Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 189 (2003) 651–675
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

High-order multi-implicit spectral deferred correction methods for problems of reactive flow

Anne Bourlioux [a], Anita T. Layton [b,*], Michael L. Minion [b]

[a] *Département de Mathématiques et Statistique, Université de Montréal, CP 6128 succ. Centre-Ville, Montréal, Qué., Canada H3C 3J7*
[b] *Department of Mathematics, University of North Carolina, Chapel Hill, NC 27599, USA*

Received 14 May 2002; received in revised form 23 April 2003; accepted 24 April 2003

Abstract

Models for reacting flow are typically based on advection–diffusion–reaction (A–D–R) partial differential equations. Many practical cases correspond to situations where the relevant time scales associated with each of the three sub-processes can be widely different, leading to disparate time-step requirements for robust and accurate time-integration. In particular, interesting regimes in combustion correspond to systems in which diffusion and reaction are much faster processes than advection. The numerical strategy introduced in this paper is a general procedure to account for this time-scale disparity. The proposed methods are high-order multi-implicit generalizations of spectral deferred correction methods (MISDC methods), constructed for the temporal integration of A–D–R equations. Spectral deferred correction methods compute a high-order approximation to the solution of a differential equation by using a simple, low-order numerical method to solve a series of correction equations, each of which increases the order of accuracy of the approximation. The key feature of MISDC methods is their flexibility in handling several sub-processes implicitly but independently, while avoiding the splitting errors present in traditional operator-splitting methods and also allowing for different time steps for each process. The stability, accuracy, and efficiency of MISDC methods are first analyzed using a linear model problem and the results are compared to semi-implicit spectral deferred correction methods. Furthermore, numerical tests on simplified reacting flows demonstrate the expected convergence rates for MISDC methods of orders three, four, and five. The gain in efficiency by independently controlling the sub-process time steps is illustrated for nonlinear problems, where reaction and diffusion are much stiffer than advection. Although the paper focuses on this specific time-scales ordering, the generalization to any ordering combination is straightforward.

AMS: 65B05; 65M20; 80A32

Keywords: Advection–diffusion–reaction equation; Operator splitting; Reactive flows; Semi-implicit methods; Spectral deferred correction methods

* Corresponding author. Tel.: +1-919-843-2550; fax: +1-919-962-9345.
E-mail addresses: bourliou@dms.umontreal.ca (A. Bourlioux), layton@amath.unc.edu (A.T. Layton), minion@amath.unc.edu (M.L. Minion).

1. Introduction

Many physical systems with dynamics that involve two or more processes with widely differing characteristic time scales are of interest to researchers in the physical and biological sciences. Well-known examples include combustion [4,20], the transport of air pollutants [15,24,26], and the movement of contaminants or microorganisms in ground water systems [18,27]. In each of these examples, the mathematical models used to describe the dynamics consist of systems of partial differential equations (PDEs), which specify the advection, diffusion, and reaction of chemical species within a moving medium. When the time scales associated with these physical processes vary significantly, standard numerical methods are often inefficient. Hence, the construction of accurate, stable, and efficient numerical methods for the solution of advection–diffusion–reaction (A–D–R) equations is a topic of significant current interest.

To illustrate the phenomenon of multiple time scales, consider the test problem studied in Section 5 of an idealized one-dimensional flame. The model equations consist of two coupled A–D–R equations for the scalars u and v that represent the fuel and oxidizer mass fractions, respectively:

$$\begin{aligned} \frac{\partial u}{\partial t} + w\frac{\partial u}{\partial x} &= \nu\frac{\partial^2 u}{\partial x^2} + g(u,v), \\ \frac{\partial v}{\partial t} + w\frac{\partial v}{\partial x} &= \nu\frac{\partial^2 v}{\partial x^2} + g(u,v). \end{aligned} \tag{1}$$

Here, the advection velocity is given by $w(x,t)$, ν is the diffusivity of both scalars (assumed to be equal), and $g(u,v)$ is the function that represents the reaction of the fuel and oxidizer. One physically relevant choice of parameters and initial conditions results from the study of unsteady non-premixed flames, in which the initial data correspond roughly to pure fuel in one portion of the domain and pure oxidizer in the other. A simple reaction model given by $g(u,v) = -Duv$, where D is the reaction rate constant, is used. In the case that the diffusivity ν is small, the reaction rate D is large, and the advection velocity w is a time-modulated function, the dynamics of the resulting system are concentrated in a thin, non-steady reaction zone or flamelet.

In the flamelet example, the largest time scale corresponds to the time dependence of the advection term. Despite a small viscosity, if the thin reaction zone (in which sharp spatial variations occur) is fully resolved, the diffusive time scale in this zone will be much smaller than the advective time scale. Furthermore, since the reaction parameter D is chosen to be large, the reaction time scale is small as well. A detailed time-scale analysis is presented in Section 5 for the specific parameters considered there. The pertinent point is that in order to accurately predict the long-time dynamics of this system through numerical simulation, it is necessary to resolve all three of the physical processes on their respective time scales.

A popular approach for constructing higher-order numerical methods for PDEs is the method of lines approach (MOL), in which the equations are first discretized spatially, resulting in a large coupled system of ordinary differential equations (ODEs). For A–D–R equations, the disparity in time scales renders the ODE system stiff. Therefore for stability considerations, implicit methods may be used to avoid a prohibitively small time step. When the characteristic time scale of diffusion and reaction also differ widely, accuracy considerations may require that a smaller time step be used for the fastest process as well. This paper presents a strategy for constructing arbitrarily high-order accurate methods for ODEs for use in the MOL context an implicit yet uncoupled treatment of multiple stiff terms, as motivated by stability, and allow the use of different time steps for each of those terms, as motivated by accuracy issues.

Many well-established numerical methods for stiff ODEs exist and have been extensively studied (see, e.g. [2,12]). When applied to a MOL approximation of A–D–R equations, popular methods such as implicit Runge–Kutta or Backward Difference Formula methods are *fully implicit* in nature, i.e., every term in the

A–D–R equation is treated implicitly. Such an implementation requires the solution of implicit equations that couple each term in the system. The computational cost in terms of operations and memory of solving these (typically nonlinear) coupled equations in a fully implicit method can be significant, especially when the number of chemical species is large. (For example, the methane–air combustible mixture considered in [25] involves 49 species and 260 reactions.)

Several strategies exist to avoid the solution of the completely coupled implicit equation that arise in fully implicit methods. One possibility is to use a *semi-implicit* (hereafter SI) approach, in which advection is handled explicitly, whereas diffusion and reaction are integrated implicitly (see, e.g. [3,14,26]). The implicit equations in the SI approach can be considerably less expensive to solve in terms of computational cost than those arising in fully implicit methods, especially when the advection term is nonlinear. Operator splitting (OS) is another strategy that leads to a more affordable numerical solution [16,28]. In the OS approach, processes are decoupled and integrated sequentially. The resulting implicit equations in the OS approach are generally even easier to solve than those in the SI approach, since the reaction usually gives rise to a spatially local equation. Moreover, the OS approach allows different time steps to be used for the advection, diffusion, and reaction terms. By integrating slower processes using larger time steps, one can usually reduce the overall computational cost with little loss in accuracy. However, in addition to the numerical errors arising in the integration of each term, the OS approach also introduces splitting errors, which reduce the accuracy of the approximation [16,23]. A popular classical approach to achieve second-order temporal accuracy in a scheme with OS is Strang's splitting [23], which describes the precise ordering in which to combine second-order accurate solvers for each of the split terms in order to achieve overall second-order accuracy. The main limitation of Strang's splitting is that the generalization to higher-order in time is not straightforward. Thus, any gain in efficiency with an OS approach must be considered relative to the potential loss in accuracy [16,23].

In [24] Sun proposes a pseudo-non-time-splitting (PNTS) method, which is similar to the SI method in that advection is treated explicitly and diffusion and reaction are integrated implicitly. In addition, the PNTS method integrates the three terms using different time steps; thus, in this respect it is similar to the OS approach. However, the PNTS method reduces splitting errors by weakly coupling the the different terms during each time step. Specifically, temporally constant approximations of terms that would normally be neglected in each step of an OS approach are included. Sun shows that the PNTS method generates results comparable to those obtained using a standard OS method with less computational time [24]. A similar strategy for reducing splitting errors is to iteratively solve a series of weakly coupled operator-split equations during each time step. Such iterative OS methods include approximations to the neglected terms in the OS equations, which are iteratively improved during each time step [15,27]. Again, the benefit of reducing the splitting error in iterative OS methods must be weighed against the increased computational cost of the iterative procedure [29].

In this study, high-order multi-implicit spectral deferred correction (MISDC) methods are presented for solving A–D–R equations. MISDC methods are similar to PNTS or OS methods in that these methods allow terms to be solved in a decoupled manner using different time steps. In theory, however, 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. MISDC methods, which are a generalization of SI spectral deferred correction (SDC) methods introduced in [19] (which are in turn modifications of the explicit and implicit SDC methods appearing in [11]), 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. The accuracy and stability of MISDC methods for A–D–R equations and its efficiency relative to SI methods are investigated in this study.

The outline of this paper is as follows. In Section 2, a review of SISDC methods is presented in the context of a MOL discretization of the A–D–R equations. In Section 3, standard OS methods and the new

MISDC methods are presented and discussed. In Section 4, the stability and accuracy of MISDC methods are studied using a linear model problem and comparison to SISDC methods are made. In Section 5, numerical results are presented to demonstrate the convergence and accuracy properties of the methods. Numerical experiments for the motivating example (1) are used to illustrate the improvement in efficiency by selectively reducing only the reaction time step. The numerical results also identify cases for which MISDC methods compare favorably with SI Runge–Kutta methods.

2. SISDC methods

This section presents a short description of SISDC methods introduced in [19] in the context of a MOL discretization of A–D–R equations. SISDC methods are similar to MISDC methods in that some of the terms in the equation are treated explicitly and some implicitly. However, unlike MISDC methods, each term in SISDC methods is integrated with the same time step, and all implicit terms are integrated together.

Because this work focuses on the time discretization accuracy only, the A–D–R problem is presented in one space dimension for simplicity without loss of generality; extension to more space dimensions is discussed in Section 6. Let $u(x,t)$ be a (possibly vector-valued) function that satisfies the A–D–R equation

$$u_t = f_A u_x + v u_{xx} + f_R. \tag{2}$$

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

A popular method for approximating the solution to (2) is the MOL, in which the equation is first discretized spatially, resulting in a system of ODEs:

$$u'(t) = F_A(t,u(t)) + F_D(t,u(t)) + F_R(t,u(t)), \tag{3}$$

$$u(a) = u_0 \tag{4}$$

for $t \in [a,b]$. The terms F_A, F_D, and F_R are obtained from the spatial discretization of $f_A u_x$, $v u_{xx}$, 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. (3) and (4) can be integrated using SISDC 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.

The integral form of the solution to (3) and (4) is given by

$$u(t) = u_0 + \int_a^t (F_A(\tau,u(\tau)) + F_D(\tau,u(\tau)) + F_R(\tau,u(\tau)))\,d\tau. \tag{5}$$

Let $\tilde{u}(t)$ be an approximation to $u(t)$. SDC methods generate high-order numerical solutions by approximating the correction $\delta(t) \equiv u(t) - \tilde{u}(t)$, which is then used to improve the accuracy of $\tilde{u}(t)$.

