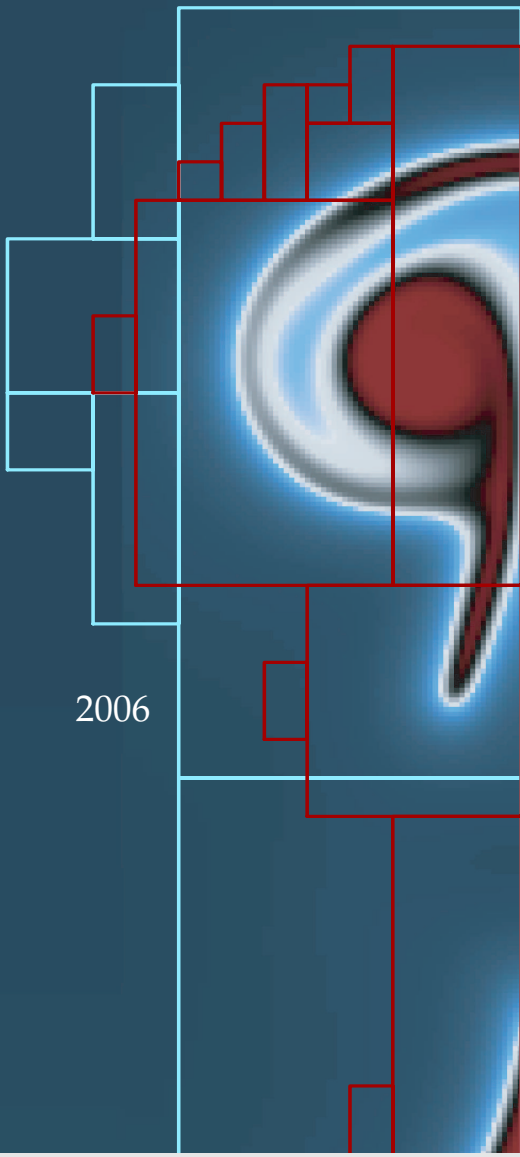
Communications in Applied Mathematics and Computational Science

Volume 1 No. 1 2006



ON THE SPECTRAL DEFERRED CORRECTION OF SPLITTING METHODS FOR INITIAL VALUE PROBLEMS

THOMAS HAGSTROM AND RUHAI ZHOU

mathematical sciences publishers

COMM. APP. MATH. AND COMP. SCI.
Vol. 1, No. 1, 2006

ON THE SPECTRAL DEFERRED CORRECTION OF SPLITTING METHODS FOR INITIAL VALUE PROBLEMS

THOMAS HAGSTROM AND RUHAI ZHOU

Spectral deferred correction is a flexible technique for constructing high-order, stiffly-stable time integrators using a low order method as a base scheme. Here we examine their use in conjunction with splitting methods to solve initial-boundary value problems for partial differential equations. We exploit their close connection with implicit Runge–Kutta methods to prove that up to the full accuracy of the underlying quadrature rule is attainable. We also examine experimentally the stability properties of the methods for various splittings of advection-diffusion and reaction-diffusion equations.

1. Introduction

Dutt et al. [7] have introduced a method of spectral deferred correction which allows one to automatically increase the accuracy of a low order time-stepping method. Defect and/or deferred correction methods for initial value problems have been known for some time [23; 9]. The main innovation in [7] is the use of spectral integration on Gaussian quadrature nodes to construct the corrections. This avoids instabilities and conditioning problems associated with repeated differentiations. They show that if forward or backward Euler methods are used as the base scheme (2–5), stable and stiffly-stable methods of very high order result. More recently, Auzinger et al. [3; 2] have analyzed similar algorithms and suggested various improvements.

Our interest here is in the use of SDC methods in conjunction with operator and/or dimensional splitting to solve initial-boundary value problems for partial differential equations. In a series of papers [21; 6; 19] Minion et al. have explored the use splitting methods as the base scheme in an SDC approach. However, their implementations have mainly been designed to achieve an order of accuracy

MSC2000: 65L06, 65M20.

Keywords: splitting methods, deferred correction, stability regions.

Hagstrom was supported in part by NASA Contract NRA-96-LeRC-2, NSF Grants DMS-9600146, DMS-9971772, DMS-0306285, Lawrence Livermore National Laboratory Subcontract B547968, and by ICOMP, NASA Glenn Research Center. Any conclusions or recommendations expressed in this paper are those of the author and do not necessarily reflect the views of NSF, LLNL, or NASA.

Zhou was supported in part by NASA Grant NRA-96-LeRC-2 and NSF Grant DMS-0308019.

approximately equal to the number of time levels stored. (For an exception see [17].) In our approach we exploit the close connection of SDC methods with implicit Runge-Kutta methods to prove that up to double this accuracy is attainable, though only for the approximate solutions at the boundaries of each correction interval. (See also [3].) Again this result is expected from the perspective of the Runge-Kutta methods as the approximate solutions on the interior nodes correspond to the Runge-Kutta stage variables. We believe this possibility for enhanced accuracy is of importance for large, memory-bound applications; our experiments indicate that the efficiency of the proposed higher-order methods is essentially the same as for the methods proposed in the above-cited works.

We also examine the stability properties of the methods for various special cases. We begin by constructing stability domains for the standard model of operator splitting applied to advection-diffusion problems (e.g., [1]). We have considered all of the typical quadratures (Gauss–Legendre, Gauss–Lobatto, and Gauss–Radau) and a variety of starting methods (2–4). We also compare consistent and inconsistent correction methods (2–5). Second, we consider what we call preconditoned splitting methods for both linear and nonlinear problems. We have used such techniques to develop fourth and higher order solvers for complex models of reacting gases [12; 11; 24]. We note that Layton and Minion [20] have carried out an extensive stability study for SDC applied to splitting methods. We will compare our results to theirs, in essence assessing the effect of our additional corrections on the stability domains.

Finally, we verify the properties of the methods in nonlinear settings through experiments with simpler reaction-diffusion and advection-diffusion problems, focusing on the requirements on the preconditioner to obtain good accuracy and stability. (A similar study for lower order splitting methods is presented in [22].) Given the difficulties in fully analyzing splitting methods for complex problems, such studies seem necessary to validate any proposed methods.

2. Spectral deferred correction with splitting

We consider the initial-value problem:

$$\frac{du}{dt} = F(u,t), \quad u(t_0) = u_0, \quad u, F \in \mathbb{R}^\kappa \tag{2–1}$$

and recall the well-known fact that given $t_0 = T_0 < T_1 < \cdots$ (2–1) can be reformulated as a sequence of integral equations:

$$u(t) = u(T_j) + \int_{T_j}^{t} F(u(\tau),\tau)d\tau, \quad t \in [T_j, T_{j+1}]. \tag{2–2}$$

Our formulation of spectral deferred correction of some splitting method for approximating (2–1) has three essentially independent ingredients. First we introduce two splittings of F:

$$F = \tilde{F}_I + \tilde{F}_E, \quad F = F_I + F_E, \tag{2–3}$$

and two associated time-stepping formulas; a pth order multistep method (e.g., an IMEX method [1]) which we call the *starting method*:

$$\sum_{j=-1}^{k-1} \alpha_j v(t_{n-j}) = h_n \sum_{j=-1}^{k-1} \beta_j^{(n)} \tilde{F}_I(v(t_{n-j}), t_{n-j}) + h_n \sum_{j=0}^{k-1} \tilde{\beta}_j^{(n)} \tilde{F}_E(v(t_{n-j}), t_{n-j}), \tag{2–4}$$

and the first order method,

$$v(t_{n+1}) = v(t_n) + h_n F_I(v(t_{n+1}), t_{n+1}) + h_n F_E(v(t_n), t_n), \tag{2–5}$$

which we call the *correction method*. Second, we introduce, as in [7], a collocation method for approximating (2–2). Setting $\Delta T = T_{j+1} - T_j$ we introduce m nodes:

$$t_{jk} = T_j + c_k \Delta T, \quad 0 \le c_1 < c_2 < \cdots < c_m \le 1. \tag{2–6}$$

A solution of the polynomial collocation approximation defined by these nodes is a set of vectors v_{jk} satisfying:

$$v_{jk} = v(T_j) + \int_{T_j}^{T_j + c_k \Delta T} \psi_j(t) dt, \tag{2–7}$$

$$= v(T_j) + \Delta T \sum_{\alpha=1}^{m} S_{k\alpha} F(v_{j\alpha}, T_j + c_\alpha \Delta T)), \tag{2–8}$$

where $\psi_j(t)$ is the unique degree-$(m-1)$ interpolant of the data

$$(T_j + c_k \Delta T, F(v_{jk}, T_j + c_k \Delta T)), \quad k = 1, \ldots, m.$$

Here, following [7], we note that the matrix S whose entries are $S_{k\alpha}$ is a well-conditioned $m \times m$ spectral integration matrix.

The evolution of the approximate solution from T_j to T_{j+1} now proceeds as follows:

i: Compute approximations, v_{jk}^0, using m steps of (2–4) with the appropriate reduced time steps,

$$h_k = (c_k - c_{k-1}) \Delta T, \quad c_0 = 0. \tag{2–9}$$

(Note: a multistep starting method may make use of data at points $t_{j-1,k}$.)

ii: Given our lth approximation, v^l_{jk}, we define residuals, r^l_{jk}, using the spectral integration matrix, S:

$$r^l_{jk} = v(T_j) + \Delta T \sum_{\alpha=1}^{m} S_{k\alpha} F(v^l_{j\alpha}, t_{j\alpha}) - v^l_{jk}, \tag{2–10}$$
$$= v(T_j) + \int_{T_j}^{T_j + c_k \Delta T} \psi^l_j(t)dt - v^l_{jk}.$$

Here $\psi^l_j(t)$ is the unique degree-$(m-1)$ interpolant of the data

$$(T_j + c_k \Delta T, F(v^l_{jk}, T_j + c_k \Delta T)), \quad k = 1, \ldots, m.$$

iii: With the residual in hand, (2–5) is used to update the approximation. The idea here is to write $v^{l+1} = v^l + \delta^l$ and note that the correction can be viewed as an approximate solution to the perturbed equation:

$$\frac{d\delta^l}{dt} = F(v^l + \delta^l, t) - F(v^l) + \frac{dr^l}{dt}, \quad \delta^l(T_j) = 0. \tag{2–11}$$

The most straightforward approach, used by Dutt et al. [7] and Minion [21], is to apply (2–5) directly to (2–11) to obtain the correction formula:

$$\begin{aligned} \delta^l_{jk} = \delta^l_{j,k-1} &+ r^l_{jk} - r^l_{j,k-1} \\ &+ h_k \left(F_I(v^l_{jk} + \delta^l_{jk}, t_{jk}) - F_I(v^l_{jk}, t_{jk}) \right) \\ &+ h_k \left(F_E(v^l_{j,k-1} + \delta^l_{j,k-1}, t_{j,k-1}) - F_E(v^l_{j,k-1}, t_{j,k-1}) \right), \end{aligned} \tag{2–12}$$

$$\delta^l_{j0} = 0, \tag{2–13}$$

$$v^{l+1}_{jk} = v^l_{jk} + \delta^l_{jk}. \tag{2–14}$$

iv: Stop the process after L steps and define the solution update by:

$$v(T_{j+1}) = v(T_j) + \int_{T_j}^{T_{j+1}} \psi^L_j(s)ds. \tag{2–15}$$

Obviously, the description above leaves room for a wide range of implementations, some of which we will discuss below.

Concerning the starting and correction methods, we note that in many cases it is possible to choose F_I and/or $\tilde{F}_I$ to be linear in v, in which case the methods are called linearly implicit. Also we assume that $p \leq m$ where m is the number of nodes in the quadrature formula underlying the correction process. Although we

will prove that the overall method typically attains an order $q > m$, our analysis indicates that there is no benefit to choosing $p > m$. Moreover, as multistep starting methods may use stage values, we are limited by the stage order, which is m. We emphasize that we could use the correction method as our starting method, and indeed that is what has been done in the references mentioned herein. Also, as the time steps will not be equally spaced, the coefficients in (2–4) will depend on n if a truly multistep method is used. (See, though, [3; 2] for a method which allows an equispaced temporal grid while maintaining the accuracy of the Gaussian quadrature rules.) Lastly, it is possible to replace (2–5) and/or (2–4) with a multisplitting or fractional step scheme as in [6], but for simplicity we will focus on the simpler splittings for our analysis here.

Concerning the nodes, Dutt et al. [7] take them to be the Gauss–Legendre nodes. Minion [21], on the other hand, suggests Gauss–Lobatto nodes, and we will also consider right-handed Gauss–Radau nodes. In [20] uniform nodes are shown to be feasible from the standpoint of stability, but their use would preclude the attainment of the higher order accuracy which is a focus of the current work.

Lastly we note that alternative correction formulas are also possible. The correction method we have employed in [12; 11; 24; 25] follows [9]:

$$\bar{v}^l_{jk} = \bar{v}^l_{j,k-1} + \delta t_k F_I(\bar{v}^l_{jk}, t_{jk}) + \delta t_k F_E(\bar{v}^l_{j,k-1}, t_{j,k-1}) + r^l_{j,k-1} - r^l_{jk}. \quad (2\text{–}16)$$

where

$$\bar{v}^l_{j0} = v(T_j). \quad (2\text{–}17)$$

Then set:

$$v^{l+1}_{jk} = v^0_{jk} + v^l_{jk} - \bar{v}^l_{jk}. \quad (2\text{–}18)$$

However, the theoretical results in this paper only apply in general to corrections based on (2–12)-(2–14). To apply them to corrections based on (2–16) we must make the starting method and the correction method coincide, which is the case in [12; 11; 24; 25]. Under those conditions, and for linear problems, the two correction methods are mathematically identical.

