

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 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 WENO discretizations.

1. Strong Stability Preserving Methods. The concept of strong stability preserving methods was first 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 step sizes $\Delta t \leq \Delta t_{\text{FE}}$. 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 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 ,

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

*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. Dept. of Energy Computational Science Graduate Fellowship and by funding from King Abdullah University of Science and Technology (KAUST).

†Department of Mathematics, University of Massachusetts Dartmouth, North Dartmouth, MA 02747 (sgottlieb@umassd.edu). This work was supported by AFOSR grant number 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 post-doctoral fellowship, NSF grant number CCF-0321917, and by Award No KUK-C1-013-04 made by King Abdullah University of Science and Technology (KAUST).

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 . We say that the method is *strong stability preserving* (SSP) if (in the solution of (1.1)) it holds that

$$\|u^{n+1}\| \leq \max \{ \|u^n\|, \|u^{n-1}\|, \dots, \|u^{n-k+1}\| \}. \quad (1.4)$$

whenever (1.3) holds and the timestep satisfies

$$\Delta t \leq \mathcal{C} \Delta t_{\text{FE}}. \quad (1.5)$$

Throughout this work, \mathcal{C} is taken to be the largest value such that (1.5) and (1.3) together always imply (1.4). This value \mathcal{C} is called the *SSP coefficient* of the method.

For example, consider explicit multistep methods [39]:

$$u^{n+1} = \sum_{i=1}^k (\alpha_i u^{n+1-i} + \Delta t \beta_i F(u^{n+1-i})). \quad (1.6)$$

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 non-negative:

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

If the forward Euler method applied to (1.1) is strongly stable under the timestep restriction $\Delta t \leq \Delta t_{\text{FE}}$ and $\alpha_i, \beta_i \geq 0$ then the solution obtained by the multistep method (1.6) satisfies the strong stability bound (1.4) under the timestep restriction

$$\Delta t \leq \min_i \frac{\alpha_i}{\beta_i} \Delta t_{\text{FE}},$$

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

In the case of a one-step method the monotonicity requirement (1.4) reduces to

$$\|u^{n+1}\| \leq \|u^n\|.$$

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

$$\begin{aligned} u^{(0)} &= u^n, \\ u^{(i)} &= \sum_{j=0}^{i-1} \left(\alpha_{ij} u^{(j)} + \Delta t \beta_{ij} F(u^{(j)}) \right), \\ u^{n+1} &= u^{(s)}. \end{aligned} \quad (1.7)$$

If all the coefficients are non-negative, 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 timestep 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 (1.7) satisfies the strong stability bound (1.2) under the timestep 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.

Research in the field of 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 timestep restrictions. For instance, to obtain a fifth-order explicit linear multistep method with a time-step restriction of $\Delta t \leq 0.2 \Delta t_{\text{FE}}$ requires nine steps; for a sixth-order method, this increases to thirteen 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].

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 include [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], 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 *et. al.* 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 multi-stage Runge–Kutta methods, which are a generalization of both linear multistep methods and Runge–Kutta methods. We have found that deriving the order conditions using a generalization of

the approach presented in [2], and formulating the optimization problem using the approach from [31] allows us to efficiently find methods of up to eighth order with relatively modest storage requirements and large 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.

The major result of this paper is the development of SSP two-step Runge–Kutta methods of up to eighth order that are efficient and practical. In Section 2, we discuss some classes of two-step Runge–Kutta (TSRK) methods and prove that all SSP TSRK methods belong to one of two simple subclasses. In Section 3, we derive order conditions and show that explicit SSP TSRK methods have order at most eight. In Section 4, we formulate the optimization problem, give an efficient form for the implementation of SSP two-step Runge–Kutta methods, and present optimal methods of up to eighth order. The properties of our methods are compared with those of existing SSP methods including Runge–Kutta, linear multi-step, and hybrid methods [21], as well as the two- and three-step methods in [8]. Numerical verification of the optimal methods and a demonstration of the need for high-order time discretizations for use with high-order spatial discretizations is presented in Section 5. Conclusions and future work are discussed in Section 6.