The procedures with which $\delta(t)$ is computed using the approximate solution $\tilde{u}(t)$ are now described. To this end, define the residual function

$$E(t,\tilde{u}(t)) = u_0 + \int_a^t \Big(F_A(\tau,\tilde{u}(\tau)) + F_D(\tau,\tilde{u}(\tau)) + F_R(\tau,\tilde{u}(\tau))\Big)\,d\tau - \tilde{u}(t). \tag{6}$$

The definition of $\delta(t)$ and the integral equation (5) can be combined to give

$$\tilde{u}(t)+\delta(t)=u_0+\int_a^t\Big(F_{\mathrm{A}}(\tau,\tilde{u}(\tau)+\delta(\tau))+F_{\mathrm{D}}(\tau,\tilde{u}(\tau)+\delta(\tau))+F_{\mathrm{R}}(\tau,\tilde{u}(\tau)+\delta(\tau))\Big)\,\mathrm{d}\tau. \tag{7}$$

From (6) and (7), one obtains the *correction equation*

$$\begin{aligned}\delta(t)=\int_a^t\Big(&F_{\mathrm{A}}(\tau,\tilde{u}(\tau)+\delta(\tau))-F_{\mathrm{A}}(\tau,\tilde{u}(\tau))+F_{\mathrm{D}}(\tau,\tilde{u}(\tau)+\delta(\tau))-F_{\mathrm{D}}(\tau,\tilde{u}(\tau))+F_{\mathrm{R}}(\tau,\tilde{u}(\tau)+\delta(\tau))\\&-F_{\mathrm{R}}(\tau,\tilde{u}(\tau))\Big)\,\mathrm{d}\tau+E(t,\tilde{u}(t)).\end{aligned} \tag{8}$$

In the numerical discretization, let $\Delta t>0$ be the time step and $t_n=n\Delta t$, for $n=0,1,2,\ldots$, be the nth time-level. 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 (8) to $(s+1)$th order. If F_{A}, F_{D}, and F_{R} are Lipschitz continuous in u, then (8) implies that $\|\delta(t)-E(t,\tilde{u})\|=\mathcal{O}(\Delta t^{s+1})$. Therefore, 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-hand side of (8). In fact, ignoring the integral term completely gives the standard Picard iteration procedure which also converges assuming Lipschitz continuity, albeit slowly if the Lipschitz constant is large.

In a MOL discretization, the situation is more subtle in that the operators F_{A} and F_{D} approximate differential operators, which are not Lipschitz continuous. Therefore, some higher spatial derivative of $\tilde{u}$ must be bounded for the above arguments to hold. This subtle interaction between temporal and spatial error is problem dependent and is particularly important in the discussion of spatial and temporal approximation of boundary conditions [1,10,21].

To advance the solution by one time step, SISDC methods described in [19] use an SI method, based on the first-order forward and backward Euler methods, to compute a provisional solution on $[t_n,t_{n+1}]$. Then the accuracy of the solution is improved by iteratively solving the correction equation (8). A summary of the method is given below.

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 $t_{n,m}$ for $m=0,1,\ldots,N_{\mathrm{A}}$ such that $t_n=t_{n,0}<t_{n,1}<\cdots<t_{n,m}<\cdots<t_{n,N_{\mathrm{A}}}=t_{n+1}$. 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_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 $F_m(u_m^k)$. An approximate solution $\tilde{u}\equiv u_m^0$, for $m=0,1,\ldots,N_{\mathrm{A}}$, is computed by means of the SI method as follows:

$$u_{m+1}^0=u_m^0+\Delta t_m\big(F_{A_m}(u_m^0)+F_{\mathrm{D}_{m+1}}(u_{m+1}^0)+F_{\mathrm{R}_{m+1}}(u_{m+1}^0)\big). \tag{9}$$

The accuracy of u_m^0 is improved by iteratively solving a similar SI discretization of the correction equation (8) for δ^k, $k=0,1,2,\ldots$, and setting $u^{k+1}=u^k+\delta^k$. Specifically,

$$\begin{aligned}\delta_{m+1}^k=\delta_m^k+\Delta t_m\big(&F_{\mathrm{A}_m}(u_m^k+\delta_m^k)-F_{\mathrm{A}_m}(u_m^k)+F_{\mathrm{D}_{m+1}}(u_{m+1}^k+\delta_{m+1}^k)-F_{\mathrm{D}_{m+1}}(u_{m+1}^k)\\&+F_{\mathrm{R}_{m+1}}(u_{m+1}^k+\delta_{m+1}^k)-F_{\mathrm{R}_{m+1}}(u_{m+1}^k)\big)+E_{m+1}(u^k)-E_m(u^k).\end{aligned} \tag{10}$$

The solution of both (9) and (10) requires the solution of an implicit equation that couples F_{D} and F_{R}.

From (10), a direct update equation for u^{k+1} can be derived. Let $I_m^{m+1}(F_{\mathrm{A}}(u^k)+F_{\mathrm{D}}(u^k)+F_{\mathrm{R}}(u^k))$ denote the numerical quadrature approximation to

$$\int_{t_m}^{t_{m+1}} \left(F_{\mathrm{A}}(\tau, u^k(\tau)) + F_{\mathrm{D}}(\tau, u^k(\tau)) + F_{\mathrm{R}}(\tau, u^k(\tau))\right) \mathrm{d}\tau. \tag{11}$$

To obtain a $(k+1)$th order approximation for u^{k+1}, the residual function $E(u^k)$, and hence the quadrature (11), must be approximated with $(k+1)$th order accuracy. In [19], the points t_m are chosen to be the Gauss–Lobatto nodes of the interval $[t_n, t_{n+1}]$ and the quadrature (11) is computed, with $\mathcal{O}(\Delta t^{k+2})$ error, as the integral of an interpolating polynomial over the subinterval $[t_m, t_{m+1}]$.

From (6), one obtains the following expression for $E_{m+1}(u^k) - E_m(u^k)$:

$$E_{m+1}(u^k) - E_m(u^k) = I_m^{m+1}(F_{\mathrm{A}}(u^k) + F_{\mathrm{D}}(u^k) + F_{\mathrm{R}}(u^k)) - u_{m+1}^k + u_m^k. \tag{12}$$

From (10) and (12), one obtains the following update equation for u^{k+1}:

$$\begin{aligned} u_{m+1}^{k+1} = {} & u_m^{k+1} + \Delta t_m \left(F_{\mathrm{A}_m}(u_m^{k+1}) - F_{\mathrm{A}_m}(u_m^k) + F_{\mathrm{D}_{m+1}}(u_{m+1}^{k+1}) - F_{\mathrm{D}_{m+1}}(u_{m+1}^k) + F_{\mathrm{R}_{m+1}}(u_{m+1}^{k+1})\right. \\ & \left. - F_{\mathrm{R}_{m+1}}(u_{m+1}^k)\right) + I_m^{m+1}\left(F_{\mathrm{A}}(u^k) + F_{\mathrm{D}}(u^k) + F_{\mathrm{R}}(u^k)\right). \end{aligned} \tag{13}$$

By explicitly integrating the advection term, which is usually nonlinear, the SI approach can be computationally less expensive than a fully implicit approach. Nonetheless, a more efficient method may be developed by further splitting the integration within the implicit step into a diffusion step and a reaction step. Moreover, the integration could proceed with different time steps for different processes, an important feature in multiple time-scale problems. The design of high-order methods along these lines is described in the next section.

3. High-order MISDC methods

This section presents MISDC methods, in which the different processes in A–D–R equations can be integrated separately while simultaneously controlling splitting errors. It is assumed that the reactive time scale in the A–D–R system is significantly shorter than the corresponding diffusive time scale, which is yet shorter than the advective time scale. (As will be explained below, MISDC methods can be extended to solve systems with a different ordering of time scales.) Before MISDC methods are introduced, a review of standard OS methods and splitting errors in the context of A–D–R equations is presented.

A popular approach used in the numerical solution of problems involving processes with widely varying time scales is the OS approach (e.g. [16,28]), in which processes are decoupled and integrated sequentially. In the present context, an OS method advances the solution one time step by approximating the following equations sequentially:

$$u_{\mathrm{A}}(t+\Delta t) = u(t) + \int_t^{t+\Delta t} F_{\mathrm{A}}(\tau, u_{\mathrm{A}}(\tau))\, \mathrm{d}\tau, \tag{14}$$

$$u_{\mathrm{D}}(t+\Delta t) = u_{\mathrm{A}}(t+\Delta t) + \int_t^{t+\Delta t} F_{\mathrm{D}}(\tau, u_{\mathrm{D}}(\tau))\, \mathrm{d}\tau, \tag{15}$$

$$u(t+\Delta t) = u_{\mathrm{D}}(t+\Delta t) + \int_t^{t+\Delta t} F_{\mathrm{R}}(\tau, u(\tau))\, \mathrm{d}\tau. \tag{16}$$

A numerical solution obtained by numerically integrating (14)–(16) has two types of numerical errors: integration errors, introduced by approximating the integrals on the right-hand side of (14)–(16), and splitting errors, introduced by separating the solution given by equation (5) into sequential steps. Unless the operators associated with F_{A}, F_{D}, and F_{R} commute, the approximation given by (14)–(16) is no more

accurate than $\mathcal{O}(\Delta t^2)$ locally and $\mathcal{O}(\Delta t)$ globally, regardless of the order of the integration errors [23]. To achieve high-order accuracy, both splitting and integration errors must be reduced.

MISDC methods achieve higher-order accuracy by simultaneously reducing splitting and integration errors during the deferred correction iterations. This is achieved, in part, via the coupling of the intermediate solutions of (14)–(16). To illustrate, (14)–(16) are rewritten as

$$u_{\mathrm{A}}(t+\Delta t) = u(t) + \int_t^{t+\Delta t} F_{\mathrm{A}}(\tau, u(\tau))\,\mathrm{d}\tau, \tag{17}$$

$$u_{\mathrm{D}}(t+\Delta t) = u(t) + \int_t^{t+\Delta t} (F_{\mathrm{A}}(\tau, u(\tau)) + F_{\mathrm{D}}(\tau, u(\tau)))\,\mathrm{d}\tau, \tag{18}$$

$$u(t+\Delta t) = u(t) + \int_t^{t+\Delta t} (F_{\mathrm{A}}(\tau, u(\tau)) + F_{\mathrm{D}}(\tau, u(\tau)) + F_{\mathrm{R}}(\tau, u(\tau)))\,\mathrm{d}\tau. \tag{19}$$

Note that in the integrals of (17)–(19), the entire target solution u appears, as opposed to (14)–(16), in which intermediate solutions are used. However, in practice F_{D} is non-local and F_{R} is nonlinear, so a numerical approximation to the coupled equations (17)–(19) requires the solution of a system of coupled nonlinear equations, which may be computationally expensive. Thus, the following weakly coupled equations are sequentially approximated instead:

$$u_{\mathrm{A}}(t+\Delta t) = u(t) + \int_t^{t+\Delta t} F_{\mathrm{A}}(\tau, u_{\mathrm{A}}(\tau))\,\mathrm{d}\tau, \tag{20}$$

$$u_{\mathrm{D}}(t+\Delta t) = u(t) + \int_t^{t+\Delta t} (F_{\mathrm{A}}(\tau, u_{\mathrm{A}}(\tau)) + F_{\mathrm{D}}(\tau, u_{\mathrm{D}}(\tau)))\,\mathrm{d}\tau, \tag{21}$$

$$u(t+\Delta t) = u(t) + \int_t^{t+\Delta t} (F_{\mathrm{A}}(\tau, u_{\mathrm{A}}(\tau)) + F_{\mathrm{D}}(\tau, u_{\mathrm{D}}(\tau)) + F_{\mathrm{R}}(\tau, u(\tau)))\,\mathrm{d}\tau. \tag{22}$$

One may notice that (14)–(16) and (20)–(22) are equivalent analytically. However, when discretized, and in particular if different time steps are used to integrate different processes, then (20)–(22) result in smaller splitting errors, as explained below. Suppose Δt_m, Δt_p, and Δt_q are used to advance u_{A}, u_{D}, and u, respectively. Then, as are u_{D} and u, F_{A} in (21) and $F_{\mathrm{A}} + F_{\mathrm{D}}$ in (22) are integrated for Δt_p and Δt_q, respectively. In (14)–(16), however, the integration of F_{A} is done in (14) for Δt_m and F_{D} in (15) for Δt_p, giving rise to larger splitting errors. How MISDC methods further reduce splitting errors is discussed below.

