods and $\Delta t = 16,8,4,2\times\Delta x$ for fifth-order SISDC methods; for Allen–Cahn equation, $\Delta t = 1,\ \frac{1}{2},\ \frac{1}{4},\ \frac{1}{8}\times\Delta x$. Results are shown in Figs. 1 and 2. (The approximations obtained for the Allen–Cahn equation using fifth-order SISDC methods and $\Delta t = \frac{1}{8}\Delta x$ are dominated by spatial errors and thus those results are not shown.) These results suggest that for these problems, using moderate-order prediction and correction steps reduce computation costs. For the same time step, SISDCK[1P1C] methods, which require the most correction steps to attain a given order of accuracy, are likely to have the highest computational costs, whereas the SISDCK[3P1C] methods, which require the least (or among the least) correction steps, are likely to have the lowest (or one of the lowest) costs. These

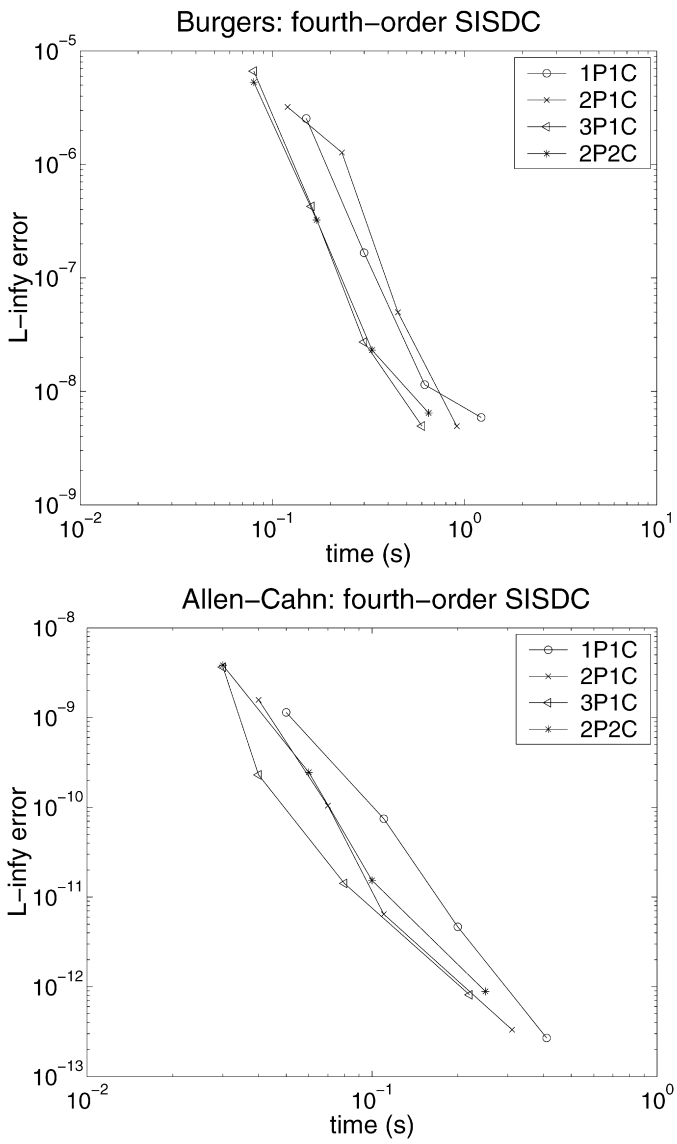

Fig. 1. Efficiency comparison of fourth-order SISDC methods using prediction and correction steps of differing orders. Results obtained for Burgers equation (top) and Allen–Cahn equation (bottom).

results are consistent with the analysis above. Nonetheless, it is noteworthy that the SISDCK[1P1C] methods considered generate the most accurate approximations (at the same order) for these problems. The relative accuracy and efficiency of these SISDC methods is likely to be problem dependent to some extent. Indeed, for the Burgers equation, SISDCK[3P1C] and SISDCK[2P2C] methods are the most efficient, whereas for the Allen–Cahn equation, SISDCK[3P1C] and SISDCK[2P1C] methods are the most efficient.

2.4. Ladder approaches

The approximations computed by the provisional step and by the initial correction steps have lower orders of accuracy than the final solution. Ladder approaches make use of this fact to reduce the computational cost of a SISDC method without compromising the overall order of the solution. This is achieved by allowing larger temporal or spatial errors in the initial SDC iterations. Indeed, ladder approaches are similar to multigrid methods, which recursively compute provisional solutions and corrections on coarser (space or time) grids.

An example of the ladder approaches—a multigrid method in time—was implemented in [16]. To obtain a Kth order solution, the quadrature $\mathcal{Q}$ in (7) must be approximated to Kth order. When Gauss–Lobbato or uniformly-spaced nodes are used, K nodes or $K-1$ substeps are required. Based on the observation that the kth correction equation computes an $\mathcal{O}(\Delta t^{k+2})$ approximation, the number of substeps used to compute the solution during the initial SDC iterations was re-