3. Order of accuracy

In [7; 21] the correction process is carried out until an error on the order of the truncation error in the approximations to (2–2) is attained. This leads to methods of order m. More general analyses of the convergence and accuracy of this process appear in recent manuscripts by Hansen and Strain [15; 16], where both single step and multistep correction formulas are considered. However, their approach does not take account of the full order of accuracy of the underlying quadrature rules which is our aim here. (We note that we could use SDC methods as analyzed in [15; 16] as our starting methods.)

In this work we focus on the classical Gauss-type quadrature methods which are of orders between $2m-2$ and $2m$. We prove here that if further corrections are made the order of accuracy of the underlying quadrature rule is in fact attainable, albeit only for the approximate solutions at the coarse grid points, T_j. (See also [3].) In practice, this allows the construction of higher order methods which are more efficient from the perspective of the number of time levels which must be stored. We also see that while the order of accuracy of the starting method affects the number of corrections needed, the accuracy of the correction method does not. This is in contrast with the results of [15; 16], which show that gains in accuracy commensurate with the order of the correction method are possible until an order m method is produced.

The accuracy result follows from the observation that if the residuals were zero, that is if the related collocation approximations to (2–2) were constructed, then the method would be equivalent to a standard implicit Runge–Kutta method (e.g., [13, Chapter II]), which has the accuracy we claim. Thus we need only show that similar conclusions follow from making these residuals sufficiently high order. We note that one could more directly use the size of the residual as a basis for terminating the correction process, as suggested in [17], but we do not consider that possibility here.

To study the local truncation error we assume $v(T_j) = u(T_j)$ and, for a multistep starting scheme, $v_{j-1,k} = u(t_{j-1,k})$. Set:

$$V^l(t) = u(T_j) + \int_{T_j}^{t} \psi_j^l(s)ds, \tag{3–1}$$

noting that $v(T_{j+1}) = V^L(T_{j+1})$. First we prove:

Lemma 3.1. *Suppose F is smooth and that the polynomial quadrature rule based on (2–6) has order q. Then there exists a constant, C, independent of ΔT, such that for a sufficiently smooth solution, u, and a sufficiently smooth solution of the SDC method, V^l,*

$$|u(T_{j+1}) - V^l(T_{j+1})| \leq C\Delta T \max_k |r_{jk}^l| + O(\Delta T^{q+1}).$$

Proof. Define the defect, $d(t)$, by:

$$d(t) = \frac{dV^l}{dt} - F(V^l(t), t) = \psi_j^l(t) - F(V^l(t), t). \tag{3–2}$$

Note that by (2–10):

$$V^l(t_{jk}) - v_{jk}^l = r_{jk}^l, \tag{3–3}$$

and by definition

$$\psi_j^l(t_{jk}) = F(v_{jk}^l, t_{jk}). \tag{3–4}$$

Thus by the Lipschitz continuity of F:

$$|d(t_{jk})| \leq K|r^l_{jk}|. \tag{3–5}$$

Let the matrix $\Phi(t, \tau, V^l(\tau))$ be defined by:

$$\Phi = D_{V^l(\tau)} w, \tag{3–6}$$

where for $t > \tau$:

$$\frac{dw}{dt} = F(w, t), \quad w(\tau) = V^l(\tau). \tag{3–7}$$

We note that standard results on the differentiability of solutions of ordinary differential equations with respect to their initial data imply that the derivatives of Φ with respect to τ can be bounded in terms of the derivatives of V^l which we have assumed (and will subsequently prove) to be bounded independent of l and ΔT. We then have the following nonlinear variation-of-constants formula, known as the Alekseev–Gröbner Lemma [13, Chapter I]:

$$V^l(T_{j+1}) - u(T_{j+1}) = \int_{T_j}^{T_{j+1}} \Phi(T_{j+1}, \tau, V^l(\tau)) d(\tau) d\tau. \tag{3–8}$$

Now replace the integral by the quadrature rule associated with the nodes. We have:

$$\begin{aligned} V^l(T_{j+1}) - u(T_{j+1}) = {} & \Delta T \sum_k \omega_k \Phi(T_{j+1}, t_{jk}, V^l(t_{jk})) \cdot d(t_{jk}) \\ & + \int_{T_j}^{T_{j+1}} \Phi(T_{j+1}, \tau, V^l(\tau)) d(\tau) d\tau \\ & - \Delta T \sum_k \omega_k \Phi(T_{j+1}, t_{jk}, V^l(t_{jk})) \cdot d(t_{jk}). \end{aligned} \tag{3–9}$$

Using (3–5) the first term is bounded by $C\Delta T \max |r^l_{jk}|$ while the difference of the second and third is $O(\Delta T^{q+1})$ by our assumption on the accuracy of the quadrature rule and the smoothness of V^l. This completes the proof of Lemma 3.1. □

To complete our analysis, we need only prove that the residual is reduced by the correction process and that the approximate degree-m polynomial solution, V^l, has derivatives bounded independent of the time step.

Lemma 3.2. *For a sufficiently smooth solution, u, and smooth functions, F, F_I, F_E, $\tilde{F}_I$, $\tilde{F}_E$, a starting method of order $p \leq m$, and corrections based on (2–12)-(2–14), there exist constants, C_l and $M_{r,l}$, independent of ΔT sufficiently small such that the residuals satisfy*:

$$\max_k |r^l_{jk}| \leq C_l \Delta T^{p+1+l}, \tag{3–10}$$

and, for $l \geq m-p-2$ *and* $0 \leq r \leq m$:

$$\max_{t\in[T_j,T_{j+1}]} \left| \frac{d^r V^l}{dt^r} \right| \leq M_{r,l}. \tag{3–11}$$

Proof. Denote by $\tilde{V}(t)$ the exact degree m polynomial solution of the collocation equations. We have by [13, Theorem 7.10]:

$$\frac{d^r u}{dt^r} - \frac{d^r \tilde{V}}{dt^r} = O(\Delta T^{m+1-r}), \quad 0 \leq r \leq m. \tag{3–12}$$

Also, the residual satisfies:

$$\begin{aligned} r^l_{jk} &= \int_{T_j}^{t_{jk}} (\psi^l_j(s) - \tilde{V}'(s))ds + \tilde{V}(t_{jk}) - v^l_{jk} \\ &= \Delta T \sum_{\alpha=1}^{m} S_{k\alpha}(F(v^l_{j\alpha}, t_{j\alpha}) - F(\tilde{V}(t_{j\alpha}), t_{j\alpha})) + \tilde{V}(t_{jk}) - v^l_{jk}. \end{aligned} \tag{3–13}$$

We proceed by induction on l. For $l = 0$ (3–10) follows directly from the consistency of (2–4). Precisely, since $p \leq m$, $u(t_{jk}) - v^0_{jk} = O(\Delta T^{p+1})$, (3–12) implies $\tilde{V}(t_{jk}) - v^0_{jk} = O(\Delta T^{p+1})$ with (3–10) following from the Lipschitz assumptions.

Denoting by R^l the residual vector,

$$R^l = \begin{pmatrix} r^l_{j1} \\ \vdots \\ r^l_{jm} \end{pmatrix} \in \mathbb{R}^{m\kappa}, \tag{3–14}$$

we recast the correction process as a fixed point iteration:

$$R^{l+1} = C(R^l), \tag{3–15}$$

and analyze the Jacobian derivative $D_R C(0)$. From (3–13) and (2–14) we have:

$$r^{l+1}_{jk} = r^l_{jk} - \delta^l_{jk} + \Delta T \sum_{\alpha=1}^{m} S_{k\alpha}(F(v^l_{j\alpha} + \delta^l_{j\alpha}, t_{j\alpha}) - F(v^l_{j\alpha}, t_{j\alpha})). \tag{3–16}$$

Taking the difference of (3–16) for consecutive values of k and using (2–12) we arrive at the formula:

$$\begin{aligned} r^{l+1}_{jk} = r^{l+1}_{j,k-1} &+ \Delta T \sum_{\alpha=1}^{m} (S_{k\alpha} - S_{k-1,\alpha})(F(v^l_{j\alpha} + \delta^l_{j\alpha}, t_{j\alpha}) - F(v^l_{j\alpha}, t_{j\alpha})) \\ &- h_k \left(F_I(v^l_{jk} + \delta^l_{jk}, t_{jk}) - F_I(v^l_{jk}, t_{jk}) \right) \\ &- h_k \left(F_E(v^l_{j,k-1} + \delta^l_{j,k-1}, t_{j,k-1}) - F_E(v^l_{j,k-1}, t_{j,k-1}) \right), \end{aligned} \tag{3–17}$$

where $S_{0\alpha} = 0$. Set:

$$G_{kk'} = D_{r^l_{jk'}} r^{l+1}_{jk}, \quad H_{kk'} = D_{r^l_{jk'}} \delta^l_{jk}, \tag{3–18}$$

where the derivatives are evaluated at $R^l = 0$. Note that $G_{kk'}, H_{kk'} \in \mathbb{R}^{\kappa\times\kappa}$ with $G_{kk'}$ being the block entries of $D_R C(0)$. Noting that $\delta^l_{jk} = 0$ if $R^l = 0$ we have from (2–12):

$$\begin{aligned} H_{kk'} &= H_{k-1,k'} + \varepsilon_{kk'} I - \varepsilon_{k-1,k'} I \\ &\quad + h_k \Big(D_u F_I(\tilde{V}(t_{jk}), t_{jk}) H_{kk'} + D_u F_E(\tilde{V}(t_{j,k-1}), t_{j,k-1}) H_{k-1,k'} \Big), \end{aligned} \tag{3–19}$$

$$H_{0k'} = 0. \tag{3–20}$$

(Here we are using ε_{ij} to denote the Kronecker δ to avoid confusion with the correction vector.) Combining (3–19) with (3–20) and solving in increasing k we conclude that $H_{kk'} = O(1)$. Moreover,

$$H_{kk'} = 0, \quad k < k'. \tag{3–21}$$

(This fact proves to be a barrier to accelerating convergence; see the remark below.) Differentiating (3–17) on the other hand we find:

$$\begin{aligned} G_{kk'} &= G_{k-1,k'} + \Delta T \sum_{\alpha=1}^{m} (S_{k\alpha} - S_{k-1,\alpha}) D_u F(\tilde{V}(t_{j\alpha}), t_{j\alpha}) H_{\alpha k'} \\ &\quad - h_k \left(D_u F_I(\tilde{V}(t_{jk}), t_{jk}) H_{kk'} + D_u F_E(\tilde{V}(t_{j,k-1}), t_{j,k-1}) H_{k-1,k'} \right), \end{aligned} \tag{3–22}$$

$$G_{0k'} = 0. \tag{3–23}$$

Solving (3–22) it is clear that $G_{kk'} = O(\Delta T)$ which is sufficient to prove (3–10).

Finally we note that a direct consequence of the Lipschitz conditions and the expression of V^l and $\tilde{V}$ in Lagrange form is that:

$$\left| \frac{d^r V^l}{dt^r} - \frac{d^r \tilde{V}}{dt^r} \right| \le C \Delta T^{1-r} \max_k |F(V^l(t_{jk}), t_{jk}) - F(\tilde{V}(t_{jk}), t_{jk})|. \tag{3–24}$$

Using (3–13) and (3–10) we find:

$$\left| \frac{d^r V^l}{dt^r} - \frac{d^r \tilde{V}}{dt^r} \right| \le C \Delta T^{p+l+2-r}. \tag{3–25}$$

By (3–12), (3–11) holds so long as $p + l + 2 \ge m$. This completes the proof of Lemma 3.2. □

Remarks. The proof of Lemma 3.2 makes no use of the assumption that $F_E + F_I = F$ and thus holds for inconsistent methods (2–5). We also see that the matrix on the righthand side of (3–22) cannot in general be $o(\Delta T)$ since it is the difference between nonzero full and block lower triangular matrices. Thus 3.2 is not directly related to the accuracy of (2–5). However, the choice of (2–5) does effect the stability of the overall method, though it may still be more efficient to use an inconsistent formula in some cases. We note that these results differ from those presented in [15; 16], where gains in accuracy commensurate with the order of the correction method are proved. A difference is that we are proving higher order convergence - in particular higher order than is attained at the interior quadrature nodes. In [17] it is shown, for linear problems, that by using GMRES to accelerate the correction process only half as many corrections are needed to attain the full accuracy. In addition, they show that the use of GMRES improves the accuracy for stiff problems.

We note that if (2–16)-(2–18) are used, then the analogue of δ^l is given by $\bar{\delta}^l = v^0 - \bar{v}^l$. This correction satisfies:

$$\begin{aligned}\bar{\delta}^l_{jk} = \bar{\delta}^l_{j,k-1} + r^l_{jk} - r^l_{j,k-1} + v^0_{jk} - v^0_{j,k-1} \\ -h_k\Big(F_I(v^0_{jk} - \bar{\delta}^l_{jk}, t_{jk}) + F_E(v^0_{j,k-1} - \bar{\delta}^l_{j,k-1}, t_{j,k-1})\Big). \quad (3\text{–}26)\end{aligned}$$

We see that unless v^0 satisfies (2–5), that is unless the correction method and the starting method coincide, the correction does not approach zero with the residual. Hence the method cannot be interpreted as an approximation to an implicit Runge–Kutta method. However, if they do coincide Lemma 3.2 also holds. The proof follows essentially line for line, so we omit it.