Given some interval $[t_n, t_{n+1}]$ on which the solution is sought and an approximation $\tilde{u}(t)$ to $u(t)$, $t \in [t_n, t_{n+1}]$, MISDC methods improve the accuracy of $\tilde{u}(t)$ by computing the correction term $\delta(t) \equiv u(t) - \tilde{u}(t)$. To this end, let $E(t, \tilde{u})$ be the residual function associated with $\tilde{u}(t)$:

$$E(t, \tilde{u}) = u(t_n) + \int_{t_n}^{t} \Big(F_{\mathrm{A}}(\tau, \tilde{u}(\tau)) + F_{\mathrm{D}}(\tau, \tilde{u}(\tau)) + F_{\mathrm{R}}(\tau, \tilde{u}(\tau))\Big)\,\mathrm{d}\tau - \tilde{u}(t). \tag{23}$$

Following [19], the correction equation that arises from (19) and (23) is

$$\begin{aligned}\delta(t) = \int_{t_n}^{t} \Big(&F_{\mathrm{A}}(\tau, \tilde{u}(\tau) + \delta(\tau)) - F_{\mathrm{A}}(\tau, \tilde{u}(\tau)) + F_{\mathrm{D}}(\tau, \tilde{u}(\tau) + \delta(\tau)) \\ &- F_{\mathrm{D}}(\tau, \tilde{u}(\tau)) + F_{\mathrm{R}}(\tau, \tilde{u}(\tau) + \delta(\tau)) - F_{\mathrm{R}}(\tau, \tilde{u}(\tau))\Big)\,\mathrm{d}\tau + E(t, \tilde{u}(\tau)).\end{aligned} \tag{24}$$

As in (17)–(19), (24) can be decoupled and approximated by δ_A and δ_D, the definitions of which are analogous to (20) and (21), respectively. That is, δ_A is an approximation to δ that preserves the advection terms inside the integral and δ_D is a more accurate approximation that preserves both the advection and diffusion terms:

$$\delta_A(t) = \int_{t_n}^{t} \Big(F_A(\tau, \tilde{\boldsymbol{u}}(\tau) + \delta(\tau)) - F_A(\tau, \tilde{\boldsymbol{u}}(\tau)) \Big)\, d\tau + E(t, \tilde{\boldsymbol{u}}(\tau)), \tag{25}$$

$$\delta_D(t) = \delta_A(t) + \int_{t_n}^{t} \Big(F_D(\tau, \tilde{\boldsymbol{u}}(\tau) + \delta(\tau)) - F_D(\tau, \tilde{\boldsymbol{u}}(\tau)) \Big)\, d\tau. \tag{26}$$

Thus,

$$\delta(t) = \delta_D(t) + \int_{t_n}^{t} \Big(F_R(\tau, \tilde{\boldsymbol{u}}(\tau) + \delta(\tau)) - F_R(\tau, \tilde{\boldsymbol{u}}(\tau)) \Big)\, d\tau. \tag{27}$$

If the correction equations (25)–(27) were to be solved simultaneously, MISDC methods would generate approximations with no splitting errors. However, as is with (17)–(19), a numerical approximation to (25)–(27) requires the computationally expensive approximation of a system of coupled nonlinear equations. Because, as mentioned in the previous section, the integral terms in the correction equation need only be approximated with a low-order method, a weakly coupled set of correction equations analogous to (20)–(22) are approximated sequentially instead:

$$\delta_A(t) = \int_{t_n}^{t} \Big(F_A(\tau, \tilde{\boldsymbol{u}}(\tau) + \delta_A(\tau)) - F_A(\tau, \tilde{\boldsymbol{u}}(\tau)) \Big)\, d\tau + E(t, \tilde{\boldsymbol{u}}(\tau)), \tag{28}$$

$$\delta_D(t) = \int_{t_n}^{t} \Big(F_A(\tau, \tilde{\boldsymbol{u}}(\tau) + \delta_A(\tau)) - F_A(\tau, \tilde{\boldsymbol{u}}(\tau)) + F_D(\tau, \tilde{\boldsymbol{u}}(\tau) + \delta_D(\tau)) - F_D(\tau, \tilde{\boldsymbol{u}}(\tau)) \Big)\, d\tau + E(t, \tilde{\boldsymbol{u}}(\tau)), \tag{29}$$

$$\begin{aligned} \delta(t) = \int_{t_n}^{t} \Big(& F_A(\tau, \tilde{\boldsymbol{u}}(\tau) + \delta_A(\tau)) - F_A(\tau, \tilde{\boldsymbol{u}}(\tau)) + F_D(\tau, \tilde{\boldsymbol{u}}(\tau) + \delta_D(\tau)) - F_D(\tau, \tilde{\boldsymbol{u}}(\tau)) \\ & + F_R(\tau, \tilde{\boldsymbol{u}}(\tau) + \delta(\tau)) - F_R(\tau, \tilde{\boldsymbol{u}}(\tau)) \Big)\, d\tau + E(t, \tilde{\boldsymbol{u}}(\tau)). \end{aligned} \tag{30}$$

For standard advection, diffusion, and reaction discrete operators, the system (28)–(30) is easier to solve than (25)–(27) because typically the advection correction equation (28) is integrated explicitly; the diffusion correction equation (29) is non-local but linear, so efficient direct or iterative solvers for large linear systems of algebraic equations can be used; the reaction correction equation (30) is nonlinear but local, resulting in a system of uncoupled ODEs, for which efficient and highly parallelizable solvers are available. However, splitting errors are introduced into (28)–(30) because the first F_A and F_D terms inside the integrals on the right-hand side of (28) and (29) are not evaluated at $\tilde{\boldsymbol{u}} + \delta$ but at $\tilde{\boldsymbol{u}} + \delta_A$ and $\tilde{\boldsymbol{u}} + \delta_D$, respectively. When F_A and F_D are Lipschitz continuous, this splitting error is one order smaller than δ itself; thus, as the order of accuracy of $\tilde{\boldsymbol{u}}$ is improved during the deferred correction iterations, the splitting error converges to zero.

In addition to splitting errors, solutions computed by both OS and MISDC methods contain integration errors that arise from numerical quadrature approximation of the integrals. The integration errors associated with the diffusion and reaction terms may be large because of their stiffness. Owing to the decoupling

of the processes in both OS and MISDC methods, it is possible to selectively reduce integration errors by using smaller time steps for fast-scale processes.

To use smaller time steps for the diffusion process and yet smaller ones for the reaction process, the subinterval $[t_m, t_{m+1}]$ is subdivided into N_D subintervals by choosing points $t_{m,p}$ for $p = 0, 1, \ldots, N_D$ such that $t_m = t_{m,0} < t_{m,1} < \cdots < t_{m,p} < \cdots < t_{m,N_D} = t_{m+1}$. 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 $t_{m,p} = t_{m,p,0} < t_{m,p,1} < \cdots < t_{m,p,q} < \cdots < t_{m,p,N_R} = t_{m,p+1}$. 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. Let $\Delta t_p \equiv t_{p+1} - t_p$ and $\Delta t_q \equiv t_{q+1} - t_q$. Fig. 1 shows an example of time-step subdivision. In this implementation, t_m, t_p, and t_q are Gauss–Lobatto nodes of the intervals $[t_n, t_{n+1}]$, $[t_m, t_{m+1}]$, and $[t_p, t_{p+1}]$, respectively.

An implementation of high-order MISDC methods is described in the remainder of this section. As in the previous section, for an arbitrary function $\psi(t)$, let ψ_m^k, ψ_p^k, and ψ_q^k denote the numerical approximations to $\psi(t_m)$, $\psi(t_p)$, and $\psi(t_q)$, respectively, after k deferred correction iterations; an analogous convention is also adopted for $F_p(\psi_p^k)$ and $F_q(\psi_q^k)$. To compute the provisional solution, the following equations are solved:

For $m = 0, \ldots, N_A - 1$

$$u^0_{A_{m+1}} = u^0_m + \Delta t_m F_{A_m}(u^0_m), \tag{31}$$

For $p = 0, \ldots, N_D - 1$

$$u^0_{D_{p+1}} = u^0_p + \Delta t_p \left(F_{A_m}(u^0_m) + F_{D_{p+1}}(u^0_{D_{p+1}}) \right), \tag{32}$$

Compute $F_{D_{p+1}}(u^0_{D_{p+1}})$,

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

$$u^0_{q+1} = u^0_q + \Delta t_q \left(F_{A_m}(u^0_m) + F_{D_{p+1}}(u^0_{D_{p+1}}) + F_{R_{q+1}}(u^0_{q+1}) \right), \tag{33}$$

End

End

Compute $F_{A_{m+1}}(u^0_{m+1})$, (34)

End

In (31) $u^0_m = u(t_n)$ for $m = 0$ and $u^0_m = u^0_{m-1,N_D-1,N_R}$ for $m > 0$; in (32) $u^0_p = u^0_m$ for $p = 0$ and $u^0_p = u^0_{m,p-1,N_R}$ for $p > 0$; and in (33) $u^0_q = u^0_{m,p-1,N_R}$ for $q = 0$ and $p > 0$, and $u^0_q = u(t_n)$ for $q = 0$ and $p = 0$. (These definitions are different from the OS methods described earlier.) Since F_A is treated explicitly, $u^0_{A_{m+1}}$ is not used; thus, in practice (31) is superfluous. Moreover, F_{A_m} is evaluated at the full provisional solution u^0_m in (34) rather than at $u^0_{A_m}$. With the coupling among processes preserved, the solution obtained by solving (31)–(34) is more accurate than the solution obtained from the analogous approximation to (14)–(16) using the same time steps Δt_m, Δt_p, and Δt_q [24].

A MI discretization of the correction equations (28)–(30) is given by

$$\delta^k_{A_{m+1}} = \delta^k_{A_m} + \Delta t_m \left(F_{A_m}(u^{k+1}_m) - F_{A_m}(u^k_m) \right) + E_{m+1}(u^k) - E_m(u^k), \tag{35}$$

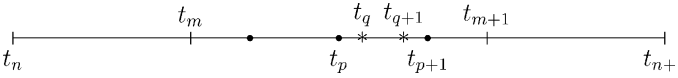

Fig. 1. Illustration of three levels of time-step subdivision: $[t_n, t_{n+1}]$ into $[t_m, t_{m+1}]$, $[t_m, t_{m+1}]$ into $[t_p, t_{p+1}]$, and $[t_p, t_{p+1}]$ into $[t_q, t_{q+1}]$. In this example, $N_A = 3$, $N_D = 4$, $N_R = 3$, $m = 1$, $p = 2$, and $q = 1$.

$$\delta^k_{\mathrm{D}_{p+1}} = \delta^k_{\mathrm{D}_p} + \Delta t_p\Big(F_{\mathrm{A}_m}(u^{k+1}_m) - F_{\mathrm{A}_m}(u^k_m) + F_{\mathrm{D}_{p+1}}(u^k_{p+1} + \delta^k_{\mathrm{D}_{p+1}}) - F_{\mathrm{D}_{p+1}}(u^k_{p+1})\Big) + E_{p+1}(u^k) - E_p(u^k), \quad (36)$$

$$\begin{aligned}\delta^k_{q+1} = \delta^k_q + \Delta t_q\Big(&F_{\mathrm{A}_m}(u^{k+1}_m) - F_{\mathrm{A}_m}(u^k_m) + F_{\mathrm{D}_{p+1}}(u^k_{p+1} + \delta^k_{\mathrm{D}_{p+1}}) - F_{\mathrm{D}_{p+1}}(u^k_{p+1}) + F_{\mathrm{R}_{q+1}}(u^k_{q+1} + \delta^k_{q+1})\\ &- F_{\mathrm{R}_{q+1}}(u^k_{q+1})\Big) + E_{q+1}(u^k) - E_q(u^k).\end{aligned} \quad (37)$$

