SIAM J. NUMER. ANAL.
Vol. 49, No. 6, pp. 2618–2639

STRONG STABILITY PRESERVING TWO-STEP RUNGE–KUTTA METHODS*

DAVID I. KETCHESON†, SIGAL GOTTLIEB‡, AND COLIN B. MACDONALD§

Abstract. We investigate the strong stability preserving (SSP) property of two-step Runge–Kutta (TSRK) methods. We prove that all SSP TSRK methods belong to a particularly simple subclass of TSRK methods, in which stages from the previous step are not used. We derive simple order conditions for this subclass. Whereas explicit SSP Runge–Kutta methods have order at most four, we prove that explicit SSP TSRK methods have order at most eight. We present explicit TSRK methods of up to eighth order that were found by numerical search. These methods have larger SSP coefficients than any known methods of the same order of accuracy and may be implemented in a form with relatively modest storage requirements. The usefulness of the TSRK methods is demonstrated through numerical examples, including integration of very high order weighted essentially non-oscillatory discretizations.

Key words. strong stability preserving, monotonicity, two-step Runge–Kutta methods

AMS subject classifications. Primary, 65M20; Secondary, 65L06

DOI. 10.1137/10080960X

1. Strong stability preserving methods.

The concept of strong stability preserving (SSP) methods was introduced by Shu and Osher in [40] for use with total variation diminishing spatial discretizations of a hyperbolic conservation law:

$$U_t + f(U)_x = 0.$$

When the spatial derivative is discretized, we obtain the system of ODEs

$$u_t = F(u), \tag{1.1}$$

where u is a vector of approximations to U, $u_j \approx U(x_j)$. The spatial discretization is carefully designed so that when this ODE is fully discretized using the forward Euler method, certain convex functional properties (such as the total variation) of the numerical solution do not increase,

$$\|u^n + \Delta t F(u^n)\| \leq \|u^n\| \tag{1.2}$$

for all small enough stepsizes $\Delta t \leq \Delta t_{\text{FE}}$. Typically, we need methods of higher order and we wish to guarantee that the higher order time discretizations will preserve this strong stability property. This guarantee is obtained by observing that if a time

*Received by the editors September 23, 2010; accepted for publication (in revised form) September 26, 2011; published electronically December 22, 2011.
http://www.siam.org/journals/sinum/49-6/80960.html

†Division of Mathematical and Computer Sciences and Engineering, 4700 King Abdullah University of Science & Technology, Thuwal 23955, Saudi Arabia (david.ketcheson@kaust.edu.sa). The work of this author was supported by a U.S. Department of Energy Computational Science Graduate Fellowship and by funding from King Abdullah University of Science and Technology.

‡Department of Mathematics, University of Massachusetts Dartmouth, North Dartmouth, MA 02747 (sgottlieb@umassd.edu). This work was supported by AFOSR grant FA9550-09-1-0208.

§Mathematical Institute, University of Oxford, Oxford OX1 3LB, UK (macdonald@maths.ox.ac.uk). The work of this author was supported by an NSERC postdoctoral fellowship, NSF grant CCF-0321917, and award KUK-C1-013-04 made by King Abdullah University of Science and Technology.

discretization can be decomposed into convex combinations of forward Euler steps, then any convex functional property (referred to herein as a *strong stability* property) satisfied by forward Euler will be *preserved* by the higher-order time discretizations, perhaps under a different time-step restriction.

Given a semi-discretization of the form (1.1) and convex functional $\|\cdot\|$, we assume that there exists a value Δt_{FE} such that for all u,

$$\|u + \Delta t F(u)\| \le \|u\| \text{ for } 0 \le \Delta t \le \Delta t_{\mathrm{FE}}. \tag{1.3}$$

A k-step numerical method for (1.1) computes the next solution value u^{n+1} from previous values $u^{n-k+1}, \dots, u^n$. A method is said to be SSP if (in the solution of (1.1)) it holds that

$$\|u^{n+1}\| \le \max\left\{\|u^n\|, \|u^{n-1}\|, \dots, \|u^{n-k+1}\|\right\} \tag{1.4}$$

whenever (1.3) holds and the time-step satisfies

$$\Delta t \le \mathcal{C} \Delta t_{\mathrm{FE}}. \tag{1.5}$$

For multistage methods, a more stringent condition is useful, in which the intermediate stages are also required to satisfy a monotonicity property. For the two-step Runge–Kutta methods (TSRK) in this work, this condition is

$$\max\left\{\|u^{n+1}\|, \max_i \|y_i^n\|\right\} \le \max\left\{\|u^n\|, \|u^{n-1}\|, \max_i \|y_i^{n-1}\|\right\}. \tag{1.6}$$

The quantities y_i^n, y_i^{n-1} are the intermediate stages from the current and previous step, respectively. This type of property is sometimes referred to as *internal strong stability* or *internal monotonicity*. Throughout this work, $\mathcal{C}$ is taken to be the largest value such that (1.5) and (1.3) together always imply (1.6). This value $\mathcal{C}$ is called the *SSP coefficient* of the method.

For example, consider explicit multistep methods [39]:

$$u^{n+1} = \sum_{i=1}^{k} \left(\alpha_i u^{n+1-i} + \Delta t \beta_i F(u^{n+1-i})\right). \tag{1.7}$$

Since $\sum_{i=1}^k \alpha_i = 1$ for any consistent method, any such method can be written as convex combinations of forward Euler steps if all the coefficients are nonnegative:

$$u^{n+1} = \sum_{i=1}^{k} \alpha_i \left(u^{n+1-i} + \frac{\beta_i}{\alpha_i} \Delta t F(u^{n+1-i})\right).$$

If the forward Euler method applied to (1.1) is strongly stable under the time-step restriction $\Delta t \le \Delta t_{\mathrm{FE}}$ and $\alpha_i, \beta_i \ge 0$, then the solution obtained by the multistep method (1.7) satisfies the strong stability bound (1.4) under the time-step restriction

$$\Delta t \le \min_i \frac{\alpha_i}{\beta_i} \Delta t_{\mathrm{FE}}.$$

(If any of the β's are equal to zero, the corresponding ratios are considered infinite.)

In the case of a Runge–Kutta method the monotonicity requirement (1.6) reduces to

$$\max\left\{\|u^{n+1}\|, \max_i \|y_i^n\|\right\} \le \|u^n\|.$$

For example, an s-stage explicit Runge–Kutta method is written in the form [40]

$$\begin{aligned} u^{(0)} &= u^n, \\ u^{(i)} &= \sum_{j=0}^{i-1} \left(\alpha_{ij} u^{(j)} + \Delta t \beta_{ij} F(u^{(j)}) \right), \\ u^{n+1} &= u^{(s)}. \end{aligned} \tag{1.8}$$

If all the coefficients are nonnegative, each stage of the Runge–Kutta method can be rearranged into convex combinations of forward Euler steps, with a modified step size:

$$\begin{aligned} \|u^{(i)}\| &= \left\| \sum_{j=0}^{i-1} \left(\alpha_{ij} u^{(j)} + \Delta t \beta_{ij} F(u^{(j)}) \right) \right\| \\ &\leq \sum_{j=0}^{i-1} \alpha_{ij} \left\| u^{(j)} + \Delta t \frac{\beta_{ij}}{\alpha_{ij}} F(u^{(j)}) \right\|. \end{aligned}$$

Now, since each $\|u^{(j)} + \Delta t \frac{\beta_{ij}}{\alpha_{ij}} F(u^{(j)})\| \leq \|u^{(j)}\|$ as long as $\frac{\beta_{ij}}{\alpha_{ij}} \Delta t \leq \Delta t_{\text{FE}}$, and since $\sum_{j=0}^{i-1} \alpha_{ij} = 1$ by consistency, we have $\|u^{n+1}\| \leq \|u^n\|$ as long as $\frac{\beta_{ij}}{\alpha_{ij}} \Delta t \leq \Delta t_{\text{FE}}$ for all i and j. Thus, if the forward Euler method applied to (1.1) is strongly stable under the time-step restriction $\Delta t \leq \Delta t_{\text{FE}}$, i.e., (1.3) holds, and if $\alpha_{ij}, \beta_{ij} \geq 0$, then the solution obtained by the Runge–Kutta method (1.8) satisfies the strong stability bound (1.2) under the time-step restriction

$$\Delta t \leq \min_{i,j} \frac{\alpha_{ij}}{\beta_{ij}} \Delta t_{\text{FE}}.$$

As above, if any of the β's are equal to zero, the corresponding ratios are considered infinite.

This approach can easily be generalized to implicit Runge–Kutta methods and implicit linear multistep methods. Thus it provides sufficient conditions for strong stability of high-order explicit and implicit Runge–Kutta and multistep methods. In fact, it can be shown from the connections between SSP theory and contractivity theory [11, 12, 19, 20] that these conditions are not only sufficient, they are necessary as well.

Much research on SSP methods focuses on finding high-order time discretizations with the largest allowable time-step. Unfortunately, explicit SSP Runge–Kutta methods with positive coefficients cannot be more than fourth-order accurate [32, 38], and explicit SSP linear multistep methods of high-order accuracy require very many steps in order to have reasonable time-step restrictions. For instance, obtaining a fifth-order explicit linear multistep method with a time-step restriction of $\Delta t \leq 0.2 \Delta t_{\text{FE}}$ requires 9 steps; for a sixth-order method, this increases to 13 steps [34]. In practice, the large storage requirements of these methods make them unsuitable for the solution of the large systems of ODEs resulting from semi-discretization of a PDE. Multistep methods with larger SSP coefficients and fewer stages have been obtained by considering special starting procedures [22, 37].

Perhaps because of the lack of practical explicit SSP methods of very high order, high-order spatial discretizations for hyperbolic PDEs are often paired with lower-order time discretizations; some examples of this are [5, 6, 7, 9, 10, 27, 33, 36, 43].

This may lead to loss of accuracy, particularly for long time simulations. In an extreme case [13], weighted essentially non-oscillatory (WENO) schemes of up to 17th order were paired with third-order SSP Runge–Kutta time integration; of course, convergence tests indicated only third-order convergence for the fully discrete schemes. Practical higher-order accurate SSP time discretization methods are needed for the time evolution of ODEs resulting from high-order spatial discretizations.

To obtain higher-order explicit SSP time discretizations, methods that include both multiple steps and multiple stages have been considered. These methods are a subclass of explicit general linear methods that allow higher order with positive SSP coefficients. Gottlieb, Shu, and Tadmor considered a class of two-step, two-stage methods [14]. Another class of such methods was considered by Spijker [41]. Huang [21] considered hybrid methods with many steps and found methods of up to seventh-order (with seven steps) with reasonable SSP coefficients. Constantinescu and Sandu [8] considered two- and three-step Runge–Kutta methods, with a focus on finding SSP methods with stage order up to four.