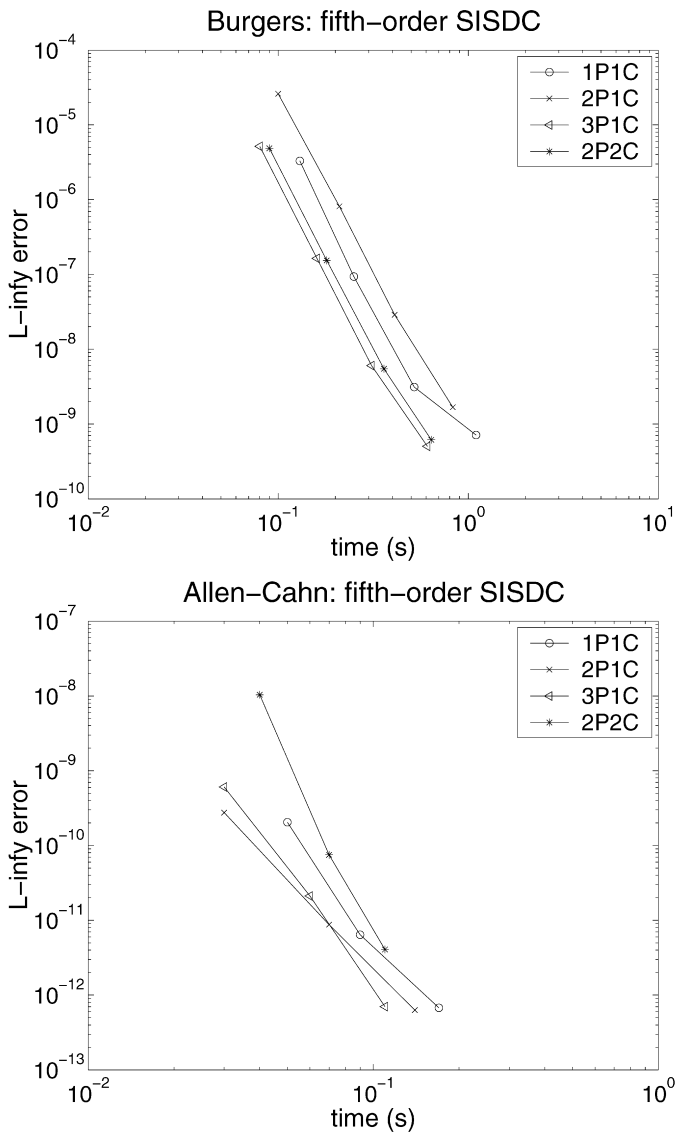

Fig. 2. Efficiency comparison of fifth-order SISDC methods using prediction and correction steps of differing orders. Results obtained for Burgers equation (top) and Allen–Cahn equation (bottom).

duced, i.e., fewer substeps were used when k is small [16]. This approach will be referred to as "Ladder: substep". When this approach was applied to a linear problem [16], the reduction in computational cost was offset by a reduction in accuracy, and no significant improvement in efficiency was observed. Nonetheless, below we further investigate the effectiveness of the "Ladder: substep" approach using nonlinear PDE problems (8) and (10).

Alternatively, one may use a less refined spatial grid or a lower-order spatial discretization method when k is small; these two approaches, which are examples of multigrid methods in space, will be referred to as "Ladder: space-res" and "Ladder: space-order", respectively. A justification that the "Ladder: space-order" approach preserves the overall order of a SISDC method and a lower bound for the orders of the spatial methods is given in Appendix A.

The lower the expected accuracy of the solution of an intermediate SDC iteration, the lower the accuracy—and lower the computational cost—of the temporal and spatial discretization needed for that iteration (i.e., fewer substeps or spatial grid points, or a lower-order spatial discretization method, are needed to generate an approximation of the required order). Thus, ladder approaches are likely to be more effective when used in conjunction with a SISDC method that uses a low-order prediction step. Another issue is that, although the ladder approaches preserve the overall order of the SISDC methods, the accuracy of the methods will likely be affected [16]. Thus, the question is whether the gain in speed is outweighed by the reduction in accuracy. Below we conduct numerical tests to assess the efficiency gain, if any, that may be obtained by combining the above ladder approaches with SISDC methods, and to determine whether the SISDC[1P1C] method using ladder approaches may be more efficient than the SISDC methods using high-order prediction steps.

2.5. Numerical results: "Ladder: space-res" greatly improves efficiency

We compare the efficiency of four implementations of the SISDC4[1P1C] methods by means of the Burgers equation: (i) base case without ladder approach, (ii) with "Ladder: substep", (iii) with "Ladder: space-res", and (iv) with "Ladder: space-order". A spatial grid of $N = 512$ subintervals was used, except in the initial SDC iterations with "Ladder: space-res" approach. In the computations of the prediction and correction steps, the "Ladder: substep" approach used 1, 2, 2, and 3 substeps; the "Ladder: space-res" approach used spatial grids of $N = 64$, 128, 256, and 512; and the "Ladder: space-order" approach used third-, fourth-, fifth-, and sixth-order centered differencing. The time steps were set to $\Delta t = 8, 4, 2, 1 \times \Delta x$, where in the "Ladder: space-res" approach the time steps were chosen using the smallest Δx.

Before reviewing the numerical results, we first estimate the computational saving of the ladder approaches under consideration. Given a spatial grid of N subintervals, it takes $\mathcal{O}(N)$ or $\sim cN$ floating-point operations to advance the solution by one substep. Assume that the number of operations does not vary substantially by the applications of ladder approaches, c can be assumed to be the same for all four SISDC4[1P1C] methods for each substep. Then for the base case, the total number of operations required for each time step is $12cN$ (4 SDC iterations, each with 3 substeps). The "Ladder: substep" approach requires $8cN$ operations. With the "Ladder: space-res" approach, each substep in the prediction step requires $cN/8$ operations, the first, second, and third SDC iteration requires $cN/4$, $cN/2$, and cN operations, respectively; thus the total number of operations required is $45/8cN$. With the "Ladder: space-order" approach, each substep in the prediction step requires $cN(3/6)$ operations, the first, second, and third SDC iteration requires $cN(4/6)$, $cN(5/6)$, and cN operations, respectively; thus the total number of operations required is $9cN$. Thus, one expects the "Ladder: space-res" approach to yield the largest reduction in computational cost.

Fig. 3, top, shows the L_∞ errors and computational times for the four SISDC4[1P1C] methods. For sufficiently large Δt, all four methods generate approximations that exhibit approximately fourth-order convergence in time. For a given time step, the base-case approximation, in which no ladder approach was used, is the most accurate. However, the computational cost associated with the base case is also highest. For this problem, the "Ladder: space-res" approach is the most effective, a result that is consistent with the analysis above.

In the next set of numerical tests, we study the accuracy and efficiency of SISDC methods that combine the "Ladder: space-res" approach with moderate-order predictors and correctors. In particular, we compare the efficiency of the SISDC4[1P1C], SISDC4[2P1C], SISDC4[3P1C], and SISDC4[2P2C] methods. For a method that requires K prediction and correction steps, a spatial grid of $N_0/2^{K-k}$ was used in the kth SDC iteration, where N_0 denotes the number of spatial intervals used to compute the final solution. The results are shown in Fig. 3, bottom. As previously mentioned, ladder approaches are likely to be more effective when applied to SISDC methods that use low-order prediction and correction steps. Indeed, compared to the base-case (no ladder approach) results in Fig. 1, top, the relative efficiency of SISDC4[1P1C] and SISDC4[2P1C] methods improved with the use of the "Ladder: space-res" approach, and for this problem, SISDC4[1P1C], SISDC4[2P1C], and SISDC4[3P1C] show comparable efficiency. The SISDC4[2P2C] method is substantially less efficient than other methods. This may be attributable to a reduction in the smoothness of the correction δ by the spatial errors that are different for each iteration. This results in larger errors when a second-order method (rather than a first-order method) is used in the correction step.