As in [19], the discretized correction equations (35)–(37) are rewritten in terms of updated values of the target function. To this end, the definition of the numerical quadrature approximation $I^{m+1}_m(\psi^k)$ is extended to $I^{p+1}_p(\psi^k) \equiv \int_{t_p}^{t_{p+1}} \psi^k(\tau)\,\mathrm{d}\tau$ and $I^{q+1}_q(\psi^k) \equiv \int_{t_q}^{t_{q+1}} \psi^k(\tau)\,\mathrm{d}\tau$. The correction terms δ_{A}, δ_{D}, and δ are used to obtain u_{A}, u_{D}, and u, respectively, from $\tilde{\boldsymbol{u}}$. That is, u_{A} is set to $\tilde{\boldsymbol{u}} + \delta_{\mathrm{A}}$, u_{D} to $\tilde{\boldsymbol{u}} + \delta_{\mathrm{D}}$, and u to $\tilde{\boldsymbol{u}} + \delta$. Combining these update relations with (36) and and (37), one arrives at the following update equations for the diffusion and reaction processes:

$$\begin{aligned}u^{k+1}_{\mathrm{D}_{p+1}} = {} & u^{k+1}_{\mathrm{D}_p} + \Delta t_p\Big(F_{\mathrm{A}_m}(u^{k+1}_{\mathrm{A}_m}) - F_{\mathrm{A}_m}(u^k_m) + F_{\mathrm{D}_{p+1}}(u^{k+1}_{\mathrm{D}_{p+1}}) - F_{\mathrm{D}_{p+1}}(u^k_{p+1})\Big)\\ &+ I^{p+1}_p\big(F_{\mathrm{A}}(u^k) + F_{\mathrm{D}}(u^k) + F_{\mathrm{R}}(u^k)\big),\end{aligned} \quad (38)$$

$$\begin{aligned}u^{k+1}_{q+1} = {} & u^{k+1}_q + \Delta t_q\Big(F_{\mathrm{A}_m}(u^{k+1}_{\mathrm{A}_m}) - F_{\mathrm{A}_m}(u^k_m) + F_{\mathrm{D}_{p+1}}(u^{k+1}_{\mathrm{D}_{p+1}}) - F_{\mathrm{D}_{p+1}}(u^k_{p+1}) + F_{\mathrm{R}_{q+1}}(u^{k+1}_{q+1}) - F_{\mathrm{R}_{q+1}}(u^k_{q+1})\Big)\\ &+ I^{q+1}_q\big(F_{\mathrm{A}}(u^k) + F_{\mathrm{D}}(u^k) + F_{\mathrm{R}}(u^k)\big).\end{aligned} \quad (39)$$

Because the advection process is treated explicitly, u^k_{A} need not be computed; thus, an update equation for u^k_{A} is not needed. In summary, the following steps improve the order of accuracy of u^k by one:

```
For m = 0, ..., N_A − 1
    For p = 0, ..., N_D − 1
      Solve (38) for u^{k+1}_{D_{p+1}}; Compute F_{D_{p+1}}(u^{k+1}_{D_{p+1}}).
      For q = 0, ..., N_R − 1
        Solve (39) for u^{k+1}_{q+1}; Compute F_{R_{q+1}}(u^{k+1}_{q+1}).
      End
      Update F_{D_{p+1}}(u^{k+1}_{p+1}).
    End
    Compute F_{A_{m+1}}(u^{k+1}_{m+1}).
End
```
(40)

The above steps are ordered to minimize splitting errors introduced by the MI scheme [22], as explained below. Although these splitting errors, which are locally $\mathcal{O}(\Delta t^{k+3})$ and globally $\mathcal{O}(\Delta t^{k+2})$, are smaller than those in the standard OS approach, they are nonetheless present in (39) because the F_{D} term in (39) is evaluated not at u^{k+1}_{q+1} but rather at $u^{k+1}_{\mathrm{D}_{p+1}}$. For an arbitrary function $F(t, u(t))$,

$$F(t,u(t)) = F(t,\tilde{\boldsymbol{u}}(t)) + (u(t) - \tilde{\boldsymbol{u}}(t))\frac{\partial}{\partial u}F(t,u(t)) + \mathcal{O}((u(t) - \tilde{\boldsymbol{u}}(t))^2). \quad (41)$$

Note that if F is a linear operator (e.g., when the advection coefficient f_{A} in (2) is independent of u, in which case F_{A} is a linear operator), then $\partial F/\partial u$ is a function of t only (i.e., independent of u) for each ODE that arises from the MOL spatial discretization of the PDE (2).

By assumption, the reaction term is stiffer than the diffusion term, which is yet stiffer than the advection term. It follows that $|\partial F_R/\partial u| \gg |\partial F_D/\partial u| \gg |\partial F_A/\partial u|$. (This is true for the numerical simulations presented in Section 5.) Thus, to minimize splitting errors, F_R, rather than F_A or F_D, is evaluated at u_{q+1}^{k+1}.

In some A–D–R systems, the diffusion term may be stiffer than the reaction term. To minimize intergration errors in this case, diffusion should be updated using the smallest time step Δt_q, reaction updated using Δt_p, and advection updated using Δt_m. To minimize splitting errors, the ordering of the diffusion and reaction steps in (40) should also be reversed, i.e., reaction is updated for every N_R diffusion updates, and advection is updated for every N_D reaction updates.

Four parameters must be chosen to specify the MISDC method, two of which determine the order of accuracy of the method, and two which determine the relative sizes of the three time steps. Following the notation in [19], let MISDC(K, P, N_D, N_R) denote the MISDC method that uses K total iterations (i.e., computation of provisional solution and the number of iterations of correction equations) and P Gauss–Lobatto nodes (or $N_A = P - 1$ advection substeps). Furthermore, recall that N_D denotes the number of subintervals within an advection substep (or, equivalently, the ratio of the advection to diffusion time step size), and N_R denotes the number of subintervals within a diffusion substep (or the ratio of diffusion to reaction time step size). Therefore, the total number of reaction time steps in one deferred correction iteration of the MISDC method is $(P-1)N_D N_R$. The formal order of accuracy of the method is $\min(K, P)$.

All four of the above parameters could be adjusted during the solution process to improve the efficiency of the overall MISDC method. Monitoring the size of the respective correction terms in the MISDC iteration provides information on the error for that piece of the equation and can be used to adjust relative time steps. A more involved examination of time step selection will be presented in a future work.

4. Linear stability and accuracy analysis

In this section, a linear stability and accuracy analysis is performed on MISDC methods using the standard model problem. Let λ be a complex constant, c be a real number with $0 < c \ll 1$, and let

$$u'(t) = \lambda u(t) \equiv \mathrm{i}\,\mathrm{Im}(\lambda)u(t) + c\,\mathrm{Re}(\lambda)u(t) + (1-c)\,\mathrm{Re}(\lambda)u(t), \tag{42}$$

$$u(0) = 1. \tag{43}$$

The imaginary part of the right-hand side of (42) is associated with the advection process and is treated implicitly; the two real parts are associated with the diffusion and reaction processes, respectively, and are treated implicitly. The number c determines the relative stiffness of the reaction term compared to the diffusion term.

4.1. Stability analysis

The stability region for a given method is defined as the set of λ such that $|\tilde{u}(\lambda)| < 1$, where $\tilde{u}(\lambda)$ represents the numerical solution resulting from one time step of the method with $\Delta t = 1$. Fig. 2 shows the stability diagrams for the MISDC$(K, K, 2, 2)$ methods for $K = 3, 4, \ldots, 7$, computed using (42) and (43) for $c = 0.1$. The lines in the figures indicate the λ values for which $|\tilde{u}(\lambda)| = 1$; the methods are stable for smaller λ. The stability diagrams for MISDC methods with different values of N_D and N_R are similar.

One might notice that the stability diagrams for MISDC and SISDC methods are similar (see [19] for the SISDC stability diagrams). Indeed, this is to be expected: the time step of a numerical method applied to the model problem (42) and (43) is limited by the imaginary part, which is treated explicitly in both methods. In the context of stability analysis, the most significant difference between MISDC and SISDC methods is the size of the time steps used for the diffusion and reaction terms (i.e., the real part of (42)), which does not

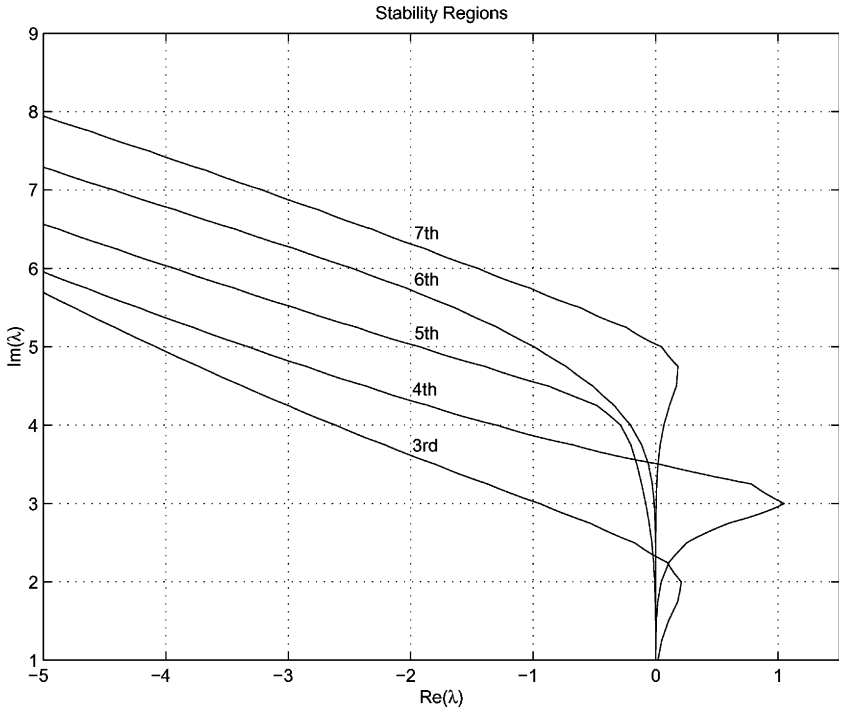

Fig. 2. Stability regions for MISDC$(K, K, 2, 2)$ methods for $K = 3, 4, \ldots, 7$.

affect the top boundary of the stability region. Therefore, the two types of SDC methods have similar stability diagrams.

4.2. Accuracy analysis

An accuracy region for a given error tolerance ϵ is defined as the set of λ such that $|\tilde{u}(\lambda) - e^{\lambda}| < \epsilon$. The accuracy regions for the MISDC$(K, K, 2, 2)$ method are shown in Fig. 3 for $K = 3, 4, \ldots, 7$. The lines in the figures indicate λ values for which $|\tilde{u}(\lambda) - e^{\lambda}| = \epsilon$ for $\epsilon = 10^{-4}$. Because accuracy increases with the order of the method, the size of the accuracy region also increases with the order. Also, the size of the accuracy regions for MISDC methods increases when N_{D} and N_{R} increase.

For a given spatial resolution, the amount of computational work per time step for a given MISDC$(K, N_{\mathrm{A}} + 1, N_{\mathrm{D}}, N_{\mathrm{R}})$ method is $\mathcal{O}(K \cdot N_{\mathrm{A}} \cdot N_{\mathrm{D}} \cdot N_{\mathrm{R}})$. In this study, work is measured by the number of implicit function evaluations, which arise in the solution of (32), (33), (38), and (39). With this criterion, work done for each time step is given by $KN_{\mathrm{A}}N_{\mathrm{D}}(1 + N_{\mathrm{R}})$. Since K and N_{A} normally scale with the order of the method, for fixed N_{D} and N_{R}, the amount of work per time step increases quadratically with the order. Fig. 4 shows the scaled accuracy diagrams for the MISDC$(K, K, 2, 2)$ methods. In the scaled accuracy diagrams, Re(λ) and Im(λ) are divided by the work done per time step. The accuracy diagrams for the higher-order methods are larger than those of the lower-order methods even after the scaling. The scaled accuracy diagrams for MISDC methods with different values of N_{D} and N_{R} are similar.