Lastly we remark that our proof, relying as it does on the Lipschitz continuity of F, fails in the stiff limit, though we will show that the order of accuracy is attained for some stiff problems. In [2] experimental studies are presented of the convergence of deferred correction of the backward Euler method to the underlying implicit Runge–Kutta method.

Combining Lemma 3.1 and Lemma 3.2 we have proven our main theorem.

Theorem 3.3. *For a sufficiently smooth solution, u, Lipschitz continuous functions, F, F_I, F_E, $\tilde{F}_I$, $\tilde{F}_E$, a starting method of order $p \le m$, and $l \ge m - p - 2$, there exists a constant C independent of ΔT sufficiently small such that*:

$$|u(T_{j+1}) - V^l(T_{j+1})| \le C\Delta T^{\min(p+l+1,q+1)}.$$

Note that we have assumed that after the desired corrections are made the solution is updated by (2–15). Of course, if T_{j+1} is a node, then nothing needs to be done; we simply take v^L_{jm} as the value at T_{j+1}.

If, on the other hand, T_{j+1} is not a node, we may typically replace (2–15) by simple polynomial extrapolation using $(T_j, v(T_j))$ and all data on the quadrature nodes, thus saving m evaluations of F. In particular, even if $c_1 \neq 0$, that is if T_j is not a node, Theorem 3.3 is still valid. To see this, define the polynomial

$$\phi(t) = v(T_j) + \int_{T_j}^{t} \psi_j^L(s)ds - r^L(t), \tag{3–27}$$

where $r^L(t)$ is the polynomial that interpolates $(T_j, 0)$ and (t_{jk}, r_{jk}^L), for $k = 1, 2, \ldots, m$. Obviously, the degree of $\phi(t)$ is m. Suppose T_j is not a node. Then since $\phi(t_{jk}) = v_{jk}^L$, $\phi(t)$ is exactly the polynomial that interpolates $(T_j, v(T_j))$ and (t_{jk}, v_{jk}^L), $k = 1, 2, \ldots, m$. The update given by (2–15) is

$$v(T_{j+1}) = \phi(T_{j+1}) + r(T_{j+1}), \tag{3–28}$$

while the update given by extrapolation is $\phi(T_{j+1})$. So the difference between these two updates is controlled by $r(T_{j+1})$. Therefore, they have the same order, though the stability characteristics may be altered.

4. Efficiency and linear stability of sample methods

We now consider, experimentally, the accuracy, efficiency and linear stability of some simple examples of the methods discussed above. For simplicity, following [1], we consider a Dahlquist-type problem modeling operator splitting applied to a spatially discretized advection-diffusion equation. Precisely we consider:

$$u' = (\alpha + i\beta)u, \quad \alpha, \beta \in R, \quad \alpha \leq 0, \tag{4–1}$$

and take

$$F_E(u) = \tilde{F}_E(u) = i\beta u \quad \text{or} \quad F_E = 0, \tag{4–2}$$

$$F_I(u) = \tilde{F}_I(u) = \alpha u. \tag{4–3}$$

The range of methods tested includes:

(1) Gauss–Lobatto, Gauss–Legendre and right-handed Gauss–Radau quadrature with $m = 3, \ldots, 10$ nodes, encompassing method orders from 5 through 20;

(2) Multistep IMEX methods [1] of orders 1 through $m - 1$ as starting methods (2–4);

(3) Consistent correction methods with $F_E = i\beta u$ and inconsistent correction methods with $F_E = 0$;

We note that the split multistep methods we use are based on backward differentiation with F_E extrapolated to the new time level. Thus they are the natural

generalizations of the SBDF methods of [1] to nonequispaced grids. Their order is k and we take $k = 1, \ldots, m-1$. When $k > 1$ we are using values, $v^L_{j-1,r}$, which are only accurate to order m. Also in that case we need to consider the eigenvalues of the amplification matrix, A, mapping between values used in subsequent starting formulas. The stability properties of the SBDF methods themselves are not directly at issue and have not been studied, though we expect they are unstable at high order.

As the number of methods considered is in the hundreds we will limit our discussion to a few representative cases. Mainly we will display results obtained using the Gauss–Legendre and Gauss–Radau nodes. In most instances the behavior of the Gauss–Lobatto methods was essentially the same. An exception is the stability regions, where all three will be compared.

4.1. ***Accuracy and efficiency.*** We first verify that the methods attain the design order even if inconsistent corrections are used. Precisely we consider (4–1) with $\alpha = -1/20$, $\beta = -2\pi$, and solve up to $T = 20$. See Figure 1 for experiments with Gauss–Legendre nodes and Figure 2 for experiments with Gauss–Radau nodes. In each case we observe convergence at the correct rate, though the use of inconsistent corrections clearly leads to less accurate results for a fixed time step. The accuracy is insensitive to the order of the starting method, indicating higher efficiency with the use of higher order starting methods.

From the point of view of computational effort, the efficiency of an ode solver is typically measured by the number of function evaluations required to attain a given accuracy. An emphasis of the current work is the possibility to achieve higher order for fixed m than in earlier implementations of SDC, presumably with significant savings in memory. However, these savings could conceivably be lost if the methods proposed here turn out to be less efficient. Thus we wish to compare the efficiency of different variations of SDC, including methods where m is chosen larger than necessary to achieve the design order.

To facilitate comparisons with previously published results we consider here, following [21], a nonstiff van der Pol equation for $0 \le t \le 4$:

$$u_1' = u_2, \quad u_2' = -u_1 + (1 - u_1^2)u_2, \tag{4–4}$$

$$u_1(0) = 2, \quad u_2(0) = \frac{2}{3}. \tag{4–5}$$

As the equations are not stiff, we do not split them and simply use explicit starting and correction methods.

We consider three comparisons. First we fix the starting method (first order) and the number of corrections while varying m. Thus we are comparing methods of the same order. The results, shown in Figure 3, show that efficiency measured in this way is essentially independent of m; it is apparently determined by the average step

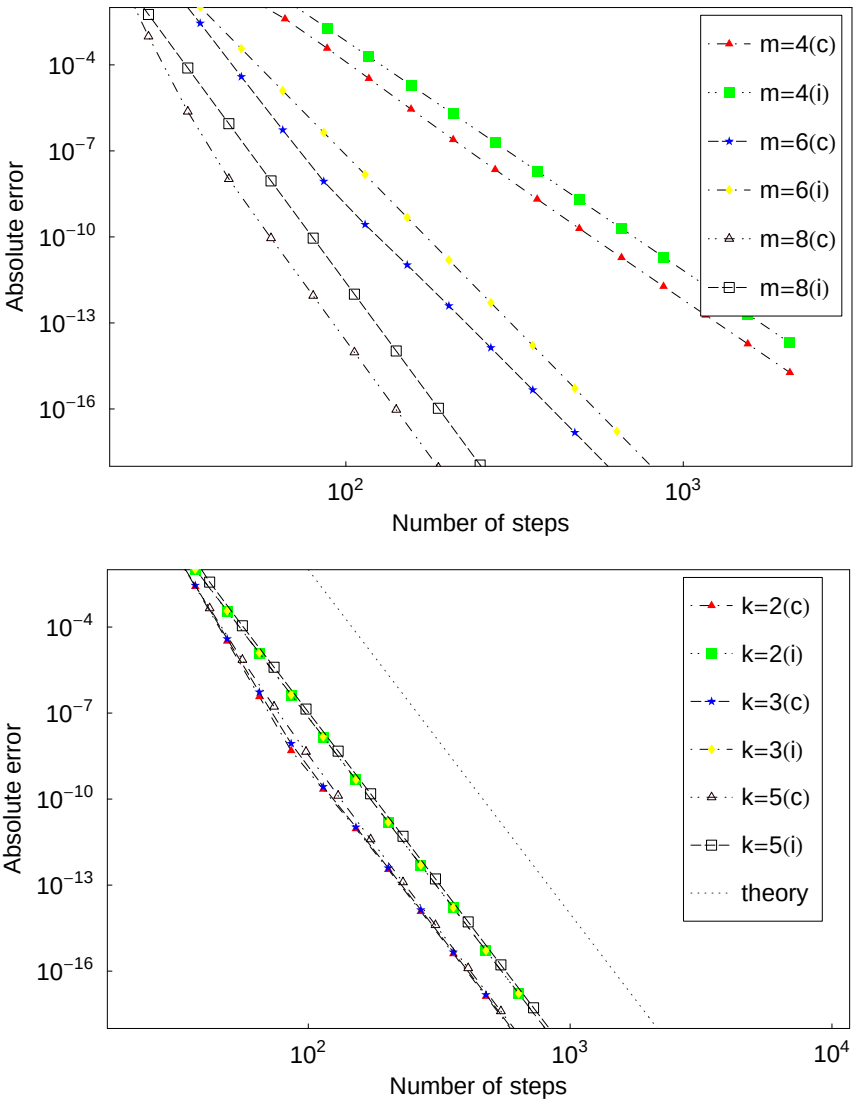


Figure 1. The top graph shows the accuracy of the split SDC methods using the same starting method (3rd order SBDF) and different numbers of Gauss–Legendre nodes m. The bottom graph shows the accuracy of SDC methods with 6 Gauss–Legendre nodes, but different starting methods (k indicates the kth-order SBDF starting method, c,i indicates consistent and inconsistent corrections, respectively). The dotted line shows the theoretical convergence order 12.

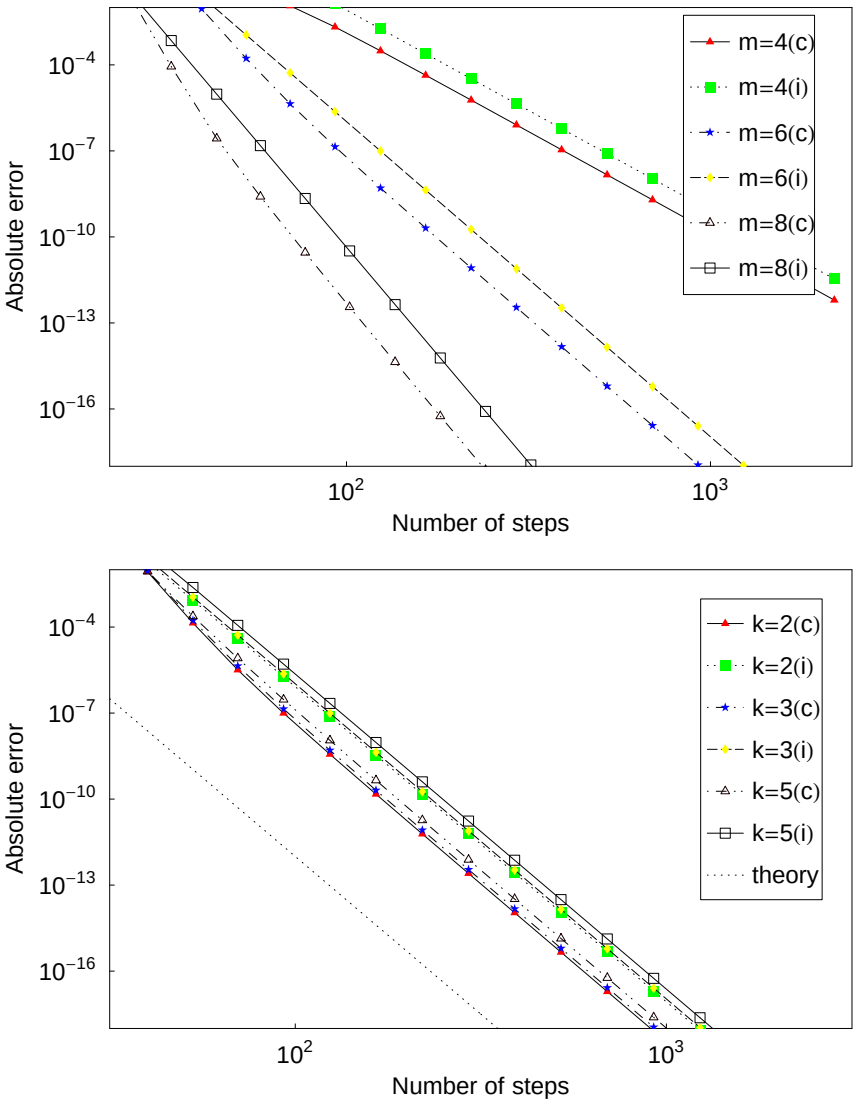


Figure 2. The top graph shows the accuracy of the split SDC methods using the same starting method (3rd order SBDF) and different numbers of Gauss–Radau nodes m. The bottom graph shows the accuracy of split SDC methods with 6 Gauss–Radau nodes, but different starting methods (k indicates the kth-order SBDF starting method, c,i indicates consistent and inconsistent corrections, respectively). The dotted line shows the theoretical convergence order 11.