The above results are obtained for one-dimensional (spatial) problems. In two dimensions, reducing the spatial resolution by a half reduces the computational cost by three-fourth, compared to the cost reduction of a half in one dimension. Thus, the "Ladder: space-res" approach is likely to be more effective in higher spatial dimensions. To demonstrate that, the efficiency of the SISDC4[1P1C], SISDC4[2P1C], SISDC4[3P1C], and SISDC4[2P2C] methods, applied in conjunction with the "Ladder: space-res" approach, is compared in the solution of the following two-dimensional problem:

$$u_t = -uu_x - u_y + \epsilon(u_{xx} + u_{yy}), \tag{17}$$

$$u(x, y, 0) = \frac{1}{2} + \frac{1}{2}\sin(2\pi x)\sin(2\pi y), \tag{18}$$

for $x \in [0, 1]$, $y \in [0, 1]$, and $t \in [0, 0.25]$. The diffusion coefficient ϵ is set to 0.01. Bi-periodic boundary conditions are assumed.

Fig. 4 shows the L_∞ errors and computational times for the four methods. With the "Ladder: space-res" approach, the SISDC4[1P1C] method is only slightly more expensive than methods using moderate-order prediction/correction steps. The relative difference in computational costs among these methods is smaller in two dimensions, because, as mentioned previously, a reduction in spatial resolution results in a larger reduction in cost. Because the SISDC4[1P1C] method is most accurate for this problem, it is also the most efficient method among the four. Thus, these results suggest that in higher spatial dimensions, the benefit of the "Ladder: space-res" approach may outweigh that of moderate-order prediction and correction methods.

3. MISDC methods

For PDEs that involve multiple fast processes of differing time scales, MISDC methods provide a flexible and efficient time integration. In this section, we introduce high-order MISDC methods that use a second- or third-order method to compute

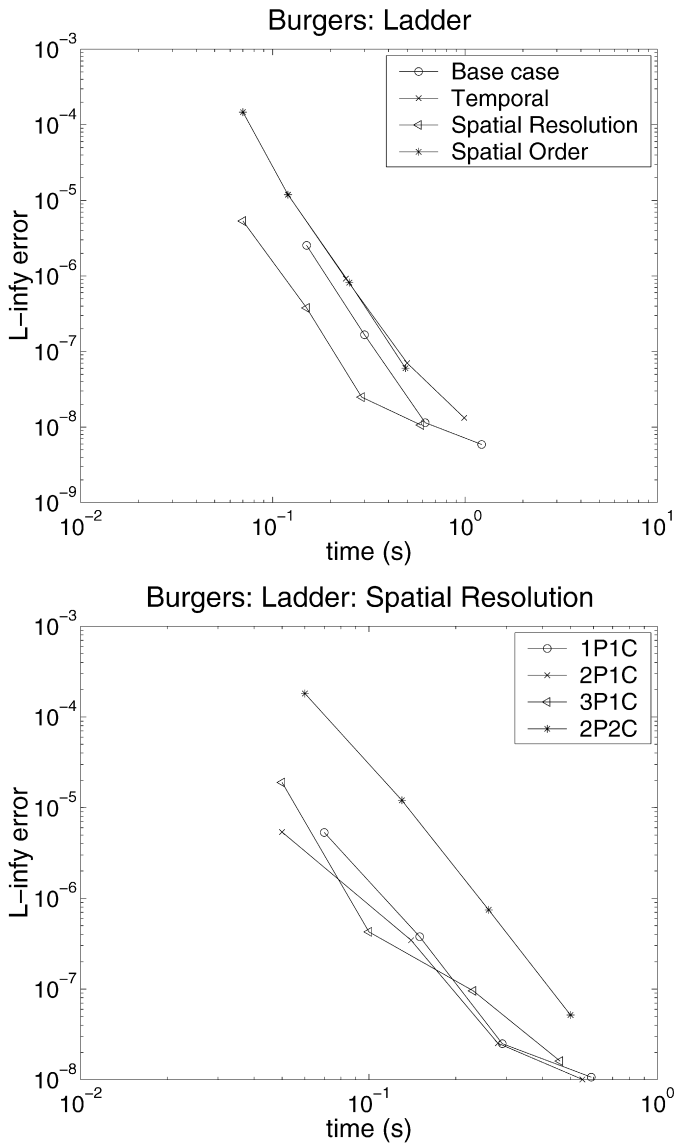

Fig. 3. Top: efficiency comparison of fourth-order SISDC methods using differing ladder approaches (base case, no ladder approach). Bottom: efficiency comparison of fourth-order SISDC methods using "Ladder: Spatial Resolution" approach, and prediction and correction steps of differing orders.

provisional solutions, and we compare these methods to the high-order MISDC methods presented in [4], which is based on forward and backward Euler method. Moderate-order correction steps, similar to those developed for SISDC methods in Section 2.2, can also be constructed, but because the effectiveness of such correctors is unclear, we will restrict our focus to the prediction step. The performance of moderate-order prediction steps will indicate whether the development of methods with moderate-order correction steps is a worthwhile pursuit.

The methods are applied to the advection–diffusion–reaction (A-D-R) equation given by

$$u_t = f_A u_x + \nu u_{xx} + f_R, \tag{19}$$

where $f_A(x,t,u(x,t))u_x(x,t)$ is the (possibly nonlinear) advection term, ν is the diffusivity, and $f_R(x,t,u(x,t))$ is the reaction term. In general, the diffusion term is given by $(\nu(x,t)u_x(x,t))_x$, but for simplicity, ν is assumed to be constant. Boundary conditions and initial conditions must be given to complete the specification of the problem.

When MISDC methods are used in conjunction with MOL to approximate the solution to (19), the equation is first discretized spatially, resulting in a system of ODEs,

$$u'(t) = F_A\big(t,u(t)\big) + F_D\big(t,u(t)\big) + F_R\big(t,u(t)\big), \tag{20}$$

$$u(a) = u_0, \tag{21}$$

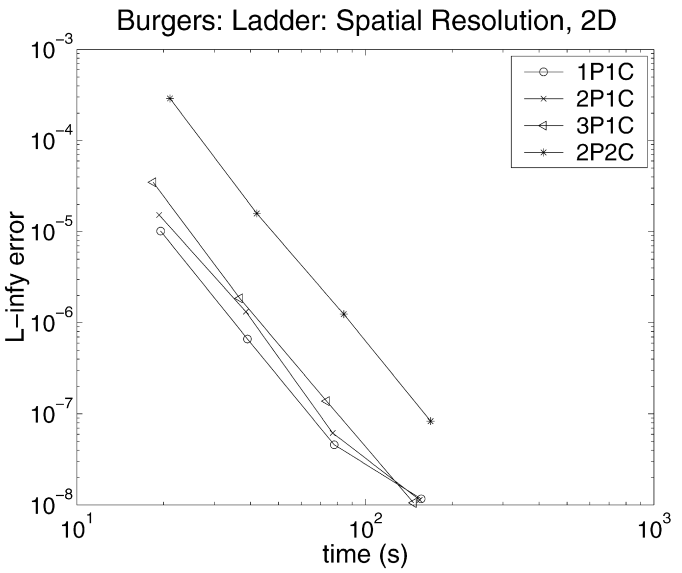

Fig. 4. Efficiency comparison of fourth-order SISDC methods using "Ladder: space-res" approach, and prediction and correction steps of differing orders, for the two-dimensional Burgers problem.