By taking smaller time steps for the diffusion and reaction processes, MISDC methods can generate approximations that are more accurate than SISDC methods of the same order. Indeed, by comparing

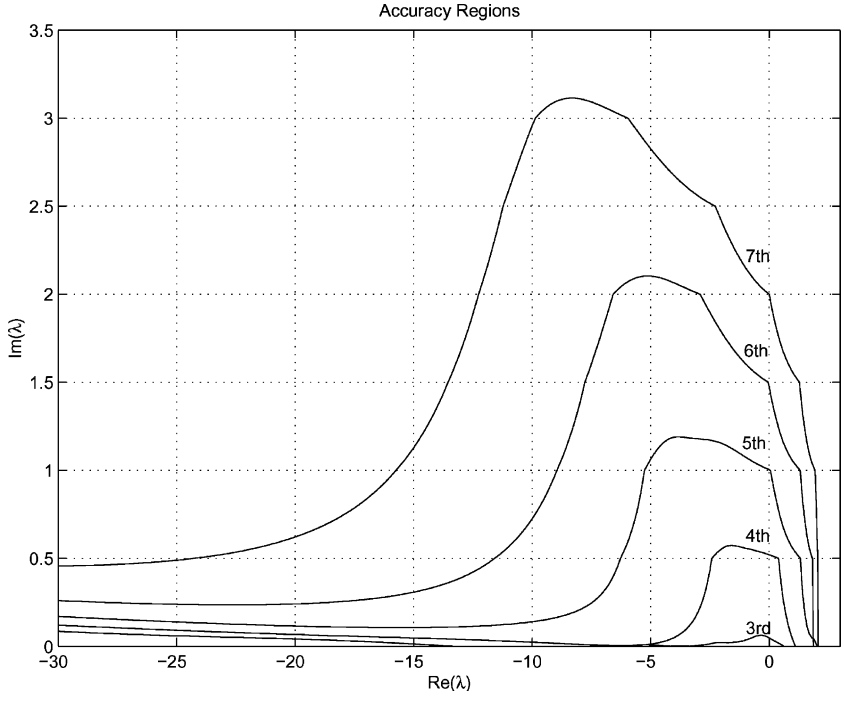

Fig. 3. Accuracy regions for MISDC$(K, K, 2, 2)$ methods for $\epsilon = 10^{-4}$ and $K = 3, 4, \ldots, 7$.

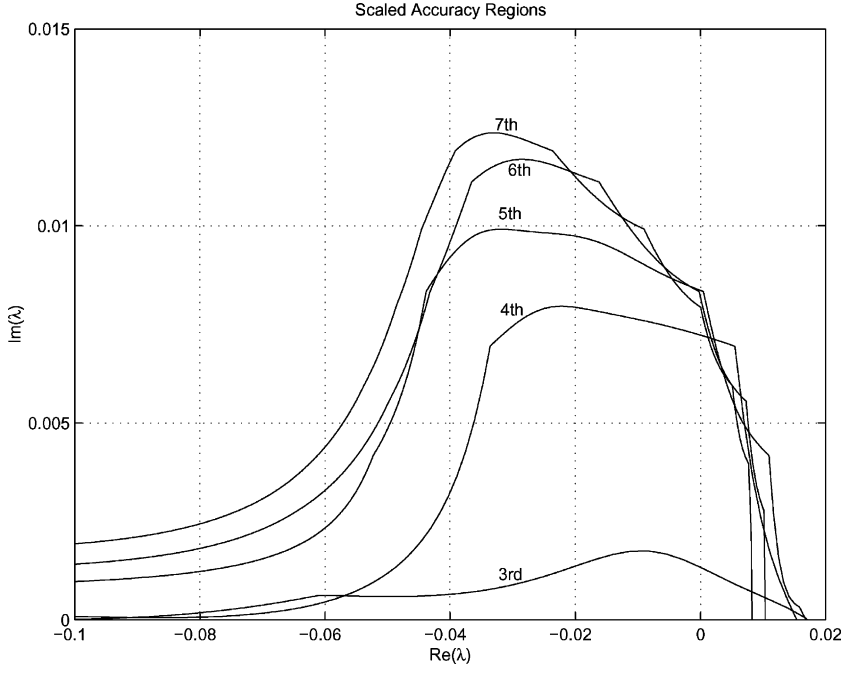

Fig. 4. Scaled accuracy regions for MISDC$(K, K, 2, 2)$ methods for $\epsilon = 10^{-4}$ and $K = 3, 4, \ldots, 7$.

Fig. 3 with Fig. 3 in [19], one sees that the accuracy regions of MISDC methods are larger than the corresponding SISDC methods. However, MISDC methods are also more expensive; thus, it is interesting to compare the efficiency of the two types of SDC methods for different λ. Fig. 5 shows the scaled accuracy regions for the MISDC(K,K,2,2) methods and the SISDC(K, K) methods for $K = 4$, 5, and 6, where SISDC(K, P) denotes a SISDC method that uses K total iterations and P Gauss–Lobatto nodes. SISDC methods integrate the diffusion and reaction terms simultaneously; thus, $K - 1$ implicit function evaluations are required for the computation of the provisional solution and $K - 1$ for each deferred correction iteration, or $K(K - 1)$ total (compared to $K(K - 1)N_D(1 + N_R)$ total implicit function evaluations per time step required by MISDC methods). However, as noted in Section 1, the implicit equations arising in SISDC methods are in general more difficult to solve than those of MISDC methods, although not in special cases such as the linear problem (42) and (43). This motivates us to count each implicit function evaluation of SISDC methods as two. In practice, these function evaluations may be more than twice as computationally expensive as those of MISDC methods. Compared to the scaled accuracy diagrams of SISDC methods, one noticeable feature of the scaled accuracy regions of MISDC methods is their long "tails" in the negative Re(λ) region. This means that, by using smaller time steps for the diffusion and reaction processes, MISDC methods generate accurate approximations and are more efficient than SISDC methods when the diffusion and reaction terms of the equation are stiff (i.e., when Re(λ) is large). In contrast, SISDC methods are more efficient for advection-dominant problems.

The decoupling of the diffusion and reaction integrations in MISDC methods gives rise to implicit equations that are easier to solve and allows accurate solutions to be computed efficiently for reaction-dominant problems. However, such decoupling introduces splitting errors. To assess the size of splitting errors, a comparison is made between a MISDC method and a SISDC method that use the same advection,

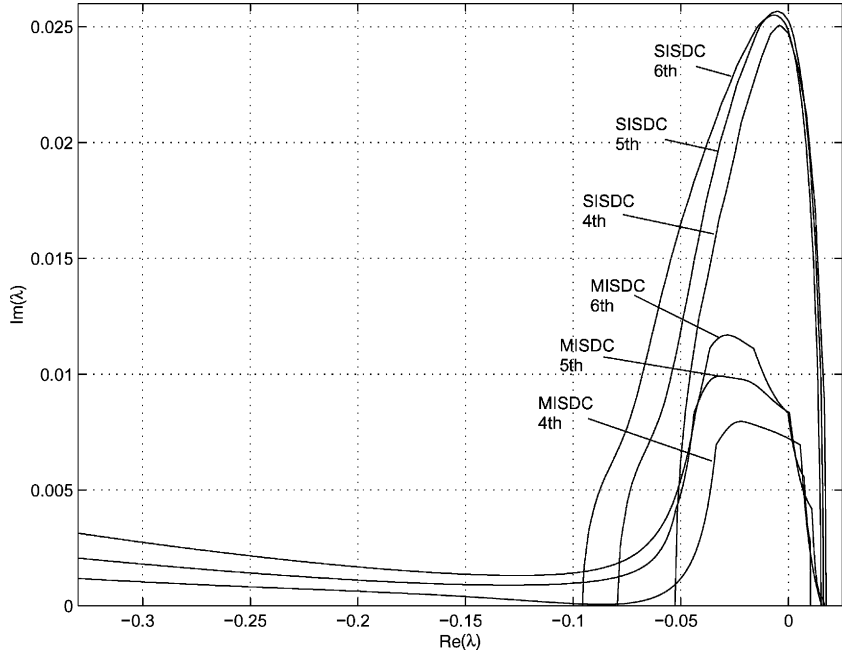

Fig. 5. Scaled accuracy regions for MISDC($K, K, 2, 2$) and SISDC(K, K) methods for $\epsilon = 10^{-4}$ and $K = 4$, 5, and 6.

diffusion, and reaction time steps. Thus, the only difference between the two methods is that, for the SISDC method, the diffusion and reaction processes are integrated together, whereas for the MISDC method, these two processes are integrated sequentially, resulting in splitting errors.

Fig. 6 shows the scaled accuracy regions of the MISDC$(4,4,1,1)$ and SISDC$(4,4)$ methods. As pointed out before, the advection, diffusion, and reaction time steps used in the MISDC$(4,4,1,1)$ method are the same as those in the SISDC$(4,4)$ method, but the approximations computed by these two methods differ because of the splitting errors introduced by the decoupling of the diffusion and reaction processes in the MISDC$(4,4,1,1)$ method. For Eq. (42), the operators associated with the advection, diffusion, and reaction terms commute. Therefore, if the decoupled equations (14)–(16) were to be integrated exactly, the resulting approximation should contain no splitting error and the accuracy regions for the MISDC$(4,4,1,1)$ and SISDC$(4,4)$ methods should coincide. However, these equations are approximated using a first-order SI method, hence splitting errors persist in the MISDC approximation and the resulting accuracy region is smaller than that of the SISDC region for most values of λ. The accuracy region of the MISDC$(4,4,1,1)$ does not have a tail like those in Fig. 5 because the time steps used for the diffusion and reaction processes are not refined, unlike the methods considered in Fig. 5.

To further demonstrate the relationship between accuracy and reaction time step size, accuracy regions are computed for the MISDC(K,K,N_D,N_R) methods for fixed K and N_D and for different N_R's. Fig. 7 shows the scaled accuracy regions of the MISDC$(4,4,2,N_R)$ methods, for $N_R = 1,2,\ldots,5$. In the advection-dominant regime, MISDC methods that use larger reactive time steps (i.e., smaller N_R) are more efficient. Nevertheless, as one moves toward the diffusion–reaction-dominant regime (i.e., larger $|\mathrm{Re}(\lambda)|$), MISDC methods using smaller reactive time steps (i.e., larger N_R) become more efficient. It is noteworthy that for $\mathrm{Re}(\lambda) \approx -10$, $u(1) \approx 10^{-4}$, so the error tolerance $\epsilon = 10^{-4}$ in this case is of the same order as the solution. The efficiency of MISDC methods in this stiff diffusion and/or reaction regime is re-examined in the next section using nonlinear problems.