In this work we consider a class of two-step multistage Runge–Kutta methods, which are a generalization of both linear multistep methods and Runge–Kutta methods. In section 2, we discuss some classes of TSRK methods and use the theory presented in [41] to prove that all SSP TSRK methods belong to a very simple subclass. In section 3, we derive order conditions using a generalization of the approach presented in [2] and show that explicit SSP TSRK methods have order at most eight. In section 4, we formulate the optimization problem using the approach from [31], give an efficient form for the implementation of SSP TSRK methods, and present optimal explicit methods of up to eighth order. These methods have relatively modest storage requirements and reasonable effective SSP coefficient. We also report optimal lower-order methods; our results agree with those of [8] for second-, third-, and fourth-order methods of up to four stages and improve upon other methods previously found both in terms of order and the size of the SSP coefficient. Numerical verification of the optimal methods and a demonstration of the need for high-order time discretizations for use with high-order spatial discretizations are presented in section 5. Conclusions and future work are discussed in section 6.

2. SSP TSRK methods. A general class of TSRK methods was studied in [25, 4, 16, 45]. TSRK methods are a generalization of Runge–Kutta methods that include values and stages from the previous step:

(2.1a)
$$y_i^n = d_i u^{n-1} + (1-d_i)u^n + \Delta t \sum_{j=1}^{s} \hat{a}_{ij} F(y_j^{n-1}) + \Delta t \sum_{j=1}^{s} a_{ij} F(y_j^n), \quad 1 \le i \le s,$$

(2.1b)
$$u^{n+1} = \theta u^{n-1} + (1-\theta)u^n + \Delta t \sum_{j=1}^{s} \hat{b}_j F(y_j^{n-1}) + \Delta t \sum_{j=1}^{s} b_j F(y_j^n).$$

Here u^n denotes the solution values at time $t = n\Delta t$ and the values y_i^n, y_i^{n-1} are intermediate stages used to compute the solution at the next time step. We use the matrices $\mathbf{A}$, $\hat{\mathbf{A}}$ and column vectors $\mathbf{b}$, $\hat{\mathbf{b}}$, and $\mathbf{d}$ to refer to the coefficients of the method in a compact way.

As we will prove in Theorem 1, only rather special TSRK methods can have positive SSP coefficient. Generally, method (2.1) cannot be strong stability preserv-

ing unless all the coefficients $\hat{a}_{ij}, \hat{b}_j$ are identically zero. A brief explanation of this requirement is as follows. Since method (2.1) does not include terms proportional to y_i^{n-1}, it is not possible to write a stage of method (2.1) as a convex combination of forward Euler steps if the stage includes terms proportional to $F(y_i^{n-1})$. This is because those stages depend on u^{n-2}, which is not available in a two-step method.

Hence we are led to consider simpler methods of the following form (compare [15, p. 362]). We call these *augmented Runge–Kutta methods*:

$$y_i^n = d_i u^{n-1} + (1-d_i)u^n + \Delta t \sum_{j=1}^{s} a_{ij} F(y_j^n), \qquad 1 \le i \le s, \tag{2.2a}$$

$$u^{n+1} = \theta u^{n-1} + (1-\theta)u^n + \Delta t \sum_{j=1}^{s} b_j F(y_j^n). \tag{2.2b}$$

2.1. The Spijker form for general linear methods. TSRK methods are a subclass of general linear methods. In this section, we review the theory of strong stability preservation for general linear methods [41]. A general linear method can be written in the form

$$w_i^n = \sum_{j=1}^{l} s_{ij} x_j^n + \Delta t \sum_{j=1}^{m} t_{ij} F(w_j^n) \qquad (1 \le i \le m), \tag{2.3a}$$

$$x_j^{n+1} = w_{J_j}^n \qquad (1 \le j \le l). \tag{2.3b}$$

The terms x_j^n are the l input values available from previous steps, while the w_j^n include both the output values and intermediate stages used to compute them. Equation (2.3b) indicates which of these values are used as inputs in the next step. The earlier definition of strong stability preservation is extended to (2.3) by requiring, in place of (1.6), that

$$\|w_j\| \le \max_i \|x_i\|, \qquad 1 \le j \le m.$$

We will frequently write the coefficients s_{ij} and t_{ij} as a $m \times l$ matrix $\mathbf{S}$ and a $m \times m$ matrix $\mathbf{T}$, respectively. Without loss of generality (see [41, section 2.1.1]) we assume that

$$\mathbf{S}\mathbf{e} = \mathbf{e}, \tag{2.4}$$

where $\mathbf{e}$ is a column vector with all entries equal to unity. This implies that every stage is a consistent approximation to the solution at some time.

Runge–Kutta methods, multistep methods, and multistep Runge–Kutta methods are all general linear methods and can be written in the form (2.3). For example, an s-stage Runge–Kutta method with Butcher coefficients $\mathbf{A}$ and $\mathbf{b}$ can be written in the form (2.3) by taking $l = 1, m = s+1$, $J = \{m\}$, and

$$\mathbf{S} = (1, 1, \ldots, 1)^{\mathrm{T}}, \qquad \mathbf{T} = \begin{pmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{b}^{\mathrm{T}} & 0 \end{pmatrix}.$$

Linear multistep methods

$$u^{n+1} = \sum_{j=1}^{l} \alpha_j u^{n+1-j} + \Delta t \sum_{j=0}^{l} \beta_j F\left(u^{n+1-j}\right)$$

admit the Spijker form

$$\mathbf{S} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 1 \\ \alpha_l & \alpha_{l-1} & \dots & \alpha_1 \end{pmatrix}, \quad \mathbf{T}^{(l+1)\times(l+1)} = \begin{pmatrix} 0 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 0 \\ \beta_l & \beta_{l-1} & \dots & \beta_0 \end{pmatrix},$$

where l is the number of steps, $m = l + 1$, and $J = \{2, \dots, l + 1\}$.

TSRK methods (2.1) can be written in Spijker form as follows: set $m = 2s + 2$, $l = s + 2$, $J = \{s + 1, s + 2, \dots, 2s + 2\}$, and

$$\mathbf{x}^n = \left(u^{n-1}, y_1^{n-1}, \dots, y_s^{n-1}, u^n\right)^{\mathrm{T}}, \tag{2.5a}$$

$$\mathbf{w}^n = \left(y_1^{n-1}, y_2^{n-1}, \dots, y_s^{n-1}, u^n, y_1^n, y_2^n, \dots, y_s^n, u^{n+1}\right)^{\mathrm{T}}, \tag{2.5b}$$

$$\mathbf{S} = \begin{pmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} \\ 0 & \mathbf{0} & 1 \\ \mathbf{d} & \mathbf{0} & \mathbf{e} - \mathbf{d} \\ \theta & \mathbf{0} & 1 - \theta \end{pmatrix}, \qquad \mathbf{T} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 0 & \mathbf{0} & 0 \\ \hat{\mathbf{A}} & \mathbf{0} & \mathbf{A} & \mathbf{0} \\ \hat{\mathbf{b}}^{\mathrm{T}} & 0 & \mathbf{b}^{\mathrm{T}} & 0 \end{pmatrix}. \tag{2.5c}$$

2.2. The SSP coefficient for general linear methods. In order to analyze the SSP property of a general linear method (2.3), we first define the column vector $\mathbf{f} = [F(w_1), F(w_2), \dots, F(w_m)]^{\mathrm{T}}$, so that (2.3a) can be written compactly as

$$\mathbf{w} = \mathbf{S}\mathbf{x} + \Delta t \mathbf{T}\mathbf{f}. \tag{2.6}$$

Adding $r\mathbf{T}\mathbf{w}$ to both sides of (2.6) gives

$$(\mathbf{I} + r\mathbf{T})\,\mathbf{w} = \mathbf{S}\mathbf{x} + r\mathbf{T}\left(\mathbf{w} + \frac{\Delta t}{r}\mathbf{f}\right).$$

Assuming that the matrix on the left is invertible we obtain

$$\begin{aligned} \mathbf{w} &= (\mathbf{I} + r\mathbf{T})^{-1}\mathbf{S}\mathbf{x} + r(\mathbf{I} + r\mathbf{T})^{-1}\mathbf{T}\left(\mathbf{w} + \frac{\Delta t}{r}\mathbf{f}\right) \\ &= \mathbf{R}\mathbf{x} + \mathbf{P}\left(\mathbf{w} + \frac{\Delta t}{r}\mathbf{f}\right), \end{aligned} \tag{2.7}$$

where we have defined

$$\mathbf{P} = r(\mathbf{I} + r\mathbf{T})^{-1}\mathbf{T}, \quad \mathbf{R} = (\mathbf{I} + r\mathbf{T})^{-1}\mathbf{S} = (\mathbf{I} - \mathbf{P})\mathbf{S}. \tag{2.8}$$

Observe that, by the consistency condition (2.4), the row sums of $[\mathbf{R}\ \mathbf{P}]$ are each equal to one:

$$\mathbf{R}\mathbf{e} + \mathbf{P}\mathbf{e} = (\mathbf{I} - \mathbf{P})\mathbf{S}\mathbf{e} + \mathbf{P}\mathbf{e} = \mathbf{e} - \mathbf{P}\mathbf{e} + \mathbf{P}\mathbf{e} = \mathbf{e}.$$

Thus, if $\mathbf{R}$ and $\mathbf{P}$ have no negative entries, each stage w_i is given by a convex combination of the inputs x_j and the quantities $w_j + (\Delta t/r)F(w_j)$. It can thus be shown (see [41]) that any strong stability property of the forward Euler method is preserved by

the method (2.6) under the time-step restriction given by $\Delta t \le \mathcal{C}(\mathbf{S},\mathbf{T})\Delta t_{\mathrm{FE}}$, where $\mathcal{C}(\mathbf{S},\mathbf{T})$ is defined as

$$\mathcal{C}(\mathbf{S},\mathbf{T}) = \sup_r \left\{ r : (\mathbf{I}+r\mathbf{T})^{-1} \text{ exists and } \mathbf{P} \ge 0, \mathbf{R} \ge 0 \right\},$$

where $\mathbf{P}$ and $\mathbf{R}$ are defined in (2.8). Hence the SSP coefficient of method (2.7) is greater than or equal to $\mathcal{C}(\mathbf{S},\mathbf{T})$.

To state precisely the conditions under which the SSP coefficient is, in fact, equal to $\mathcal{C}(\mathbf{S},\mathbf{T})$, we must introduce the concept of reducibility. The definition used here was proposed in [41, Remark 3.2]).

