A numerical study of diagonally split Runge–Kutta methods for PDEs with discontinuities

Colin B. Macdonald*, Sigal Gottlieb†, and Steven J. Ruuth‡

June 3, 2007

Abstract

Diagonally split Runge–Kutta (DSRK) time discretization methods [1] are a class of implicit time-stepping schemes which offer both high-order convergence and a form of strong stability known as unconditional contractivity. This combination is not possible within the classes of Runge–Kutta or linear multistep methods and therefore appears promising for the strong stability preserving (SSP) time-stepping community [10] which is generally concerned with computing oscillation-free numerical solutions of PDEs. Using a variety of numerical test problems, we show that although second- and third-order DSRK methods do preserve the strong stability property for all time step-sizes, they suffer from order reduction at large step-sizes. Indeed, for time-steps larger than those typically chosen for explicit methods, DSRK methods behave like first-order implicit methods. This is unfortunate, since it is precisely to allow a large Δt that we choose to

*Department of Mathematics, Simon Fraser University, Burnaby, British Columbia, V5A 1S6 Canada (cbm@sfu.ca). The work of this author was partially supported by an NSERC Canada PGS-D scholarship, a grant from NSERC Canada, and a scholarship from the Pacific Institute of Mathematics (PIMS).

†Department of Mathematics, University of Massachusetts Dartmouth, North Dartmouth MA 02747 USA (sgottlieb@umassd.edu). This work was supported by AFOSR grant number FA9550-06-1-0255.

‡Department of Mathematics, Simon Fraser University, Burnaby, British Columbia, V5A 1S6 Canada (sruuth@sfu.ca). The work of this author was partially supported by a grant from NSERC Canada.

use implicit methods. These results suggest that unconditionally contractive DSRK methods are limited in usefulness as they are unable to compete with either the first-order backward Euler method for large step-sizes or with Crank-Nicolson or high-order explicit SSP Runge–Kutta methods for smaller step-sizes.

1 Introduction

Strong stability preserving (SSP) high-order time discretizations [36, 38, 14] were developed for the solution of semi-discrete method-of-lines approximations of hyperbolic partial differential equations (PDEs) with discontinuous solutions. In such cases, carefully constructed spatial discretization methods guarantee a desired strong stability property (for example, that the solution be free of oscillations) when coupled with first-order forward Euler (FE) time-stepping. However, for practical computation, higher-order time discretizations are usually needed, and there is no guarantee that the nonlinearly stable spatial discretization will produce stable results when coupled with a linearly stable higher-order time discretization. In fact, numerical evidence [13, 14] shows that oscillations may occur when using a linearly stable, high-order time discretization which does not preserve the stability properties of forward Euler, even if the same spatial discretization is total variation diminishing (TVD) when combined with the first-order forward Euler time-discretization. SSP methods are high-order time discretization methods that preserve the strong stability properties—in any norm or semi-norm—of the spatial discretization coupled with forward Euler time-stepping.

The idea behind SSP methods is to assume that the spatial discretization is strongly stable under a certain semi-norm when coupled with the forward Euler time discretization, for a suitably restricted time-step, and then try to find a higher-order time discretization that maintains strong stability for the same semi-norm, perhaps under a different time-step restriction. The class of high-order SSP time discretization methods for the semi-discrete method-of-lines approximations of PDEs was developed in [38, 36] and was at that time known as TVD time discretizations. This class of methods was further studied in [13, 11, 33, 20, 40, 41, 34, 32, 37, 10]. These methods preserve the stability properties of forward Euler in any norm or semi-norm. In fact, since the stability arguments are based on convex decompositions of high-order methods in terms of the forward Euler method, any convex function (such as the cell entropy stability property of high-order schemes studied in [31, 29]) will be preserved by SSP high-order time discretizations. These SSP time discretizations can then be safely used with any spatial discretization which has the required stability properties when coupled with forward Euler.

The drawback of explicit SSP methods is that they suffer from restrictive time-step conditions. To obviate these difficulties we turn to implicit time-stepping methods with SSP properties. It was shown in [20] and [16], that any spatial discretization which is strongly stable in some semi-norm for the explicit forward Euler method under a certain time restriction will also be strongly stable, in the same semi-norm, with the implicit backward Euler (BE) method, *without* a time-step restriction. In previous work [14], efforts have been made to design higher-order implicit methods which share the strong stability properties of backward Euler, without any restriction on the time-step. Unfortunately, this goal cannot be realized for methods within the class of Runge–Kutta (RK) or linear multistep methods. For both implicit Runge–Kutta and multistep methods it has been proved that any higher-order SSP method, even for linear constant coefficient problems, will have some time-step restriction [14, 39]. This step-size restriction becomes apparent even in the simplest computations. An example of this is seen in Section 2, Figure 1 and Section 4, Table 3 where the solution to a linear advection equation is discretized using a TVD forward difference spatial discretization and the implicit Crank–Nicolson (CN) time discretization. The numerical solution develops oscillations when the time-step restriction is exceeded. However, when the first-order, unconditionally SSP backward Euler method is used with this spatial discretization, the numerical solution remains TVD even for large step sizes.

To identify methods with no step-size restriction, we must extend our search beyond the standard Runge–Kutta and linear multistep methods. One such class, in particular, is the family of diagonally split Runge–Kutta methods (DSRK) [1, 2, 18, 21], which have been shown to allow for unconditional contractivity. In this paper, we study these DSRK methods—which may satisfy the SSP property with no step-size restriction for certain classes of problems—and compare their performance to standard implicit and explicit time-stepping methods. The paper is structured as follows: in Section 2, we describe the construction of SSP RK methods and review the results for explicit and implicit SSP RK methods. In Section 3 we introduce the DSRK methods and study their properties. In Section 4 we present numerical studies comparing DSRK with implicit and explicit RK methods, in terms of both accuracy and efficiency. In Section 5, we draw conclusions about the use of DSRK methods and future research directions.

2 Implicit and Explicit SSP Runge–Kutta Methods

We wish to approximate the solution of the ODE system

$$\boldsymbol{u}_t = L(\boldsymbol{u}), \tag{1}$$

with initial conditions $\boldsymbol{u}(t_0) = \boldsymbol{u}_0$, typically arising from the spatial discretization of the PDE

$$u_t + f(u)_x = 0,$$

in which case $\boldsymbol{u} = (u_j)$ is a vector which gives the numerical solution of the PDE at spatial points x_j, $j = 1, \ldots, m$. The spatial discretization $L(\boldsymbol{u})$ is often chosen so that forward Euler

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + \Delta t L(\boldsymbol{u}^n),$$

satisfies the strong stability requirement $||\boldsymbol{u}^{n+1}|| \leq ||\boldsymbol{u}^n||$ in some norm or semi-norm $||\cdot||$, under the CFL condition

$$\Delta t \leq \Delta t_{\mathrm{FE}}.$$

As in [38, 6], a general m-stage Runge–Kutta (RK) method for (1) is written in Shu–Osher form

$$\begin{aligned} \boldsymbol{u}^{(0)} &= \boldsymbol{u}^n, \\ \boldsymbol{u}^{(i)} &= \sum_{k=0}^{m} \left(\alpha_{ik} \boldsymbol{u}^{(k)} + \Delta t \beta_{ik} L(\boldsymbol{u}^{(k)}) \right), \quad \alpha_{ik} \geq 0, \quad i = 1, \ldots, m, \\ \boldsymbol{u}^{n+1} &= \boldsymbol{u}^{(m)}. \end{aligned} \tag{2}$$

Consistency requires that $\sum_{k=0}^{m} \alpha_{ik} = 1$. Thus, if $\alpha_{ik} \geq 0$ and $\beta_{ik} \geq 0$, all the intermediate stages in (2), $\boldsymbol{u}^{(i)}$, are simply convex combinations of backward Euler and forward Euler operators, with Δt replaced by $\frac{\beta_{ik}}{\alpha_{ik}} \Delta t$. Therefore, any norm, semi-norm or convex function property satisfied by both the backward Euler and forward Euler methods will be preserved by the RK method, under the time-step restriction

$$\Delta t \leq \min_{i \neq k} \frac{\alpha_{ik}}{\beta_{ik}} \Delta t_{\mathrm{FE}}, \tag{3}$$

where $\frac{\alpha_{ik}}{\beta_{ik}} = \infty$ if $\beta_{ik} = 0$. If the method consists of only forward Euler steps, we call it an *explicit Runge–Kutta method*, otherwise it is known as an *implicit Runge–Kutta method*. In the case where any of the $\beta_{ik} < 0$, some modification is necessary [36]; for further details on the development of schemes under this relaxed condition see [12, 34, 14].

Much of the research in the field of SSP methods centers around the search for high-order SSP methods where the allowable time-step is as large as possible. If a method has a SSP time-step restriction $\Delta t \leq \mathcal{C} \Delta t_{\mathrm{FE}}$, then we will often use $\mathcal{C}$, the *SSP coefficient* or *CFL coefficient*, to measure the allowable time-step of a method relative to that of forward Euler.

2.1 Explicit SSP Runge–Kutta methods

Many optimal methods have been found in [38, 13, 14, 11, 33, 41]. These methods include the case where there are more stages than the minimum required for the desired order, so as to maximize the allowable time-step. Although the additional stages increase the computational cost, this is often offset by the larger step-size that may be taken. The most popular explicit SSP RK methods are given below.

Two-stage, second-order SSPRK (SSPRK(2,2)) An optimal second-order SSP Runge–Kutta method is given by

$$\begin{aligned} \boldsymbol{u}^{(1)} &= \boldsymbol{u}^n + \Delta t L(\boldsymbol{u}^n), \\ \boldsymbol{u}^{n+1} &= \frac{1}{2}\boldsymbol{u}^n + \frac{1}{2}\boldsymbol{u}^{(1)} + \frac{1}{2}\Delta t L(\boldsymbol{u}^{(1)}). \end{aligned}$$

The step-size restriction for this method is $\Delta t \leq \Delta t_{\mathrm{FE}}$, which means that it has a SSP coefficient of $\mathcal{C} = 1$. However, note that the computational work required is doubled compared to forward Euler.