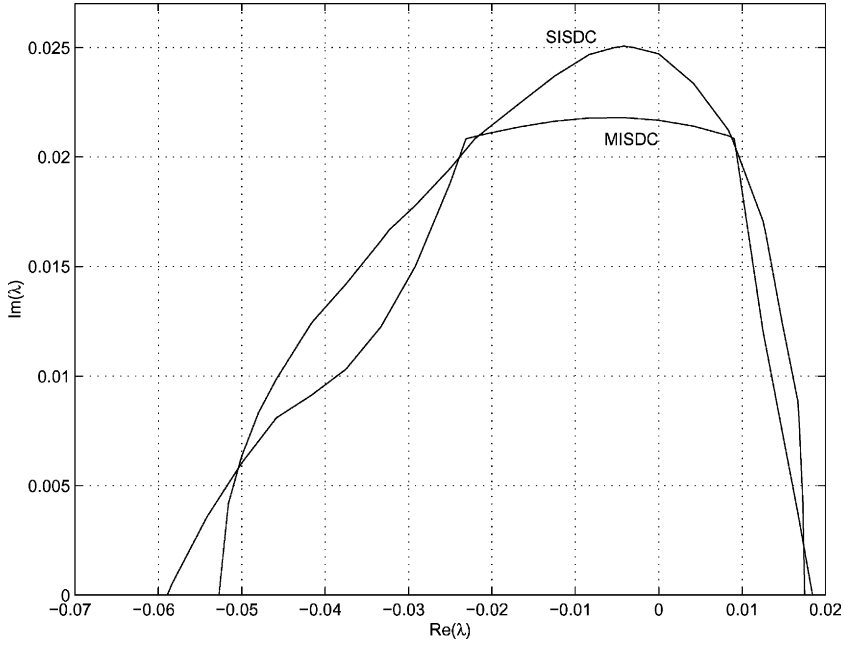

Fig. 6. Scaled accuracy regions for MISDC$(4,4,1,1)$ and SISDC$(4,4)$ methods for $\epsilon = 10^{-4}$.

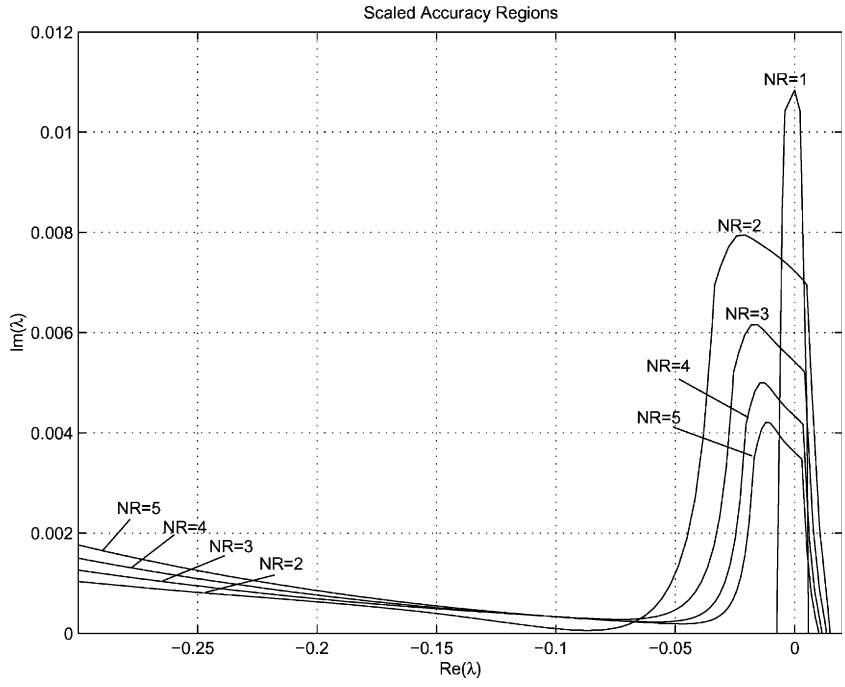

Fig. 7. Scaled accuracy regions for MISDC$(4,4,2,N_R)$ methods for $\epsilon = 10^{-4}$ and $N_R = 1,2,\ldots,5$.

5. Numerical examples

In this section, the accuracy, efficiency, and convergence behavior of MISDC methods is studied using the one-dimensional Burgers' equation, with a reaction term, and a coupled A–D–R system describing a simple, one-dimensional model of flamelets. 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 MISDC methods.

5.1. Burgers' equation with viscosity and reaction

In the first example, the Burgers' equation with reaction is used

$$u_t + uu_x = \frac{\delta(1-\gamma)}{2}u_{xx} + \frac{2(2\gamma-1)}{\delta}u(u-1)^2, \tag{44}$$

for $x \in [-2,2]$ and $t \in [0,0.5]$, with initial conditions

$$u(x,0) = \frac{1}{2} - \frac{1}{2}\tanh\left(\frac{x}{\delta}\right), \tag{45}$$

and Dirichlet boundary conditions $u(-2,t) = 1$ and $u(2,t) = 0$.

Given initial conditions (45) 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 (44) is given by

$$u(x,t) = \frac{1}{2} - \frac{1}{2}\tanh\left(\frac{x-\gamma t}{\delta}\right). \tag{46}$$

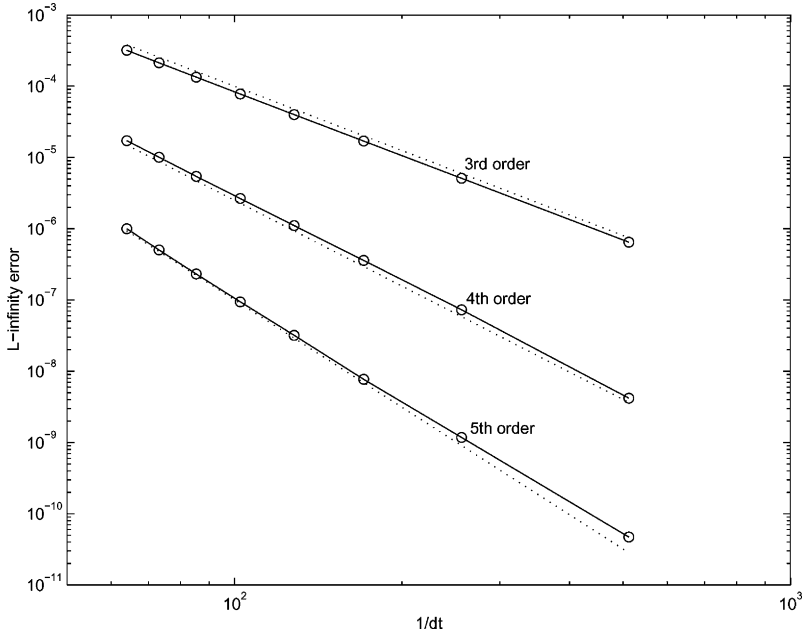

Fig. 8. Log–log plot of L_∞ error versus $1/\Delta t$. Solid lines are linear interpolants of numerical results, indicated by open circles; dotted lines indicate error lines with expected convergence rates.

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 indistinguishable from 1 and 0 at the left and right boundary points ($x = -2$ and 2), respectively, within double-precision machine accuracy. Thus, (46) 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.

In a spatial discretization of (44) using finite differences, the diffusion term gives rise to a banded matrix. Thus, the integration of the diffusion process using (32) and (38) requires the solution of a linear system, which is done in $\mathcal{O}(N)$ time, where N denotes the number of spatial subintervals. The nonlinear equations (33) and (39) associated with the reaction term are solved using Newton's method. In the convergence tests reported below, a spatial grid of $N = 1024$ is used. The time steps are chosen to be $\Delta t = r\Delta x$, where $r = 8, 7, \ldots, 1$. In this example, the diffusion and reaction coefficients are 0.00625 and 20, respectively. The Courant number, which gives a measure of the stiffness of the advection term, is given by $\gamma\Delta t/\Delta x = 0.75r$; the non-dimensional diffusion coefficient, which gives a measure of the diffusive stiffness, is given by $(\delta(1-\gamma)/2)(\Delta t/\Delta x^2) = 0.8r$.

Fig. 8 shows L_∞ errors computed using the MISDC$(K, K, 2, 2)$ methods for $K = 3$, 4, and 5. The errors are computed using the reference solution (46). In all cases, the approximations converge at the expected rate K.

5.2. A simple model of flamelets

The second example focuses on two coupled one-dimensional A–D–R equations, similar in form to those used in models of flamelets [4]. Let u and v represent the fuel and oxidizer mass fractions, respectively. The equations that describe the evolution of u and v are given by

$$\frac{\partial u}{\partial t} + w\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2} - Duv, \tag{47}$$

$$\frac{\partial v}{\partial t} + w\frac{\partial v}{\partial x} = \nu\frac{\partial^2 v}{\partial x^2} - Duv, \tag{48}$$

where w is the advection rate and D is the reaction coefficient. The reaction terms represent the one-step irreversible reaction $u + v \to p$, where p is the mass fraction of the product. As is done in [4], the passive scalar $z \equiv (u - v)/2$, which satisfies the simpler advection–diffusion equation

$$\frac{\partial z}{\partial t} + w\frac{\partial z}{\partial x} = \nu\frac{\partial^2 z}{\partial x^2} \tag{49}$$

is introduced. Eq. (49) can be integrated in conjunction with the modified equation for the fuel u:

$$\frac{\partial u}{\partial t} + w\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2} - Du(u - 2z). \tag{50}$$

The convergence behavior of MISDC methods as applied to this model is first studied. Eqs. (49) and (50) are integrated for $x \in [-1, 1]$ and $t \in [0, 0.5]$. Dirichlet boundary conditions are assumed: $z(-1, t) = -0.5$, $z(1, t) = 0.5$, $u(-1, t) = 1$, $u(1, t) = 0$, $v(-1, t) = 0$, $v(1, t) = 1$. The advection velocity is set to be $w(x, t) = -0.5x(1 + 5\cos(8\pi t))$, and the diffusion and reaction coefficients ν and D are chosen to be 0.01 and 500, respectively. To generate the initial conditions, (49) and (50) are integrated by means of the MISDC(5, 5, 2, 6) method for $t \in [0, 0.1]$ using initial conditions $z(x, 0) = 0.5\,\mathrm{erf}(x/\sqrt{2\nu})$, $u(x, 0) = z(x, 0) + |z(x, 0)|$, and $v(x, 0) = -z(x, 0) + |z(x, 0)|$. ($u(x, 0)$ and $v(x, 0)$ given by these initial conditions have discontinuous first derivatives at $x = 0$, which may lead to large spatial errors.) The solution is then used as initial conditions for all other simulations.

An analysis of the relevant time scales can be performed by focusing first on the steady solution corresponding to the time-average value $w_{\mathrm{ave}}(x, t) = -0.5x$. With the specific values for D and ν above, the reaction zone is very thin and concentrated in the mixing region around $x = 0$, where the maximum reaction rate occurs. Defining the flame thickness as the layer where 99% of the reaction takes place, numerical results for the average velocity field with the data above correspond to a flame thickness of 0.131. For the case with the time-modulated velocity field, one expects a slowly modulated correction to that average response. A spatial resolution of $N = 1024$ is selected, which corresponds approximately to 50 points across the average reactive layer. Using the corresponding mesh size Δx as a reference length scale, the following relevant time scales of the problem can be identified (once the relaxation to a periodic solution is complete):

- the largest time scale corresponds to the forcing period from the advection term: $t_{\mathrm{forcing}} = 1/4$;
- the characteristic diffusion time scale based on Δx is given by $t_{\mathrm{diff}} = \Delta x^2/\nu = 3.82\mathrm{e}-4$. A detailed asymptotic analysis of the significant processes in the thin reactive layer indicates that this time scale must roughly correspond to the reaction time scale as well, as diffusion and reaction are the only relevant mechanisms in the thin layer;
- there is also an intermediate time scale corresponding to the transport by advection on the spatial scale of the flame thickness $t_{\mathrm{adv}} = \Delta x/0.5 = 3.91\mathrm{e}-3$.

In the numerical simulation, the time steps are chosen to be $2^{-r}\Delta x$, where $r = 1, 2, \ldots, 5$, as motivated by the advection time scale. This choice of time step satisfies the stability condition for an explicit advection time-stepping scheme, but requires the use of implicit schemes for the faster processes, in particular, for diffusion with the non-dimensional diffusion coefficient given by $\nu\Delta t/\Delta x^2 = 5.12 \times 2^{-r}$.