DEFINITION 1 (row-reducibility). *A method in form* (2.3) *is* row-reducible *if there exist indices i, j, with $i \ne j$, such that all of the following hold:*

1. *Rows i and j of $\mathbf{S}$ are equal.*
2. *Rows i and j of $\mathbf{T}$ are equal.*
3. *Column i of $\mathbf{T}$ is not identically zero.*
4. *Column j of $\mathbf{T}$ is not identically zero.*

Otherwise, we say the method is row-irreducible.

The following result, which formalizes the connection between the algebraic conditions above and the SSP coefficient, was stated in [41, Remark 3.2].

LEMMA 1. *Consider a row-irreducible general linear method with coefficients $\mathbf{S},\mathbf{T}$. The SSP coefficient of the method is $\mathcal{C} = \mathcal{C}(\mathbf{S},\mathbf{T})$.*

2.3. Restrictions on the coefficients of SSP TSRK methods. In light of Lemma 1, we are interested in methods with $\mathcal{C}(\mathbf{S},\mathbf{T}) > 0$. The following lemma characterizes such methods.

LEMMA 2 (see [41, Theorem 2.2(i)]). *$\mathcal{C}(\mathbf{S},\mathbf{T}) > 0$ if and only if all the following hold:*

$$\mathbf{S} \ge \mathbf{0}, \tag{2.9a}$$

$$\mathbf{T} \ge \mathbf{0}, \tag{2.9b}$$

$$\mathrm{Inc}(\mathbf{TS}) \le \mathrm{Inc}(\mathbf{S}), \tag{2.9c}$$

$$\mathrm{Inc}(\mathbf{T}^2) \le \mathrm{Inc}(\mathbf{T}), \tag{2.9d}$$

where all the inequalities are elementwise and the incidence matrix of a matrix $\mathbf{M}$ with entries m_{ij} is

$$\mathrm{Inc}(\mathbf{M})_{ij} = \begin{cases} 1 & \text{if } m_{ij} \ne 0, \\ 0 & \text{if } m_{ij} = 0. \end{cases}$$

Combining Lemmas 1 and 2, we find several restrictions on the coefficients of SSP TSRK methods. It is known that irreducible strong stability preserving Runge–Kutta methods have positive stage coefficients $a_{ij} \ge 0$ and strictly positive weights $b_j > 0$. The following theorem shows that similar properties hold for SSP TSRK methods. The theorem and its proof are very similar to [32, Theorem 4.2]. In the proof, we will use a second irreducibility concept. A method is said to be *DJ-reducible* if it involves one or more stages whose value does not affect the output. If a TSRK method is neither row-reducible nor DJ-reducible, we say it is *irreducible*.

THEOREM 1. *Let $\mathbf{S},\mathbf{T}$ be the coefficients of a s-stage TSRK method* (2.1) *that is row-irreducible in form* (2.5) *with positive SSP coefficient $\mathcal{C} > 0$. Then the coefficients*

of the method satisfy the following properties:

$$\mathbf{0} \le \mathbf{d} \le \mathbf{1}, \qquad 0 \le \theta \le 1, \tag{2.10a}$$

$$\mathbf{A} \ge \mathbf{0}, \qquad \mathbf{b} \ge \mathbf{0}, \tag{2.10b}$$

$$\hat{\mathbf{A}} = \mathbf{0}, \qquad \hat{\mathbf{b}} = \mathbf{0}. \tag{2.10c}$$

Furthermore, if the method is also DJ-irreducible, the weights must be strictly positive:

$$\mathbf{b} > \mathbf{0}. \tag{2.11}$$

All these inequalities should be interpreted componentwise.

Proof. Consider a method satisfying the stated assumptions. Lemma 1 implies that $\mathcal{C}(\mathbf{S}, \mathbf{T}) > 0$, so Lemma 2 applies. Then (2.10a)–(2.10c) follow from the corresponding conditions (2.9a)–(2.9c).

To prove (2.11), observe that condition (2.9d) of Lemma 2 means that if $b_j = 0$ for some j, then

$$\sum_i b_i a_{ij} = 0. \tag{2.12}$$

Since $\mathbf{A}, \mathbf{b}$ are nonnegative, (2.12) implies that either b_i or a_{ij} is zero for each value of i. Now partition the set $\mathcal{S} = \{1, 2, \ldots, s\}$ into $\mathcal{S}_1, \mathcal{S}_2$ such that $b_j > 0$ for all $j \in \mathcal{S}_1$ and $b_j = 0$ for all $j \in \mathcal{S}_2$. Then $a_{ij} = 0$ for all $i \in \mathcal{S}_1$ and $j \in \mathcal{S}_2$. This implies that the method is DJ-reducible, unless $\mathcal{S}_2 = \emptyset$. ∎

Remark 1. Theorem 1 and, in particular, condition (2.10c), dramatically reduce the class of methods that must be considered when searching for optimal SSP TSRK methods. They also allow the use of very simple order conditions, derived in the next section.

3. Order conditions and a barrier. Order conditions for TSRK methods up to order 6 have previously been derived in [25]. Alternative approaches to order conditions for TSRK methods, using trees and B-series, have also been identified [4, 16]. In this section we derive order conditions for augmented Runge–Kutta methods (2.2). The order conditions derived here are not valid for the general class of methods given by (2.1). Our derivation follows Albrecht's approach [2] and leads to very simple conditions, which are almost identical in appearance to order conditions for RK methods. For simplicity of notation, we consider a scalar ODE only. For more details and justification of this approach for systems, see [2].

3.1. Derivation of order conditions. When applied to the trivial ODE $u'(t) = 0$, any TSRK scheme reduces to the recurrence $u^{n+1} = \theta u^{n-1} + (1-\theta)u^n$. For an SSP TSRK scheme, we have $0 \le \theta \le 1$ (by Theorem 1) and it follows that the method is zero-stable. Hence to prove convergence of order p, it is sufficient to prove consistency of order p (see, e.g., [24, Theorem 2.3.4]).

We can write an augmented Runge–Kutta method (2.2) compactly as

$$\mathbf{y}^n = \mathbf{d}u^{n-1} + (\mathbf{e} - \mathbf{d})u^n + \Delta t \mathbf{A}\mathbf{f}^n, \tag{3.1a}$$

$$u^{n+1} = \theta u^{n-1} + (1-\theta)u^n + \Delta t \mathbf{b}^{\mathrm{T}}\mathbf{f}^n. \tag{3.1b}$$

Let $\tilde{u}(t)$ denote the exact solution at time t and define

$$\tilde{\mathbf{y}}^n = [\tilde{u}(t_n + c_1\Delta t), \ldots, \tilde{u}(t_n + c_s\Delta t)]^{\mathrm{T}},$$
$$\tilde{\mathbf{f}}^n = [F(\tilde{u}(t_n + c_1\Delta t)), \ldots, F(\tilde{u}(t_n + c_s\Delta t))]^{\mathrm{T}},$$

where $c_1, \dots, c_s$ are the abscissae of the TSRK scheme. The vectors $\tilde{\mathbf{y}}^n, \tilde{\mathbf{f}}^n$ denote the true solution stage values and the true solution derivatives (values of F). Then the *truncation error* τ^n and *stage truncation errors* $\boldsymbol{\tau}^n$ are implicitly defined by

$$\tilde{\mathbf{y}}^n = \mathbf{d}\tilde{u}^{n-1} + (\mathbf{e} - \mathbf{d})\tilde{u}^n + \Delta t \mathbf{A}\tilde{\mathbf{f}}^n + \Delta t \boldsymbol{\tau}^n, \tag{3.2a}$$

$$\tilde{u}(t_{n+1}) = \theta\tilde{u}^{n-1} + (1-\theta)\tilde{u}^n + \Delta t \mathbf{b}^{\mathrm{T}}\tilde{\mathbf{f}}^n + \Delta t \tau^n. \tag{3.2b}$$

To find formulas for the truncation errors, we make use of the Taylor expansions

$$\tilde{u}(t_n + c_i\Delta t) = \sum_{k=0}^{\infty} \frac{1}{k!}\Delta t^k c_i^k \tilde{u}^{(k)}(t_n),$$

$$F(\tilde{u}(t_n + c_i\Delta t)) = \tilde{u}'(t_n + c_i\Delta t) = \sum_{k=1}^{\infty} \frac{1}{(k-1)!}\Delta t^{k-1} c_i^{k-1} \tilde{u}^{(k)}(t_n),$$

$$\tilde{u}(t_{n-1}) = \sum_{k=0}^{\infty} \frac{(-\Delta t)^k}{k!}\tilde{u}^{(k)}(t_n).$$

Substitution gives

$$\boldsymbol{\tau}^n = \sum_{k=1}^{\infty} \boldsymbol{\tau}_k \Delta t^{k-1} \tilde{u}^{(k)}(t_n), \tag{3.3a}$$

$$\tau^n = \sum_{k=1}^{\infty} \tau_k \Delta t^{k-1} \tilde{u}^{(k)}(t_n), \tag{3.3b}$$

where

$$\boldsymbol{\tau}_k = \frac{1}{k!}\left(\mathbf{c}^k - (-1)^k\mathbf{d}\right) - \frac{1}{(k-1)!}\mathbf{A}\mathbf{c}^{k-1},$$

$$\tau_k = \frac{1}{k!}\left(1 - (-1)^k\theta\right) - \frac{1}{(k-1)!}\mathbf{b}^{\mathrm{T}}\mathbf{c}^{k-1}.$$

Subtracting (3.2) from (3.1) gives

$$\boldsymbol{\epsilon}^n = \mathbf{d}\epsilon^{n-1} + (\mathbf{e} - \mathbf{d})\epsilon^n + \Delta t \mathbf{A}\boldsymbol{\delta}^n - \Delta t \boldsymbol{\tau}^n, \tag{3.4a}$$

$$\epsilon^{n+1} = \theta\epsilon^{n-1} + (1-\theta)\epsilon^n + \Delta t \mathbf{b}^{\mathrm{T}}\boldsymbol{\delta}^n - \Delta t \tau^n, \tag{3.4b}$$

where $\epsilon^{n+1} = u^{n+1} - \tilde{u}(t_{n+1})$ is the global error, $\boldsymbol{\epsilon}^n = \mathbf{y}^n - \tilde{\mathbf{y}^n}$, is the global stage error, and $\boldsymbol{\delta}^n = \mathbf{f}^n - \tilde{\mathbf{f}}^n$ is the right-hand-side stage error.