for $t \in [a, b]$. The terms F_A, F_D, and F_R are obtained from the spatial discretization of $f_A u_x$, νu_{xx}, and f_R, respectively, and hence do not depend on the spatial derivatives of u. Therefore, F_D is simply a linear operator, as is F_A in the linear case where f_A does not depend on u. Eqs. (20) and (21) are then integrated using MISDC methods, in which the non-stiff advection term F_A is treated explicitly and the stiff diffusion and reaction terms F_D and F_R are integrated implicitly. Moreover, like the operating-splitting approach, processes are decoupled and integrated sequentially, possibly using different time steps.

The MISDC methods developed in [4] are based on forward and backward Euler methods and on a first-order operator-splitting approach. Results in Section 2 suggest that one may construct a more efficient SISDC method by using a second- or third-order method to compute the provisional solution; numerical tests will be conducted to determine whether analogous results hold for MISDC methods.

3.1. Second-order prediction step

A multi-implicit, operator-splitting procedure is described here for obtaining a provisional solution of the spatially discretized A-D-R equation (19) with second-order temporal accuracy. The procedure is based on Strang splitting [17]. In the integration of the solution from t_n to t_{n+1}, the time interval $[t_n, t_{n+1}]$ is divided into N_A subintervals by choosing points t_m for $m = 0, 1, \ldots, N_A$. Assume for now that all three processes are integrated using the same time step $\Delta t_m \equiv t_{m+1} - t_m$. To advance the solution from t_m to t_{m+1}, one solves following equations sequentially

$$u_A\left(t_m + \frac{\Delta t_m}{2}\right) = u(t_m) + \int_{t_m}^{t_m + \frac{\Delta t_m}{2}} F_A\big(u(\tau)\big)\, d\tau, \tag{22}$$

$$u_D\left(t_m + \frac{\Delta t_m}{2}\right) = u_A\left(t_m + \frac{\Delta t_m}{2}\right) + \int_{t_m}^{t_m + \frac{\Delta t_m}{2}} F_D\big(u_D(\tau)\big)\, d\tau, \tag{23}$$

$$u_R(t_m + \Delta t_m) = u_D\left(t_m + \frac{\Delta t_m}{2}\right) + \int_{t_m}^{t_m + \Delta t_m} F_R\big(u_R(\tau)\big)\, d\tau, \tag{24}$$

$$u_D(t_m + \Delta t_m) = u_R(t_m + \Delta t_m) + \int_{t_m + \frac{\Delta t_m}{2}}^{t_{m+1}} F_D\big(u_D(\tau)\big)\, d\tau, \tag{25}$$

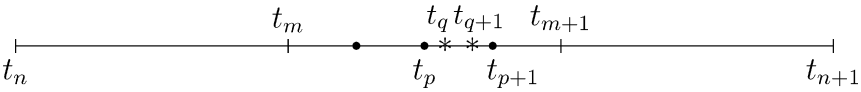

Fig. 5. Illustration of three levels of time-step subdivision: $[t_n, t_{n+1}]$ into $[t_m, t_{m+1}]$, $[t_m, t_{m+1}]$ into $[t_p, t_{p+1}]$, and $[t_p, t_{p+1}]$ into $[t_q, t_{q+1}]$. In this example, $N_A = 3$, $N_D = 4$, $N_R = 3$, $m = 1$, $p = 2$, and $q = 1$.

$$u(t_m + \Delta t_m) = u_D(t_m + \Delta t_m) + \int_{t_m + \frac{\Delta t_m}{2}}^{t_{m+1}} F_A\big(u_D(\tau)\big)\, d\tau. \tag{26}$$

Note that Eq. (26) is an advection step, but because it yields the full solution, the result is denoted by u rather than u_A.

To attain overall $\mathcal{O}(\Delta t^2)$ accuracy, the integrals on the right must be approximated to $\mathcal{O}(\Delta t^2)$. To reduce overall computational cost with little loss in accuracy, one may integrate slower processes using larger time steps. To use smaller time steps for the diffusion process and yet smaller ones for the reaction process, the substep $[t_m, t_{m+1}]$ is subdivided into N_D subintervals by choosing points $t_{m,p}$ for $p = 0, 1, \ldots, N_D$ such that $t_{m,p} = t_m + p\Delta t_p$ where $\Delta t_p = \Delta t_m / N_D$. Then $[t_{m,p}, t_{m,p+1}]$ is further subdivided into N_R subintervals by choosing points $t_{m,p,q}$ for $q = 0, 1, \ldots, N_R$ such that $t_{m,p,q} = t_{m,p} + q\Delta t_q$ where $\Delta t_q = \Delta t_p / N_R$. Where there is no ambiguity, the subscript m is omitted in $t_{m,p}$, and m and p omitted in $t_{m,p,q}$ for notational simplicity; i.e., $t_{m,p}$ and $t_{m,p,q}$ are written as t_p and t_q, respectively. Fig. 5 shows an example of time-step subdivision.

To improve accuracy, we couple the intermediate solutions of (22) and (23), and of (25) and (26). The diffusion and reaction processes are integrated using time steps Δt_p and Δt_q, respectively, (23)–(25) become

$$u_D(t_p + \Delta t_p) = u_D(t_p) + \int_{t_p}^{t_p + \Delta t_p} F_A\big(u(\tau)\big) + F_D\big(u_D(\tau)\big)\, d\tau, \tag{27}$$

$$u_R(t_q + \Delta t_q) = u_R(t_q) + \int_{t_q}^{t_q + \Delta t_q} F_R\big(u_R(\tau)\big)\, d\tau, \tag{28}$$

$$u_D(t_p + 2\Delta t_p) = u_R(t_p + \Delta t_p) + \int_{t_p + \Delta t_p}^{t_p + 2\Delta t_p} F_A\big(u(\tau)\big) + F_D\big(u_D(\tau)\big)\, d\tau. \tag{29}$$

In (28), $u_R(t_q) = u_D(t_p + \Delta t_p)$ for $q = 0$. Eq. (28) is solved for two diffusion time steps, i.e., $q = 0, 1, \ldots, 2N_R - 1$. Eqs. (27)–(29) are repeatedly solved in sequence for one advection time step, i.e., for $p = 0, 2, 4, \ldots, N_D - 2$.