The nonlinear equations associated with the reaction term are solved using Newton's method. The reaction equation for this system is in fact quadratic; therefore, it is possible to solve the reaction equations

analytically. Nevertheless, a Newton solver is used here to realistically represent the general case in which analytic solutions are not known. Because the number of reaction substeps, rather than CPU time, is chosen to be the measure of computational work associated with the reaction integration steps, the use of an iterative solver does not affect the efficiency measures.

Because no analytic solution is known for the system, the numerical solution computed using the MISDC$(5, 5, 2, 6)$ method with $\Delta t = 2^{-6}\Delta x$ is used as the reference solution. Fig. 9 exhibits the reference solution u for $x \in [-0.5, 0.5]$ as a function of time; for $x \in [-1, -0.5)$, $u \approx 1$, and for $x \in (0.5, 1]$, $u \approx 0$. Fig. 10 shows the L_∞ errors for u computed using the MISDC$(K, K, 2, 6)$ methods for $K = 3$, 4, and 5. These results indicate that numerical solutions obtained using MISDC methods are converging at the expected rates. Similar results are also obtained for z and v (not shown).

In the next set of experiments, the efficiency of MISDC methods with varying relative sizes of reaction time steps is compared. Figs. 11 and 12 show L_∞ errors for u, computed using the MISDC$(3, 3, 2, N_R)$ methods for $N_R = 2$, 4, and 6, versus total numbers of implicit reaction and diffusion equations solved, respectively. The three curves in Fig. 11 lie in close proximity to each other, which implies that for the same number of reaction integration substeps, the three MISDC methods generate solutions of similar accuracy. In contrast, Fig. 12 shows that for the same number of diffusion substeps, the MISDC$(3, 3, 2, 6)$ method, which takes the smallest reaction time step, generates the most accurate solutions, whereas the MISDC$(3, 3, 2, 2)$ method generates the least accurate solution. These results suggest that for problems with stiff reactions, provided that the integration of the diffusion process is expensive compared to the advection and reaction processes, then MISDC methods that take smaller reaction time steps are more efficient than those using larger reaction time steps. (Indeed, this model, which gives rise to quadratic reaction equations, meets this criterion.) On the other hand, if the integration of the reaction process is expensive compared to the solution of the diffusion equation, then methods with $N_R = 1$ are more efficient. This is further discussed in the next section.

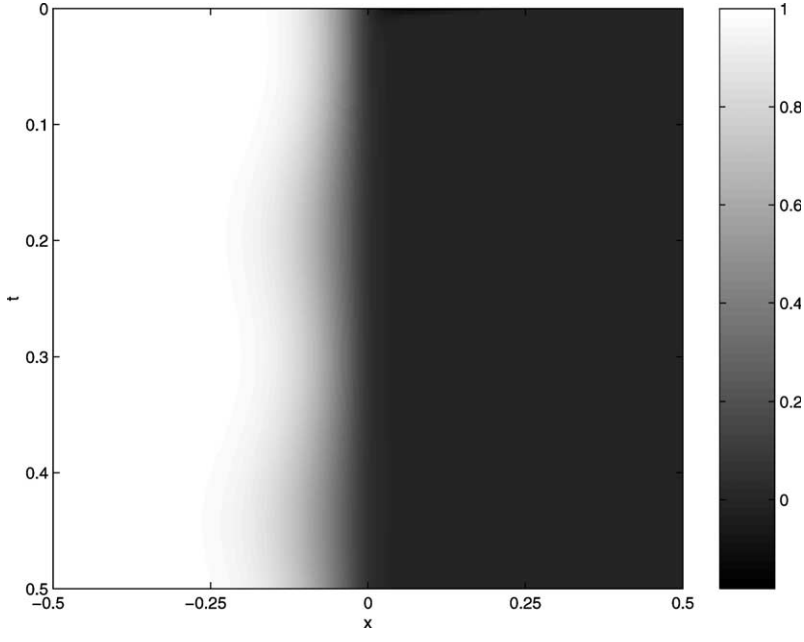

Fig. 9. Reference solution for u as a function of time.

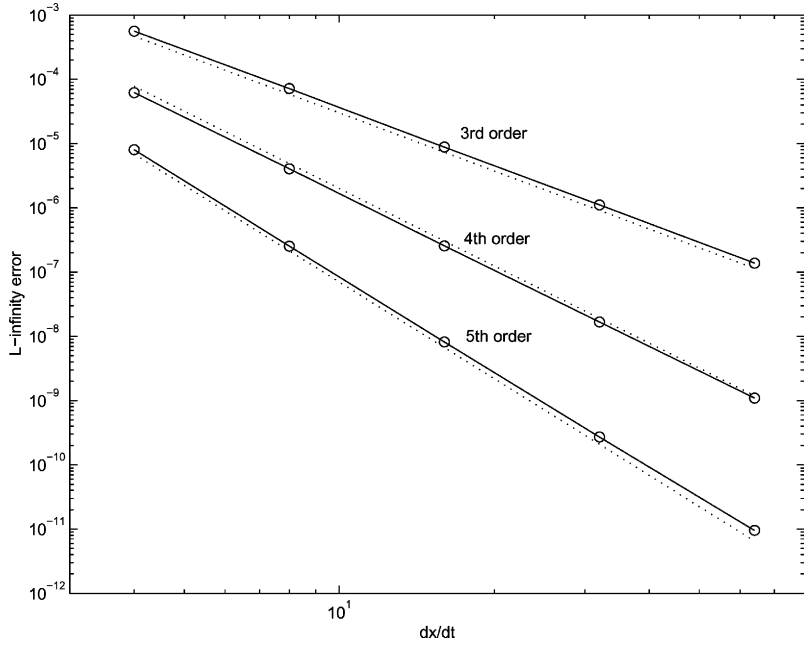

Fig. 10. Log–log plot of L_∞ error versus $1/\Delta x$ for u. Notations are the same as for Fig. 8.

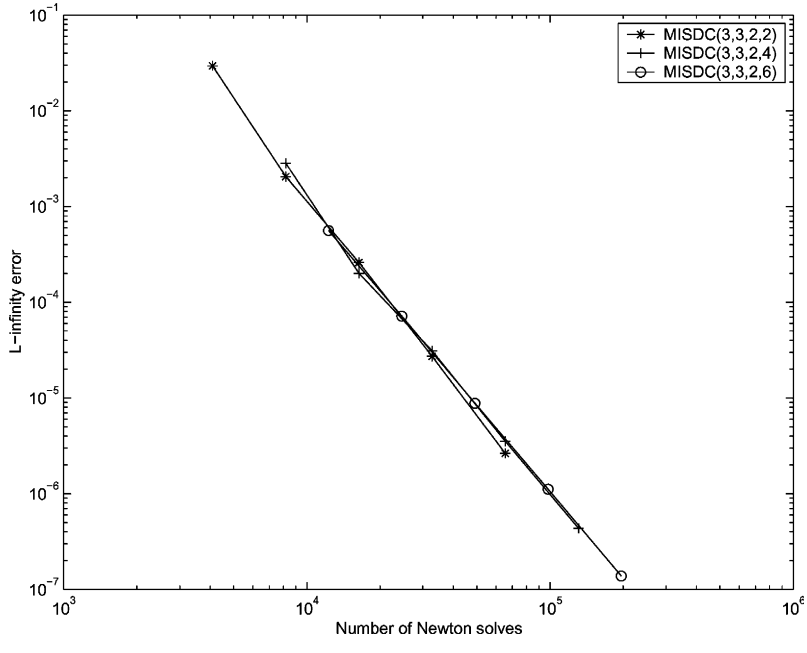

Fig. 11. Log–log plot of L_∞ error versus number of Newton solves for u. Notations are the same as for Fig. 8.

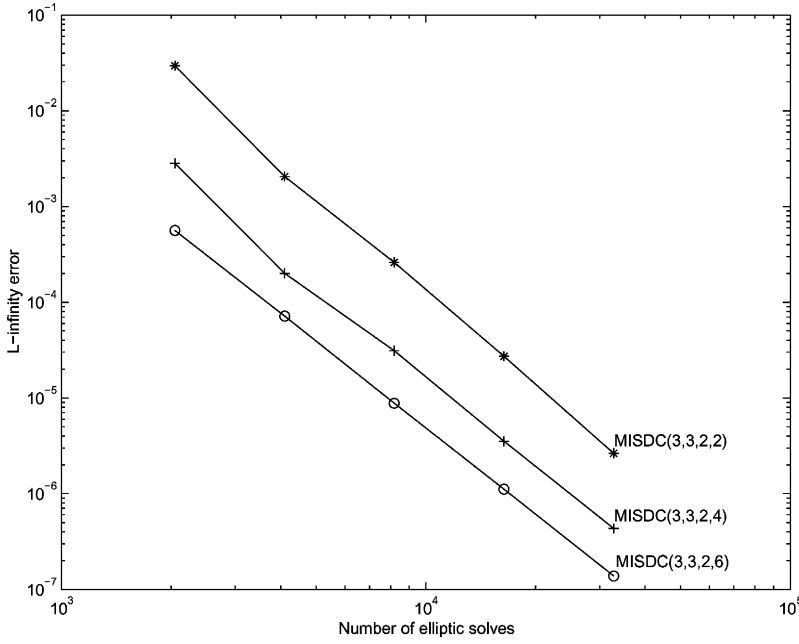

Fig. 12. Log–log plot of L_∞ error versus number of elliptic solves for u. Notations are the same as for Fig. 8.

5.3. A comparison with semi-implicit Runge–Kutta methods

In the last set of numerical experiments, the efficiency of MISDC methods is compared to SI Additive Runge–Kutta (ARK) methods from [14]. Of course, any such comparison very much depends on the selected test cases, because the efficiency of a numerical method for an A–D–R equation depends on the relative stiffness of the individual terms and the relative cost of the individual implicit or coupled implicit equations. These costs further depend on the spatial discretization, the dimension of the problem, and the degree of parallelism in the implementation. Therefore, the purpose of this comparison is only to illustrate that MISDC methods compare favorably with ARK methods on a model A–D–R equation, and not to determine which method is more efficient in general.

For this comparison, the flamelet model problem of the previous section is used. The initial conditions are $u(x,0) = 0.5(1+\cos(2\pi x))$ and $v(x,0) = 0.9u(x,0)$; the advection velocity is set to be $w(x,t) = 0.5 + \cos(2\pi x)$, and the diffusion and reaction coefficients ν and D are chosen to be 0.25 and 10,000, respectively. For the ARK methods, the diffusion and reaction terms are coupled and integrated with the same time step, while in the MISDC method, all terms are decoupled. It should be noted that in [14], the ARK methods are applied to reacting flow problems in which the magnitude of the diffusion is sufficiently small that it can be treated explicitly with the advection. The nonlinear implicit equations for the ARK methods are solved by means of Newton's method using the previous stage answer as an initial guess. The size of the time step is hence set by numerically determining the maximum time step which produces convergence of the Newton iteration during the initial time step.