2. SSP Two-step Runge–Kutta Methods. The principal focus of this work is on the strong stability preserving properties of two-step Runge–Kutta (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:

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

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

Here u^n and u^{n-1} denote solution values at the times $t = n\Delta t$ and $t = (n-1)\Delta t$, while the values y_i^n are intermediate stages used to compute the solution at the next time step. We will use the matrices and vectors \mathbf{A} , $\hat{\mathbf{A}}$, \mathbf{b} , $\hat{\mathbf{b}}$, and \mathbf{d} to refer to the coefficients of the method.

We are interested only in TSRK methods that have the strong stability preserving property. As we will prove in Theorem 1, this greatly reduces the set of methods relevant to our study. Except in special cases, the method (2.1) cannot be strong stability preserving unless all of 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 of the form 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 of the form $F(y_i^{n-1})$. 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 **Type I** methods:

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

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

Now consider the special case in which the method (2.1) has some stage y_i^n identically equal to u^n . Then including terms proportional to $F(u^n)$ will not prevent the method from being written as a convex combination of forward Euler steps; furthermore, since $y_i^{n-1} = u^{n-1}$, terms of the form $F(u^{n-1})$ can also be included. This leads to what we will call **Type II** methods, which have the form:

$$y_1^n = u^n, \quad (2.3a)$$

$$y_i^n = d_i u^{n-1} + (1 - d_i) u^n + \hat{a}_i \Delta t F(u^{n-1}) + \Delta t \sum_{j=1}^s a_{ij} F(y_j^n), \quad 2 \leq i \leq s, \quad (2.3b)$$

$$u^{n+1} = \theta u^{n-1} + (1 - \theta) u^n + \hat{b}_1 \Delta t F(u^{n-1}) + \Delta t \sum_{j=1}^s b_j F(y_j^n). \quad (2.3c)$$

Here we have assumed that the first stage is the one equal to u^n , which involves no loss of generality. We can refer to the coefficients of Type II methods in the matrix/vector notation of (2.1) except that matrix $\hat{\mathbf{A}}$ reduces to vector $\hat{\mathbf{a}}$ and we have $d_1 = \hat{a}_1 = 0$ and $a_{1j} = 0$ for all $1 \leq j \leq s$.

REMARK 1. *From a theoretical perspective, the distinction between Type I and Type II methods may seem artificial, since the class of all Type I methods is equivalent to the class of all Type II methods. From a practical perspective, however, the distinction is very useful. Transforming a given method from one type to the other generally requires adding a stage. Thus the class of s -stage Type I methods and the class of s -stage Type II methods are distinct (though not disjoint). So it is natural to refer to a method as being of Type I or Type II, depending on which representation uses fewer stages; this convention is used throughout the present work.*

REMARK 2. *Type I methods (2.2) and Type II methods (2.3) are equivalent to the (two-step) methods of Type 4 and Type 5, respectively, considered in [8].*

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

$$w_i^n = \sum_{j=1}^l s_{ij} x_j^n + \Delta t \sum_{j=1}^m t_{ij} F(w_j^n), \quad (1 \leq i \leq m), \quad (2.4a)$$

$$x_j^{n+1} = w_{j_j}^n, \quad (1 \leq j \leq l). \quad (2.4b)$$

The terms x_j^n are the l input values available from previous steps, while the w_j^n includes both the output values and intermediate stages used to compute them. Equation (2.4b) indicates which of these values are used as inputs in the next step.

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

$$\mathbf{S}\mathbf{e} = \mathbf{e}, \quad (2.5)$$

where \mathbf{e} is a 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, multi-step methods, and multi-step Runge–Kutta methods are all subclasses of general linear methods, and can be written in the form (2.4). For example, an s -stage Runge–Kutta method with Butcher coefficients \mathbf{A} and \mathbf{b} can be written in form (2.4) by taking $l = 1, m = s + 1, J = \{m\}$, and

$$\mathbf{S} = (1, 1, \dots, 1)^\top, \quad \mathbf{T} = \begin{pmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{b}^\top & 0 \end{pmatrix}.$$

Linear multistep methods

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

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 $J = \{2, \dots, l + 1\}$.