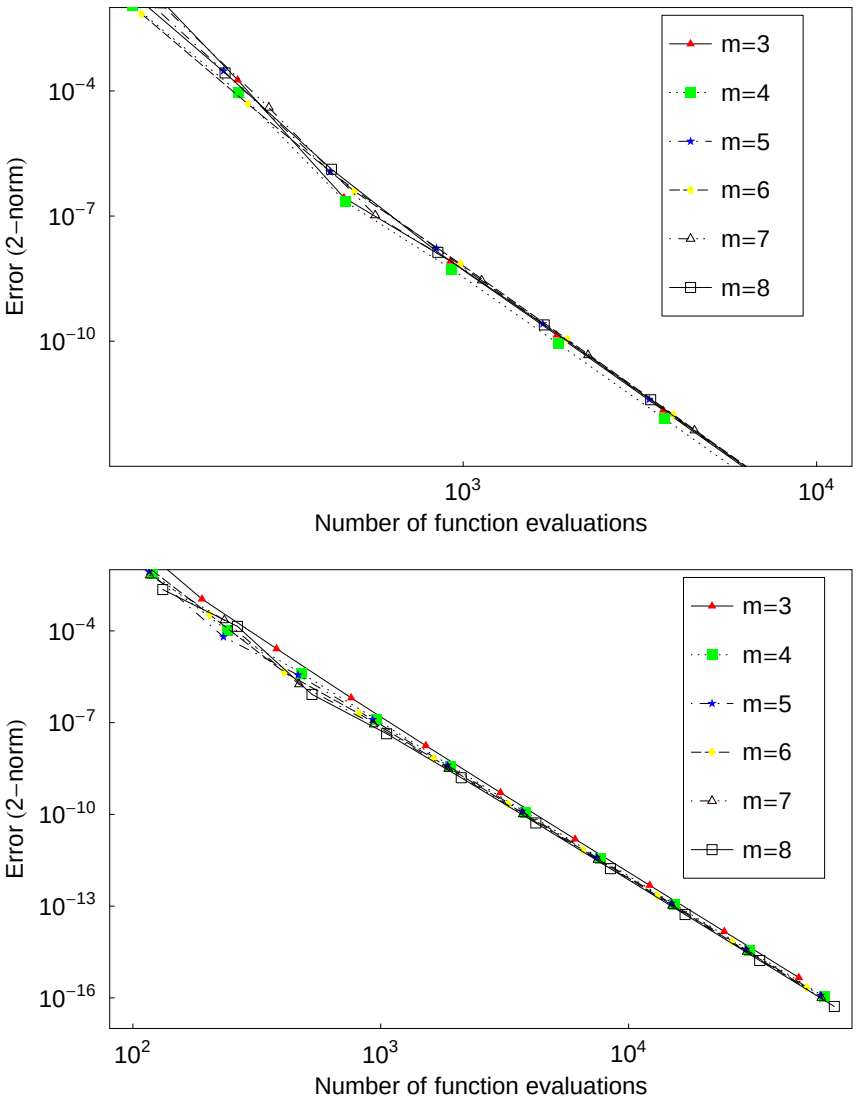


Figure 3. Efficiency in terms of function evaluations for 6th order Gauss–Legendre and 5th order Gauss–Radau methods with first order starting methods and varying numbers of quadrature nodes, m.

size and the number of corrections. Thus the gains in memory utilization resulting from the exploitation of the full order of the quadrature rules are fully realized, at least for this example.

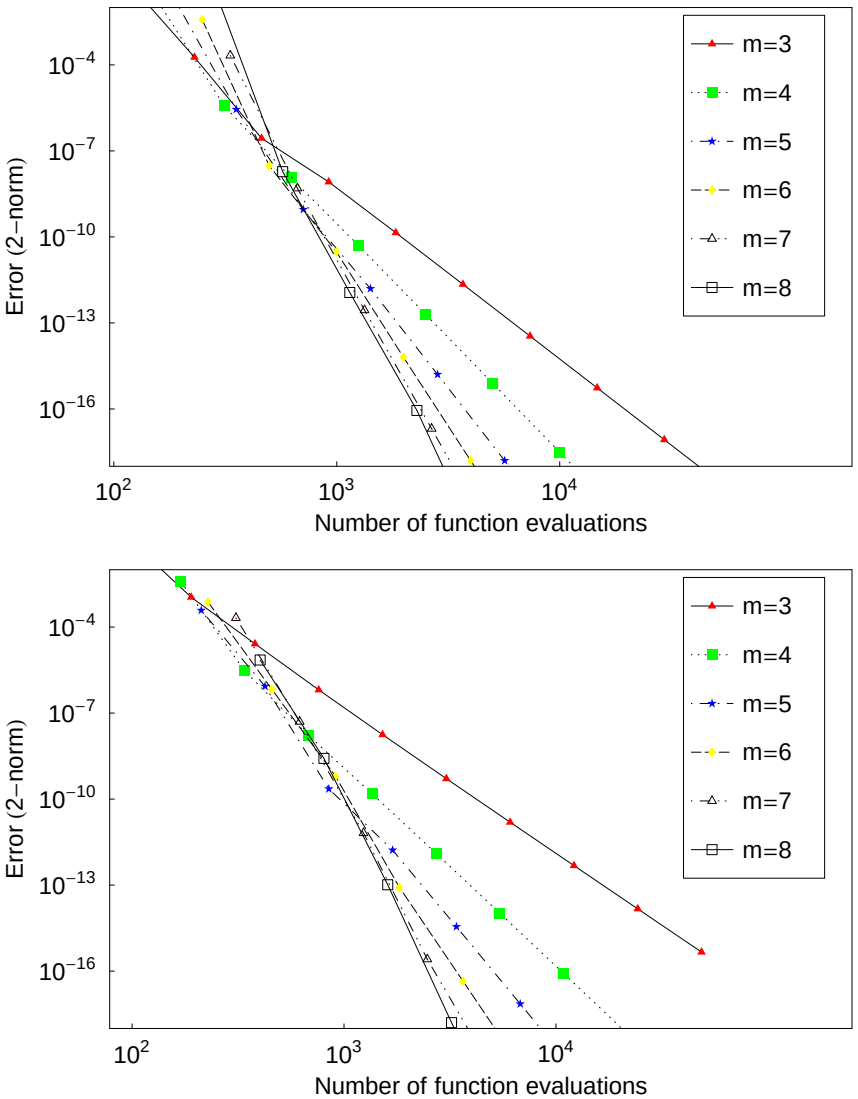


Figure 4. The efficiency of Legendre and Radau methods for various orders. In all cases we use a first order starting method.

Second, in Figure 4 we compare efficiency for differing orders and quadratures, in each case using the full order of the quadrature rule. As expected, the "optimal" method order depends on the desired tolerance, with the higher order methods favored as the tolerances decrease.

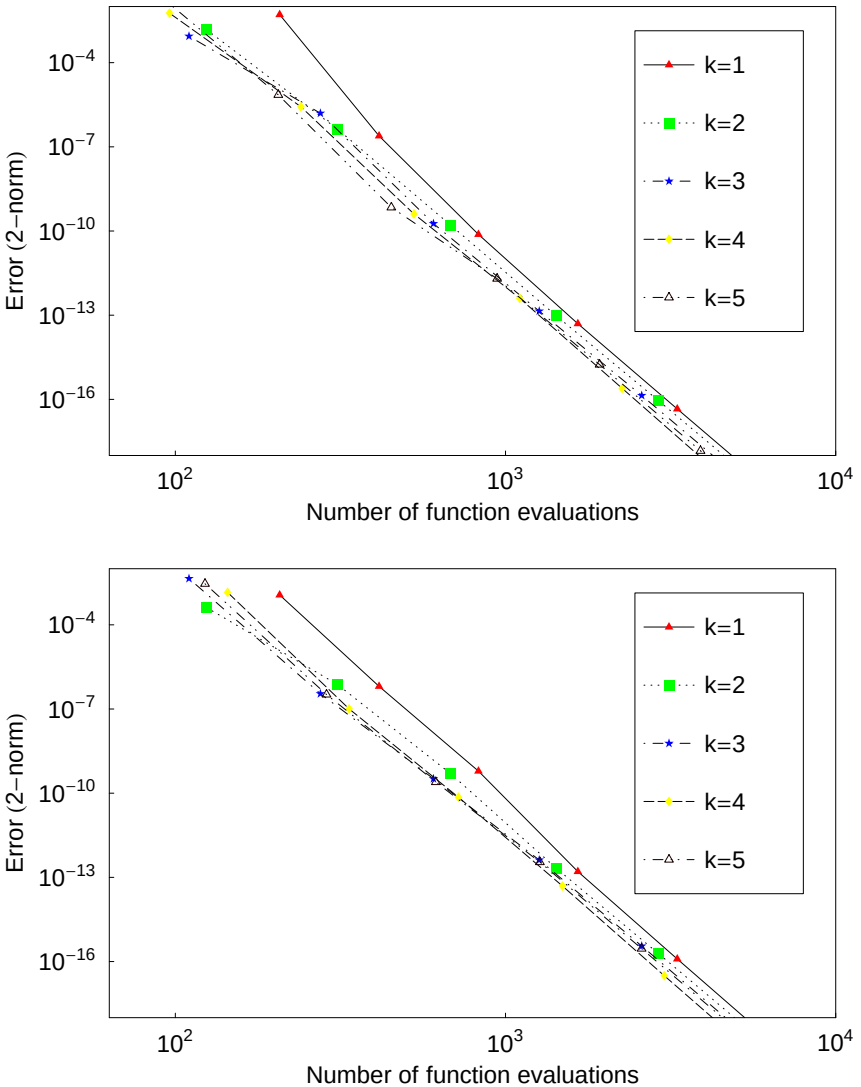


Figure 5. The efficiency of 12th order Legendre and 11th order Radau methods varying the order of the starting method.

Lastly we consider the effect of varying the order of the starting method, fixing $m = 6$. The results, shown in Figure 5, demonstrate a substantial gain in efficiency as the starting method order is increased from 1 to 2 with modest, but measurable, gains when it is increased from 2 to 3. Beyond third order there seems to be no advantage in further increases.

4.2. ***Stability for*** (4–1)**.** The analysis of the stability of splitting methods in general is difficult. Even for linear, constant coefficient systems, the fact that the matrices defining the split operators cannot be expected to commute limits the predictive value of analyzing a split version of Dahlquist's model problem. Nonetheless, at least to establish some basis for the comparison of stability for our different splitting procedures, we will follow [1] and plot experimentally determined stability domains associated with (4–1). In the following sections we will consider a more general stability problem motivated by what we call preconditioned splitting methods.

We note that a more interesting definition of stability domains for splitting methods has been proposed by Frank et al. [8]. Their idea is to consider stability for a scalar, split system under the assumption that the time step is chosen so that the explicit method is stable. This allows a clean definition of a stability domain for split methods and a generalization of many of the standard notions of $A(\alpha)$ and $L(\alpha)$ stability. Layton and Minion [20] have applied this definition to the SDC of splitting methods and shown that quadrature rules excluding the left endpoint such as the Gauss–Legendre and righthand Gauss–Radau rules lead to $L(\alpha)$-stable methods with $\alpha \approx \pi/2$. However, this analysis is not general as one might often want to use methods in regimes where the explcit scheme by itself is unstable. The simple case of (4–1) illustrates this; the stability domain of explicit Euler contains no points on the imaginary axis except the origin. Thus with the splitting considered here the consistent explicit correction method is always unstable, so the results of [8; 20] do not apply. Nonetheless, as in [20] we find that the stability properties of the Gauss–Lobatto methods are clearly inferior to those based on Gauss–Legendre or Gauss–Radau quadrature.

Figures 6 and 7 show stability domains for various Legendre and Radau-based methods. The overall results are quite similar. The stability domains are somewhat larger if inconsistent rather than consistent corrections are used. The domains increase in size with increasing m but decrease with increasing k. Obviously, except for k large, they contain a very large region near the negative real axis.

Lastly in Figure 8 we compare the stability of 8th order methods with $m = 5$. We clearly see that the stability domain obtained using Gauss–Radau nodes is slightly larger than that obtained with Gauss–Legendre nodes, but both are much larger than the stability domain obtained using Gauss–Lobatto nodes.

4.3. ***Relative accuracy.*** Lastly we make some relative accuracy comparisons fixing m and the order of the methods in Figure 9. Note that we are thus not carrying out the full number of corrections when Legendre or Radau nodes are used. We find that under these restrictions the Legendre nodes yield the most accurate results, followed by the Radau nodes.