Fig. 13 displays the L_∞ errors of the solutions computed using the MISDC$(4,4,2,6)$, MISDC$(5,5,2,5)$, ARK4(3)6L[2]SA (hereafter ARK4), and ARK5(4)8LSA (hereafter ARK5) methods for $N = 32$, 64, 128, 256, and 512. The errors are plotted against total computational cost which is measured by the number of floating-point operations (flops) using MATLAB. The numerical solution computed using the ARK5 method with $N = 1024$ and $\Delta t = 1/40960$ is used as the reference solution

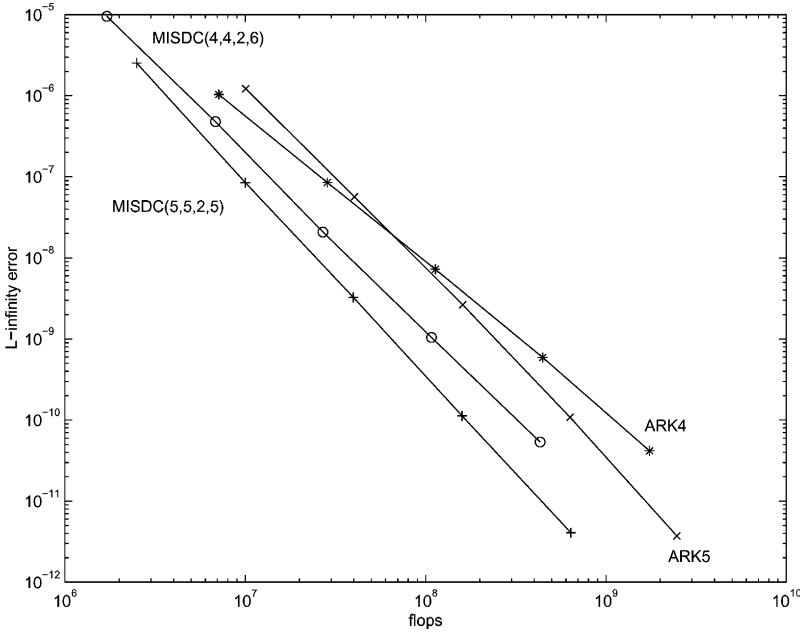

Fig. 13. Log–log plot of L_∞ error versus number of flops for u, computed using the MISDC$(4,4,2,6)$, MISDC$(5,5,2,5)$, ARK4, and ARK5 methods.

for approximating the errors. The time step for the MISDC methods is set to be $\Delta t = \Delta x$, whereas a much smaller time step of $\Delta t = \Delta x/40$ is used for the ARK4 and ARK5 methods to ensure convergence of the Newton iterations. With these choices, the reaction time steps for the four methods are approximately the same, specifically $\Delta x/36$ and $\Delta x/40$ for the MISDC$(4,4,2,6)$ and MISDC$(5,5,2,5)$, respectively, and $\Delta x/40$ for the ARK methods.

For the same spatial resolution, the approximations computed using the ARK methods are more accurate than the MISDC methods of the same order. This is not surprising since each term in the A–D–R equation is evaluated at the reaction time scale. However, in this example, the ARK methods are also more expensive owing to the fact that the implicit equations for u (but not for z) involve both non-local diffusion terms and nonlinear reaction terms and are thus made expensive to solve. In contrast, MISDC methods require the solution of a local implicit reaction equation and a linear implicit diffusion equation (which is also solved 6 and 5 fewer times than in the ARK method). The numerical results shown in Fig. 13 suggest that for problems in which both the diffusion and the reaction terms are stiff, MISDC methods can be constructed which compare favorably with SI Runge–Kutta methods.

6. Discussion

In this report, MISDC methods for the temporal integration of ODEs with multiple time scales have been presented. When applied to A–D–R equations, MISDC methods integrate the advection term explicitly, and the diffusion and reaction terms implicitly, independently, and possibly with different time steps. Unlike standard OS methods, MISDC methods can easily be constructed to generate numerical solutions with arbitrarily high-order of temporal accuracy. This is achieved by simultaneously reducing integration and splitting errors in the deferred correction iterations.

The potential advantage of MISDC methods for real-world problems in the physical and biological sciences is certainly problem dependent. In general, higher-order methods are only more efficient than lower-order methods when the solution is sufficiently smooth in time, when a high level of accuracy is desired, and/or when the temporal integration interval is very long. For A–D–R equations, MISDC methods offer the most possible benefit for applications in which the stiffness of the diffusion and reaction operators differs widely and in which the decoupled diffusion and reaction problems is much simpler than solving the coupled diffusion-reaction equation. For many problems, the reaction term is nonlinear but spatially local, whereas the implicit diffusion equation is global. In this case, there is a great benefit in solving the reaction equations independently at each node of the computational grid, especially in terms of performance on parallel computers.

In some applications, the reaction terms are computed using chemistry packages, which makes solving coupled diffusion-reaction equations even more difficult. MISDC schemes provide a systematic procedure to incorporate such black-box solvers into a more complex time integration scheme and to still achieve high-order accuracy. In such cases, however, it is not clear whether one may significantly improve the efficiency of a given MISDC method by only reducing reaction time steps due to the relatively high expense of solving the reaction equation. In fact, because the cost of using such black-box solver at each time step might be high, MISDC methods with $N_R = 1$ (i.e., equal reaction and diffusion time step size) are likely to be more efficient for such applications.

In MISDC methods, the correction term associated with each process can be used to dynamically determine the appropriate time-step size for each process to meet certain accuracy requirements. Thus, MISDC methods are (as are other SDC methods [11]) suitable candidates for adaptive time-marching. An adaptive formulation of MISDC methods is currently being developed by the authors.

In the derivation of the MISDC methods, it is assumed that the reactive time scale is much smaller than that of diffusion and advection; thus, the reactive time step is chosen to be the smallest. However, in applications with large diffusivities or solutions with large spatial variations, or when spatial resolution is sufficiently high, the diffusive stiffness may exceed that of reaction. For such problems, the diffusive time step should be the smallest, and the ordering of the diffusion and reaction updates in the algorithm described in Section 3 should be reversed. Indeed, since the time scales of the processes may vary over time, an adaptive formulation of the methods can be developed in which the ordering of the updates is dynamically determined.

Because the focus of this study is on time discretization accuracy, the A–D–R problems considered are in one space dimension. Because MISDC methods are constructed in the context of MOL, the extension to more space dimensions is straightforward when spatial discretization is done using finite differencing—spatial derivatives are approximated using finite differencing, resulting in a system of ODEs, which can then be integrated using MISDC methods. If finite-volume methods are used in space (e.g., for applications with sharp moving fronts in solutions), extension to more space dimensions can be done by means of dimensional-splitting [6,8].

In the current implementation, spatial discretization is done using sixth-order centered differencing, which assumes that the solution is smooth. In many examples of interest, however, solutions may have sharp gradients, which may be accurately resolved by centered differencing by adopting a sufficiently refined spatial grid or adaptive mesh refinement. An alternative approach for capturing the advection term is to compute cell-averaged solution of the model equations using a conservative method. LeVeque and Yee [9] compared two approaches for solving a model advection equation with a stiff source term, specifically, predictor–corrector methods with flux limiters and splitting methods. They found that splitting methods generally perform better, except for stiff problems, in which a numerical phenomenon of incorrect propagation speeds of discontinuities was observed. A conservative formulation of the MISDC methods, which incorporates a method developed for hyperbolic conservation laws [5], has been developed [7] and will be compared to the approaches studied in [9]. To be useful for real-world problems, extensions to problems

with Arrhenius kinetics [17], which gives rise to more realistic temperature-dependent reaction rates, and with nonlinear diffusion coefficients [13] will also be developed.

Acknowledgements

This work 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.

References

[1] S. Abarbanel, D. Gottlieb, 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.
[2] U.M. Ascher, L.R. Petzold, Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations, SIAM, Philadelphia, PA, 2000.
[3] U.M. Ascher, S.J. Ruuth, R.J. Spiter, Implicit–explicit Runge–Kutta methods for time-dependent partial differential equations, Appl. Numer. Math. 25 (2–3) (1997) 151–167.
[4] A. Bourlioux, A.J. Majda, An elementary model for the validation of flamelet approximations in non-premixed turbulent combustion, Combust. Theor. Model. 4 (4) (2000) 189–210.
[5] P. Colella, P.R. Woodward, The piecewise parabolic method (PPM) for gas-dynamical simulations, J. Comput. Phys. 54 (1984) 174–201.
[6] P. Colella, Multidimensional upwind methods for hyperbolic conservation laws, J. Comput. Phys. 87 (1990) 171–200.
[7] A.T. Layton, M.L. Minion, Conservative multi-implicit spectral deferred correction methods for reacting gas dynamics, J. Comput. Phys. (2003) (submitted).
[8] R.J. LeVeque, K.-M. Shyue, Two-dimensional front tracking based on high resolution wave propagation methods, J. Comput. Phys. 123 (1996) 354–368.
[9] R.J. LeVeque, H.C. Yee, A study of numerical methods for hyperbolic conservation laws with stiff source terms, J. Comput. Phys. 86 (1990) 187–210.
[10] M.H. Carpenter, D. Gottlieb, S. Abarbanel, W.-S. Don, The theoretical accuracy of Runge–Kutta time discretization for the initial boundary value problem: a study of the boundary error, SIAM J. Sci. Comput. 16 (1995) 1241–1252.
[11] A. Dutt, L. Greengard, V. Rokhlin, Spectral deferred correction methods for ordinary differential equations, BIT 40 (2) (2000) 241–266.
[12] E. Hairer, G. Wanner, Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic Problems, Springer, Berlin, 1991.
[13] J.F. Kanney, C.T. Miller, D.A. Barry, Comparison of fully coupled approaches for approximating nonlinear transport and reaction problems, Adv. Water Resour. 26 (2003) 353–372.
[14] C.A. Kennedy, M.H. Carpenter, Additive Runge–Kutta schemes for convection–diffusion–reaction equations, Appl. Numer. Math. 44 (1–2) (2002) 139–181.
[15] A. Koziol, J. Pudykiewicz, Global-scale environmental transport of persistent organic pollutants, Chemosphere 45 (8) (2001) 1181–1200.
[16] D. Lanser, J.G. Verwer, Analysis of operator splitting for advection–diffusion–reaction problems from air pollution modelling, J. Comput. Appl. Math. 111 (1–2) (1999) 201–216.
[17] A. Linan, F.A. Williams, Fundamental Aspects of Combustion, Oxford Press, New York, 1993.
[18] C.T. Miller, G. Christakos, P.T. Imhoff, J.F. McBride, J.A. Pedit, J.A. Trangenstein, Multiphase flow and transport modeling in heterogeneous porous media: challenges and approaches, Adv. Water Resour. 21 (2) (1998) 77–120.
[19] M.L. Minion, Semi-implicit spectral deferred correction methods for ordinary differential equations. Commun. Math. Sci. (2002) (submitted).
[20] H.N. Najm, P.S. Wyckoff, O.M. Knio, A semi-implicit numerical scheme for reacting flow. I. Stiff chemistry, J. Comput. Phys. 143 (1998) 381–402.
[21] R.D. Richtmyer, K.W. Morton, Difference Methods for Initial-value Problems, second ed., Wiley–Interscience, New York, 1967.
[22] B. Sportisse, An analysis of operating splitting techniques in the stiff case, J. Comput. Phys. 161 (2000) 140–168.
[23] G. Strang, On the construction and comparison of difference schemes, SIAM J. Numer. Anal. 8 (3) (1968) 506–517.
[24] P. Sun, A pseudo-non-time-splitting method in air quality modeling, J. Comput. Phys. 127 (1996) 152–157.

[25] M. Valorani, D.A. Goussis, Explicit time-scale splitting algorithm for stiff problems: auto-ignition of gaseous mixtures behind a steady shock, J. Comput. Phys. 169 (2001) 44–79.
[26] J.G. Verwer, J.G. Blom, W. Hundsdorfer, An implicit explicit approach for atmospheric transport-chemistry problems, Appl. Numer. Math. 20 (1–2) (1996) 191–209.
[27] A.L. Walter, E.O. Frind, D.W. Blowes, C.J. Ptacek, J.W. Molson, Modeling of multicomponent reactive transport in groundwater. 1. Modeling development and evaluation, Water Resour. Res. 30 (11) (1994) 3137–3148.
[28] N.N. Yanenko, The Method of Fractional Steps, Springer, New York, 1971.
[29] G.T. Yeh, V.S. Tripathi, A critical evaluation of recent developments in hydrogeochemical transport models of reactive multichemical components, Water Resour. Res. 25 (1) (1989) 93–108.