General 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

$$\mathbf{x}^n = (u^{n-1}, y_1^{n-1}, \dots, y_s^{n-1}, u^n)^\top, \quad (2.6a)$$

$$\mathbf{w}^n = (y_1^{n-1}, y_2^{n-1}, \dots, y_s^{n-1}, u^n, y_1^n, y_2^n, \dots, y_s^n, u^{n+1})^\top, \quad (2.6b)$$

$$\mathbf{S} = \begin{pmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} \\ 0 & \mathbf{0} & 1 \\ \mathbf{d} & \mathbf{0} & \mathbf{e} - \mathbf{d} \\ \theta & \mathbf{0} & 1 - \theta \end{pmatrix}, \quad \mathbf{T} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & 0 \\ \hat{\mathbf{A}} & \mathbf{0} & \mathbf{A} & \mathbf{0} \\ \hat{\mathbf{b}}^\top & \mathbf{0} & \mathbf{b}^\top & 0 \end{pmatrix}. \quad (2.6c)$$

Type I methods (2.2) can be written in a simpler form with $m = s + 2, l = 2, J = \{1, s + 2\}$, and

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

Type II methods (2.3) can also be written in a simpler form with $m = s + 2, l = 2, J = \{2, s + 2\}$:

$$\mathbf{x}^n = (u^{n-1}, u^n)^\top, \quad \mathbf{w}^n = (u^{n-1}, u^n, y_2^n, \dots, y_s^n, u^{n+1})^\top, \\ \mathbf{S} = \begin{pmatrix} 1 & 0 \\ \mathbf{d} & \mathbf{e} - \mathbf{d} \\ \theta & 1 - \theta \end{pmatrix}, \quad \mathbf{T} = \begin{pmatrix} 0 & \mathbf{0} & 0 \\ \hat{\mathbf{a}} & \mathbf{A} & \mathbf{0} \\ \hat{b}_1 & \mathbf{b}^\top & 0 \end{pmatrix}.$$

2.2. The SSP Coefficient for General Linear Methods. In order to analyze the SSP property of a general linear method (2.4), we first define the vector $\mathbf{f} = [F(w_1), F(w_2), \dots, F(w_m)]^T$, so that (2.4a) can be written compactly as

$$\mathbf{w} = \mathbf{S}\mathbf{x} + \Delta t \mathbf{T}\mathbf{f}. \quad (2.7)$$

Adding $r\mathbf{T}\mathbf{w}$ to both sides of (2.7) 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,

$$\begin{aligned} \mathbf{w} &= (\mathbf{I} + r\mathbf{T})^{-1}\mathbf{S}\mathbf{x} + r(\mathbf{I} + r\mathbf{T})^{-1}\mathbf{T}\left(\mathbf{w} + \frac{\Delta t}{r}\mathbf{f}\right) \\ &= \mathbf{R}\mathbf{x} + \mathbf{P}\left(\mathbf{w} + \frac{\Delta t}{r}\mathbf{f}\right), \end{aligned} \quad (2.8)$$

where we have defined

$$\mathbf{P} = r(\mathbf{I} + r\mathbf{T})^{-1}\mathbf{T}, \quad \mathbf{R} = (\mathbf{I} + r\mathbf{T})^{-1}\mathbf{S} = (\mathbf{I} - \mathbf{P})\mathbf{S}. \quad (2.9)$$

Observe that, by the consistency condition (2.5), 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 $w_j + (\Delta t/r)F(w_j)$. In other words, this method is a convex combination of forward Euler steps. Hence any strong stability property of the forward Euler method is preserved by the method (2.7) under the time step restriction given by $\Delta t \leq \mathcal{C}(\mathbf{S}, \mathbf{T})\Delta t_{\text{FE}}$ where $\mathcal{C}(\mathbf{S}, \mathbf{T})$ is defined as

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

where \mathbf{P} and \mathbf{R} are defined in (2.9). By the foregoing observation, it is clear that the SSP coefficient of method (2.8) 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. A Runge–Kutta method is said to be *reducible* if there exists a method with fewer stages that always produces the same output. One kind of reducibility is known as *HS-reducibility*; a Runge–Kutta method is HS-reducible if two of its stages are identically equal. This definition is extended in a natural way to general linear methods in [41, Theorem 3.1]; hereafter we refer to the reducibility concept defined there also as *HS-reducibility*.