To compute a second-order provisional solution, the advection term is extrapolated to second order, and the diffusion and reaction terms are averaged in time. Eqs. (27), (28), and (29) are approximated to second order by

$$u_{D_{p+1}} = u_{D_p} + \Delta t_p \left(\left(\frac{N_A + p - \frac{1}{2}}{N_A} \right) F_A(u_m) - \left(\frac{p - \frac{1}{2}}{N_A} \right) F_A(u_{m-1}) + \frac{1}{2} F_D(u_{D_p}) + \frac{1}{2} F_D(u_{D\,p+1}) \right), \tag{30}$$

$$u_{R_{q+1}} = u_{R_q} + \frac{\Delta t_q}{2} \big(F_R(u_{R_q}) + F_R(u_{R_{q+1}}) \big). \tag{31}$$

The procedures for computing a second-order provisional solution are summarized below:

For $m = 0, \ldots, N_A - 1$
  For $p = 0, 2, 4, \ldots, N_D - 2$

$$u^0_{D_{p+1}} = u^0_{D_p} + \Delta t_p \left(\left(\frac{N_A + p - \frac{1}{2}}{N_A} \right) F_A\big(u^0_m\big) - \left(\frac{p - \frac{1}{2}}{N_A} \right) F_A\big(u^0_{m-1}\big) + \frac{1}{2} F_D\big(u^0_{D_p}\big) + \frac{1}{2} F_D\big(u^0_{D_{p+1}}\big) \right), \tag{32}$$

$$u^0_{R_0} = u^0_{D_{p+1}}$$

For $q = 0, \ldots, 2N_R - 1$

$$u^0_{R_{q+1}} = u^0_{R_q} + \frac{1}{2} \Delta t_q \big(F_R\big(u^0_{R_q}\big) + F_R\big(u^0_{R_{q+1}}\big) \big) \tag{33}$$

End

$$u^0_{p+2} = u^0_{R\,2N_R} + \Delta t_p \left(\left(\frac{N_A + p + \frac{1}{2}}{N_A} \right) F_A\big(u^0_m\big) - \left(\frac{p + \frac{1}{2}}{N_A} \right) F_A\big(u^0_{m-1}\big) + \frac{1}{2} F_D\big(u^0_{R_{2N_R}}\big) + \frac{1}{2} F_D\big(u^0_{D_{p+2}}\big) \right) \tag{34}$$

Compute $F_D(u^0_{p+2})$, $F_R(u^0_{p+2})$
End
Compute $F_A(u^0_{m+1})$.
End

The above algorithm generates second-order approximations at every other diffusion node, i.e., at t_p where $p = 0, 2, 4, \ldots, N_D - 2$. Second-order approximations are obtained at other locations by means of second-order Lagrange interpolations.

3.2. Third-order prediction step

We now present a MISDC method that uses a third-order prediction step. Because a limitation of Strang's splitting is that its generalization to higher than second order in time is not straightforward, our approach is not based on Strang splitting. The strategy for approximating the solution in the subinterval $[t_p, t_{p+1}]$

$$u(t_p + \Delta t_p) = u(t_p) + \int_{t_p}^{t_p+\Delta t_p} \big(F_A\big(\tau, u(\tau)\big) + F_D\big(\tau, u(\tau)\big) + F_R\big(\tau, u(\tau)\big)\big)\, d\tau \tag{35}$$

is as follows: first an approximation $u_{R_{p+1}}$ is computed by treating both advection and diffusion explicitly, and by integrating reaction implicitly, using the diffusion time step Δt_p, i.e.,

$$u_{R_{p+1}} = \sum_{j=0}^{s} b_j u_{p-j} + \mathcal{E}_p^{p+1}\big(F_A(\tau, u) + F_D(\tau, u)\big) + \mathcal{I}_p^{p+1}\big(F_R(\tau, u)\big). \tag{36}$$

In this first step, diffusion is treated explicitly to avoid coupling it with reaction, which would require the solution of a system of coupled nonlinear equations. The error in $u_{R_{p+1}}$ is likely to be large because of the stiffness in the diffusion term. Thus, a more accurate approximation $u_{D_{p+1}}$ is then computed by treating advection explicitly, and by integrating reaction and diffusion implicitly. Again, we want to avoid the coupling of the global diffusion and nonlinear reaction terms. To that end, F_R is evaluated using known u_R values at t_{p+1} as follows:

$$u_{D_{p+1}} = \sum_{j=0}^{s} b_j u_{p-j} + \mathcal{E}_p^{p+1}\big(F_A(\tau, u)\big) + \mathcal{I}_p^{p+1}\big(F_D(\tau, u_D) + F_R(\tau, u_R)\big). \tag{37}$$

Although u_D is likely more accurate than u_R because of the implicit treatment of F_D in (37), the error in u_D may still be large because the stiffness in F_R may require the use of a smaller time step. Thus, u_D is used only indirectly in the computation of the provisional solution. Specifically, $F_D(t_{p+1}, u_{D_{p+1}})$ is computed and assumed known in $[t_p, t_{p+1}]$. Then the provisional solution u is computed by treating advection explicitly, diffusion implicitly (but known), and reaction implicitly, using the small reaction time step Δt_q, i.e., one solves

$$u_{q+1} = \sum_{j=0}^{s} b_j u_{q-j} + \mathcal{E}_q^{q+1}\big(F_A(\tau, u)\big) + \mathcal{I}_q^{q+1}\big(F_D(\tau, u_D) + F_R(\tau, u)\big). \tag{38}$$

The integrals in (36)–(38) are approximated using a third-order IMEX BDF method. Standard IMEX methods (e.g., [2]) are formulated for uniform time steps. Although t_m, t_p, and t_q are uniformly spaced, Δt_m, Δt_p, and Δt_q are different. Thus, in the IMEX approximation of (36) and (37), F_A may not be known at every diffusion node, and, similarly in (38), F_A and F_D may not be known at every reaction node. To avoid the need of computing non-standard IMEX coefficients for variable time steps (indeed, a different set of IMEX coefficients would be required for each diffusion step in (36) and (37), and for each reaction step in (38)), third-order Lagrange interpolation and extrapolation are used to approximate F_A at diffusion and reaction nodes and F_D values at reaction nodes.