If we assume an expansion for the right-hand-side stage errors $\boldsymbol{\delta}^n$ as a power series in Δt

$$\boldsymbol{\delta}^n = \sum_{k=0}^{p-1} \boldsymbol{\delta}_k^n \Delta t^k + \mathcal{O}(\Delta t^p), \tag{3.5}$$

then substituting the expansions (3.5) and (3.3) into the global error formula (3.4) yields

$$\boldsymbol{\epsilon}^n = \mathbf{d}\epsilon^{n-1} + (\mathbf{e} - \mathbf{d})\epsilon^n + \sum_{k=0}^{p-1} \mathbf{A}\boldsymbol{\delta}_k^n \Delta t^{k+1} - \sum_{k=1}^{p} \boldsymbol{\tau}_k \tilde{u}^{(k)}(t_n)\Delta t^k + \mathcal{O}(\Delta t^{p+1}), \tag{3.6a}$$

$$\epsilon^{n+1} = \theta\epsilon^{n-1} + (1-\theta)\epsilon^n + \sum_{k=0}^{p-1} \mathbf{b}^{\mathrm{T}}\boldsymbol{\delta}_k^n \Delta t^{k+1} - \sum_{k=1}^{p} \tau_k \tilde{u}^{(k)}(t_n)\Delta t^k + \mathcal{O}(\Delta t^{p+1}). \tag{3.6b}$$

Hence we find the method is consistent of order p if

$$\mathbf{b}^{\mathrm{T}}\boldsymbol{\delta}_k^n = 0 \quad (0 \le k \le p-1) \qquad \text{and} \qquad \tau_k = 0 \quad (1 \le k \le p). \tag{3.7}$$

It remains to determine the vectors $\boldsymbol{\delta}_k^n$ in the expansion (3.5). In fact, we can relate these recursively to the $\boldsymbol{\epsilon}_k$. First we define

$$\mathbf{t}_n = t_n\mathbf{e} + \mathbf{c}\Delta t,$$
$$\mathbf{F}(\mathbf{y}, \mathbf{t}) = [F(y_1(t_1)), \ldots, F(y_s(t_s))]^{\mathrm{T}},$$

where $\mathbf{c}$ is the vector of abscissae. Then we have the Taylor series

$$\begin{aligned}\mathbf{f}^n = \mathbf{F}(\mathbf{y}^n, \mathbf{t}_n) &= \tilde{\mathbf{f}}^n + \sum_{j=1}^{\infty} \frac{1}{j!}(\mathbf{y}^n - \tilde{\mathbf{y}}^n)^j \cdot \mathbf{F}^{(j)}(\tilde{\mathbf{y}}^n, \mathbf{t}_n) \\ &= \tilde{\mathbf{f}}^n + \sum_{j=1}^{\infty} \frac{1}{j!}(\boldsymbol{\epsilon}^n)^j \cdot \mathbf{g}_j(\mathbf{t}_n),\end{aligned}$$

where

$$\mathbf{F}^{(j)}(\mathbf{y}, \mathbf{t}) = [F^{(j)}(y_1(t_1)), \ldots, F^{(j)}(y_s(t_s))]^{\mathrm{T}},$$
$$\mathbf{g}_j(\mathbf{t}) = [F^{(j)}(y(t_1)), \ldots, F^{(j)}(y(t_s))]^{\mathrm{T}},$$

and the dot product denotes componentwise multiplication. Thus

$$\boldsymbol{\delta}^n = \mathbf{f}^n - \tilde{\mathbf{f}}^n = \sum_{j=1}^{\infty} \frac{1}{j!}(\boldsymbol{\epsilon}^n)^j \cdot \mathbf{g}_j(t_n\mathbf{e} + \mathbf{c}\Delta t).$$

Since

$$\mathbf{g}_j(t_n\mathbf{e} + \mathbf{c}) = \sum_{l=0}^{\infty} \frac{\Delta t^l}{l!}\mathbf{C}^l\mathbf{g}_j^{(l)}(t_n),$$

where $\mathbf{C} = \mathrm{diag}(\mathbf{c})$, we finally obtain the desired expansion:

$$\boldsymbol{\delta}^n = \sum_{j=1}^{\infty}\sum_{l=0}^{\infty} \frac{\Delta t^l}{j!l!}\mathbf{C}^l(\boldsymbol{\epsilon}^n)^j \cdot \mathbf{g}_j^{(l)}(t_n). \tag{3.8}$$

To determine the coefficients $\boldsymbol{\delta}_k$, we alternate recursively between (3.8) and (3.6a). Typically, the abscissae $\mathbf{c}$ are chosen as $\mathbf{c} = \mathbf{A}\mathbf{e}$ so that $\boldsymbol{\tau}_1 = 0$. With these choices, we collect the terms relevant for up to fifth-order accuracy:

Terms appearing in $\boldsymbol{\delta}_1$:	$\emptyset$
Terms appearing in $\boldsymbol{\epsilon}_2$:	$\boldsymbol{\tau}_2$
Terms appearing in $\boldsymbol{\delta}_2$:	$\boldsymbol{\tau}_2$
Terms appearing in $\boldsymbol{\epsilon}_3$:	$\mathbf{A}\boldsymbol{\tau}_2, \boldsymbol{\tau}_3$
Terms appearing in $\boldsymbol{\delta}_3$:	$\mathbf{C}\boldsymbol{\tau}_2, \mathbf{A}\boldsymbol{\tau}_2, \boldsymbol{\tau}_3$
Terms appearing in $\boldsymbol{\epsilon}_4$:	$\mathbf{AC}\boldsymbol{\tau}_2, \mathbf{A}^2\boldsymbol{\tau}_2, \mathbf{A}\boldsymbol{\tau}_3, \boldsymbol{\tau}_4$
Terms appearing in $\boldsymbol{\delta}_4$:	$\mathbf{AC}\boldsymbol{\tau}_2, \mathbf{A}^2\boldsymbol{\tau}_2, \mathbf{A}\boldsymbol{\tau}_3, \boldsymbol{\tau}_4, \mathbf{CA}\boldsymbol{\tau}_2, \mathbf{C}\boldsymbol{\tau}_3, \mathbf{C}^2\boldsymbol{\tau}_2, \boldsymbol{\tau}_2^2$

The order conditions are then given by (3.7). In fact, we are left with order conditions identical to those for Runge–Kutta methods, except that the definitions of the stage truncation errors $\boldsymbol{\tau}_k, \tau_k$ and of the abscissas $\mathbf{c}$ are modified. For a list of the order conditions up to eighth order, see [1, Appendix A].

3.2. Order and stage order of TSRK methods. The presence of the term $\boldsymbol{\tau}_2^2$ in $\boldsymbol{\delta}_4$ leads to the order condition $\mathbf{b}^{\mathrm{T}}\boldsymbol{\tau}_2^2 = 0$. For irreducible SSP methods, since $\mathbf{b} > 0$ (by Theorem 1), this implies that $\boldsymbol{\tau}_2^2 = 0$, i.e., fifth-order SSP TSRK methods must have at least stage order two. The corresponding condition for Runge–Kutta methods leads to the well-known result that no explicit RK method can have order greater than four and a positive SSP coefficient. Similarly, the conditions for seventh order include $\mathbf{b}^{\mathrm{T}}\boldsymbol{\tau}_3^2 = 0$, which leads (together with the nonnegativity of $\mathbf{A}$) to the result that implicit SSP RK methods have order at most six. In general, the conditions for order $2k+1$ include the condition $\mathbf{b}^{\mathrm{T}}\boldsymbol{\tau}_k^2 = 0$. Thus, like SSP Runge–Kutta methods, SSP TSRK methods have a lower bound on the stage order and an upper bound on the overall order.

THEOREM 2. *Any irreducible TSRK method* (2.1) *of order p with positive SSP coefficient has stage order at least $\lfloor \frac{p-1}{2} \rfloor$.*

Proof. Consider an irreducible TSRK method with $\mathcal{C} > 0$. Following the procedure outlined above, we find that for order p, the coefficients must satisfy

$$\mathbf{b}^{\mathrm{T}}\boldsymbol{\tau}_k^2 = 0, \qquad k = 1, 2, \ldots, \left\lfloor \frac{p-1}{2} \right\rfloor.$$

Since $\mathbf{b} > 0$ by Theorem 1, this implies that

$$\boldsymbol{\tau}_k^2 = 0, \qquad k = 1, 2, \ldots, \left\lfloor \frac{p-1}{2} \right\rfloor. \qquad \square$$

Application of Theorem 2 dramatically simplifies the order conditions for high-order SSP TSRKs. This is because increased stage order leads to the vanishing of many of the order conditions. Additionally, Theorem 2 leads to an upper bound on the order of explicit SSP TSRKs.

THEOREM 3. *The order of an irreducible explicit SSP TSRK method is at most eight. Furthermore, if the method has order greater than six, it has a stage y_j^n identically equal to u^{n-1}.*

Proof. To prove the second part, consider an explicit irreducible TSRK method with order greater than six. By Theorem 2, this method must have stage order at least three. Solving the conditions for stage y_2 to have stage order three gives that c_2 must be equal to -1 or 0. Taking $c_2 = -1$ implies that $y_2 = u^{n-1}$. Taking $c_2 = 0$ implies $y_2 = y_1 = u^n$; in this case, there must be some stage y_j not equal to u^n and we find that necessarily $c_j = -1$ and hence $y_j = u^{n-1}$.

To prove the first part, suppose there exists an irreducible SSP TSRK method of order nine. By Theorem 2, this method must have stage order at least four. Let j be the index of the first stage that is not identically equal to u^{n-1} or u^n. Solving the conditions for stage y_j to have order four reveals that c_j must be equal to $-1, 0$, or 1. The cases $c_j = -1$ and $c_j = 0$ lead to $y_j = u^{n-1}$ and $y_j = u^n$, contradicting our assumption. Taking $c_j = 1$ leads to $d_j = 5$. By Theorem 1, this implies that the method is not SSP. $\square$

We remark here that other results on the structure of SSP TSRK methods may be obtained by similar use of the stage order conditions and Theorems 1 and 2. We list some examples here, but omit the proofs since these results are not essential to our present purpose.

1. Any irreducible SSP TSRK method (implicit or explicit) of order greater than four must have a stage equal to u^{n-1} or u^n.
2. Each abscissa c_i of any irreducible SSP TSRK method of order greater than four must be nonnegative or equal to -1.

3. Irreducible (implicit) SSP TSRK methods with $p > 8$ must have a stage y_j^n identically equal to u^{n-1}.

4. Optimal SSP explicit TSRK methods. Our objective in this section is to find explicit TSRK methods that have the largest possible SSP coefficient. A method of order p with s stages is said to be optimal if it has the largest value of $\mathcal{C}$ over all explicit TSRK methods of order at least p with no more than s stages. The methods presented were found via numerical search as described below. Although we do not know in general if these methods are globally optimal, our search recovered the global optimum in every case for which it was already known.