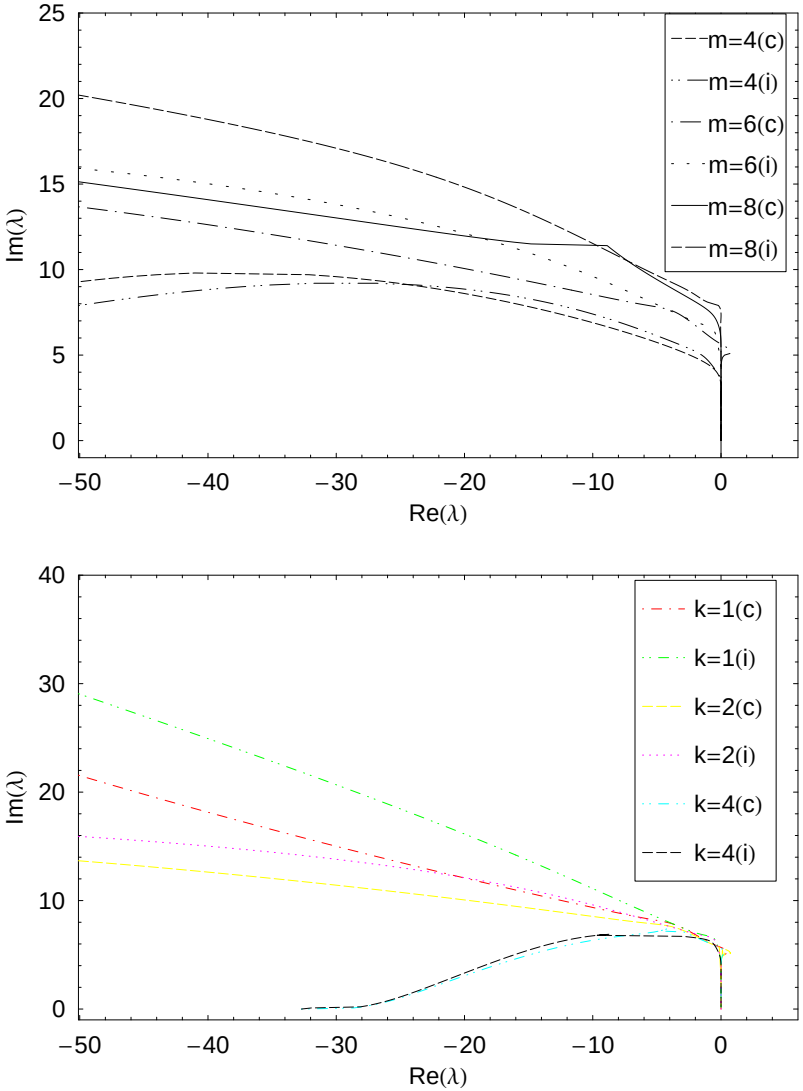


Figure 6. The top graph shows the stability of the SDC methods using same starting method (2nd order SBDF) and different numbers of Gauss–Legendre nodes m. The bottom graph shows the stability of SDC methods with 6 Gauss–Legendre nodes, but different starting methods (k indicates the kth-order SBDF starting method).

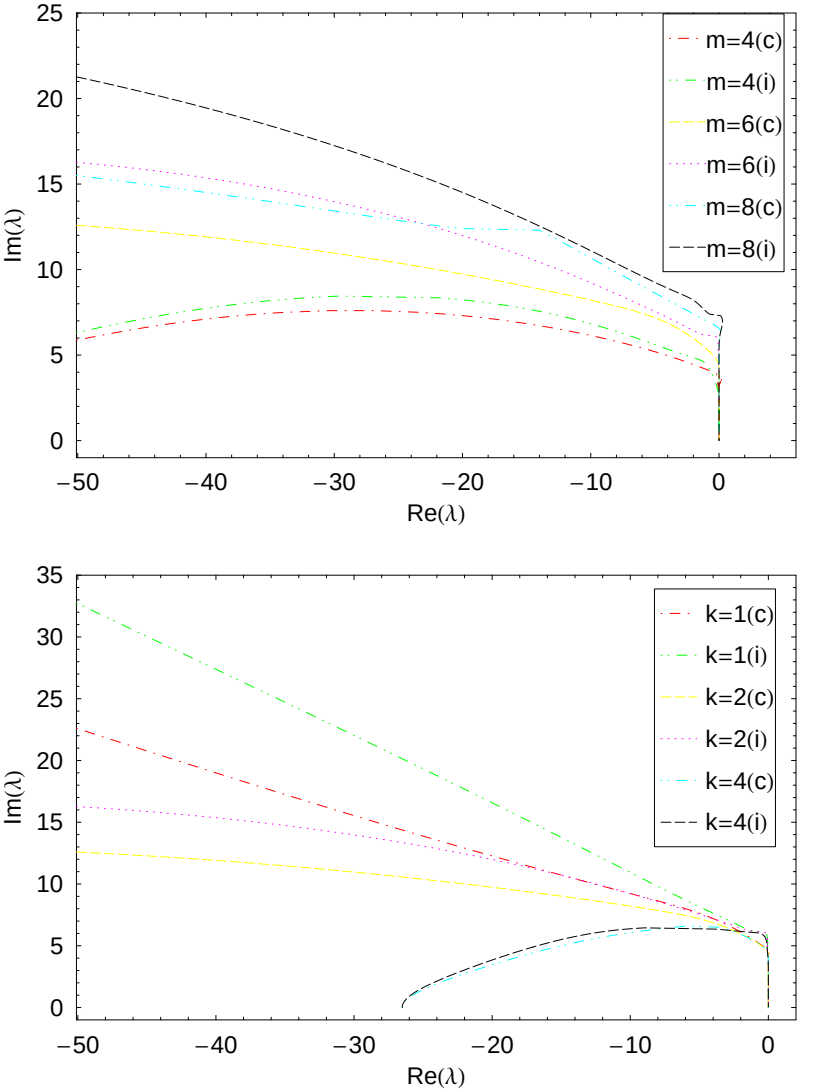


Figure 7. The top graph shows the stability of the SDC methods using same starting method (2nd order SBDF) and different numbers of Gauss–Radau nodes m. The bottom graph shows the stability of SDC methods with 6 Gauss–Radau nodes, but different starting methods (k indicates the kth-order SBDF starting method).

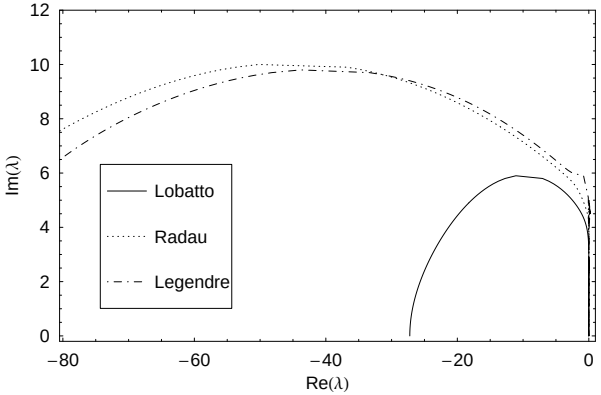


Figure 8. Stability domains for 8th order methods (2nd order SBDF starting method and 6 consistent corrections) with $m = 5$ and various quadrature nodes.

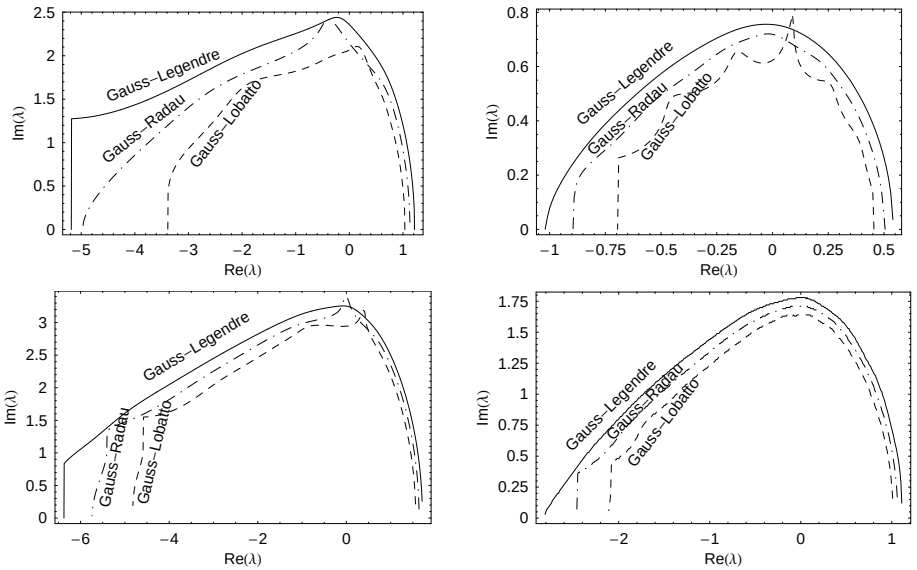


Figure 9. In all figures, we use the single step starting method with consistent corrections. For the top two figures, $m = 5$, the order is 8 for all methods. The top-left figure shows the accuracy region for $\epsilon = 10^{-4}$; while for the top-right figure, $\epsilon = 10^{-9}$. For the bottom two figures, $m = 9$, the order is 16 for all methods. For the bottom-left figure, $\epsilon = 10^{-9}$, while for the bottom-right figure, $\epsilon = 10^{-14}$.

5. Preconditioned splitting methods

Although the stability and accuracy of the SDC methods used in conjunction with advection-diffusion splittings is reasonable, we believe it is worthwhile to pursue a more general and flexible approach. Introduce a preconditioning matrix P and the splitting:

$$F_I = -Pv, \quad F_E = F + Pv. \tag{5–1}$$

Here, P will generally be dependent on t and local values of v. It should satisfy the requirements:

i: $P + P^* \geq 0$;

ii: $I + h_k P$ inexpensively invertible;

iii: the split methods have good stability and accuracy properties.

Of course the difficult property to satisfy is the third. For a simple model problem, we will see that stability of the base method is ensured by choosing P sufficiently large compared with the Jacobian of F, and then good accuracy follows from not making it too large. (We suspect that generalizations of the stability analysis to nonlinear problems satisfying appropriate one-sided Lipschitz conditions would be straightforward.)

5.1. ***Linear stability for the scalar problem.*** We repeat the stability analysis for Dahlquist's equation (4–1) but now with the general preconditioner:

$$P = \mu + i\eta, \quad \mu, \eta \in R, \quad \mu \geq 0. \tag{5–2}$$

Note that we are allowing an imaginary part in P, corresponding to the inclusion of a linear advection term in the preconditioner. The amplification factor of the first order splitting (2–5) is then given by:

$$r^2 = \left| \frac{1 + h(\alpha + \mu + i(\beta + \eta))}{1 + h(\mu + i\eta)} \right|^2 = \frac{(1 + h(\alpha + \mu))^2 + h^2(\beta + \eta)^2}{(1 + h\mu)^2 + h^2\eta^2}. \tag{5–3}$$

For $\eta = 0$ we have A-stability if:

$$\mu \geq \frac{\alpha^2 + \beta^2}{2|\alpha|}. \tag{5–4}$$

For a discretized advection-diffusion equation with Peclet number Pe we obtain:

$$P \geq C \left(-\frac{1}{Pe} D_x^2 + Pe \right), \tag{5–5}$$

where D_x^2 is an approximation to the Laplacian and the inequality is in the usual sense of matrices. For large Peclet number such a choice is likely to have a negative

impact on accuracy as the preconditioner is large. Of course one can give up on A-stability. For example if $\mu \geq |\alpha|/2$ we have:

$$h \leq \frac{2|\alpha|}{\beta^2} = O(Pe^{-1}), \tag{5–6}$$

independent of the spatial mesh width Δx, which is acceptable if Pe is not too large.

Much better results can be obtained if we choose η to be nonzero and of the opposite sign of β. Then we have A-stability if:

$$\mu \geq \frac{|\alpha|}{2}, \quad |\eta| \geq \frac{|\beta|}{2}, \quad \alpha\beta \leq 0. \tag{5–7}$$

We will show nonlinear examples where a linear advection term is included in P. Of course the sign condition can be difficult to satisfy where the advection term nearly vanishes, but then the local Peclet number is not large. A-stability is not generally preserved when the correction process is included, but we will see below that the stability domains can be quite large.

We again note that we do not in general expect that our time step is chosen so that the explicit method is stable. Thus the stability analyses of [8; 20] are not directly applicable.

5.2. ***Linear stability domains of the preconditioned methods.*** In Figures 10, 11, and 12 we plot linear stability domains of the consistently corrected methods assuming:

$$\mu = d_r\alpha, \quad \eta = d_i\beta, \tag{5–8}$$

with d_r and d_i chosen from $\{\frac{3}{4}, \frac{5}{4}\}$. Clearly, d_i determines stability along the imaginary axis and d_r along the real axis.

As before, the stability characteristics of the Gauss–Legendre and Gauss–Radau methods are quite similar, with the stability domains of the Gauss–Legendre methods being generally a little larger. The Gauss–Lobatto methods, on the other hand, show superior stability along the real axis for $d_r = \frac{3}{4}$.

We also tested multistep starting methods and inconsistent corrections. Except in the case of second order starting methods, the stability domains are significantly reduced when consistent corrections are used. However, with inconsistent corrections and $d_r = d_i = \frac{5}{4}$ they are enlarged. We plot below (Figure 13) results using an 11th order Radau scheme with the 3rd order starting method.