The procedures for computing a third-order provisional solution are summarized below:

For $m = 0, \ldots, N_A - 1$
Approximate F_A values at diffusion nodes t_p's in $[t_m, t_{m+1}]$.
For $p = 0, \ldots, N_D - 1$

$$\begin{aligned}\frac{11}{6}u^0_{R_{p+1}} &= 3u^0_p - \frac{3}{2}u^0_{p-1} + \frac{1}{3}u^0_{p-2} + \Delta t_p\big(3F_A\big(u^0_p\big) - 3F_A\big(u^0_{p-1}\big) + F_A\big(u^0_{p-2}\big)\\ &\quad + 3F_D\big(u^0_p\big) - 3F_D\big(u^0_{p-1}\big) + F_D\big(u^0_{p-2}\big) + F_R\big(u_{R}{}^0_{p+1}\big)\big)\\ \frac{11}{6}u^0_{D_{p+1}} &= 3u^0_p - \frac{3}{2}u^0_{p-1} + \frac{1}{3}u^0_{p-2} + \Delta t_p\big(3F_A\big(u^0_p\big) - 3F_A\big(u^0_{p-1}\big) + F_A\big(u^0_{p-2}\big)\\ &\quad + F_D\big(u_{D}{}^0_{p+1}\big) + F_R\big(u_{R}{}^0_{p+1}\big)\big)\end{aligned}$$

Approximate F_A and F_D values at reaction nodes t_q's in $[t_p, t_{p+1}]$.
$u_0^0 = u_{D}{}_{p+1}^{0}$
For $q = 0, \ldots, N_R - 1$

$$\frac{11}{6}u_{q+1}^0 = 3u_q^0 - \tfrac{3}{2}u_{q-1}^0 + \tfrac{1}{3}u_{q-2}^0 + \Delta t_q\big(3F_A(u_q^0) - 3F_A(u_{q-1}^0) + F_A(u_{q-2}^0) + F_D(u_{D}{}_{q+1}^{0}) + F_R(u_{q+1}^0)\big)$$

Compute $F_R(u_{q+1}^0)$,
End
Compute $F_D(u_{p+2}^0)$,
End
Compute $F_A(u_{m+1}^0)$.
End

3.3. Numerical results

The efficiency of MISDC methods is assessed using the Burgers' equation with reaction

$$u_t + uu_x = \epsilon u_{xx} + c_R u(u-1)^2, \tag{39}$$

for $x \in [-2, 2]$ and $t \in [0, 0.5]$, where $\epsilon > 0$ and $c_R > 0$ are the diffusion and reaction coefficients, respectively. The initial conditions are given by

$$u(x, 0) = \frac{1}{2} - \frac{1}{2}\tanh\left(\frac{x}{\delta}\right). \tag{40}$$

The system (39) and (40) are completed with Dirichlet boundary conditions $u(-2, t) = 1$ and $u(2, t) = 0$.

If the diffusion coefficient and reaction coefficient are set to $\epsilon = \delta(1-\gamma)/2$ and $c_R = 2(2\gamma - 1)/\delta$, then given initial conditions (40) on the spatial domain $x \in [-\infty, \infty]$ and with boundary conditions $u(-\infty, t) = 1$ and $u(\infty, t) = 0$, the analytic solution for (39) is given by

$$u(x, t) = \frac{1}{2} - \frac{1}{2}\tanh\left(\frac{x - \gamma t}{\delta}\right). \tag{41}$$

Technically speaking, $u(x, t) = 1$ and $u(x, t) = 0$ only as $x \to -\infty$ and $x \to \infty$ for finite t. Nevertheless, for the time interval considered in this example, u is practically indistinguishable from 1 and 0 at the left and right boundary points ($x = -2$ and 2), respectively, within double-precision machine accuracy. Thus, (41) is used as the reference solution even though the simulations are done using a finite spatial domain. The parameters γ and δ are chosen to be 0.75 and 0.05, respectively, which gives $\epsilon = 0.0125$ and $c_R = 20$.

Fig. 6 shows L_∞ errors versus computational times for approximations obtained by means of fourth-order MISDC methods using first-, second-, and third-order prediction steps and using first-order correction steps. A spatial grid of $N = 512$ was used. Time steps for each method were chosen to fall within the region where the method is stable and where temporal truncation errors dominate spatial errors. For the MISDC4[1P1C] method, $\Delta t = 8, 4, 2, 1 \times \Delta x$; for MISDC4[2P1C], $\Delta t = 2, 1, 0.5, 0.25 \times \Delta x$; for MISDC4[3P1C], $\Delta t = 4, 2, 1, 0.5 \times \Delta x$. The errors are computed using the reference solution (41).

In all cases, the computed solutions exhibit approximately $\mathcal{O}(\Delta t^4)$ convergence. These results also indicate that the MISDC4[3P1C] method is the most efficient among the three, whereas the MISDC4[2P1C] method is the least efficient. Indeed, owing to the ordering of the diffusion and reaction steps in the second-order prediction step, approximations computed using the MISDC4[2P1C] method has the largest splitting errors. By assumption, and by our choice of parameter values, the reaction term is stiffer than the diffusion term. It follows that $|\partial F_R/\partial u| \gg |\partial F_D/\partial u|$. Thus, to minimize splitting errors, F_R, rather than F_D, should be evaluated at u_{q+1}^{k+1}, as done in the first- and third-order prediction steps (but not the second-order step).

Splitting errors in the MISDC[3P1C] methods increase with the stiffness of the diffusion and reaction terms; thus, the stiffer F_D and F_R are, the less accurate the approximations u_R and u_D, and consequently u are. In another set of experiments, the efficiency of the methods is compared using a stiff problem, in which the diffusion coefficient ϵ and reaction coefficients c_R were set to 0.01 and 2000, respectively. The initial conditions were set to

$$u(x, 0) = \frac{1}{2} - \frac{1}{2}\tanh\left(\frac{x}{20}\right). \tag{42}$$

With this set of ϵ, c_R, and initial conditions, however, an analytic solution is not known.

Eqs. (39) and (42) were integrated using the MISDC4[1P1C], MISDC4[2P1C], and MISDC4[3P1C] methods using a spatial grid of $N = 512$ and time steps of $\Delta t = 0.32$, 0.16, 0.08, and $0.04 \times \Delta x$. The approximation computed using the MISDC4[1P1C] method with $\Delta t = 0.01 \times \Delta x$ was used as the reference solution.