4.1. Cost of explicit TSRK methods. In comparing explicit methods with different numbers of stages, one is usually interested in the relative time advancement per computational cost. For this purpose, we define the *effective SSP coefficient*

$$\mathcal{C}_{\text{eff}} = \frac{\mathcal{C}}{s}.$$

This normalization enables us to compare the cost of integration up to a given time, assuming that the time-step is chosen according to the SSP restriction.

For multistep, multistage methods like those considered here, the cost of the method must be determined carefully. For example, consider a method satisfying the following property:

$$(d_i, a_{i,1}, \ldots, a_{i,s}) = (1, 0, \ldots, 0, 0, \ldots, 0) \text{ for some } i, \text{ and} \tag{4.1a}$$

$$(d_j, a_{j,1}, \ldots, a_{j,s}) = (0, 0, \ldots, 0, 0, \ldots, 0) \text{ for some } j. \tag{4.1b}$$

In this case the TSRK method has stages y_i^n, y_j^n identically equal to u^{n-1}, u^n, respectively. Since $F(u^{n-1})$ has been computed at the previous step, it can be used again for free. Therefore, for such methods, we consider the number of stages (when determining $\mathcal{C}_{\text{eff}}$) to be one less than the apparent value. Separate searches were performed to find optimal methods that do or do not satisfy condition (4.1). It turns out that all optimal explicit methods found do satisfy this condition. It is convenient to write such methods by numbering the stages from zero, so that s accurately represents the cost of the method:

$$y_0^n = u^{n-1}, \tag{4.2a}$$

$$y_1^n = u^n, \tag{4.2b}$$

$$y_i^n = d_i u^{n-1} + (1 - d_i)u^n + \Delta t \sum_{j=0}^{s} a_{ij} F(y_j^n), \qquad 2 \le i \le s, \tag{4.2c}$$

$$u^{n+1} = \theta u^{n-1} + (1 - \theta)u^n + \Delta t \sum_{j=0}^{s} b_j F(y_j^n). \tag{4.2d}$$

Note that this can still be written in the compact form (3.1) where $\mathbf{A}$ is then an $s + 1 \times s + 1$ matrix with the first two rows identically zero and the vector $\mathbf{d}$ has $d_0 = 1$ and $d_1 = 0$. In the remainder of this paper, we deal only with methods of the form (4.2) and refer to (4.2) as an s-stage method.

Remark 2. General augmented Runge-Kutta methods (2.2) are equivalent to the two-step methods of Type 4 considered in [8]. The special augmented Runge–Kutta methods (4.2) are equivalent to the two-step methods of Type 5 considered in [8].

4.2. Formulating the optimization problem. The optimization problem is formulated using the theory of section 2:

$$\max_{\mathbf{S},\mathbf{T}} r,$$
$$\text{subject to} \quad \begin{cases} (\mathbf{I}+r\mathbf{T})^{-1}\mathbf{S} \geq 0, \\ (\mathbf{I}+r\mathbf{T})^{-1}\mathbf{T} \geq 0, \\ \Phi_p(\mathbf{S},\mathbf{T}) = 0, \end{cases}$$

where the inequalities are understood componentwise and $\Phi_p(\mathbf{S},\mathbf{T})$ represents the order conditions up to order p. This formulation, solved numerically in MATLAB using a sequential quadratic programming approach (fmincon in the optimization toolbox), was used to find the methods given below.

Remark 3. By virtue of Theorem 1, optimal (explicit) SSP methods found in the class of augmented Runge–Kutta methods are in fact optimal over the larger class of (explicit) TSRK methods (2.1). Also, because they do not use intermediate stages from previous time-steps, special conditions on the starting method (important for general TSRK methods (2.1) [16, 45, 44]) are unnecessary. Instead, augmented Runge–Kutta methods can be started with any SSP Runge–Kutta method of the appropriate order.

Remark 4. In the theory of [41], numerical processes with arbitrary input vectors are considered. When applying that theory to methods of the form (4.2), it is important to use a Spijker form that accounts for the fact that y_0^n, y_1^n are identically equivalent to u^{n-1}, u^n. When optimizing such methods, we therefore use the following form (with $m = s+2$, $l = 2$, $J = \{2, s+2\}$):

$$\mathbf{x}^n = \left(u^{n-1}, u^n\right)^{\mathrm{T}}, \quad \mathbf{w}^n = \left(y_0^n, y_1^n, y_2^n, \ldots, y_s^n, u^{n+1}\right)^{\mathrm{T}},$$
$$\mathbf{S} = \begin{pmatrix} \mathbf{d} & \mathbf{e}-\mathbf{d} \\ \theta & 1-\theta \end{pmatrix}, \qquad \mathbf{T} = \begin{pmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{b}^{\mathrm{T}} & 0 \end{pmatrix}.$$

Here again, the first two rows of $\mathbf{A}$ are identically zero, and $d_0 = 1, d_1 = 0$.

4.3. Low-storage implementation of augmented Runge–Kutta methods. The form (2.7), with $r = \mathcal{C}(\mathbf{S},\mathbf{T})$, typically yields very sparse coefficient matrices for optimal methods. This form is useful for a low-storage implementation. This form can be written out explicitly as

(4.3a)
$$y_i^n = \tilde{d}_i u^{n-1} + \left(1 - \tilde{d}_i - \sum_{j=0}^{s} q_{ij}\right) u^n + \sum_{j=0}^{s} q_{ij}\left(y_j^n + \frac{\Delta t}{r} F(y_j^n)\right), \quad (0 \leq i \leq s),$$

(4.3b)
$$u^{n+1} = \tilde{\theta} u^{n-1} + \left(1 - \tilde{\theta} - \sum_{j=0}^{s} \eta_j\right) u^n + \sum_{j=0}^{s} \eta_j \left(y_j^n + \frac{\Delta t}{r} F(y_j^n)\right),$$

where the coefficients are given by (using the relations (2.8)):

$$\mathbf{Q} = r\mathbf{A}(\mathbf{I}+r\mathbf{A})^{-1}, \qquad \eta = r\mathbf{b}^{\mathrm{T}}(\mathbf{I}+r\mathbf{A})^{-1},$$
$$\tilde{\mathbf{d}} = \mathbf{d} - \mathbf{Q}\mathbf{d}, \qquad \tilde{\theta} = \theta - \eta^{\mathrm{T}}\mathbf{d}.$$

Because the optimal methods turn out to satisfy (4.1), we have again written the sums in (4.3) from zero instead of one. Note that the entries of $\mathbf{Q}, \eta, \tilde{\mathbf{d}}$ are thus indexed from 0 to s.

TABLE 4.1

Effective SSP coefficients $\mathcal{C}_{\text{eff}}$ of optimal explicit TSRK methods of orders two to four. Results known to be optimal from [8] *or* [31] *are shown in* ***bold****.*

$s \setminus p$	2	3	4
2	**0.707**	**0.366**	
3	**0.816**	**0.550**	**0.286**
4	**0.866**	**0.578**	**0.398**
5	**0.894**	0.598	0.472
6	**0.913**	0.630	0.509
7	**0.926**	0.641	0.534
8	**0.935**	0.653	0.562
9	**0.943**	0.667	0.586
10	**0.949**	0.683	0.610

When implemented in this form, many of the methods presented in the next section have modest storage requirements, despite using large numbers of stages. The analogous form for Runge–Kutta methods was used in [28].

In the following sections we discuss the numerically optimal methods, and in Tables A.1–A.5 we give the coefficients in the low-storage form (4.3) for some numerically optimal methods.

4.4. Optimal explicit methods of orders one to four. In the case of first-order methods, one can do no better (in terms of effective SSP coefficient) than the forward Euler method. For orders two to four, SSP coefficients of optimal methods found by numerical search are listed in Table 4.1. We list these mainly for completeness, since SSP Runge–Kutta methods with good properties exist up to order four.

In [29], upper bounds for the values in Table 4.1 are found by computing optimally contractive general linear methods for linear systems of ODEs. Comparing the present results to the two-step results from that work, we see that this upper bound is achieved (as expected) for all first- and second-order methods, and even for the two- and three-stage third-order methods.

Optimal methods found in [8] include two-step general linear methods of up to fourth order using up to four stages. By comparing Table 4.1 with the results therein, we see that the SSP coefficients of the optimal methods among the classes examined in both works (namely, for $1 \le s \le 4$, $2 \le p \le 4$) agree. The methods found in [8] are produced by software that guarantees global optimality.

All results listed in bold are thus known to be optimal because they match those obtained in [8], [31], or both. This demonstrates that our numerical optimization approach was able to recover all known globally optimal methods and suggests that the remaining methods found in the present work may also be globally optimal.

The optimal s-stage, second-order SSP TSRK method has SSP coefficient $\mathcal{C} = \sqrt{s(s-1)}$ and nonzero coefficients

$$\begin{aligned} q_{i,i-1} &= 1 && (2 \le i \le s),\\ \eta_s &= 2(\sqrt{s(s-1)} - s + 1),\\ \tilde{\mathbf{d}} &= [1, 0, \ldots, 0],\\ \tilde{\theta} &= 2(s - \sqrt{s(s-1)}) - 1. \end{aligned}$$

Note that these methods have $\mathcal{C}_{\text{eff}} = \sqrt{\frac{s-1}{s}}$, whereas the corresponding optimal Runge–Kutta methods have $\mathcal{C}_{\text{eff}} = \frac{s-1}{s}$. Using the low-storage assumption intro-

TABLE 4.2
Effective SSP coefficients $\mathcal{C}_{\text{eff}}$ of optimal explicit two-step Runge–Kutta methods of orders five to eight.

$s \setminus p$	5	6	7	8
4	0.214			
5	0.324			
6	0.385	0.099		
7	0.418	0.182		
8	0.447	0.242	0.071	
9	0.438	0.287	0.124	
10	0.425	0.320	0.179	
11	0.431	0.338	0.218	0.031
12	0.439	0.365	0.231	0.078

duced in [28], these methods can be implemented with just three storage registers, just one register more than is required for the optimal second-order SSP Runge–Kutta methods.

The optimal nine-stage, third-order method is remarkable in that it is a Runge–Kutta method. In other words, allowing the freedom of using an additional step does not improve the SSP coefficient in this case.

4.5. Optimal explicit methods of orders five to eight. Table 4.2 lists effective SSP coefficients of numerically optimal TSRK methods of orders five to eight. Although these methods require many stages, it should be remembered that high-order (non-SSP) Runge–Kutta methods also require many stages. Indeed, some of our SSP TSRK methods have fewer stages than the minimum number required to achieve the corresponding order for a Runge–Kutta method (regardless of SSP considerations).

The fifth-order methods present an unusual phenomenon: when the number of stages is allowed to be greater than 8, it is not possible to achieve a larger effective SSP coefficient than the optimal 8-stage method, even allowing as many as 12 stages. This appears to be accurate and not simply due to failure of the numerical optimizer, since in the 9-stage case the optimization scheme recovers the apparently optimal method in less than 1 minute but fails to find a better result after several hours.