Three-stage, third-order SSPRK (SSPRK(3,3)) An optimal third-order SSP Runge–Kutta method is given by

$$\begin{aligned}
\boldsymbol{u}^{(1)} &= \boldsymbol{u}^n + \Delta t L(\boldsymbol{u}^n), \\
\boldsymbol{u}^{(2)} &= \frac{3}{4}\boldsymbol{u}^n + \frac{1}{4}\boldsymbol{u}^{(1)} + \frac{1}{4}\Delta t L(\boldsymbol{u}^{(1)}), \\
\boldsymbol{u}^{n+1} &= \frac{1}{3}\boldsymbol{u}^n + \frac{2}{3}\boldsymbol{u}^{(2)} + \frac{2}{3}\Delta t L(\boldsymbol{u}^{(2)}).
\end{aligned}$$

The step-size restriction for this method is $\Delta t \leq \Delta t_{\mathrm{FE}}$, so it has a value of $\mathcal{C} = 1$. However, the computational work in this method is three times that of forward Euler. This method is very commonly used and is also known as the third-order TVD Runge-Kutta scheme or the Shu–Osher method.

Five-stage, fourth-order SSPRK (SSPRK(5,4)) An optimal method developed in [40, 32] with coefficients expressed to 15 digits is

$$\begin{aligned}
\boldsymbol{u}^{(1)} &= \boldsymbol{u}^n + 0.391752226571890\Delta t L(\boldsymbol{u}^n), \\
\boldsymbol{u}^{(2)} &= 0.444370493651235\boldsymbol{u}^n + 0.555629506348765\boldsymbol{u}^{(1)} \\
&\quad + 0.368410593050371\Delta t L(\boldsymbol{u}^{(1)}), \\
\boldsymbol{u}^{(3)} &= 0.620101851488403\boldsymbol{u}^n + 0.379898148511597\boldsymbol{u}^{(2)} \\
&\quad + 0.251891774271694\Delta t L(\boldsymbol{u}^{(2)}), \\
\boldsymbol{u}^{(4)} &= 0.178079954393132\boldsymbol{u}^n + 0.821920045606868\boldsymbol{u}^{(3)} \\
&\quad + 0.544974750228521\Delta t L(\boldsymbol{u}^{(3)}), \\
\boldsymbol{u}^{n+1} &= 0.517231671970585\boldsymbol{u}^{(2)} \\
&\quad + 0.096059710526146\boldsymbol{u}^{(3)} + 0.063692468666290\Delta t L(\boldsymbol{u}^{(3)}) \\
&\quad + 0.386708617503269\boldsymbol{u}^{(4)} + 0.226007483236906\Delta t L(\boldsymbol{u}^{(4)}).
\end{aligned}$$

The step-size restriction for this method is approximately $\Delta t \leq 1.508\Delta t_{\mathrm{FE}}$, which means that it has a value of $\mathcal{C} \approx 1.508$. The computational work in this method is five times that of forward Euler, but the allowable time-step makes this method almost as efficient as the SSPRK(3,3) method, yet higher order.

In the development of new methods and in the numerical tests below, these explicit methods will serve as the gold standard, to be compared to implicit methods in terms of the time-step allowed and the computational cost required.

2.2 Implicit SSP methods

Often, total variation diminishing (TVD) spatial discretizations are constructed in conjunction with the forward Euler method. The implicit backward Euler method will also preserve this property for all step-sizes, but a different time-discretization, such as the second-order Crank–Nicolson (CN) method, may only preserve the TVD property for a limited range of step-sizes. For example, consider the case of the linear wave equation

$$u_t + au_x = 0,$$

with $a = -2\pi$, a step-function initial condition

$$u(x, 0) = \begin{cases} 1 & \text{if } \frac{\pi}{2} \le x \le \frac{3\pi}{2}, \\ 0 & \text{otherwise,} \end{cases}$$

and periodic boundary conditions on the domain $(0, 2\pi]$. The solution is a step function convected around the domain. For a simple first-order forward-difference TVD spatial discretization $L(\boldsymbol{u})$ of $-au_x$, the result will be TVD for all sizes of Δt when using the implicit backward Euler method. If we use the forward Euler time-stepping, the result is TVD for $\Delta t \le \Delta t_{\text{FE}} = \frac{\Delta x}{|a|}$. On the other hand, consider the Crank–Nicolson method

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + \frac{1}{2}\Delta t L(\boldsymbol{u}^n) + \frac{1}{2}\Delta t L(\boldsymbol{u}^{n+1}). \tag{4}$$

As (4) is in Shu–Osher form [36], we see from (3) that it is SSP only for values $\Delta t \le 2\Delta t_{\text{FE}}$. This restriction is illustrated in Figure 1 where an excessively large Δt leads to oscillations and a clear violation of the TVD property.

Crank–Nicolson requires extra computational cost due to the solution of an implicit system, but with respect to strong stability only allows a doubling of the step-size compared to forward Euler or the second-order SSPRK(2,2). This means that, in general, it will not be efficient to use this method. The major focus of our work is the search for implicit RK methods which have no time-step restriction. The first-order backward Euler method is one such method. Unfortunately, there are no Runge–Kutta or linear multistep methods of order greater than one which will satisfy this property [39, 19]. However, if we search outside these classes, we can find higher-order methods which are unconditionally SSP. One such class is the family of diagonally split Runge–Kutta (DSRK) methods.

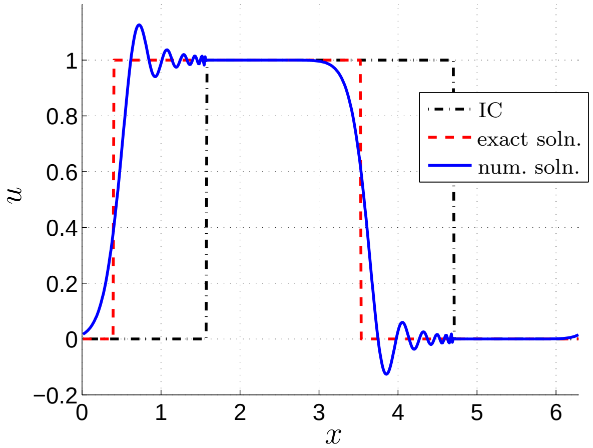


Figure 1: Oscillations from Crank–Nicolson time-stepping in the advection of a square wave with $\Delta t = 8\Delta t_{\mathrm{FE}} = 8\Delta x$ and $\Delta x = \frac{2\pi}{512}$.

3 Diagonally Split Runge–Kutta Methods

DSRK methods [1, 2, 21, 18] are one-step methods which are based on a Runge–Kutta formulation, but where the ODE operator L in (1) has different inputs used for the diagonal and off-diagonal components. We define the *diagonal splitting function* of L as

$$\mathfrak{L}_j(\boldsymbol{u}, \boldsymbol{z}) = L(z_1, z_2, \ldots, z_{j-1}, u_j, z_{j+1}, \ldots, z_m), \quad j = 1, \ldots, m, \tag{5}$$

that is, the j^{th} component of $\mathfrak{L}(\boldsymbol{u}, \boldsymbol{z})$ is computed using the j^{th} component of $\boldsymbol{u}$ for the j^{th} input of L and components of $\boldsymbol{z}$ for the other inputs of L.

The general DSRK method is

$$\boldsymbol{U}^i = \boldsymbol{u}^n + \Delta t \sum_{j=1}^{m} a_{ij} \mathfrak{L}(\boldsymbol{U}^j, \boldsymbol{Z}^j), \tag{6a}$$

$$\boldsymbol{Z}^i = \boldsymbol{u}^n + \Delta t \sum_{j=1}^{m} w_{ij} \mathfrak{L}(\boldsymbol{U}^j, \boldsymbol{Z}^j), \tag{6b}$$

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + \Delta t \sum_{j=1}^{m} b_j \mathfrak{L}(\boldsymbol{U}^j, \boldsymbol{Z}^j), \tag{6c}$$

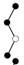
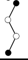
order 1	$\boldsymbol{b}^{\mathrm{T}}\boldsymbol{e} = 1$	
order 2	$\boldsymbol{b}^{\mathrm{T}}\mathbf{C}\boldsymbol{e} = \frac{1}{2}$	
order 3	$\boldsymbol{b}^{\mathrm{T}}\mathbf{C}^2\boldsymbol{e} = \frac{1}{3}$	
	$\boldsymbol{b}^{\mathrm{T}}\mathbf{W}\mathbf{C}\boldsymbol{e} = \frac{1}{6}$	$\boldsymbol{b}^{\mathrm{T}}\mathbf{A}\mathbf{C}\boldsymbol{e} = \frac{1}{6}$
order 4	$\boldsymbol{b}^{\mathrm{T}}\mathbf{C}^3\boldsymbol{e} = \frac{1}{4}$	
	$\boldsymbol{b}^{\mathrm{T}}\mathbf{C}\mathbf{W}\mathbf{C}\boldsymbol{e} = \frac{1}{8}$	$\boldsymbol{b}^{\mathrm{T}}\mathbf{C}\mathbf{A}\mathbf{C}\boldsymbol{e} = \frac{1}{8}$
	$\boldsymbol{b}^{\mathrm{T}}\mathbf{W}\mathbf{C}^2\boldsymbol{e} = \frac{1}{12}$	$\boldsymbol{b}^{\mathrm{T}}\mathbf{A}\mathbf{C}^2\boldsymbol{e} = \frac{1}{12}$
	$\boldsymbol{b}^{\mathrm{T}}\mathbf{W}^2\mathbf{C}\boldsymbol{e} = \frac{1}{24}$	 $\boldsymbol{b}^{\mathrm{T}}\mathbf{A}\mathbf{W}\mathbf{C}\boldsymbol{e} = \frac{1}{24}$
	$\boldsymbol{b}^{\mathrm{T}}\mathbf{W}\mathbf{A}\mathbf{C}\boldsymbol{e} = \frac{1}{24}$	 $\boldsymbol{b}^{\mathrm{T}}\mathbf{A}^2\mathbf{C}\boldsymbol{e} = \frac{1}{24}$

Table 1: The 14 order conditions for fourth-order DSRK schemes written in matrix form where $\mathbf{C} = \mathrm{diag}(\boldsymbol{c})$. See [2] for an explanation of the trees.

where the coefficients $(\mathbf{A}, \boldsymbol{b}^{\mathrm{T}}, \boldsymbol{c}, \mathbf{W})$ of the method must satisfy the order conditions in Table 1. We note that these include the order conditions of the so-called *underlying Runge–Kutta method* (i.e., conditions only on $\mathbf{A} = (a_{ij})$, $\boldsymbol{b}$, and $\boldsymbol{c}$) and are augmented by additional order conditions on the coefficients $\mathbf{W} = (w_{ij})$.