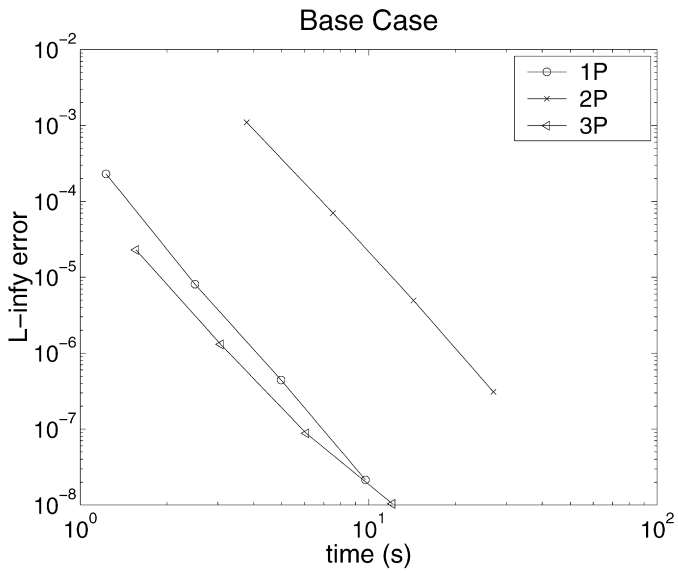

Fig. 6. Efficiency comparison of fourth-order MISDC methods using first-, second-, and third-order prediction steps.

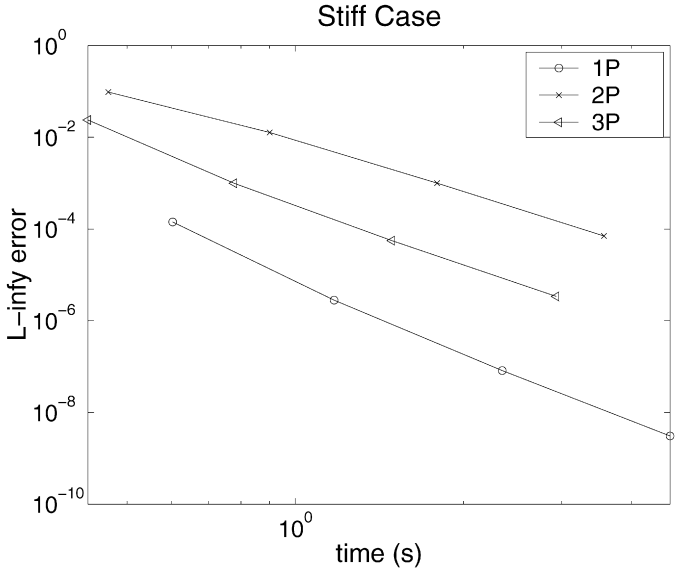

Fig. 7. Efficiency comparison of fourth-order MISDC methods using first-, second-, and third-order prediction steps for stiff parameters.

Fig. 7 shows L_∞ errors versus computational times. For the same time step, the MISDC4[3P1C] method is the least computationally expensive, whereas the MISDC4[1P1C] method is the most expensive. However, owing to larger splitting errors, the solution computed by MISDC4[3P1C] method is less accurate (but still more accurate than the MISDC4[2P1C] method), and thus, for this stiff problem, the MISDC4[1P1C] method is the most efficient. These results suggest that while moderate-order predictors reduce computational costs, the effects of splitting errors must be taken into account when evaluating the efficiency of the overall methods.

4. Discussion

The goal of this study is to assess the efficiency of a number of implementations of SISDC and MISDC methods for the temporal integration of PDEs with multiple time scales. In these implementations, second- and third-order integration methods are used in the prediction steps of SISDC and MISDC methods, and also in the correction steps of SISDC methods. Strang splitting is used to compute a temporally second-order provisional solution of a MISDC method, and a novel operator-splitting approach is developed to compute a temporally third-order provisional solution. Using time-dependent PDE test

problems, we assess the accuracy and efficiency of the resulting SISDC and MISDC methods and compare their performance to the traditional implementation [4,16], which is based on forward and backward Euler methods.

Because of the computational costs associated with the deferred correction steps, a SISDC method is likely computationally more expensive than a BDF method of the same order. However, at sufficiently high orders of accuracy (e.g., higher than 6th order), BDF methods become unstable, whereas stable SISDC methods can be constructed at arbitrarily high orders. Thus, as shown in our previous study [15], a stable and high-order method can be constructed using a moderate-order BDF method in the prediction step of a SISDC method, and then the overall order of the resulting method increased via deferred correction steps. In addition, numerical results in this study suggest that, taken in isolation, using a moderate-order BDF method in the prediction step of a SISDC method can improve efficiency, compared to using an Euler method in the prediction step, by reducing the number of correction steps needed to attain a given overall order of accuracy. In terms of efficiency, the benefit that one may gain by using BDF2 in the correction steps is not clear. These results are consistent with the results in [12,15] obtained using ODE problems.

For time-dependent PDE problems, the benefits of moderate-order predictors and correctors should be re-evaluated in the context of the cost-saving ladder approaches. With ladder approaches, one typically allows larger solution errors in the initial deferred correction iterations, either by using a less refined temporal or spatial grid, or a lower-order spatial discretization method. Among the ladder approaches considered, our numerical results indicate that the "Ladder: space-res" approach, in which less refined spatial resolutions are used in the initial iterations, gives rise to the most efficient SISDC methods. Indeed, the reduction in computational costs generated by the "Ladder: space-res" approach exceeds that of moderate-order predictors, particularly in sufficiently high spatial dimensions. The "Ladder: space-res" approach, which is an example of a multigrid approach, can be effectively implemented in a massively-parallel environment using parallel multigrid techniques [5,18]. When applied to a two-dimensional problem, the SISDC4[1P1C] method, together with the "Ladder: space-res" approach, is the most efficient among the four fourth-order SISDC methods considered. Nonetheless, for applications with large spatial variations, an insufficiently refined spatial resolution may generate solutions with large errors. For such applications, other Ladder approaches or methods using higher-order steps may be more effective.

MISDC methods allow implicit treatment of multiple fast-scale processes using differing time steps. Through the deferred correction procedure, during which both integration and splitting errors are reduced, MISDC methods can compute approximations of arbitrarily high orders of accuracy while using operator-splitting. Thus, unlike SISDC methods, the benefits of which become clear over IMEX BDF methods only at sufficiently high orders of accuracy, MISDC methods are appealing in problems which involves multiple fast-scale processes, at moderate and high orders, owing to its flexibility and accuracy.

For MISDC methods, a third-order predictor gives rise to the most efficient method, compared to MISDC methods with second- or first-order predictor, when applied to the Burgers equation with mildly-stiff parameters. However, the accuracy of the MISDC[3P1C] method, and, consequently, its efficiency are reduced when stiff parameters are used. This is attributable to the larger splitting errors present in the intermediate solution. Such splitting errors may be reduced by iterating on (38): u_D is set to u, then (38) is solved again, and the procedure is repeated if needed. This reduction in splitting errors comes at the expense of increased computational cost, thus it is not clear that the above procedure will lead to an improvement in efficiency. Thus, for problems that are sufficiently stiff, the MISDC[1P1C] methods are likely more efficient. The advantage of MISDC[1P1C] methods can likely be augmented by incorporating the "Ladder: space-res" approach.