The only existing SSP methods of order greater than four are the hybrid methods of Huang [21]. Comparing the best TSRK methods of each order with the best hybrid methods of each order, the TSRK methods have substantially larger effective SSP coefficients.

The effective SSP coefficient is a fair metric for comparison between methods of the same order of accuracy. Furthermore, our 12-stage TSRK methods have sparse coefficient matrices and can be implemented in the low-storage form (4.3). Specifically, the fifth- through eighth-order methods of 12 stages require only 5, 7, 7, and 10 memory locations per unknown, respectively, under the low-storage assumption employed in [28, 30]. Typically, the methods with fewer stages require the same or more storage, so there is no reason to prefer methods with fewer stages if they have lower effective SSP coefficients. Thus, for the sixth through the eighth order, the 12-stage methods seem preferable. The SSP TSRK methods recommended here even require less storage than what (non-SSP one-step) Runge–Kutta methods of the corresponding order would typically use.

In the case of fifth-order methods, the 8-stage method has a larger effective SSP coefficient than the 12-stage method, so the 8-stage method seems best in terms of efficiency. However the 8-stage method requires more storage registers (six) than the

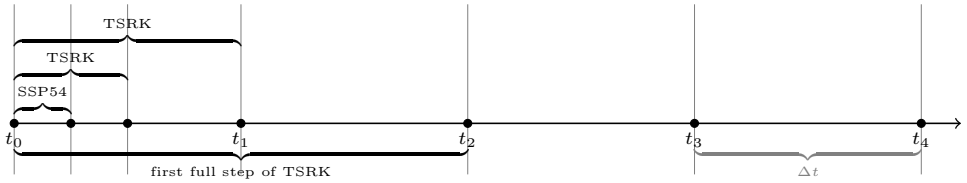

FIG. 5.1. *One possible startup procedure for SSP TSRK schemes. The first step from t_0 to t_1 is subdivided into substeps (here there are three substeps of sizes $\frac{h}{4}$, $\frac{h}{4}$, and $\frac{h}{2}$). An SSP Runge–Kutta scheme is used for the first substep. Subsequent substeps are taken with the TSRK scheme itself, doubling the stepsizes until reaching t_1. We emphasize that the startup procedure is not critical for this class of TSRK methods.*

12-stage method (five). So while the 8-stage method might be preferred for efficiency, the 12-stage method is preferred for low storage considerations.

5. Numerical experiments.

5.1. Start-up procedure. As mentioned in section 4.2, TSRK methods are not self-starting. Starting procedures for general linear methods can be complicated but for augmented Runge–Kutta methods they are straightforward. We only require that the starting procedure be of sufficient accuracy and that it also be SSP.

Figure 5.1 demonstrates one possible start-up procedure, which is employed in all numerical tests that follow. The first step of size Δt from t_0 to t_1 is subdivided into substeps in powers of two. The SSPRK(5,4) scheme [42, 32] or the SSPRK(10,4) scheme [28] is used for the first substep, with the stepsize Δt^* chosen small enough so that the local truncation error of the Runge–Kutta scheme is smaller than the global error of the TSRK scheme. For an order p method, this can be achieved by taking

$$\Delta t^* = \frac{\Delta t}{2^\gamma},\ \gamma \in \mathbb{Z}, \quad \text{and} \quad (\Delta t^*)^5 = A\Delta t^p = O(\Delta t^p). \tag{5.1}$$

Subsequent substeps are taken with the TSRK scheme itself, doubling the stepsizes until reaching t_1. From there, the TSRK scheme repeatedly advances the solution from t_n to t_{n+1} using previous step values u_{n-1} and u_n.

5.2. Order verification. Convergence studies on two ODE test problems confirm that the SSP TSRK methods achieve their design orders. The first is the Dahlquist test problem $u' = \lambda u$, with $u^0 = 1$ and $\lambda = 2$, solved until $t_f = 1$. Figure 5.2 shows a sample of TSRK methods achieving their design orders on this problem. The starting procedure used SSPRK(10,4) with the constant A in (5.1) set, respectively, to $[\frac{1}{2}, \frac{1}{2}, 10^{-2}, 10^{-3}, 10^{-3}]$ for orders $p = 4$, 5, 6, 7, and 8.

The nonlinear van der Pol problem (e.g., [35]) can be written as an ODE initial value problem consisting of two components

$$\begin{aligned} u_1' &= u_2, \\ u_2' &= \frac{1}{\epsilon}\left(-u_1 + (1 - u_1^2)u_2\right). \end{aligned}$$

We take $\epsilon = 0.01$ and initial condition $u^0 = [2; -0.6654321]$ and solve until $t_f = \frac{1}{2}$. The starting procedure is based on SSPRK(10,4) with constant $A = 1$ in (5.1). The maximum norm error is computed by comparing against a highly accurate reference solution calculated with MATLAB's ODE45 routine. Figure 5.2 shows a sample of the TSRK schemes achieving their design orders on this problem.

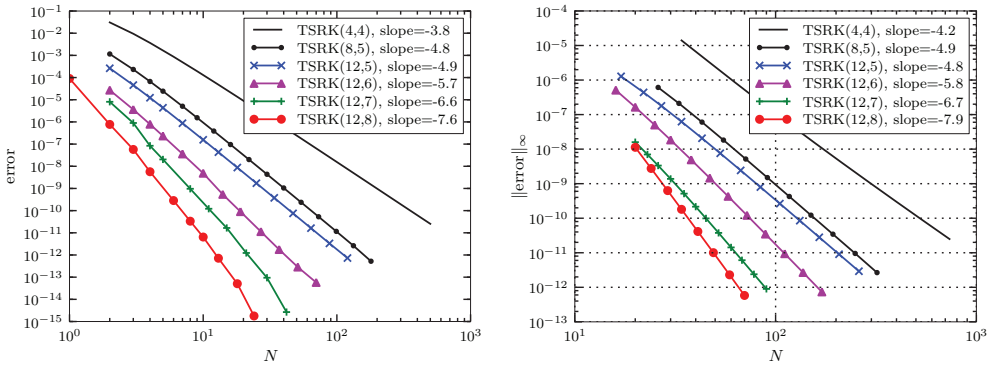

FIG. 5.2. *Convergence results for some TSRK schemes on the Dahlquist test problem (left) and van der Pol problem (right). The slopes of the lines confirm the design orders of the TSRK methods.*

5.3. High-order WENO. WENO schemes [18, 17, 26] are finite difference or finite volume schemes that use a combination of lower-order fluxes to obtain higher-order approximations, while ensuring non-oscillatory solutions. This is accomplished by using adaptive stencils which approach centered difference stencils in smooth regions and one-sided stencils near discontinuities. Many WENO methods exist, and the difference between them is in the computation of the stencil weights. WENO methods can be constructed to be high order [13, 3]. In [13], WENO methods of up to 17th order were implemented and tested. However, the authors note that in some of their computations the error was limited by the order of the time integration, which was relatively low (third-order SSPRK(3,3)). In Figure 5.3, we reproduce the numerical experiment of [13, Fig. 15], specifically the two-dimensional linear advection of a sinusoidal initial condition $u_0(x,y) = \sin(\pi(x+y))$, in a periodic square using various high-order WENO methods and our TSRK integrators of order five, seven, and eight using 12 stages. Compared with [13, Fig. 15], we note that the error is no longer dominated by the temporal error. Thus the higher-order SSP TSRK schemes allow us to see the behavior of the high-order WENO spatial discretization schemes.

5.4. Buckley–Leverett. The Buckley–Leverett equation is a model for two-phase flow through porous media and consists of the conservation law

$$U_t + f(U)_x = 0 \quad \text{with} \quad f(U) = \frac{U^2}{U^2 + a(1-U)^2}.$$

We use $a = \frac{1}{3}$ and initial conditions

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

on $x \in [0,1)$ with periodic boundary conditions. Our spatial discretization uses 100 points and following [23, 31] we use a conservative scheme with Koren limiter. We compute the solution until $t_f = \frac{1}{8}$. For this problem, the Euler solution is total variation diminishing (TVD) for $\Delta t \le \Delta t_{\text{FE}} = 0.0025$ [31]. As discussed above, we must also satisfy the SSP time-step restriction for the starting method.

Figure 5.4 shows typical solutions using an TSRK scheme with time-step $\Delta t = \sigma \Delta t_{\text{FE}}$. Table 5.1 shows the maximal TVD time-step sizes, expressed as $\Delta t = \sigma_{\text{BL}} \Delta t_{\text{FE}}$, for the Buckley–Leverett test problem. The results show that the SSP

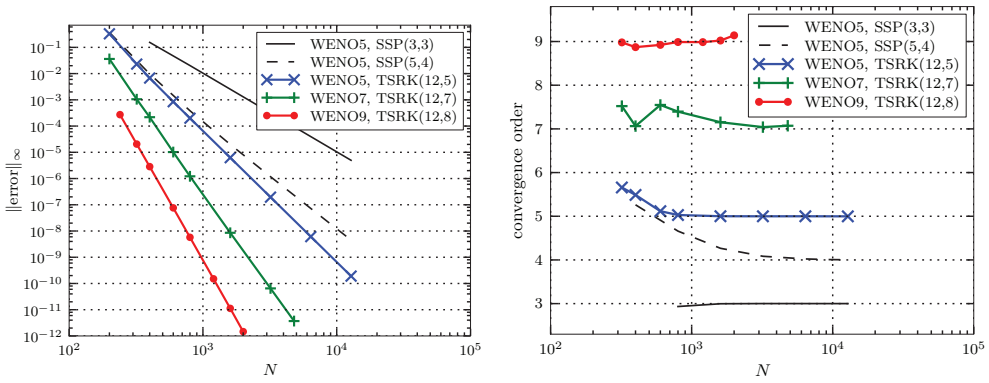

FIG. 5.3. *Convergence results for two-dimensional advection using rth-order WENO discretizations and the TSRK integrators (c.f.,* [13, Fig. 15]*). Maximum error versus number of spatial grid points in each direction (left). Observed orders of accuracy calculated from these errors (right). Computed using the same parameters as* [13, Fig. 15] *(final time* $t_f = 20$*,* $\Delta t = 0.5\Delta x$*, mapped WENO spatial discretization with* $p_\beta = r$*). Starting procedure is as described in section* 5.1 *using the SSPRK*(5,4) *scheme for the initial substep.*