3.1 Dissipative systems and contractivity

Bellen et al. [1] introduced the class of DSRK methods for *dissipative systems* $\boldsymbol{u}_t = L(t, \boldsymbol{u})$. In the special case of the maximum norm $\|\cdot\|_\infty$, a dissipative system is characterized (see [2], following Theorem 4.1) by the condition

$$\sum_{j=1, j\neq i}^{m} \left| \frac{\partial L_i(t, \boldsymbol{u})}{\partial u_j} \right| \leq -\frac{\partial L_i(t, \boldsymbol{u})}{\partial y_i}, \qquad i = 1, \ldots, m,$$

for all $t \leq t_0$ and $\boldsymbol{u} \in \mathbb{R}^m$. We note in particular that our numerical test problems in Sections 4.1 and 4.2 satisfy this condition.

If the ODE system is dissipative, then solutions satisfy a *contractivity* property [39, 23, 43]. Specifically, if $\boldsymbol{u}(t)$ and $\boldsymbol{v}(t)$ are two solutions corresponding to initial conditions $\boldsymbol{u}(t_0)$ and $\boldsymbol{v}(t_0)$ then

$$\|\boldsymbol{u}(t) - \boldsymbol{v}(t)\| \leq \|\boldsymbol{u}(t_0) - \boldsymbol{v}(t_0)\|,$$

in some norm of interest. Naturally, if solutions to the ODE system obey a contractivity property then it is desirable that a numerical method for solving the problem be contractive as well, i.e., that given numerical solutions $\boldsymbol{u}_n$ and $\tilde{\boldsymbol{u}}_n$, $||\tilde{\boldsymbol{u}}_{n+1} - \boldsymbol{u}_{n+1}|| \leq ||\tilde{\boldsymbol{u}}_n - \boldsymbol{u}_n||$ (possibly subject to a time-step restriction).

It was shown in Theorem 3.3 of [2] that a nonconfluent DSRK method for which $(\mathbf{I} - \mathbf{AX})^{-1}$ exists for any diagonal matrix $\mathbf{X} = \mathrm{diag}(x_1, x_2, \ldots, x_s) \leq 0$ is unconditionally contractive in the maximum norm for any dissipative system if and only if:

$$|1 + \boldsymbol{b}^{\mathrm{T}}\mathbf{X}(\mathbf{I} - \mathbf{AX})^{-1}\boldsymbol{e}| + \|\boldsymbol{b}^{\mathrm{T}}\mathbf{X}(\mathbf{I} - \mathbf{AX})^{-1}\|_1 = 1, \tag{7a}$$

$$|1 + \mathbf{W}_j^{\mathrm{T}}\mathbf{X}(\mathbf{I} - \mathbf{AX})^{-1}\boldsymbol{e}| + \|\mathbf{W}_j^{\mathrm{T}}\mathbf{X}(\mathbf{I} - \mathbf{AX})^{-1}\|_1 = 1, \quad \text{for } j = 1, \ldots, s, \tag{7b}$$

for all $\mathbf{X} = \mathrm{diag}(x_1, x_2, \ldots, x_s) \leq 0$, where $\mathbf{W}_j$ indicates the j^{th} column of $\mathbf{W}$.

In [21], in ’t Hout showed that if a DSRK method is unconditionally contractive in the maximum norm, the underlying RK method is of classical order $p \leq 4$, and has stage order $\tilde{p} \leq 1$. In [18], Horváth studied the positivity of RK and DSRK methods, and showed that DSRK schemes can be unconditionally positive.

The results on DSRK methods in terms of positivity and contractivity appear promising when searching for implicit SSP schemes, because positivity, contractivity, and the SSP condition are all very closely related [16, 17, 5, 6, 23]. For example, a loss of positivity implies the loss of the max-norm SSP property. For Runge–Kutta methods a link has also been established between time-step restrictions under the SSP condition and contractivity, namely that the time-step restrictions under either property agree [5], thereby enabling the possibility of transferring results established for the contractive case to the SSP case [16], and vice versa. For multistep methods, the time-step restrictions coming from either an SSP or contractivity analysis are the same, as can be seen by examining the proofs appearing in [26, 25, 36]. If we include the starting procedure into the analysis, or if we consider boundedness (a related nonlinear stability property) rather than the SSP property, significantly milder time-step restrictions may arise [20]. However, even with this less restrictive boundedness property, we find that unconditional nonlinear stability is impossible for schemes that are more than first order [19]. The promise of DSRK method is that there exist higher-order implicit unconditionally contractive methods, and therefore possibly DSRK methods which are unconditionally SSP, in this class.

3.2 DSRK schemes

It is illustrative to examine (6) when the ODE operator L is linear. In this case, with matrix $\mathbf{L}$ decomposed into $\mathbf{L} = \mathbf{L}_{\mathrm{D}} + \mathbf{L}_{\mathrm{N}}$ where $\mathbf{L}_{\mathrm{D}} = \mathrm{diag}(\mathbf{L})$, we have $\mathfrak{L}(\boldsymbol{u}, \boldsymbol{z}) = \mathbf{L}_{\mathrm{D}}\boldsymbol{u} + \mathbf{L}_{\mathrm{N}}\boldsymbol{z}$ and (6) becomes

$$\boldsymbol{U}^i = \boldsymbol{u}^n + \Delta t \sum_{j=1}^{m} a_{ij} \left(\mathbf{L}_{\mathrm{D}} \boldsymbol{U}^j + \mathbf{L}_{\mathrm{N}} \boldsymbol{Z}^j \right), \tag{8a}$$

$$\boldsymbol{Z}^i = \boldsymbol{u}^n + \Delta t \sum_{j=1}^{m} w_{ij} \left(\mathbf{L}_{\mathrm{D}} \boldsymbol{U}^j + \mathbf{L}_{\mathrm{N}} \boldsymbol{Z}^j \right), \tag{8b}$$

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + \Delta t \sum_{j=1}^{m} b_j \left(\mathbf{L}_{\mathrm{D}} \boldsymbol{U}^j + \mathbf{L}_{\mathrm{N}} \boldsymbol{Z}^j \right), \tag{8c}$$

and thus we see that for a linear ODE system), DSRK methods decompose the system into diagonal and off-diagonal components and treat each differently.

We now list the DSRK schemes which are used in Section 4 for our numerical tests.

Second-order DSRK ("DSRK2") This second-order DSRK from [1] is based on the underlying two-stage, second-order implicit RK method specified by the Butcher tableau

$$\begin{array}{c|c} \boldsymbol{c} & \mathbf{A} \\ \hline & \boldsymbol{b}^{\mathrm{T}} \end{array} = \begin{array}{c|cc} 0 & \frac{1}{2} & -\frac{1}{2} \\ 1 & \frac{1}{2} & \frac{1}{2} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}, \qquad \text{combined with } \mathbf{W} = \begin{bmatrix} 0 & 0 \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}. \tag{9a}$$

Thus the DSRK2 scheme is

$$\boldsymbol{U}^1 = \boldsymbol{u}^n + \frac{1}{2}\Delta t \mathfrak{L}(\boldsymbol{u}^n, \boldsymbol{U}^1) - \frac{1}{2}\Delta t L(\boldsymbol{u}^{n+1}), \tag{9b}$$

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + \frac{1}{2}\Delta t \mathfrak{L}(\boldsymbol{u}^n, \boldsymbol{U}^1) + \frac{1}{2}\Delta t L(\boldsymbol{u}^{n+1}). \tag{9c}$$

Note that the $\boldsymbol{u}^{n+1}$ terms are not split. For linear problems, (9) becomes

$$\boldsymbol{U}^1 = \boldsymbol{u}^n + \frac{1}{2}\Delta t\left[\mathbf{L}_{\mathrm{N}}\boldsymbol{u}^n + \mathbf{L}_{\mathrm{D}}\boldsymbol{U}^1\right] - \frac{1}{2}\Delta t\left[\mathbf{L}\boldsymbol{u}^{n+1}\right], \tag{10a}$$

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + \frac{1}{2}\Delta t\left[\mathbf{L}_{\mathrm{N}}\boldsymbol{u}^n + \mathbf{L}_{\mathrm{D}}\boldsymbol{U}^1\right] + \frac{1}{2}\Delta t\left[\mathbf{L}\boldsymbol{u}^{n+1}\right]. \tag{10b}$$

Note also in the special case when $\mathbf{L}_{\mathrm{D}} = \mathbf{0}$, (10) decouples and (10b) is exactly the Crank–Nicolson method.