We conclude with a comparison among SISDC, MISDC, and IMEX BDF methods. As previously noted, at moderate orders of accuracy, a IMEX BDF method is expected to be computationally more efficient than a SISDC method of the same order. While BDF methods of sufficiently high orders of accuracy suffer from numerical instability, SISDC methods can be constructed at arbitrarily high orders. Comparing SISDC and MISDC methods, one can consider SISDC methods a simplification of MISDC methods that involves only two processes, that does not include operator-splitting, and that uses the same time-step for both processes. That is, SISDC methods do not allow splitting of multiple stiff processes or the use of different time-steps for different processes. Nonetheless, their simplicity renders them easier to analyze and to understand.

Acknowledgements

This work was supported in part by the National Science Foundation, grant DMS-0715021.

Appendix A. "Ladder: Space-Order" approach

The integral form of the solution to (2)–(3) is given by

$$u(t) = u_0 + \int_a^t \big(F_E\big(\tau, u(\tau)\big) + F_I\big(\tau, u(\tau)\big)\big)\, d\tau. \tag{A.1}$$

Let $\tilde{u}(t)$ be an approximation to $u(t)$. Suppose in the computation of $\tilde{u}(t)$ a less accurate spatial discretization method is used such that

$$u'(t) = \tilde{F}_E\big(t, u(t)\big) + \tilde{F}_I\big(t, u(t)\big) + \mathcal{O}\big(\Delta x^p\big), \tag{A.2}$$

where p is smaller than the overall order of the SISDC method. The integral form of the solution to (A.2) and (3) is given by

$$\tilde{u}(t) = u_0 + \int_a^t \left(\tilde{F}_E\left(\tau, u(\tau)\right) + \tilde{F}_I\left(\tau, u(\tau)\right) + \mathcal{O}\left(\Delta x^p\right)\right) d\tau. \tag{A.3}$$

To derive an expression for the correction $\delta(t) \equiv u(t) - \tilde{u}(t)$, define the residual function by

$$E\left(t, \tilde{u}(t)\right) = u_0 + \int_a^t \left(\tilde{F}_E\left(\tau, \tilde{u}(\tau)\right) + \tilde{F}_I\left(\tau, \tilde{u}(\tau)\right) + \mathcal{O}\left(\Delta x^p\right)\right) d\tau - \tilde{u}(t). \tag{A.4}$$

The definition of $\delta(t)$ and the integral equation (A.1) can be combined to give

$$\tilde{u}(t) + \delta(t) = u_0 + \int_a^t \left(F_E\left(\tau, \tilde{u}(\tau) + \delta(\tau)\right) + F_I\left(\tau, \tilde{u}(\tau) + \delta(\tau)\right)\right) d\tau. \tag{A.5}$$

From (A.4) and (A.5), one obtains the correction equation

$$\begin{aligned}\delta(t) = \int_a^t &\left(F_E\left(\tau, \tilde{u}(\tau) + \delta(\tau)\right) - \tilde{F}_E\left(\tau, \tilde{u}(\tau)\right) + F_I\left(\tau, \tilde{u}(\tau) + \delta(\tau)\right)\right) \\ &- \tilde{F}_I\left(\tau, \tilde{u}(\tau) - \mathcal{O}\left(\Delta x^p\right)\right) d\tau + E\left(t, \tilde{u}(t)\right).\end{aligned} \tag{A.6}$$

Given a sth-order approximate solution $\tilde{u}$ (i.e., $\|u - \tilde{u}\| = \mathcal{O}(\Delta t^{s+1})$) on the time interval $[t_n, t_{n+1}]$, a $(s+1)$th-order approximation can be computed by estimating the correction $\delta(t)$ in (A.6) to $(s+1)$th order. If F_E and F_I are Lipschitz continuous in u, then (A.6) implies that $\|\delta(t) - E(t, \tilde{u})\| = \mathcal{O}(\Delta t^{s+1})$.

Therefore, provided that $p \geqslant s+1$ and that Δx scales with Δt, a $(s+1)$th order approximation for $\delta(t)$ can be computed from a $(s+1)$th order approximation for $E(t, \tilde{u})$ and a simple first-order rectangle rule approximation to the integral on the right side of (A.6) ignoring the $\mathcal{O}(\Delta x^p)$ term.

References

[1] G. Akrivis, M. Crouzeix, C. Makridakis, Implicit–explicit multistep methods for quasilinear parabolic equations, Numer. Math. 82 (1999) 521–541.
[2] 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.
[3] 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.
[4] 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.
[5] A. Brandt, Multigrid solvers on parallel computers, in: Elliptic Problem Solvers, Academic Press, 1981, pp. 39–83.
[6] A. Dutt, L. Greengard, V. Rokhlin, Spectral deferred correction methods for ordinary differential equations, BIT 40 (2000) 241–266.
[7] 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.
[8] J.-F. Huang, J. Jia, M.L. Minion, Accelerating the convergence of spectral deferred correction methods, J. Comput. Phys. 214 (2) (2006) 633–656.
[9] K.J. in't Hout, On the contractivity of implicit–explicit linear multistep methods, Appl. Numer. Math. 42 (2002) 201–212.
[10] C.A. Kennedy, M.H. Carpenter, Additive Runge–Kutta schemes for convection–diffusion–reaction equations, Appl. Numer. Math. 44 (1–2) (2003) 139–181.
[11] W. Kress, Error estimates for deferred correction methods in time, Appl. Numer. Math. 57 (3) (2007) 335–353.
[12] A.T. Layton, On the choice of correctors for semi-implicit Picard deferred correction methods, Appl. Numer. Math. 58 (6) (2008) 845–858.
[13] A.T. Layton, M.L. Minion, Conservative multi-implicit spectral deferred methods for reacting gas dynamics, J. Comput. Phys. 194 (2) (2004) 697–715.
[14] 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.
[15] A.T. Layton, M.L. Minion, Implications of the choice of predictors for semi-implicit Picard integral deferred correction methods, Comm. Appl. Math. Comput. Sci. 2 (1) (2006) 1–34.
[16] M.L. Minion, Semi-implicit spectral deferred correction methods for ordinary differential equations, Comm. Math. Sci. 1 (3) (2003) 471–500.
[17] G. Strang, On the construction and comparison of difference schemes, SIAM J. Numer. Anal. 8 (3) (1968) 506–517.
[18] D.E. Womble, B.C. Young, Multigrid on massively parallel computers, in: Distributed Memory Computing Conference, vol. 5, 1990, pp. 559–563.