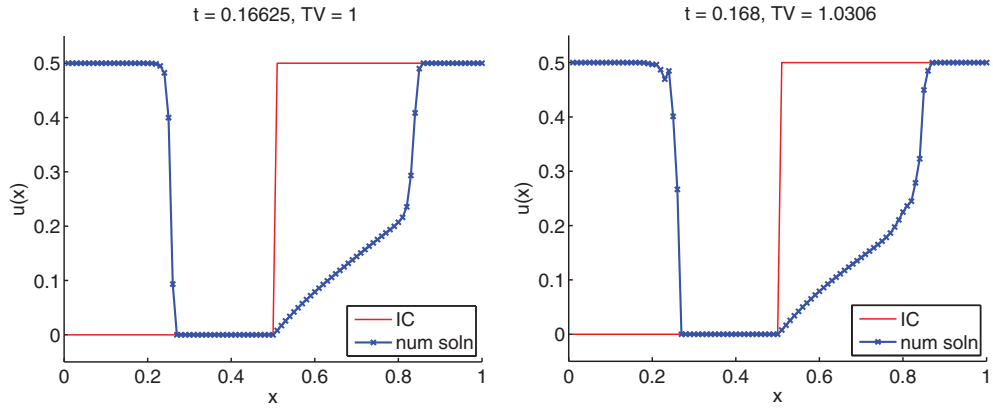

FIG. 5.4. *Two numerical solutions of the Buckley–Leverett test problem. Left: time-step satisfies the SSP time-step restriction (TSRK*(8,5) *using* $\Delta t = 3.5\Delta t_{\mathrm{FE}}$*). Right: time-step does not satisfy the restriction* ($\Delta t = 5.6\Delta t_{\mathrm{FE}}$) *and visible oscillations have formed, increasing the total variation of the solution.*

TABLE 5.1

SSP coefficients versus largest time-steps exhibiting the TVD property ($\Delta t = \sigma_{BL}\Delta t_{\mathrm{FE}}$) *on the Buckley–Leverett example for some of the explicit SSP TSRK*(*s*,*p*) *schemes. The effective SSP coefficient* $\mathcal{C}_{\mathrm{eff}}$ *should be a lower bound for* σ_{BL}/s *and indeed this is observed. SSPRK*(10,4) [28] *is used as the first step in the starting procedure.*

Method	Theoretical		Observed	
	$\mathcal{C}$	$\mathcal{C}_{\mathrm{eff}}$	σ_{BL}	σ_{BL}/s
TSRK(4,4)	1.5917	0.398	2.16	0.540
TSRK(8,5)	3.5794	0.447	4.41	0.551
TSRK(12,5)	5.2675	0.439	6.97	0.581
TSRK(12,6)	4.3838	0.365	6.80	0.567
TSRK(12,7)	2.7659	0.231	4.86	0.405
TSRK(12,8)	0.9416	0.079	4.42	0.368

coefficient is a lower bound for what is observed in practice, confirming the theoretical importance of the SSP coefficient.

6. Conclusions. In this paper we analyzed the strong stability preserving property of two-step Runge–Kutta (TSRK) methods. We found that SSP TSRK methods have a relatively simple form and that explicit methods in this class are subject to a maximal order of eight. We have presented numerically optimal explicit SSP TSRK methods of order up to this bound of eight. These methods overcome the fourth-order barrier for (one-step) SSP Runge–Kutta methods and allow larger SSP coefficients than the corresponding order multistep methods. The discovery of these methods was facilitated by our formulation of the optimization problem in an efficient form, aided by simplified order conditions and constraints on the coefficients derived by using the SSP theory for general linear methods. These methods feature favorable storage properties and are easy to implement and start, as they do not use stage values from previous steps.

We show that high-order SSP TSRK methods are useful for the time integration of a variety of hyperbolic PDEs, especially in conjunction with high-order spatial discretizations. In the case of a Buckley–Leverett numerical test case, the SSP coefficient of these methods is confirmed to provide a lower bound for the actual time-step needed to preserve the total variation diminishing property.

The order conditions and SSP conditions we have derived for these methods extend in a very simple way to methods with more steps. Future work will investigate methods with more steps and will further investigate the use of start-up methods for use with SSP multistep Runge–Kutta methods.

Appendix A. Coefficients of numerically optimal methods.

TABLE A.1
Coefficients of the optimal explicit eight-stage fifth-order SSP TSRK method (written in form (4.3)*).*

$\tilde{\theta} = 0$	$q_{3,0} = 0.058121281984411$	$q_{8,1} = 0.110261531523242$
$\tilde{d}_0 = 1.000000000000000$	$q_{7,0} = 0.020705281786630$	$q_{3,2} = 0.941878718015589$
$\tilde{d}_7 = 0.003674184820260$	$q_{8,0} = 0.008506650138784$	$q_{8,2} = 0.030113037742445$
$\eta_2 = 0.179502832154858$	$q_{2,1} = 0.914669227052357$	$q_{4,3} = 0.802870131352638$
$\eta_3 = 0.073789956884809$	$q_{4,1} = 0.036365639242841$	$q_{5,4} = 0.508785659339445$
$\eta_6 = 0.017607159013167$	$q_{5,1} = 0.491214340660555$	$q_{6,5} = 0.433864768368758$
$\eta_8 = 0.729100051947166$	$q_{6,1} = 0.566135231631241$	$q_{7,6} = 0.883974453741544$
$q_{2,0} = 0.085330772947643$	$q_{7,1} = 0.091646079651566$	$q_{8,7} = 0.851118780595529$

TABLE A.2
Coefficients of the optimal explicit 12*-stage fifth-order SSP TSRK method (written in form* (4.3)*).*

$\tilde{\theta} = 0$	$q_{6,1} = 0.041456384663457$	$q_{6,5} = 0.897622496599848$
$\tilde{d}_0 = 1$	$q_{7,1} = 0.893102584263455$	$q_{7,6} = 0.106897415736545$
$\eta_1 = 0.010869478269914$	$q_{9,1} = 0.103110842229401$	$q_{8,6} = 0.197331844351083$
$\eta_6 = 0.252584630617780$	$q_{10,1} = 0.109219062395598$	$q_{8,7} = 0.748110262498258$
$\eta_{10} = 0.328029300816831$	$q_{11,1} = 0.069771767766966$	$q_{9,8} = 0.864072067200705$
$\eta_{12} = 0.408516590295475$	$q_{12,1} = 0.050213434903531$	$q_{10,9} = 0.890780937604403$
$q_{2,0} = 0.037442206073461$	$q_{3,2} = 0.750941165462252$	$q_{11,10} = 0.928630488244921$
$q_{3,0} = 0.004990369159650$	$q_{4,3} = 0.816192058725826$	$q_{12,11} = 0.949786565096469$
$q_{2,1} = 0.962557793926539$	$q_{5,4} = 0.881400968167496$	

TABLE A.3

Coefficients of the optimal explicit 12*-stage sixth-order SSP TSRK method (written in form* (4.3)*).*

$\tilde{\theta} = 2.455884612148108e-04$	$q_{3,2} = 0.590319496200531$	$q_{8,7} = 0.873267220579217$
$\tilde{d}_0 = 1$	$q_{4,3} = 0.729376762034313$	$q_{9,8} = 0.877348047199139$
$\tilde{d}_1 0 = 0.000534877909816$	$q_{5,4} = 0.826687833242084$	$q_{10,9} = 0.822483564557728$
$q_{2,0} = 0.030262100443273$	$q_{10,4} = 0.091956261008213$	$q_{11,10} = 0.587217894186976$
$q_{2,1} = 0.664746114331100$	$q_{11,4} = 0.135742974049075$	$q_{12,11} = 0.621756047217421$
$q_{6,1} = 0.656374628865518$	$q_{6,5} = 0.267480130553594$	$\eta_1 = 0.012523410805564$
$q_{7,1} = 0.210836921275170$	$q_{11,5} = 0.269086406273540$	$\eta_6 = 0.094203091821030$
$q_{9,1} = 0.066235890301163$	$q_{12,5} = 0.344231433411227$	$\eta_9 = 0.318700620499891$
$q_{10,1} = 0.076611491217295$	$q_{7,6} = 0.650991182223416$	$\eta_{10} = 0.107955864652328$
$q_{12,1} = 0.016496364995214$	$q_{12,6} = 0.017516154376138$	$\eta_{12} = 0.456039783326905$

TABLE A.4

Coefficients of the optimal explicit 12*-stage seventh-order SSP TSRK method (written in form* (4.3)*).*

$\tilde{\theta} = 1.040248277612947e-04$	$q_{2,1} = 0.849449065363225$	$q_{4,3} = 0.166320497215237$
$\tilde{d}_0 = 1.000000000000000$	$q_{3,1} = 0.120943274105256$	$q_{10,3} = 0.032851385162085$
$\tilde{d}_2 = 0.003229110378701$	$q_{4,1} = 0.368587879161520$	$q_{5,4} = 0.343703780759466$
$\tilde{d}_4 = 0.006337974349692$	$q_{5,1} = 0.222052624372191$	$q_{6,5} = 0.519758489994316$
$\tilde{d}_5 = 0.002497954201566$	$q_{6,1} = 0.137403913798966$	$q_{7,6} = 0.598177722195673$
$\tilde{d}_8 = 0.017328228771149$	$q_{7,1} = 0.146278214690851$	$q_{8,7} = 0.488244475584515$
$\tilde{d}_1 2 = 0.000520256250682$	$q_{8,1} = 0.444640119039330$	$q_{10,7} = 0.356898323452469$
$\eta_0 = 0.000515717568412$	$q_{9,1} = 0.143808624107155$	$q_{11,7} = 0.508453150788232$
$\eta_1 = 0.040472655980253$	$q_{10,1} = 0.102844296820036$	$q_{12,7} = 0.496859299069734$
$\eta_6 = 0.081167924336040$	$q_{11,1} = 0.071911085489036$	$q_{9,8} = 0.704865150213419$
$\eta_7 = 0.238308176460039$	$q_{12,1} = 0.057306282668522$	$q_{10,9} = 0.409241038172241$
$\eta_8 = 0.032690786323542$	$q_{3,2} = 0.433019948758255$	$q_{11,10} = 0.327005955932695$
$\eta_{12} = 0.547467490509490$	$q_{7,2} = 0.014863996841828$	$q_{12,11} = 0.364647377606582$
$q_{2,0} = 0.147321824258074$	$q_{9,2} = 0.026942009774408$	

TABLE A.5

Coefficients of the optimal explicit 12*-stage eighth-order SSP TSRK method (written in form* (4.3)*).*