6. Nonlinear numerical experiments

Finally we consider the actual accuracy and stability of one of the methods discussed above for a collection of nonlinear parabolic initial-boundary value problems in

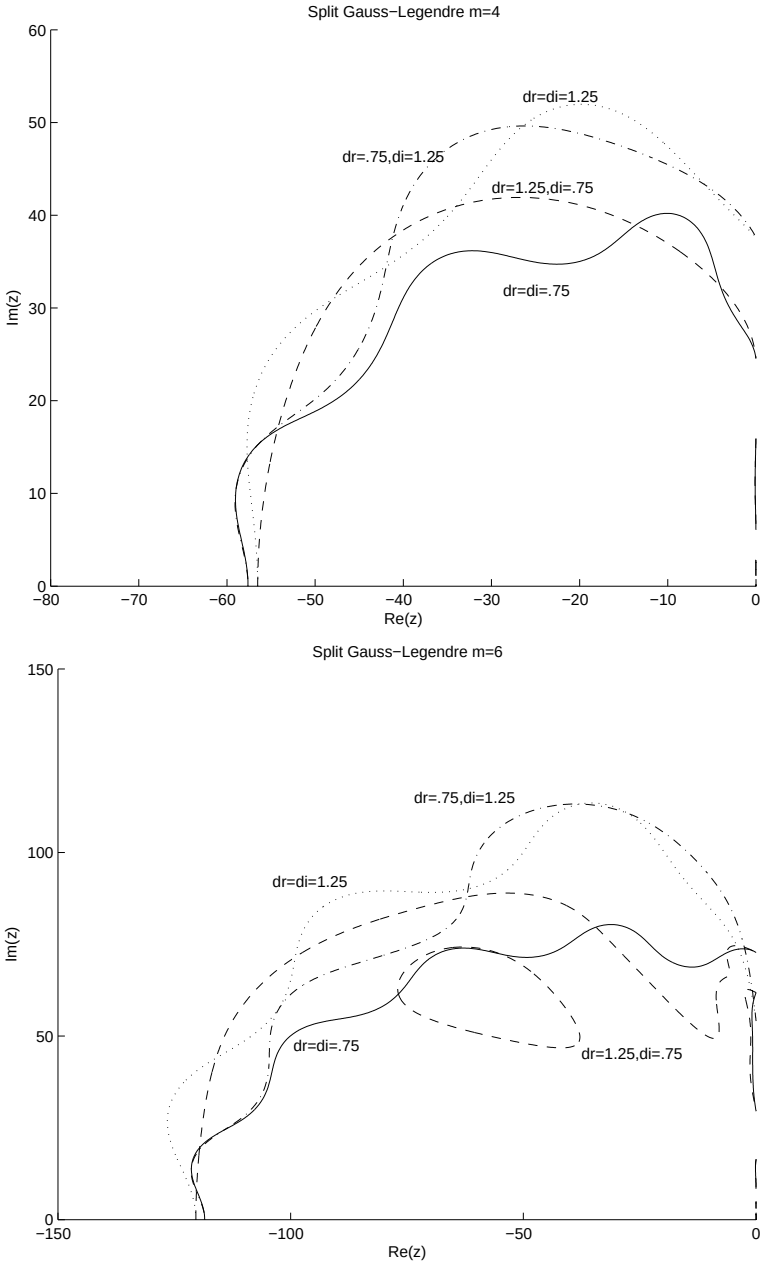


Figure 10. Stability domains for Gauss–Legendre methods with preconditioned splittings.

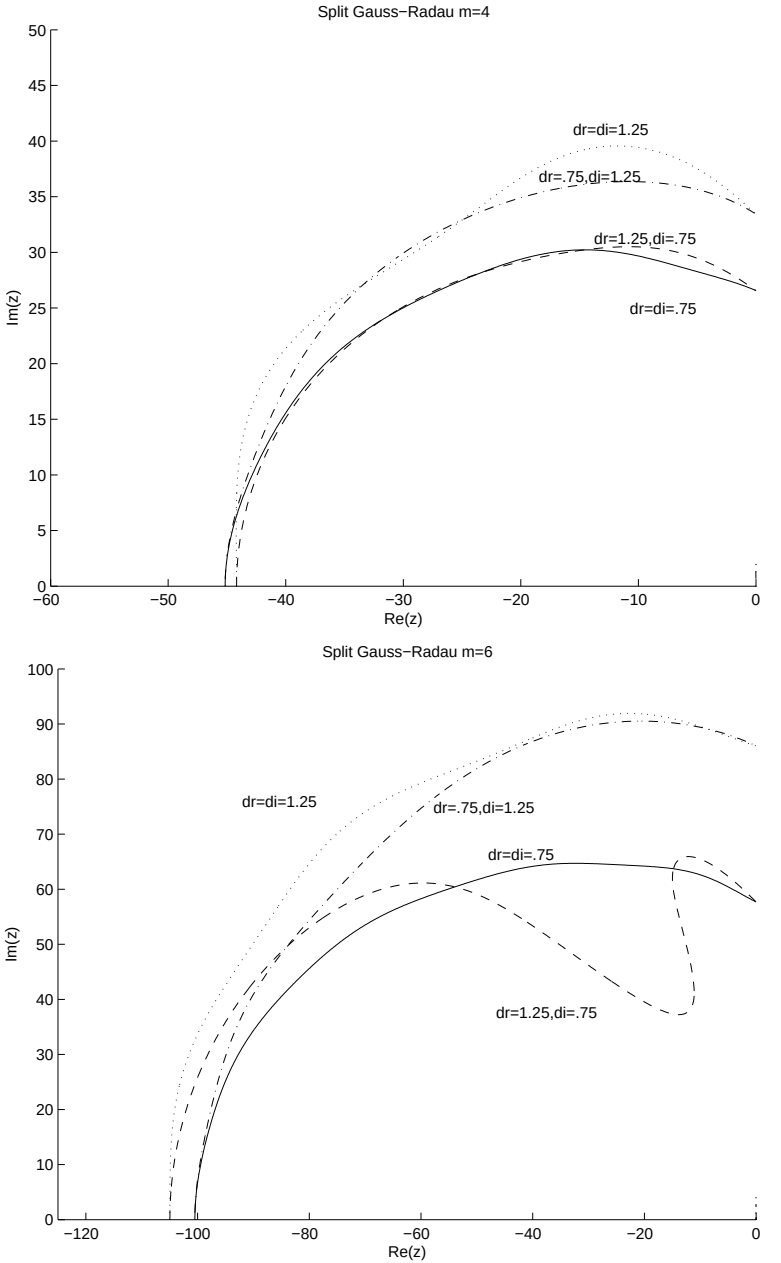


Figure 11. Stability domains for Gauss–Radau methods with preconditioned splittings.

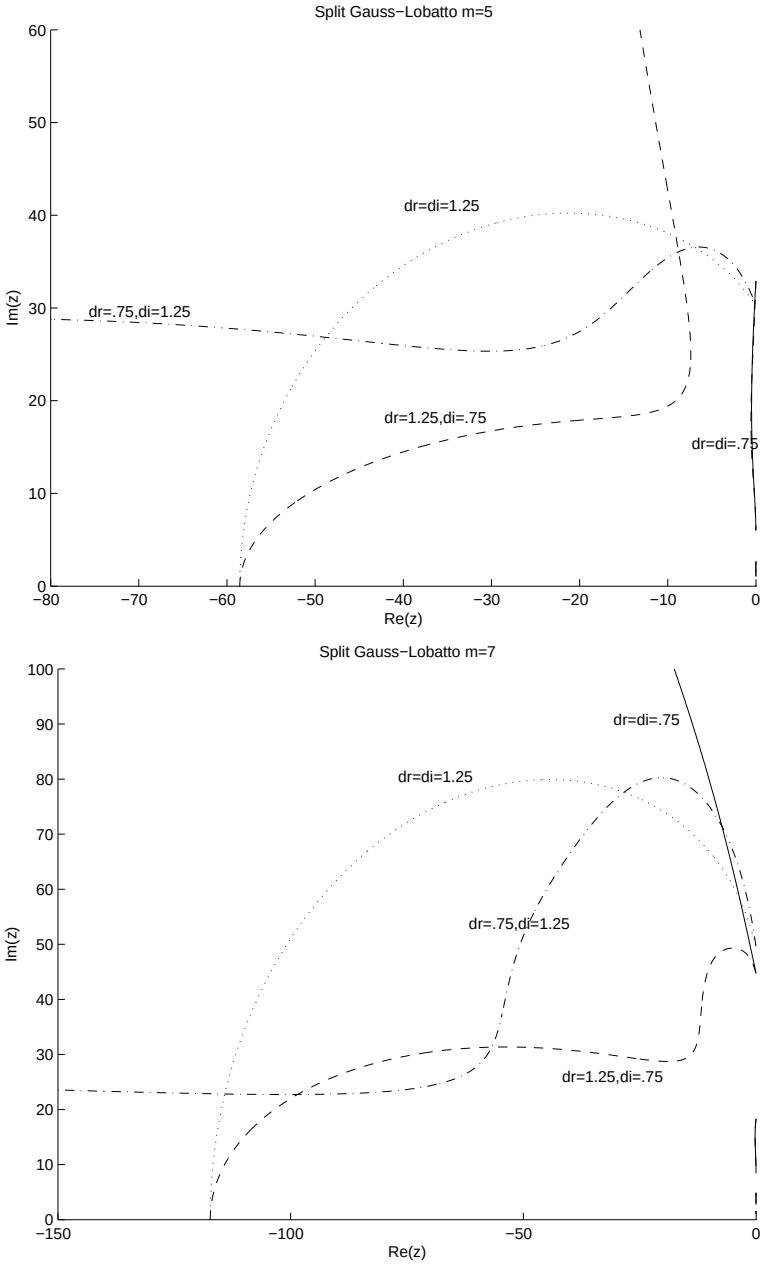


Figure 12. Stability domains for Gauss–Lobatto methods with preconditioned splittings.

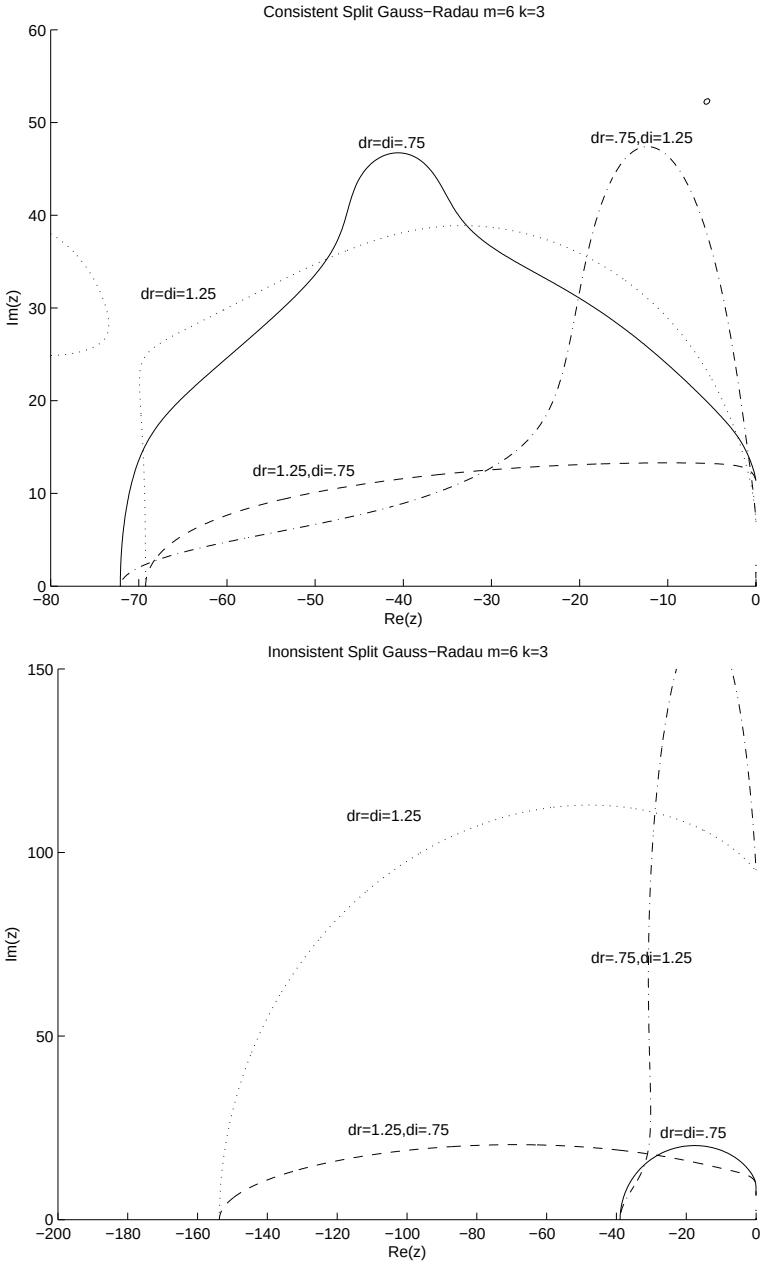


Figure 13. Stability domains for 7th order Gauss–Radau methods with preconditioned splittings and a 3rd order multistep starting method.

$1+1$ dimensions:

$$u_t = F(u, u_x, u_{xx}), \quad x \in (x_L, x_R), \tag{6–1}$$

supplemented by boundary conditions. In all cases we use the 7th order Radau method with a multistep SBDF preconditioned starting method and consistent corrections. Our spatial discretizations are 8th order; one-sided differencing at the boundaries is stabilized by the addition of a single sub-cell point at $x_L + 0.2\Delta x$ and $x_R - 0.2\Delta x$ with central differences used in the interior. See [10] for details.

Various preconditioners are considered, but in all cases the spatial differencing used in the preconditioning is limited to a 3-point stencil to minimize bandwidth. In the interior, then, we are preconditioning 8th order differences by multiples, γ_d, γ_c, of 2nd order differencing. To better correlate the results with the simple linear stability domains shown above we note that the symbols of the qth order central difference approximations to $\frac{d^j}{dx^j}$, $\hat{d}_{j,q}$, satisfy:

$$\max_{|\omega h| \le \pi} \frac{\hat{d}_{1,8}(\omega)}{\hat{d}_{1,2}(\omega)} \approx 2.66, \tag{6–2}$$

$$\max_{|\omega h| \le \pi} \frac{\hat{d}_{2,8}(\omega)}{\hat{d}_{2,2}(\omega)} \approx 1.68. \tag{6–3}$$

Thus, for example, a preconditioned approximation to the heat equation is positive independent of time step only if the damping factor, γ, is chosen to be larger than 1.68.