Third-order DSRK ("DSRK3") This formally third-order DSRK scheme [1, 2, 21] is based on the underlying RK method:

$$\begin{array}{c|c} \boldsymbol{c} & \mathbf{A} \\ \hline & \boldsymbol{b}^{\mathrm{T}} \end{array} = \begin{array}{c|ccc} 0 & \frac{5}{2} & -2 & -\frac{1}{2} \\ \frac{1}{2} & -1 & 2 & -\frac{1}{2} \\ 1 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array}, \qquad \text{combined with } \mathbf{W} = \begin{bmatrix} 0 & 0 & 0 \\ \frac{7}{24} & \frac{1}{6} & \frac{1}{24} \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{bmatrix}.$$

In our numerical experiments in Section 4.1, we will show that these DSRK methods behave as expected in terms of contractivity. Unfortunately, however, these methods suffer from order reduction. This is not completely unexpected, as [21] showed that the underlying RK methods must have stage order at most one, and low stage order—at least in RK schemes—is known to lead to order reduction [15]. We discuss order reduction further in Section 4.5 where, for comparison, we use a DSRK method which is based on the two-stage, second-order implicit RK method

$$\begin{array}{c|c} \boldsymbol{c} & \mathbf{A} \\ \hline & \boldsymbol{b}^{\mathrm{T}} \end{array} = \begin{array}{c|cc} \frac{1}{2} & \frac{3}{4} & -\frac{1}{4} \\ 1 & 1 & 0 \\ \hline & 1 & 0 \end{array}, \qquad \text{combined with } \mathbf{W} = \begin{bmatrix} \frac{1}{2} & 0 \\ 1 & 0 \end{bmatrix}. \tag{11}$$

We call this method DSRK2so2. The underlying RK scheme here has stage order two. Because the underlying method has stage order larger than one, the DSRK method based on it cannot be unconditionally contractive [21]. In numerical tests (not included) we observed that indeed, this DSRK2so2 violated the SSP condition for large enough Δt.

3.2.1 Higher-order DSRK schemes

Although unconditionally contractive second- and third-order DSRK methods such as DSRK2 and DSRK3 exist, so far no unconditionally contractive fourth-order DSRK methods have been found. Theorem 2.4 of [21] presents necessary conditions for DSRK schemes to be unconditionally contractive in the maximum norm. Specifically, in the proof, the following two necessary conditions are given:

$$\text{all principal minors of } \mathbf{A} - \boldsymbol{e}\boldsymbol{b}^{\mathrm{T}} \text{ are nonnegative,} \tag{12a}$$

$$\begin{aligned} &\text{for each } i \in \{1, 2, \ldots, s\},\ \det[(\mathbf{A} \leftarrow_i \boldsymbol{b}^{\mathrm{T}})(\mathcal{I})] \geq 0 \\ &\qquad\qquad\qquad \text{for every } \mathcal{I} \subset \{1, 2, \ldots, s\} \text{ with } i \in \mathcal{I}, \end{aligned} \tag{12b}$$

where the notation $\mathbf{M}(\mathcal{I})$ indicates the principal submatrix formed by selecting from $\mathbf{M}$ only those rows and columns indexed by $\mathcal{I}$. These necessary conditions are simpler than (7) because they involve neither the matrix $\mathbf{X}$ nor any matrix inverses. This latter property ensures the conditions can be written out as polynomial expressions which is ideal for the optimization software discussed next.

As a first step towards finding an unconditionally contractive four-stage fourth-order DSRK (DSRK44), we employ the proprietary Branch and Reduce Optimization Navigator (BARON) software [35] to search for DSRK44 satisfying conditions (12).

In [27, 32] BARON was used to find optimal explicit SSPRK schemes because the branch-and-reduce algorithm used by BARON can guarantee global optimality under certain factorable and boundedness conditions [42]. In [27, 32], the optimization was done by maximizing the SSP coefficient while constraining based on the Runge–Kutta order conditions. We begin by searching for *any* feasible methods by minimizing the sum of the squares of the $\boldsymbol{b}$ coefficients (because we anticipate that very large $\boldsymbol{b}$ coefficients would give poor schemes) while imposing the 14 order conditions in Table 1 and the 48 necessary conditions (12) as constraints. BARON ran for 30 days (on an Athlon MP 2800+ with 1 GiB of RAM) and was unable to find any feasible DSRK44 schemes satisfying even the necessary conditions (12). Constrained only by the order conditions, BARON was able to quickly find DSRK44 schemes. Thus while DSRK44 schemes certainly exist, BARON was unable to find any schemes within this class that satisfy the necessary conditions (12) for unconditional contractivity. On the other hand, BARON was able—during the first few minutes of its preprocessing step—to find five-stage fourth-order DSRK methods satisfying the necessary conditions (12). Altogether, this is a strong indication that unconditionally contractive DSRK44 methods do not exist. However, this does not constitute a proof because even after 30 days the program had not run to completion. We leave open the question of the existence of unconditionally contractive five-stage fourth-order DSRK schemes. However such schemes are still likely to suffer from the order reduction noted in Section 4.

3.3 Numerical implementation of DSRK

For linear problems, we implement DSRK using (8) by re-arranging all the unknowns into a larger linear system, in general $(2sm)\times(2sm)$ where m is the size of the linear system (1) and s is the number of stages in the underlying Runge–Kutta scheme, although particular coefficients may result in a smaller system. For example, DSRK2 (10) can be written as the $2m \times 2m$ system

$$\left[\begin{array}{c|c} \mathbf{I} - \frac{1}{2}\Delta t\mathbf{L}_{\mathrm{D}} & \frac{1}{2}\Delta t\mathbf{L} \\ \hline -\frac{1}{2}\Delta t\mathbf{L}_{\mathrm{D}} & \mathbf{I} - \frac{1}{2}\Delta t\mathbf{L} \end{array}\right] \begin{pmatrix} \boldsymbol{U}^1 \\ \boldsymbol{u}^{n+1} \end{pmatrix} = \begin{pmatrix} \boldsymbol{u}^n + \frac{1}{2}\Delta t\mathbf{L}_{\mathrm{N}}\boldsymbol{u}^n \\ \boldsymbol{u}^n + \frac{1}{2}\Delta t\mathbf{L}_{\mathrm{N}}\boldsymbol{u}^n \end{pmatrix},$$

where $\mathbf{I}$ represents the $m \times m$ identity. We then simply solve this linear system to advance one time-step. As is usually the case, nonlinear systems are considerably more difficult. For the non-linear problems, we use a numerical zero-finding method to solve the nonlinear equations.

All numerical computations are performed with Matlab versions 7.0 and 7.3 using double precision on x86 and x86-64 architectures. Linear systems were solved using Matlab's `backslash` operator, whereas for the nonlinear problems in Sections 4.4 and 4.5, we implement the diagonal splitting function (5), and use a black-box equation solver (Matlab's `fsolve`) directly on (6).

4 Numerical Results

We focus our numerical experiments on three types of problems: convection, diffusion, and convection-diffusion. SSP methods are perhaps most important for convection driven problems, such as hyperbolic problems with discontinuous solutions. The methods have also been used to treat problems where the slope or some derivative of the solution is discontinuous and, in particular, SSP schemes have been used widely to treat Hamilton–Jacobi equations (see, e.g., [30]). Many other problems of reaction-advection-diffusion type also can benefit strongly from nonlinearly stable time-stepping. For example time-stepping a spatially discretized Black–Scholes equation (an equation we consider in Section 4.3) can lead to spurious oscillations in the solution. These oscillations are particularly undesirable in option-pricing problems since they can lead to highly oscillatory results in the derivative based quantities (e.g., "the Greeks" γ and δ) that end-users are interested in.

4.1 Convection driven problems

An important prototype problem for SSP methods is the linear wave equation, or *advection equation*

$$u_t + au_x = 0, \qquad 0 \leq x \leq 2\pi \tag{13}$$

We consider (13) with $a = -2\pi$, periodic boundary conditions and various initial conditions. We use a method-of-lines approach, discretizing the interval $(0, 2\pi]$ into m points $x_j = j\Delta x$, $j = 1, \dots, m$, and then discretizing $-au_x$ with first-order upwind finite differences. We solve the resulting system (1) using the time-stepping schemes described in Sections 2 and 3.

4.1.1 Smooth initial conditions

Table 2 shows a convergence study for (13) with a fixed Δx and smooth initial data

$$u(0, x) = \sin(x).$$

The implicit time-discretization methods used are backward Euler (BE), Crank–Nicolson (CN), DSRK2 and DSRK3. We also evolve the system with several explicit methods: forward Euler (FE), SSPRK(2,2), SSPRK(3,3), and SSPRK(5,4). To isolate the effect of the time-discretization error, we exclude the effect of the error associated with the spatial discretization by comparing the numerical solution to the exact solution of the ODE system (1), rather than to the exact solution of the underlying PDE. In lieu of the exact solution we use a very accurate numerical solution obtained using MATLAB's `ode45` with minimal tolerances (`AbsTol` $= 1 \times 10^{-14}$, `RelTol` $= 1 \times 10^{-13}$). Table 2 shows that all the methods achieve their design order when Δt is

		discrete error, l_∞-norm							
c	N	BE	order	CN	order	DSRK2	order	DSRK3	order
4	16	0.518		0.0582		0.408		0.395	
2	32	0.336	0.62	0.0147	1.98	0.194	1.08	0.178	1.15
1	64	0.194	0.79	3.70e-3	2.00	0.0714	1.44	0.0590	1.59
$\frac{1}{2}$	128	0.105	0.89	9.25e-4	2.00	0.0223	1.68	0.0152	1.95
...		...		...		...		...	
$\frac{1}{32}$	2048	7.04e-3		3.61e-6		1.09e-4		1.21e-5	
$\frac{1}{64}$	4096	3.53e-3	1.00	9.04e-7	2.00	2.74e-5	1.99	1.61e-6	2.91
$\frac{1}{128}$	8192	1.77e-3	1.00	2.26e-7	2.00	6.87e-6	1.99	2.09e-7	2.95
c	N	FE	order	SSP22	order	SSP33	order	SSP54	order
2	32	unstable		unstable		unstable		2.66e-5	
1	64	0.265		7.43e-3		1.82e-4		1.66e-6	4.00
$\frac{1}{2}$	128	0.122	1.12	1.85e-3	2.01	2.27e-5	3.00	1.03e-7	4.01

Table 2: Convergence study for the linear advection of a sine wave to $t_f = 1$ using N time-steps, $m = 64$ points and a first-order upwinding spatial discretization. Here c measures the size of the step relative to Δt_{FE}.