$\tilde{\theta} = 4.796147528566197e-05$	$q_{2,0} = 0.017683145596548$	$q_{9,3} = 0.029907847389714$
$\tilde{d}_0 = 1.000000000000000$	$q_{3,0} = 0.001154189099465$	$q_{5,4} = 0.165254103192244$
$\tilde{d}_2 = 0.036513886685777$	$q_{6,0} = 0.000065395819685$	$q_{7,4} = 0.005039627904425$
$\tilde{d}_4 = 0.004205435886220$	$q_{9,0} = 0.000042696255773$	$q_{8,4} = 0.069726774932478$
$\tilde{d}_5 = 0.000457751617285$	$q_{11,0} = 0.000116117869841$	$q_{9,4} = 0.022904196667572$
$\tilde{d}_7 = 0.007407526543898$	$q_{12,0} = 0.000019430720566$	$q_{12,4} = 0.130730221736770$
$\tilde{d}_8 = 0.000486094553850$	$q_{2,1} = 0.154785324942633$	$q_{6,5} = 0.229847794524568$
$\eta_1 = 0.033190060418244$	$q_{4,1} = 0.113729301017461$	$q_{9,5} = 0.095367316002296$
$\eta_2 = 0.001567085177702$	$q_{5,1} = 0.061188134340758$	$q_{7,6} = 0.252990567222936$
$\eta_3 = 0.014033053074861$	$q_{6,1} = 0.068824803789446$	$q_{9,6} = 0.176462398918299$
$\eta_4 = 0.017979737866822$	$q_{7,1} = 0.133098034326412$	$q_{10,6} = 0.281349762794588$
$\eta_5 = 0.094582502432986$	$q_{8,1} = 0.080582670156691$	$q_{11,6} = 0.327578464731509$
$\eta_6 = 0.082918042281378$	$q_{9,1} = 0.038242841051944$	$q_{12,6} = 0.149446805276484$
$\eta_7 = 0.020622633348484$	$q_{10,1} = 0.071728403470890$	$q_{8,7} = 0.324486261336648$
$\eta_8 = 0.033521998905243$	$q_{11,1} = 0.053869626312442$	$q_{9,8} = 0.120659479468128$
$\eta_9 = 0.092066893962539$	$q_{12,1} = 0.009079504342639$	$q_{10,9} = 0.166819833904944$
$\eta_{10} = 0.076089630105122$	$q_{3,2} = 0.200161251441789$	$q_{11,10} = 0.157699899495506$
$\eta_{11} = 0.070505470986376$	$q_{6,2} = 0.008642531617482$	$q_{12,11} = 0.314802533082027$
$\eta_{12} = 0.072975312278165$	$q_{4,3} = 0.057780552515458$	

Acknowledgments. The authors thank Marc Spijker for helpful conversations regarding reducibility of general linear methods. The authors are also grateful to an anonymous referee, whose careful reading and detailed comments improved several technical details of the paper.

REFERENCES

[1] P. Albrecht, *A new theoretical approach to Runge–Kutta methods*, SIAM J. Numer. Anal., 24 (1987), pp. 391–406.
[2] P. Albrecht, *The Runge–Kutta theory in a nutshell*, SIAM J. Numer. Anal., 33 (1996), pp. 1712–1735.
[3] D. S. Balsara and C.-W. Shu, *Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy*, J. Comput. Phys., 160 (2000), pp. 405–452.
[4] J. C. Butcher and S. Tracogna, *Order conditions for two-step Runge–Kutta methods*, Appl. Numer. Math., 24 (1997), pp. 351–364.
[5] J. Carrillo, I. M. Gamba, A. Majorana, and C.-W. Shu, *A WENO-solver for the transients of Boltzmann–Poisson system for semiconductor devices: performance and comparisons with Monte Carlo methods*, J. Comput. Phys., 184 (2003), pp. 498–525.
[6] L.-T. Cheng, H. Liu, and S. Osher, *Computational high-frequency wave propagation using the level set method, with applications to the semi-classical limit of Schrödinger equations*, Comm. Math. Sci., 1 (2003), pp. 593–621.
[7] V. Cheruvu, R. D. Nair, and H. M. Turfo, *A spectral finite volume transport scheme on the cubed-sphere*, Appl. Numer. Math., 57 (2007), pp. 1021–1032.
[8] E. Constantinescu and A. Sandu, *Optimal explicit strong-stability-preserving general linear methods*, SIAM J. Sci. Comput., 32 (2010), pp. 3130–3150.
[9] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell, *A hybrid particle level set method for improved interface capturing*, J. Comput. Phys., 183 (2002), pp. 83–116.
[10] L. Feng, C. Shu, and M. Zhang, *A hybrid cosmological hydrodynamic/N-body code based on a weighted essentially nonoscillatory scheme*, Astrophys. J., 612 (2004), pp. 1–13.
[11] L. Ferracina and M. N. Spijker, *Stepsize restrictions for the total-variation-diminishing property in general Runge–Kutta methods*, SIAM J. Numer. Anal., 42 (2004), pp. 1073–1093.
[12] L. Ferracina and M. N. Spijker, *An extension and analysis of the Shu–Osher representation of Runge–Kutta methods*, Math. Comput., 249 (2005), pp. 201–219.
[13] G. Gerolymos, D. Sénéchal, and I. Vallet, *Very-high-order WENO schemes*, J. Comput. Phys., 228 (2009), pp. 8481–8524.
[14] S. Gottlieb, C.-W. Shu, and E. Tadmor, *Strong stability preserving high-order time discretization methods*, SIAM Rev., 43 (2001), pp. 89–112.
[15] E. Hairer and G. Wanner, *Solving ordinary differential equations* II: *Stiff and differential-algebraic problems*, Springer Series in Computational Mathematics, Vol. 14, Springer-Verlag, Berlin, 1991.
[16] E. Hairer and G. Wanner, *Order conditions for general two-step Runge–Kutta methods*, SIAM J. Numer. Anal., 34 (1997), pp. 2087–2089.
[17] A. Harten, B. Engquist, S. Osher, and S. R. Chakravarthy, *Uniformly high-order accurate essentially nonoscillatory schemes.* III, J. Comput. Phys., 71 (1987), pp. 231–303.
[18] A. Harten, B. Engquist, S. Osher, and S. R. Chakravarthy, *Uniformly high order essentially non-oscillatory schemes.* I, SIAM J. Numer. Anal., 24 (1987), pp. 279–309.
[19] I. Higueras, *On strong stability preserving time discretization methods*, J. Sci. Comput., 21 (2004), pp. 193–223.
[20] I. Higueras, *Representations of Runge–Kutta methods and strong stability preserving methods*, SIAM J. Numer. Anal., 43 (2005), pp. 924–948.
[21] C. Huang, *Strong stability preserving hybrid methods*, Appl. Numer. Math., 59 (2009), pp. 891–904.
[22] W. Hundsdorfer and S. J. Ruuth, *On monotonicity and boundedness properties of linear multistep methods*, Math. Comput., 75 (2005), pp. 655–672.
[23] W. H. Hundsdorfer and J. G. Verwer, *Numerical solution of time-dependent advection-diffusion-reaction equations*, Springer Ser. Comput. Math. 33, Springer-Verlag, Berlin, 2003.

[24] Z. Jackiewicz, *General Linear Methods for Ordinary Differential Equations*, Wiley, New York, 2009.
[25] Z. Jackiewicz and S. Tracogna, *A general class of two-step Runge–Kutta methods for ordinary differential equations*, SIAM J. Numer. Anal., 32 (1995), pp. 1390–1427.
[26] G.-S. Jiang and C.-W. Shu, *Efficient implementation of weighted ENO schemes*, J. Comput. Phys., 126 (1996), pp. 202–228.
[27] S. Jin, H. Liu, S. Osher, and Y.-H. R. Tsai, *Computing multivalued physical observables for the semiclassical limit of the Schrödinger equation*, J. Comput. Phys., 205 (2005), pp. 222–241.
[28] D. I. Ketcheson, *Highly efficient strong stability preserving Runge–Kutta methods with low-storage implementations*, SIAM J. Sci. Comput., 30 (2008), pp. 2113–2136.
[29] D. I. Ketcheson, *Computation of optimal monotonicity preserving general linear methods*, Math. Comput., 78 (2009), pp. 1497–1513.
[30] D. I. Ketcheson, *Runge-Kutta methods with minimum storage implementations*, J. Comput. Phys., 229 (2010), pp. 1763–1773.
[31] D. I. Ketcheson, C. B. Macdonald, and S. Gottlieb, *Optimal implicit strong stability preserving Runge–Kutta methods*, Appl. Numer. Math., 52 (2009), p. 373.
[32] J. F. B. M. Kraaijevanger, *Contractivity of Runge–Kutta methods*, BIT, 31 (1991), pp. 482–528.
[33] S. Labrunie, J. Carrillo, and P. Bertrand, *Numerical study on hydrodynamic and quasi-neutral approximations for collisionless two-species plasmas*, J. Comput. Phys., 200 (2004), pp. 267–298.
[34] H. W. J. Lenferink, *Contractivity-preserving explicit linear multistep methods*, Numer. Math., 55 (1989), pp. 213–223.
[35] C. B. Macdonald, S. Gottlieb, and S. J. Ruuth, *A numerical study of diagonally split Runge–Kutta methods for PDEs with discontinuities*, J. Sci. Comput., 36 (2008), pp. 89–112.
[36] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang, *A PDE-based fast local level set method*, J. Comput. Phys., 155 (1999), pp. 410–438.
[37] S. J. Ruuth and W. Hundsdorfer, *High-order linear multistep methods with general monotonicity and boundedness properties*, J. Comput. Phys., 209 (2005), pp. 226–248.
[38] S. J. Ruuth and R. J. Spiteri, *Two barriers on strong-stability-preserving time discretization methods*, J. Sci. Comput., 17 (2002), pp. 211–220.
[39] C.-W. Shu, *Total-variation diminishing time discretizations*, SIAM J. Sci. Statist. Comp., 9 (1988), pp. 1073–1084.
[40] C.-W. Shu and S. Osher, *Efficient implementation of essentially non-oscillatory shock-capturing schemes*, J. Comput. Phys., 77 (1988), pp. 439–471.
[41] M. Spijker, *Stepsize conditions for general monotonicity in numerical initial value problems*, SIAM J. Numer. Anal., 45 (2007), pp. 1226–1245.
[42] R. J. Spiteri and S. J. Ruuth, *A new class of optimal high-order strong-stability-preserving time discretization methods*, SIAM J. Numer. Anal., 40 (2002), pp. 469–491.
[43] M. Tanguay and T. Colonius, *Progress in modeling and simulation of shock wave lithotripsy (SWL)*, in Proceedings of the Fifth International Symposium on Cavitation (CAV2003), 2003.
[44] J. H. Verner, *Improved starting methods for two-step Runge–Kutta methods of stage-order p-3*, Appl. Numer. Math., 56 (2006), pp. 388–396.
[45] J. H. Verner, *Starting methods for two-step Runge–Kutta methods of stage-order* 3 *and order* 6, J. Comput. Appl. Math., 185 (2006), pp. 292–307.