Of course the examples are primarily meant to illustrate a viable preconditioning strategy and to provide some experience in the method's performance under a variety of conditions. With experience for a given system we would expect that better preconditioners could be found leading to further improvements in efficiency. Most of our examples would benefit from the use of an adaptive spatial mesh, but here we simply employ sufficiently fine uniform discretizations. We also compare our results with those obtained using a standard second order Strang splitting in time (e.g [22]) and a time step chosen so that the number of evaluations of the nonlinearities are comparable. For example, if we use a fourth order starting method an entire SDC step entails sixteen substeps, so we choose the time step for the Strang method to be $1/16$ times that of the SDC method. However, recall that our method is linearly implicit while the Strang splitting employs Newton iterations; thus the SDC method is noticeably faster for the time steps compared. We note, of course, that we could have used the Strang splitting as our starting method or even as our correction method, but we have not yet implemented this.

Table 1. Error data for the Brusselator problem.

ΔT	Δx	γ_d	$e_{\max}(u)$	$q(u)$	$e_{\max}(v)$	$q(v)$
$2E(-1)$	$2E(-2)$	2	3.13(−4)		1.51(−4)	
$1E(-1)$	$1E(-2)$	2	2.10(−6)	7.2	1.73(−6)	6.4

6.1. ***Brusselator.*** We consider for $(x,t) \in (0,1)\times(0,10)$:

$$u_t = 1 + u^2 v - 4u + 2\cdot 10^{-3} u_{xx}, \tag{6–4}$$

$$v_t = 3u - u^2 v + 2\cdot 10^{-3} v_{xx}, \tag{6–5}$$

$$u(x,0) = 1 + \sin 20\pi x, \quad v(x,0) = 3, \tag{6–6}$$

with Dirichlet boundary conditions. With this data the solution is known to oscillate; see the graph of the fine grid solution in Figure 14 as well as [14; 22].

Here there is no convective term to be included in the preconditioner, but the Jacobian of the reaction terms is included along with the scaled three point diffusion approximation. That is with $\nu = 2\cdot 10^{-3}$:

$$P_i = -\begin{pmatrix} 2u_i v_i - 4 + \nu\gamma_d d_{2,2} & u_i^2 \\ 3 - 2u_i v_i & -u_i^2 + \nu\gamma_d d_{2,2} \end{pmatrix}. \tag{6–7}$$

We employ a fourth order starting method with three correction steps.

The results, displayed in Table 1, are consistent with the design accuracy. Error data is obtained by comparison with a solution computed using $\Delta T = 2.5E(-4)$ and $\Delta x = 1E(-3)$.

By way of comparison, with $\Delta x = 2E(-2)$ and $\Delta T = 1.25(-2)$ the maximum errors with Strang splitting were $(1.70(-3), 9.60(-4))$, about six times larger than those reported above. Halving the grid and step sizes the Strang errors are reduced by about a factor of four to $(3.56(-4), 2.19(-4))$, about two orders of magnitude larger than were obtained with the SDC time stepping.

We also determined apparent time step stability limits. For $\gamma_d = 2$ these were weakly dependent on Δx, but we could always take rather large steps; $\Delta T = \frac{1}{2}$ for $\Delta x = \frac{1}{50}$, $\Delta T = \frac{1}{3}$ for $\Delta x = \frac{1}{100}$ and $\Delta T = \frac{1}{8}$ for $\Delta x = \frac{1}{150}$. For $\gamma_d = 1$, on the other hand, they clearly took the form $\Delta T \le c\Delta x^2$. With $\Delta x = \frac{1}{150}$ it was necessary to take $\Delta T = \frac{1}{101}$.

6.2. ***Smoothed angiogenesis model.*** Here we consider a smoothed verison of a tumor angiogenesis model presented in [18]:

$$\rho_t = 10^{-3}\rho_{xx} - .75(\rho c_x)_x + 10^2\rho(1-\rho)K(c) - 4\rho, \tag{6–8}$$

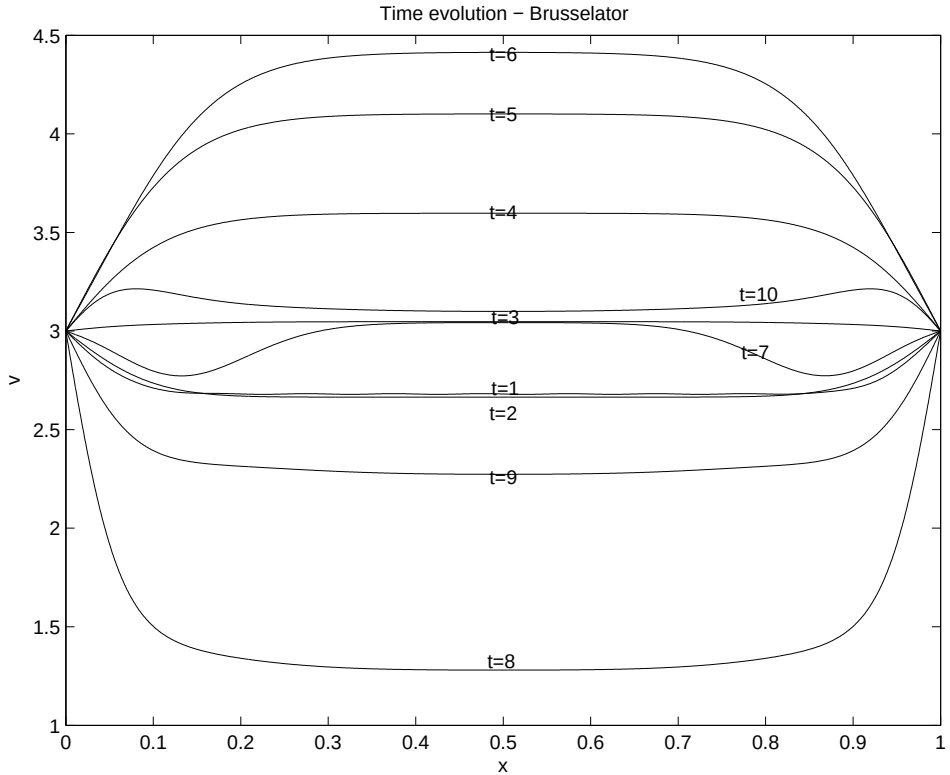


Figure 14. Fine grid solution of the Brusselator equation: v.

$$c_t = c_{xx} - c - \frac{10\rho c}{1+c}, \tag{6–9}$$

where $(x,t) \in (0,1)\times(0,.7)$ and:

$$K(c) = 5\cdot 10^{-3}(100(c-.2) + \ln(\cosh(100(c-.2)))), \tag{6–10}$$

$$\rho(x,0) = e^{-288(x-1)(x-1.08\bar{3})}, \quad c(x,0) = \cos \pi x/2, \tag{6–11}$$

and $\rho(0,t) = c(1,t) = 0$, $\rho(1,t) = c(0,t) = 1$.

The evolution of ρ for a fine grid solution computed with $\Delta T = 5E(-4)$, $\Delta x = 1E(-3)$ is shown in Figure 15. Comparison with the figures in [18] show that the smoothing has had little effect on the solution.

As these equations involve both first and second order spatial derivatives both γ_d and γ_c must be chosen. Precisely we used a block diagonal preconditioner:

$$P_{i,11} = -\left(10^{-3}\gamma_d d_{2,2} - .75\gamma_c (d_{1,2}c)_i d_{1,2} - 10^2(1-2\rho_i)K(c_i) - 4\right), \tag{6–12}$$

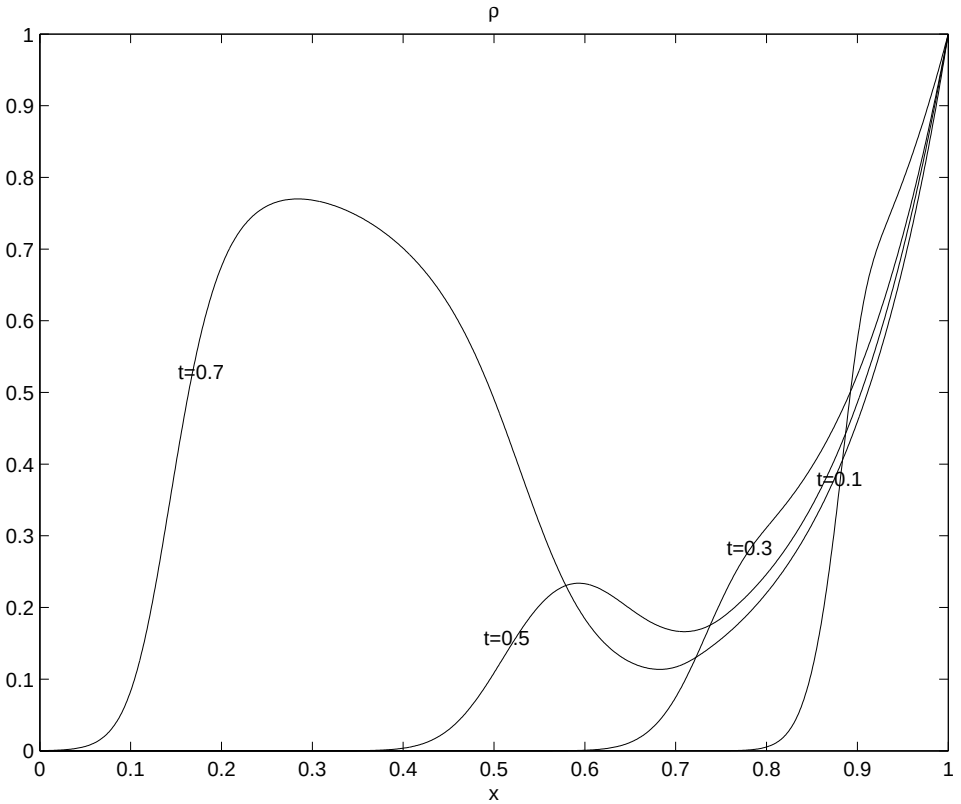


Figure 15. Fine grid solution of the angiogenesis equation: ρ.

$$P_{i,22} = -\left(\gamma_d d_{2,2} - \frac{10\rho_i}{(1+c_i)^2} - 1\right). \tag{6–13}$$

Due, we believe, to the presence of the additional second order term, acceptable stability results led us to use a second order rather than a fourth order starting method. We tested for stability with $\Delta x = 1E(-2)$ and $\Delta x = 1E(-3)$ for $\gamma_d = 1, 2$ and $\gamma_c = 1, 2, 3$. As in the previous example, with $\gamma_d = 1$ it was necessary to take $\Delta T \propto \Delta x^2$. For $\gamma_d = 2$, on the other hand, it was possible to choose ΔT independent of Δx. However, in contrast with the previous case, it was not possible to take ΔT large. With $\gamma_c = 2$ we found $\Delta T \leq 1E(-2)$ while with $\gamma_c = 0, 1$ we could choose $\Delta T \leq 1.7E(-2)$. However, we did observe better accuracy for large steps with $\gamma_c = 0$ than with $\gamma_c = 1$.

The accuracy of the computed solutions with large steps, $\gamma_d = 2$, and $\gamma_c = 0$ is displayed in Table 2. Obviously the results are consistent with the design accuracy.

Table 2. Error data for the angiogenesis problem: ρ.

ΔT	Δx	$e_{\max,\rho}$	q_ρ	$e_{\max,c}$	q_c
$1.\bar{6}E(-2)$	$1E(-2)$	$4.2(-3)$		$1.4(-4)$	
$8.\bar{3}E(-3)$	$5E(-3)$	$1.5(-5)$	8.1	$1.5(-6)$	6.6

As we now require more corrections, Strang splitting was carried out with 24 times as many steps as taken by the SDC solver - precisely 1008 and 2016 steps compared with the 42 and 84 which produced the results in Table 2. For the coarser grid, the results with Strang splitting were slightly more accurate than those obtained with the proposed method. On the finer grid, however, the SDC results were about an order of magnitude better.

6.3. ***Pulsating flame with stiff kinetics.*** Lastly, we consider a simplified thermo-diffusive combustion model with a stiff, intermediate reaction (e.g., [4]):

$$Y_t = \frac{1}{\mathcal{L}_Y}\left(Y_{xx} + \frac{1}{x}Y_x\right) - \frac{V}{x}Y_x - k_1, \tag{6–14}$$

$$W_t = \frac{1}{\mathcal{L}_W}\left(W_{xx} + \frac{1}{x}W_x\right) - \frac{V}{x}W_x + k_1 - k_2, \tag{6–15}$$

$$\Theta_t = \Theta_{xx} + \frac{1}{x}\Theta_x - \frac{V}{x}\Theta_x + k_1 + \gamma(k_2 - k_1), \tag{6–16}$$

$$k_1 = A_1 Y \exp\left(E_1 \frac{(1-\sigma)(\Theta-1)}{\sigma + \Theta(1-\sigma)}\right). \tag{6–17}$$

$$k_2 = A_2 W^2 \exp\left(E_2 \frac{(1-\sigma)(\Theta-1)}{\sigma + \Theta(1-\sigma)}\right). \tag{6–18}$$

The parameters are taken to be:

$$V = 11.7, \quad E_1 = 40, \quad E_2 = 2, \quad \mathcal{L}_Y = 2, \quad \mathcal{L}_W = \frac{3}{2}, \tag{6–19}$$

$$\sigma = \frac{1}{2}, \quad A_1 = 2\times 10^2, \quad A_2 = 2\times 10^6. \tag{6–20}$$

Note that if we assume the fast reaction is in balance with the slow reaction, that is if we assume $k_1 = k_2$, we obtain a reduced model with one species at a Lewis number, $\mathcal{L}_Y = 2$, with a pulsating solution (e.g., [5]). Initial and Dirichlet boundary conditions were obtained by interpolating a pulsating solution of the reduced problem on the spatial domain $5 \le x \le 35$. The initial W profile is then obtained through the quasiequilibrium assumption and the full system is evolved

Table 3. Observed maximum relative errors for $t \leq 20$, $m = 4$ preconditioned Radau methods applied to the pulsating flame problem. The order is calculated by $q = \log(e_{N_2}/e_{N_1})/\log(N_1/N_2)$ where N is the number of time steps and e_N is the maximum absolute error.