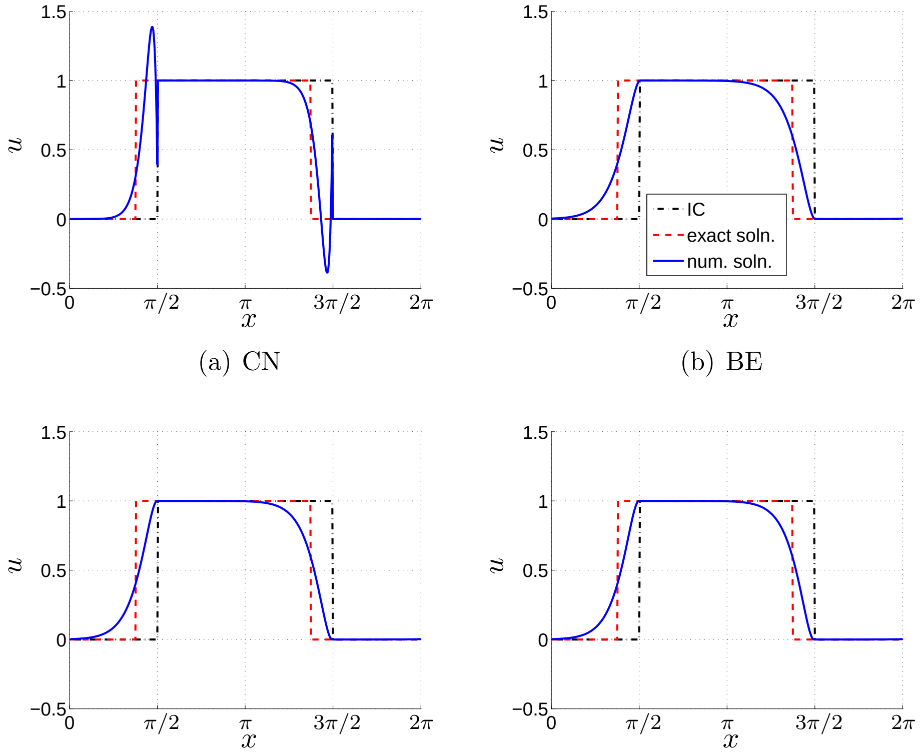
sufficiently small. However, the errors from CN are typically smaller than the errors produced by the other implicit methods. For large Δt, the second- and third-order DSRK schemes are far worse than CN. If we broaden our experiments to include explicit schemes, and take time-steps which are within the stability time-step restriction, we obtain smaller errors still. Given the relatively inexpensive cost of explicit time-stepping, it would appear that high-order explicit schemes (e.g., SSPRK(5,4)) are preferred for this smooth problem, unless, perhaps, very large time-steps are preferred over accuracy considerations.

4.1.2 Discontinuous initial conditions

We now consider the important case of advection of discontinuous data

$$u(x,0) = \begin{cases} 1 & \text{if } \frac{\pi}{2} \leq x \leq \frac{3\pi}{2}, \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

Figure 2 shows typical results. Note that oscillations are observed in the Crank–Nicolson results, while the DSRK schemes are free of such oscillations. In fact, Table 3 shows that for any time-step size BE, DSRK2 and DSRK3



(a) CN

(b) BE

(c) DSRK2

(d) DSRK3

Figure 2: Advection of a square wave after two time-steps, showing oscillations from Crank–Nicolson and none in the other methods. Here $c = 16$ and we take a first-order upwinding spatial discretization with $m = 512$ points in space.

preserve the TVD property of the spatial discretization coupled with forward Euler. In contrast, Crank–Nicolson exhibits oscillations for time-steps larger than $\Delta t = \frac{2}{|a|}\Delta x$ (i.e., $c > 2$).

4.1.3 Order reduction and scheme selection

We now delve deeper into the observed convergence rates of our smooth and nonsmooth problems.

Figures 3 and 4 show that for large time-steps, the DSRK methods exhibit behavior similar to backward Euler in that they exhibit large errors and as we decrease size of the time-steps, the error decreases at a rate which appears

		$\max_t TV(\boldsymbol{u})$				
c	N	exact	CN	BE	DSRK2	DSRK3
32	16	2	8.78	2	2	2
16	32	2	6.64	2	2	2
8	64	2	4.73	2	2	2
4	128	2	3.33	2	2	2
2	256	2	2	2	2	2
1	512	2	2	2	2	2

Table 3: Total variation of the solution for the advection of a square wave (N time-steps, $t_f = 1$). The spatial discretization uses $m = 512$ points, first-order upwinding, and periodic BCs.

only first order. As the time-steps are taken smaller still, the convergence rate increases to the design order of the DSRK schemes. In contrast, we note that Crank–Nicolson shows consistent second-order convergence over a wide range of time-steps.

On the discontinuous problem (Figure 4) we note the DSRK schemes do not produce significantly improved errors over backward Euler until the time-step sizes are small enough that Crank–Nicolson no longer exhibits spurious oscillations ($c = 2$ in Figure 4). In fact, once the time-steps are small enough that DSRK are competitive, we are almost within the nonlinear stability constraint of explicit methods such as SSPRK(2,2) ($c = 1$ in Figure 4) .

We note that neither Figure 3 nor Figure 4 takes into account the differences in computational work required by the various methods. The costs for DSRK2 and DSRK3 are significantly larger than BE and CN, because the underlying systems are larger. In the linear case, the size of the DSRK2 system is $2m \times 2m$ and the DSRK3 system is $5m \times 5m$ whereas the BE and CN systems are only $m \times m$. Even if the cost of solving the system rose only linearly with the size of the system, the cost is doubled for DSRK2 and increased five-fold for DSRK3. In reality, the cost may increase more rapidly, depending on the structure of the implicit system and the method used to solve the implicit equations. Furthermore, if a nonlinear system is solved, this cost may increase even further. It is even more difficult to quantify the increased cost of an implicit method over that of an explicit method. However, it is clear that implicit methods in general and DSRK methods in particular are significantly more costly than explicit methods.

We note that phase errors were also investigated to see if the DSRK

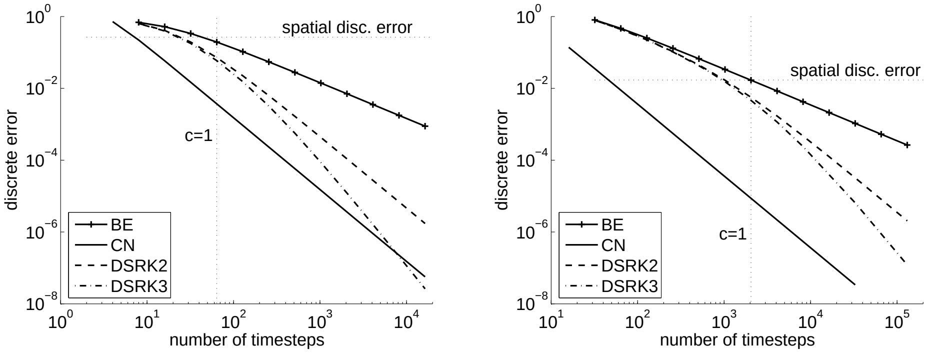


Figure 3: Convergence study for linear advection of a sine wave to $t_f = 1$. The spatial discretization here is first-order upwinding with 64 points (left) and 2048 points (right). We indicate the spatial discretization error with a dotted horizontal line.

schemes had improved phase error properties compared to BE but they do not. In general, for large Δt, DSRK methods behave similarly in many aspects to backward Euler.

In summary, our results on the advection equation show that although the DSRK method is formally high order, in practice we encounter a reduction of order for large time-steps. If one requires large time-steps and no oscillations, backward Euler is a good choice. If on the other hand, one requires accuracy, an explicit high-order SSP method is probably better suited. We will see that these results are typical for DSRK schemes.

4.2 Diffusion driven problems

Consider the diffusion or *heat equation*

$$u_t = \nu u_{xx}, \tag{15}$$

with heat coefficient ν on a periodic domain $(0, 2\pi]$. We begin by discretizing the u_{xx} term with second-order centered finite differences to obtain ODE system (1).

In Figure 5 and Table 4, we consider (15) with smooth initial conditions

$$u(0, x) = \sin(x) + \cos(2x).$$

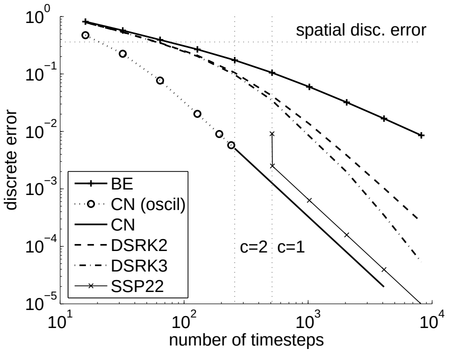


Figure 4: Linear advection of a square wave to time $t_f = 1$ using first-order upwinding and 512 points in space. Note that Crank–Nicolson produces oscillations during the computation for $c > 2$. We indicate the spatial discretization error with a dotted horizontal line.

Once again, we note that the DSRK schemes achieve their design order as Δt gets smaller, but for large time-steps they exhibit large errors and reduced convergence rates.

Figure 6 shows that Crank–Nicolson produces spurious oscillations in the solution to the heat equation with discontinuous initial conditions (14). Also, Figure 6 shows that the DSRK schemes are not competitive with backward Euler until the time-steps are smaller than the explicit stability limit (in this case, the restrictive $\Delta t \leq \frac{\Delta x^2}{2\nu}$ shown by the dotted vertical line). Clearly, DSRK methods exhibit order reduction for this parabolic problem as well.

4.3 The Black–Scholes equation

The Black–Scholes equation [3, 7, 8]

$$V_\tau = \frac{\sigma}{2} S^2 V_{SS} + rSV_S - rV, \tag{16}$$

is a PDE used in computational finance for determining the fair price V of an option at stock price S, where σ is the volatility and r is the risk-free interest rate. Note S is the independent (we can think "spatial") variable on the positive half-line and τ is a rescaled time (the actual time runs backwards from "final conditions").

We note that for our purposes, (16) is a linear non-constant coefficient advection-reaction-diffusion equation and we treat it as the ODE system (1)

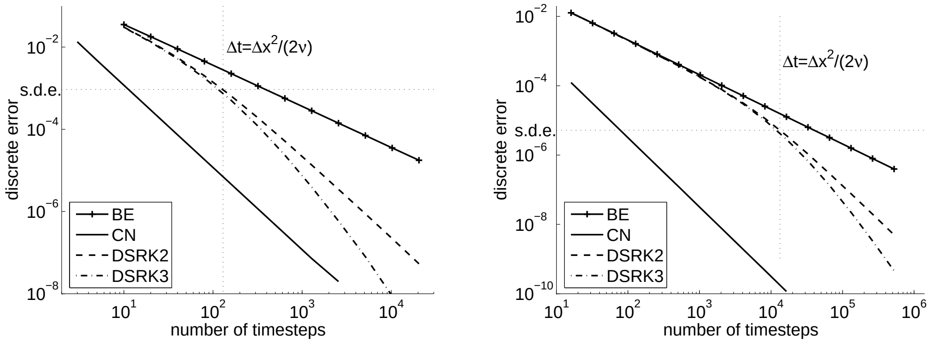


Figure 5: Convergence studies for the heat equation with smooth initial conditions. Left: $m = 64$, $t_f = 10$, $\nu = \frac{1}{16}$. Right: $m = 1024$, $t_f = 1$, $\nu = \frac{1}{4}$. Spatial discretization uses second-order centered differences and the level of spatial discretization error is indicated by the horizontal dotted line labeled "s.d.e."

		discrete error l_∞-norm			
c	N	BE	CN	DSRK2	DSRK3
830	16	0.0127	1.24e-4	0.0127	0.0127
415	32	0.00643	3.09e-5	0.00640	0.00640
...		...	...	...	...
12.97	1024	2.03e-4	3.02e-8	1.76e-4	1.74e-4
6.48	2048	1.02e-4	7.55e-9	7.77e-5	7.55e-5
1	13280	1.57e-5	1.80e-10	5.23e-6	4.28e-6
		FE	SSP(2,2)	SSP(3,3)	SSP(5,4)
2	6640	unstable	unstable	unstable	8.74e-13
1	13280	1.57e-5	3.59e-10	4.42e-13	1.32e-12

Table 4: Convergence study for the heat equation with smooth initial conditions. Here $\nu = 1/4$, $m = 1024$, $t_f = 1$. The discrete error is computed against the ODE solution calculated with MATLAB's `ode15s`. For comparison explicit methods are shown near their stability limits around $c = 1$.

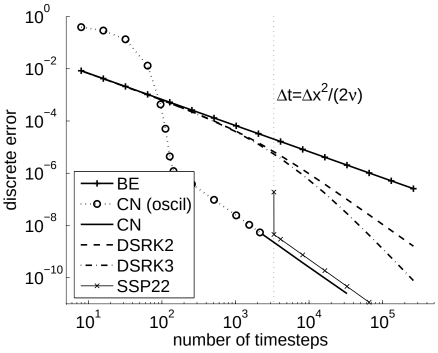


Figure 6: Heat equation with discontinuous initial conditions using $m = 512$, $t_f = 1$, and $\nu = \frac{1}{4}$. The spatial discretization in this example is second-order centered differences.

by approximating the V_S and V_{SS} terms with second-order centered finite differences. We use $\sigma = 0.8$, $r = 0.1$ and for this choice we did not notice any significant difference between upwind and centered differences for the advection term. We take the strike price $K = 100$ and consider a "put" option where the computational domain and initial conditions are shown in Figure 7. The right-hand boundary condition is an approximation to $\lim_{S\to\infty} V(S) = 0$, specifically $V(S_{\max}) = 0$. At the left-hand end of the domain, we note that (16) reduces to

$$\dot{V}_0 = -rV_0,$$

and thus it is both natural and convenient to simply solve this ODE coupled with the other components V_j as part of our method-of-lines computation.

Figure 8 shows the problem of oscillations which show up in a Crank–Nicolson calculation of the Black–Scholes problem. The oscillations are amplified in "the Greeks" i.e., the first and second spatial derivatives. We note this is a well-known phenomenon [4] associated with the CN numerical solution of (16); in practice, Rannacher time-stepping consisting of several initial steps of BE followed by CN steps [9] is often used to avoid these oscillations. DSRK schemes also avoid oscillations but are not likely competitive with Rannacher time-stepping in terms of efficiency due to the order reduction illustrated in Table 5. A great number of time-steps ($N = 17800$ in the case considered in Table 5) are required before the Crank–Nicolson calculation is completely oscillation-free in "the Greeks".

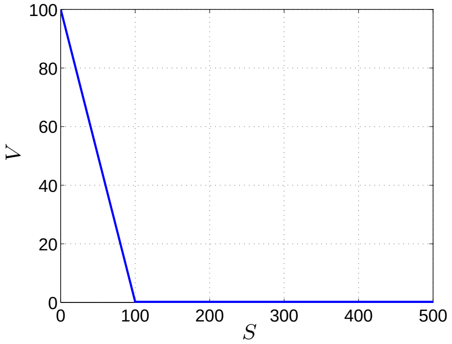


Figure 7: Computational domain and initial conditions for the Black–Scholes problem.

	discrete error l_∞-norm							
N	BE	order	CN	order	DSRK2	order	DSRK3	order
32	0.0655		0.115 *		0.0654		0.0653	
64	0.0328	1.00	0.0451 *	1.35	0.0327	1.00	0.0326	1.00
128	0.0164	1.00	8.63e-3 *	2.39	0.0163	1.00	0.0163	1.00
256	8.20e-3	1.00	8.74e-5 *	6.63	8.07e-3	1.01	8.06e-3	1.02
512	4.10e-3	1.00	1.95e-6 *	5.49	3.97e-3	1.02	3.96e-3	1.03
1024	2.05e-3	1.00	4.88e-7 *	2.00	1.92e-3	1.05	1.91e-3	1.05
...	...		...		...		...	
8192	2.56e-4		7.62e-9 *		1.60e-4		1.51e-4	
16384	1.28e-4	1.00	1.90e-9 *	2.00	5.66e-5	1.50	4.98e-5	1.60
32768	6.41e-5	1.00	4.74e-10	2.00	1.78e-5	1.67	1.67e-5	1.58

Table 5: Black–Scholes convergence study. * indicates oscillations in "the Greeks". Here, $m = 1600$, $S_{\max} = 400$, $\Delta x = \frac{1}{4}$, $t_f = 0.25$, $\sigma = 0.8$, $r = 0.1$, and strike price $K = 100$. The discrete error is calculated against a numerical solution from MATLAB's `ode15s` with `AbsTol` $= 1 \times 10^{-14}$, `RelTol` $= 1 \times 10^{-13}$.

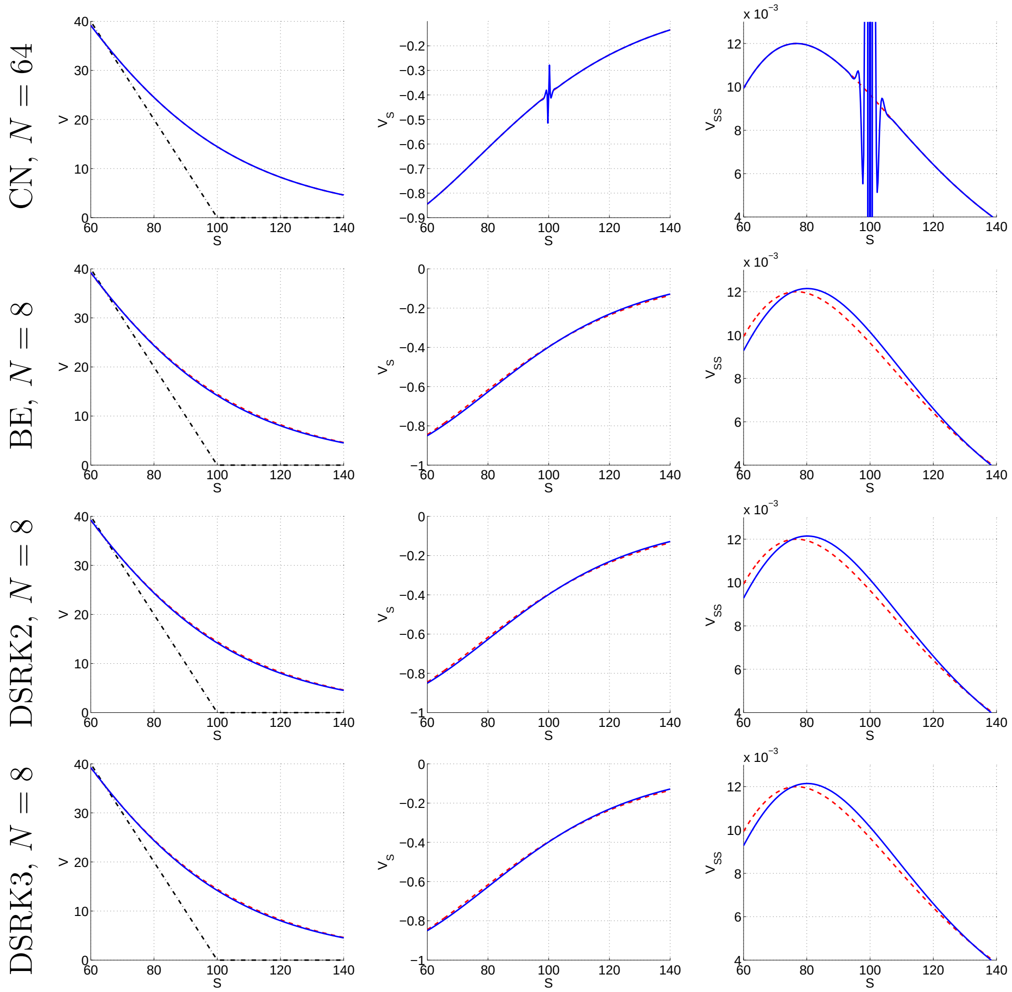


Figure 8: Numerical solutions of the Black–Scholes problem magnified near the strike price of $K = 100$ with $m = 2000$, $T_f = 0.25$, $\sigma = 0.8$, $r = 0.1$, and $S_{\max} = 500$ using N time-steps. From left-to-right: the option price V, the option δ (i.e., V_S) and the option γ (i.e., V_{SS}). Note that Crank–Nicolson exhibits oscillations with $N = 64$ whereas BE and the DSRK schemes appear free of oscillation even with the larger time-steps corresponding to $N = 8$.