$\Delta T = 3.2E(-3)$, $\Delta x = 1.2E(-2)$			$\Delta T = 1.6E(-3)$, $\Delta x = 6.0E(-3)$			q		
Y	W	Θ	Y	W	Θ	Y	W	Θ
3.1(−4)	5.0(−3)	1.0(−4)	2.5(−5)	3.2(−4)	7.9(−6)	3.6	4.0	3.7

up to $t = 20$. Plots of the computed profiles on the finest grids, $\Delta T = 4.1\bar{6}E(-5)$, $\Delta x = 1.25E(-3)$, illustrating the flame oscillation are presented in Figure 16.

Note that around $t = 3.5$ the quasisteady initial flame destabilizes and moves towards the fuel source. An oscillation is set up between times 13 and 19.

The preconditioner in this case is as in the previous examples; second order spatial derivatives are approximated by $\gamma_d d_{2,2}$ and first order by $\gamma_c d_{1,2}$. We also include the Jacobian of the reaction terms. As in the preceding case we found it better to use a second order starting method. Choosing $\gamma_d = 2$ the time step limits were independent of γ_c and Δx, with a minimum time step of approximately $\Delta T = 3.8E(-3)$. Choosing $\gamma_d = 1$, on the other hand, required $\Delta T \propto \Delta x^2$ as in the previous examples.

We compare the accuracy of results obtained with $\Delta T = 3.2E(-3)$, $\Delta x = 1.2E(-2)$ and $\Delta T = 1.6E(-3)$, $\Delta x = 6E(-3)$. Here we have taken $\gamma_d = \gamma_c = 2$. The observed maximum errors, listed in Table 3, are consistent with 4th rather than 7th order convergence. This is the order of convergence expected for highly stiff problems, being equal to the stage order of the associated Runge–Kutta method. We note that, as might be expected for an oscillatory solution, the maximum errors are out of phase and occur at very different times for the two resolutions. Recently, Huang et al. [17] have shown how the order reduction phenomenon can be eliminated through the use of GMRES-based convergence acceleration which would no doubt improve our results in this case.

We have also solved this problem using Strang splitting and 24 times as many steps. As in the previous example, the results are slightly more accurate for the coarse resolution but less accurate for the fine resolution. We are confident that an improved implementation of the SDC method as in [17] would prove to be significantly more efficient than the traditional method.

7. Conclusion

In summary, we have shown that spectral deferred correction applied to a first order splitting method can:

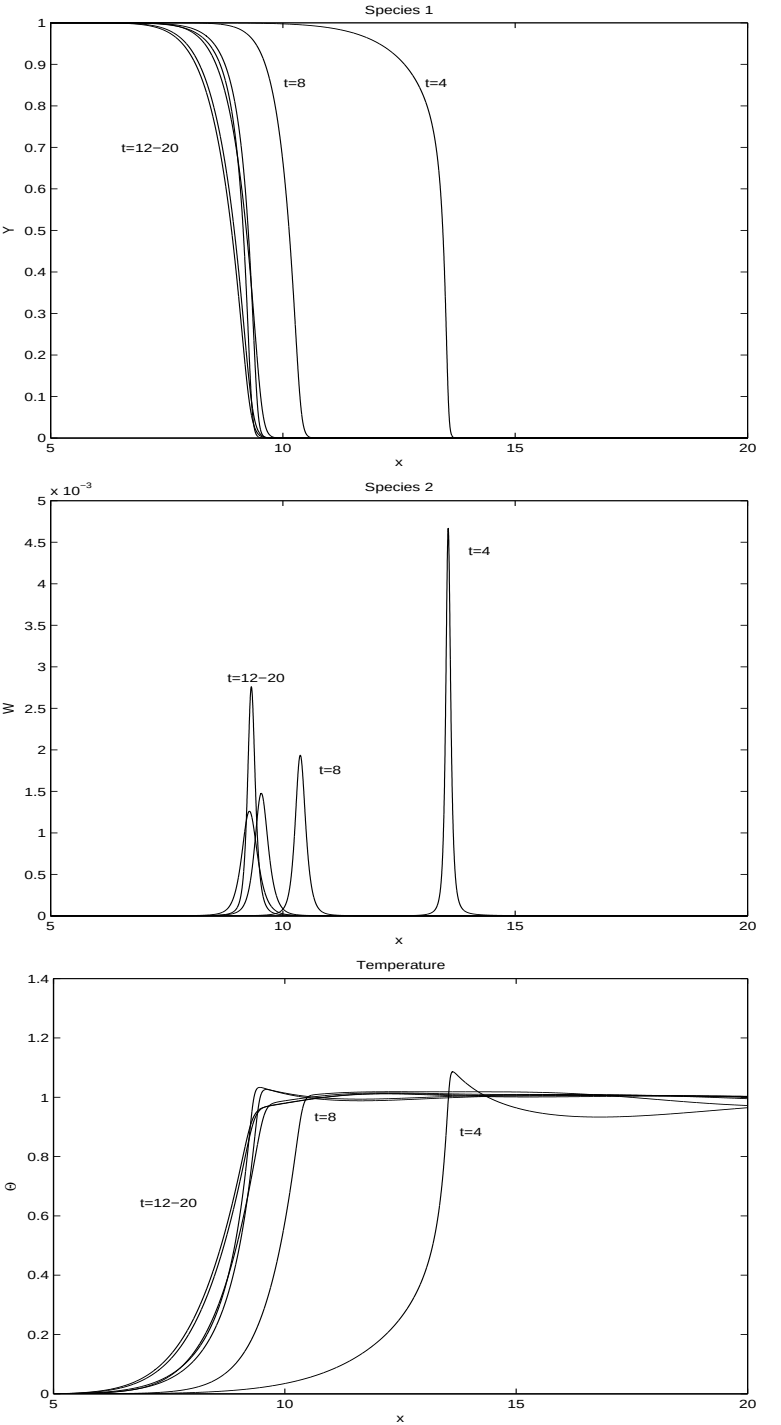


Figure 16. Fine grid solution of the flame equation.

i: Attain the full accuracy of the underlying quadrature rule;

ii: Have large stability domains.

We have also explored a general and flexible technique based on the concept of splitting by preconditioning. We have demonstrated the effectiveness of a particular instance of this strategy for reaction-advection-diffusion equations in one space dimension where high order difference approximations were preconditioned by lower order approximations with far narrower bandwidths. So long as the preconditioner was large enough in comparison with the true Jacobian, time step stability constraints independent of the spatial deiscretization were observed. This is in line with our experience solving complex combustion models [12; 11; 24]. Moreover, despite the very simple choice for the preconditioner and the fact that no convergence acceleration was employed, the methods were always as efficient and in some instances far more efficient than the standard Strng splitting approach.

Of course the greatest potential payoffs in terms of efficiency are for problems in multiple space dimensions. The fundamental issue is how simple (that is inexpensive) a preconditioner can be used without sacrificing too much accuracy or stability. It is also of interest to combine the preconditioned time-stepping strategy with the GMRES-based acceleration techniques described in [17]. We believe these issues deserve further study.

References

[1] U. M. Ascher, S. J. Ruuth, and B. T. R. Wetton, *Implicit-explicit methods for time-dependent partial differential equations*, SIAM J. Numer. Anal. **32** (1995), no. 3, 797–823. MR 96j:65076 Zbl 0841.65081

[2] W. Auzinger, H. Hofstätter, W. Kreuzer, and E. Weinmüller, *Modified defect correction algorithms for ODEs, II: Stiff initial value problems*, Tech. Report ANUM 2/03, Vienna University of Technology, 2003.

[3] ———, *Modified defect correction algorithms for ODEs, I: General theory*, Tech. report, Numer. Alg., 2004.

[4] A. Bayliss, M. Garbey, and B. Matkowsky, *Adaptive pseudo-spectral domain decomposition and the approximation of multiple layers*, J. Comput. Phys. **119** (1995), 132–141.

[5] A. Bayliss, D. Gottlieb, B. Matkowsky, and M. Minkoff, *An adaptive pseudo-spectral method for reaction diffusion problems*, J. Comput. Phys. **81** (1989), 421–443.

[6] A. Bourlioux, A. T. Layton, and M. L. Minion, *High-order multi-implicit spectral deferred correction methods for problems of reactive flow*, J. Comput. Phys. **189** (2003), no. 2, 651–675. MR 2004f:76084 Zbl 1061.76053

[7] A. Dutt, L. Greengard, and V. Rokhlin, *Spectral deferred correction methods for ordinary differential equations*, BIT **40** (2000), no. 2, 241–266. MR 2001e:65104 Zbl 0959.65084

[8] J. Frank, W. Hundsdorfer, and J. G. Verwer, *On the stability of implicit-explicit linear multistep methods*, Appl. Numer. Math. **25** (1997), no. 2-3, 193–205. MR 98m:65126 Zbl 0887.65094

[9] R. Frank, J. Hertling, and H. Lehner, *Defect correction algorithms for stiff ordinary differential equations*, Defect correction methods (Oberwolfach, 1983), Comput. Suppl., no. 5, Springer, Vienna, 1984, pp. 33–41. MR 86e:65094

[10] A. Hagstrom and G. Hagstrom, *Grid stabilization of high-order one-sided differencing, I: First order hyperbolic systems*, preprint, 2005.

[11] T. Hagstrom, K. Radhakrishnan, S. Steinberg, and R. Zhou, *Simulation of unsteady combustion phenomena using complex models*, Tech. Report 99-2397, AIAA, 1999.

[12] T. Hagstrom, K. Radhakrishnan, and R. Zhou, *Computation of steady and unsteady laminar flames: theory*, Tech. Report 1998-3246, AIAA, 1998.

[13] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving ordinary differential equations. I*, Springer Series in Computational Mathematics, no. 8, Springer, Berlin, 1993. MR 94c:65005

[14] E. Hairer and G. Wanner, *Solving ordinary differential equations. II*, Springer Series in Computational Mathematics, no. 14, Springer, Berlin, 1996. MR 97m:65007

[15] A. Hansen and J. Strain, *Convergence theory for spectral deferred correction*, Preprint, 2006.

[16] ———, *On the order of deferred correction*, Preprint, 2006.

[17] J. Huang, J. Jia, and M. Minion, *Accelerating the convergence of spectral deferred correction methods*, J. Comput. Phys. **214** (2006), 633–656.

[18] W. Hundsdorfer and J. Verwer, *Numerical solution of time-dependent advection-diffusion-reaction equations*, Springer Series in Computational Mathematics, no. 33, Springer, Berlin, 2003. MR 2004g:65001

[19] A. T. Layton and M. L. Minion, *Conservative multi-implicit spectral deferred correction methods for reacting gas dynamics*, J. Comput. Phys. **194** (2004), no. 2, 697–715. MR 2004k:76089 Zbl 02056059

[20] ———, *Implications of the choice of quadrature nodes for Picard integral deferred corrections methods for ordinary differential equations*, BIT **45** (2005), no. 2, 341–373. MR 2006h:65087 Zbl 1078.65552

[21] M. L. Minion, *Semi-implicit spectral deferred correction methods for ordinary differential equations*, Commun. Math. Sci. **1** (2003), no. 3, 471–500. MR 2005f:65085 Zbl 1088.65556

[22] D. L. Ropp, J. N. Shadid, and C. C. Ober, *Studies of the accuracy of time integration methods for reaction-diffusion equations*, J. Comput. Phys. **194** (2004), no. 2, 544–574. MR MR2034857 Zbl 1039.65069

[23] P. E. Zadunaisky, *On the estimation of errors propagated in the numerical integration of ordinary differential equations*, Numerische Math. **27** (1976/77), no. 1, 21–39. MR 55 #4691 Zbl 0324.65035

[24] R. Zhou, T. Hagstrom, K. Radhakrishnan, and S. Steinberg, *Numerical methods for reaction-diffusion equations with complex models*, In preparation, 2006.

[25] R. Zhou, M. G. Forest, and Q. Wang, *Kinetic structure simulations of nematic polymers in plane Couette cells. I. The algorithm and benchmarks*, Multiscale Model. Simul. **3** (2005), no. 4, 853–870. MR 2006c:76010 Zbl 02212496

Received August 23, 2005.

THOMAS HAGSTROM: hagstrom@math.unm.edu
Department of Mathematics and Statistics, The University of New Mexico, Albuquerque, NM 87131, United States

RUHAI ZHOU: *Department of Mathematics and Statistics, Old Dominion University, Norfolk, VA 23529, United States*
rzhou@odu.edu