We note that explicit methods are not practical for this problem because of the excessive linear stability restriction imposed by the diffusion term in (16). If an oscillation-free calculation is desired, then backward Euler is preferred over DSRK methods since DSRK methods cost more and offer essentially the same first-order convergence rates for step-sizes of practical interest. Moreover, DSRK schemes can offer little practical advantage over current Rannacher time-stepping techniques which attempt to combine the best aspects of backward Euler and Crank–Nicolson.

4.4 Hyperbolic conservation laws: Burgers' equation

Up to now we have dealt exclusively with linear problems. In this Section we consider Burgers' equation

$$u_t = -f(u)_x = -\left(\frac{1}{2}u^2\right)_x,$$

with initial condition $u(0,x) = \frac{1}{2} - \frac{1}{4}\sin(\pi x)$ on the periodic domain $x \in [0,2)$. The solution is a right-travelling, steepening shock. We discretize $-f(u)_x$ using a conservative simple upwind approximation

$$-f(u)_x \approx -\frac{1}{\Delta x}\left(\tilde{f}_{i+\frac{1}{2}} - \tilde{f}_{i-\frac{1}{2}}\right) = -\frac{1}{\Delta x}\left(f(u_i) - f(u_{i-})\right).$$

Figure 9 shows that Crank–Nicolson produces spurious oscillations in the wake of the shock, for $c = 8$ (in fact, we observe oscillations from CN for $c \geq 4$ as noted in Table 6). As expected, BE, DSRK2 and DSRK3 produce a non-oscillatory TVD solution. Table 6 shows a convergence study for this problem which illustrates the familiar pattern of order reduction.

Notice, in particular, that for any time-step size considered, one of BE or CN gives non-oscillatory results with smaller errors than the DSRK schemes considered here. However, for small time-steps, the explicit methods clearly outperform the other choices.

4.5 The van der Pol equation

The appearance of order reduction in DSRK computations is a disappointing phenomenon. It implies that DSRK methods are not likely a appropriate choice for a time-stepping scheme, because they cannot compete with BE

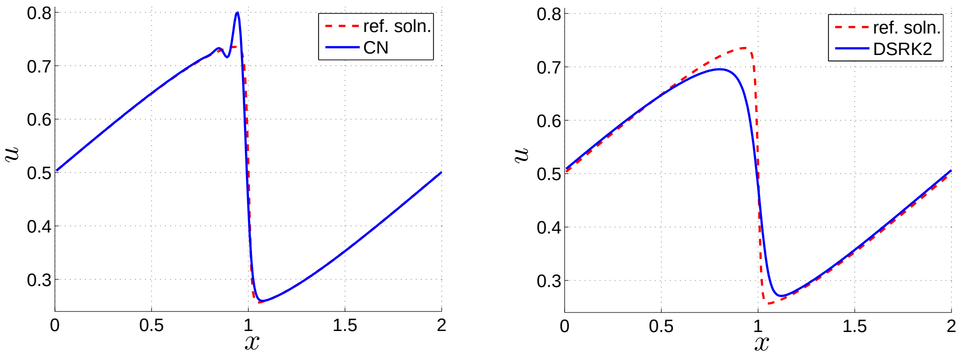


Figure 9: Burgers' equation with Crank–Nicolson (left) and DSRK2 (right) with $m = 256$ spatial points and $t_f = 2$, $N = 32$ $(c = 8)$. For CN, the solution appears smooth until the shock develops, then an oscillation develops at the trailing edge of the shock. Note that DSRK2 appears overly dissipative. The reference solution is calculated with CN and $N = 8192$.

		error (l_∞-norm against ref. soln.)							
c	N	BE	order	CN	order	DSRK2	order	DSRK3	order
16	16	0.192		0.193 *		0.195		0.195	
8	32	0.173	0.15	0.109 *	0.82	0.153	0.35	0.154	0.34
4	64	0.140	0.31	0.0399 *	1.45	0.110	0.47	0.114	0.43
2	128	0.0964	0.54	0.0124	1.68	0.0644	0.78	0.0673	0.76
1	256	0.0589	0.71	3.11e-3	2.00	0.0273	1.24	0.0249	1.43
0.5	512	0.0320	0.88	7.72e-4	2.01	8.72e-3	1.65	6.79e-3	1.87
0.25	1024	0.0165	0.96	1.90e-4	2.02	2.45e-3	1.83	1.39e-3	2.29
		FE	order	SSP22	order	SSP33	order	SSP54	order
4	64	unstable		unstable		unstable		unstable	
2	128	unstable		unstable		unstable		2.50e-4	
1	256	0.0880		5.98e-3		3.54e-4		1.36e-5	4.20
0.5	512	0.0377	1.22	1.45e-3	2.04	4.32e-5	3.03	7.63e-7	2.88
0.25	1024	0.0172	1.13	3.63e-4	2.00	5.34e-6	3.02	4.46e-8	4.10
0.125	2048	8.43e-3	1.03	9.08e-5	2.00	6.61e-7	3.01	2.68e-9	4.06

Table 6: Burgers' equation convergence study. Values for which oscillations appear are indicated with *. The setup here is the same as in Figure 9 except the reference solution is calculated with SSPRK(5,4) and $N = 8192$.

for large time-steps or with SSP explicit methods for smaller time-steps. To further study this order reduction, we apply the DSRK methods to the van der Pol equation. The van der Pol equation is an interesting problem for testing for reduction of order [24, 22, 28]. The problem can be written as an ODE initial value problem consisting of two components

$$y_1' = y_2, \tag{17a}$$

$$y_2' = \frac{1}{\epsilon}\left(-y_1 + (1 - y_1^2)y_2\right), \tag{17b}$$

with ϵ-dependent initial conditions (see Table 5.1 in [24]) and becomes increasingly stiff as ϵ is decreased.

Figure 10 shows the distinctive "flattening" [24] that occurs during the convergence studies whereby the error exhibits a region (depending on ϵ) of first-order behaviour as the step-size decreases before eventually approaching the design order of the method. This suggests that DSRK schemes suffer from order reduction whereas Crank–Nicolson clearly does not. Before the flattened region, all the high-order methods produce similar errors. In particular DSRK3 does no better than the second-order Crank–Nicolson until after the flattening region. In these figures we observe that DSRK2so2 seems to suffer from order reduction as well despite its underlying RK scheme having stage order two. Higher stage order of the underlying RK scheme is not sufficient to avoid the order reduction.

5 Conclusions and Future Directions

We studied the performance of unconditionally contractive diagonally split Runge–Kutta (DSRK) schemes of orders two and three on a variety of archetypal test cases. The numerical tests verified the asymptotic order of the schemes as well as the unconditional contractivity property. However, in every numerical experiment, the DSRK methods were out-performed by the first-order backward Euler (BE) scheme when $\Delta t > 2\Delta t_{\text{FE}}$, and by explicit Runge–Kutta methods or Crank-Nicolson (CN) when $\Delta t \leq 2\Delta t_{\text{FE}}$. At larger time-steps, the DSRK schemes are strong stability preserving (SSP) but suffer from order reduction, making BE a better choice. At small step-sizes, CN and explicit SSPRK methods are SSP, and produce far more accurate results at a smaller computational cost. It is tempting to assume that the order reduction occurs because unconditionally contractive DSRK methods

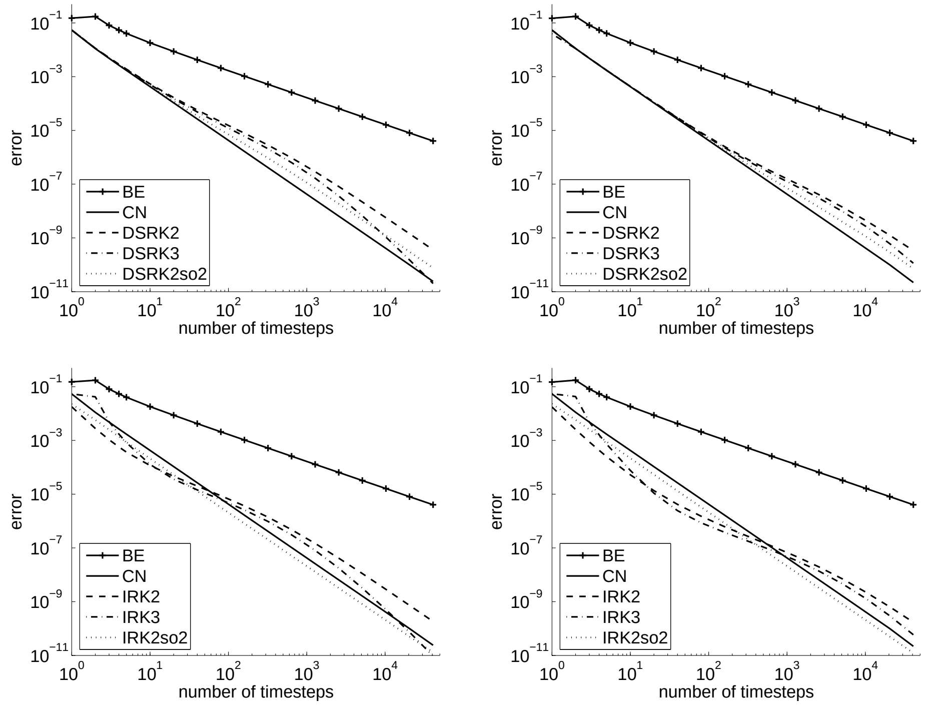


Figure 10: A convergence study on the van der Pol equation. Error shown is in the second component, where we have taken $\epsilon = 1 \times 10^{-3}$ (left) and $\epsilon = 1 \times 10^{-4}$ (right). Top row shows the methods studies in this work. Bottom row shows the behavior of the RK schemes underlying each of the DSRK schemes.

have underlying Runge–Kutta schemes which necessarily have low stage order; however our numerical experiments with the (conditionally contractive) DSRK2so2 method show that order reduction occurs even when the underlying method has higher stage order. It is therefore reasonable to assume that the splitting itself may be responsible for the order reduction, and perhaps that new stage order conditions involving the **W** coefficients must be introduced.

We investigated the class of unconditionally contractive DSRK methods using the BARON optimization software and found that an unconditionally contractive four-stage fourth-order method is unlikely to exist. A more promising avenue would be to search for a five-stage fourth-order DSRK scheme. However, such a method would likely suffer from order reduction as well, and will therefore not be of much use.

The class of unconditionally contractive DSRK methods does not produce viable alternatives to well-established conditionally SSP Runge–Kutta and linear multistep methods. Future research will focus on implicit Runge–Kutta and DSRK methods which are not unconditionally contractive, but which may have a large allowable step-size, ideally without suffering from order reduction.

Acknowledgements

The authors thank Z. Jackiewicz for suggesting we look at stage order of the underlying Runge–Kutta schemes, and suggesting the method (11).

References

[1] A. Bellen, Z. Jackiewicz, and M. Zennaro. Contractivity of waveform relaxation Runge–Kutta iterations and related limit methods for dissipative systems in the maximum norm. *SIAM J. Numer. Anal.*, 31(2):499–523, 1994.

[2] A. Bellen and L. Torelli. Unconditional contractivity in the maximum norm of diagonally split Runge–Kutta methods. *SIAM J. Numer. Anal.*, 34(2):528–543, 1997.

[3] F. Black and M. Scholes. The Pricing of Options and Corporate Liabilities. *The Journal of Political Economy*, 81(3):637–654, 1973.

[4] Thomas Coleman. Option pricing: The hazards of computing delta and gamma. website, 2006. `http://www.fenews.com/fen49/where_num_matters/numerics.htm`.

[5] L. Ferracina and M. N. Spijker. Stepsize restrictions for the total-variation-diminishing property in general Runge–Kutta methods. *SIAM J. Numer. Anal.*, 42(3):1073–1093 (electronic), 2004.

[6] L. Ferracina and M. N. Spijker. An extension and analysis of the Shu–Osher representation of Runge–Kutta methods. *Math. Comp.*, 74(249):201–219 (electronic), 2005.

[7] P.A. Forsyth. An introduction to computational finance without agonizing pain. Available on author's website, `http://www.cs.uwaterloo.ca/~paforsyt/agon.pdf`, February 2005.

[8] P.A. Forsyth and K.R. Vetzal. Numerical PDE methods for pricing path dependent options. Notes from a short course at the Fields Institute, February 2002.

[9] Michael B. Giles and Rebecca Carter. Convergence analysis of Crank–Nicolson and Rannacher time-marching. *Journal of Computational Finance*, 9(4):89–112, 2006.

[10] Sigal Gottlieb. On high order strong stability preserving Runge–Kutta and multi step time discretizations. *J. Sci. Comput.*, 25(1-2):105–128, 2005.

[11] Sigal Gottlieb and Lee-Ad J. Gottlieb. Strong stability preserving properties of Runge–Kutta time discretization methods for linear constant coefficient operators. *J. Sci. Comput.*, 18(1):83–109, 2003.

[12] Sigal Gottlieb and Steven J. Ruuth. Optimal strong-stability-preserving time-stepping schemes with fast downwind spatial discretizations. *J. Sci. Comput.*, 27(1-3):289–303, 2006.

[13] Sigal Gottlieb and Chi-Wang Shu. Total variation diminishing Runge–Kutta schemes. *Math. Comp.*, 67(221):73–85, 1998.

[14] Sigal Gottlieb, Chi-Wang Shu, and Eitan Tadmor. Strong stability-preserving high-order time discretization methods. *SIAM Rev.*, 43(1):89–112 (electronic), 2001.

[15] E. Hairer and G. Wanner. *Solving ordinary differential equations II: Stiff and differential-algebraic problems*, volume 14 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 1991.

[16] Inmaculada Higueras. On strong stability preserving time discretization methods. *J. Sci. Comput.*, 21(2):193–223, 2004.

[17] Inmaculada Higueras. Representations of Runge–Kutta methods and strong stability preserving methods. *SIAM J. Numer. Anal.*, 43(3):924–948 (electronic), 2005.

[18] Zoltán Horváth. Positivity of Runge–Kutta and diagonally split Runge–Kutta methods. *Appl. Numer. Math.*, 28(2-4):309–326, 1998. Eighth Conference on the Numerical Treatment of Differential Equations (Alexisbad, 1997).

[19] Willem Hundsdorfer and Steven J. Ruuth. On monotonicity and boundedness properties of linear multistep methods. *Math. Comp.*, 75(254):655–672 (electronic), 2006.

[20] Willem Hundsdorfer, Steven J. Ruuth, and Raymond J. Spiteri. Monotonicity-preserving linear multistep methods. *SIAM J. Numer. Anal.*, 41(2):605–623 (electronic), 2003.

[21] K. J. in 't Hout. A note on unconditional maximum norm contractivity of diagonally split Runge–Kutta methods. *SIAM J. Numer. Anal.*, 33(3):1125–1134, 1996.

[22] Christopher A. Kennedy and Mark H. Carpenter. Additive Runge–Kutta schemes for convection-diffusion-reaction equations. *Appl. Numer. Math.*, 44(1-2):139–181, 2003.

[23] J. F. B. M. Kraaijevanger. Contractivity of Runge–Kutta methods. *BIT*, 31(3):482–528, 1991.

[24] Anita T. Layton and Michael L. Minion. Implications of the choice of quadrature nodes for Picard integral deferred corrections methods for ordinary differential equations. *BIT*, 45(2):341–373, 2005.

[25] H. W. J. Lenferink. Contractivity preserving explicit linear multistep methods. *Numer. Math.*, 55(2):213–223, 1989.

[26] H. W. J. Lenferink. Contractivity-preserving implicit linear multistep methods. *Math. Comp.*, 56(193):177–199, 1991.

[27] Colin B. Macdonald. Constructing high-order Runge–Kutta methods with embedded strong-stability-preserving pairs. Master's thesis, Simon Fraser University, August 2003.

[28] Michael L. Minion. Semi-implicit spectral deferred correction methods for ordinary differential equations. *Commun. Math. Sci.*, 1(3):471–500, 2003.

[29] Haim Nessyahu and Eitan Tadmor. Nonoscillatory central differencing for hyperbolic conservation laws. *J. Comput. Phys.*, 87(2):408–463, 1990.

[30] Stanley Osher and Ronald Fedkiw. *Level set methods and dynamic implicit surfaces*, volume 153 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 2003.

[31] Stanley Osher and Eitan Tadmor. On the convergence of difference approximations to scalar conservation laws. *Math. Comp.*, 50(181):19–51, 1988.

[32] Steven J. Ruuth. Global optimization of explicit strong-stability-preserving Runge–Kutta methods. *Math. Comp.*, 75(253):183–207 (electronic), 2006.

[33] Steven J. Ruuth and Raymond J. Spiteri. Two barriers on strong-stability-preserving time discretization methods. *J. Sci. Comput.*, 17(1-4):211–220, 2002. Proceedings of the Fifth International Conference on Spectral and High Order Methods (ICOSAHOM-01) (Uppsala).

[34] Steven J. Ruuth and Raymond J. Spiteri. High-order strong-stability-preserving Runge–Kutta methods with downwind-biased spatial discretizations. *SIAM J. Numer. Anal.*, 42(3):974–996 (electronic), 2004.

[35] N. V. Sahinidis and M. Tawarmalani. *BARON 7.2: Global Optimization of Mixed-Integer Nonlinear Programs,* User's Manual, 2004. Available at `http://www.gams.com/dd/docs/solvers/baron.pdf`.

[36] Chi-Wang Shu. Total-variation-diminishing time discretizations. *SIAM J. Sci. Statist. Comput.*, 9(6):1073–1084, 1988.

[37] Chi-Wang Shu. A survey of strong stability preserving high-order time discretizations. In D. Estep and S. Tavener, editors, *Collected Lectures on the Preservation of Stability under Discretization*, volume 109 of *Proceedings in Applied Mathematics*, pages 51–65. SIAM, 2002.

[38] Chi-Wang Shu and Stanley Osher. Efficient implementation of essentially nonoscillatory shock-capturing schemes. *J. Comput. Phys.*, 77(2):439–471, 1988.

[39] M. N. Spijker. Contractivity in the numerical solution of initial value problems. *Numer. Math.*, 42(3):271–290, 1983.

[40] Raymond J. Spiteri and Steven J. Ruuth. A new class of optimal high-order strong-stability-preserving time discretization methods. *SIAM J. Numer. Anal.*, 40(2):469–491, 2002.

[41] Raymond J. Spiteri and Steven J. Ruuth. Non-linear evolution using optimal fourth-order strong-stability-preserving Runge–Kutta methods. *Math. Comput. Simulation*, 62(1-2):125–135, 2003. Nonlinear waves: computation and theory, II (Athens, GA, 2001).

[42] M. Tawarmalani and N. V. Sahinidis. Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. *Mathematical Programming*, 99:563–591, 2004.

[43] M. Zennaro. Contractivity of Runge–Kutta methods with respect to forcing terms. *Appl. Numer. Math.*, 11(4):321–345, 1993.