LEMMA 1. ([41, Theorem 3.1]) *Let \mathbf{S}, \mathbf{T} be an HS-irreducible representation of a general linear method. Then 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. ([41, Theorem 2.2(i)]) $\mathcal{C}(\mathbf{S}, \mathbf{T}) > 0$ if and only if all of the following hold:

$$\mathbf{S} \geq 0, \quad (2.10a)$$

$$\mathbf{T} \geq 0, \quad (2.10b)$$

$$\text{Inc}(\mathbf{TS}) \leq \text{Inc}(\mathbf{S}), \quad (2.10c)$$

$$\text{Inc}(\mathbf{T}^2) \leq \text{Inc}(\mathbf{T}). \quad (2.10d)$$

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

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

To apply Lemma 1, it is necessary to write a TSRK method in HS-irreducible form. A trivial type of HS-reducibility is the case where two stages y_i^n, y_j^n are identically equal; i.e., where the following condition holds for some $i \neq j$:

$$d_i = d_j, \quad \text{rows } i, j \text{ of } \mathbf{A} \text{ are identical, and rows } i, j \text{ of } \hat{\mathbf{A}} \text{ are identical} \quad (2.11)$$

This type of reducibility can be dealt with by simply combining the two stages; hence in the following theorem we assume any such stages have been eliminated already. Combining Lemma 1 and Lemma 2, we find that all SSP TSRK methods can be represented as Type I and Type II methods, introduced in Section 1.

THEOREM 1. Let \mathbf{S}, \mathbf{T} be the coefficients of a s -stage TSRK method (2.1) in the form (2.6) with positive SSP coefficient $\mathcal{C} > 0$ such that (2.11) does not hold for any $i \neq j$. Then the method can be written as an s -stage HS-irreducible method of Type I or Type II.

Proof. Consider a method satisfying the stated assumptions. Examination of \mathbf{S}, \mathbf{T} reveals that the method is HS-irreducible in form (2.6) if there is no y_j identically equal to u^n . In this case, we can apply Lemma 1 to obtain that $\mathcal{C}(\mathbf{S}, \mathbf{T}) > 0$. Then condition (2.10c) of Lemma 2 implies that $\hat{\mathbf{A}} = \hat{\mathbf{b}} = \mathbf{0}$. Under this restriction, methods of the form (2.1) simplify to Type I methods (2.2).

Now consider the case in which $y_j = u^n$ for some j . If necessary, reorder the stages so that $y_1^n = u^n$. Then rows $s+1$ and $s+2$ of $[\mathbf{S} \ \mathbf{T}]$ in the representation (2.6) are equal. Thus we can rewrite the method in form (noting that also $y_1^{n-1} = u^{n-1}$) as follows: Set $m = 2s+1, l = s+1, J = \{s, s+1, s+2, \dots, 2s+2\}$, and

$$\mathbf{x}^n = (u^{n-1}, y_2^{n-1}, \dots, y_s^{n-1}, u^n)^T, \quad (2.12a)$$

$$\mathbf{w}^n = (u^{n-1}, y_2^{n-1}, y_3^{n-1}, \dots, y_s^{n-1}, y_1^n, y_2^n, \dots, y_s^n, u^{n+1})^T, \quad (2.12b)$$

$$\mathbf{S} = \begin{pmatrix} 1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{d} & \mathbf{0} & \mathbf{e} - \mathbf{d} \\ \theta & \mathbf{0} & 1 - \theta \end{pmatrix}, \quad \mathbf{T} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 0 & \mathbf{0} & 0 \\ \hat{\mathbf{A}}_1 & \hat{\mathbf{A}}_{2:s} & \mathbf{A} & \mathbf{0} \\ \hat{b}_1 & \hat{\mathbf{b}}_{2:s}^T & \mathbf{b}^T & 0 \end{pmatrix}. \quad (2.12c)$$

Here $\hat{\mathbf{A}}_1$ and \hat{b}_1 represent the first column and first element of $\hat{\mathbf{A}}$ and $\hat{\mathbf{b}}$, respectively, while $\hat{\mathbf{A}}_{2:s}$ and $\hat{\mathbf{b}}_{2:s}^T$ represent the remaining columns and remaining entries. Since (2.12) is HS-irreducible, we can apply Lemma 1. Then we have that $\mathcal{C}(\mathbf{S}, \mathbf{T}) > 0$, so

that Lemma 2 applies. Applying condition (2.10c) of Lemma 2 to the representation (2.12), we find that $\hat{\mathbf{A}}_{2:s}$ and $\hat{\mathbf{b}}_{2:s}^T$ must vanish, but $\hat{\mathbf{A}}_1$ and \hat{b}_1 may be non-zero. The resulting methods are HS-irreducible Type II methods (2.3). \square

REMARK 3. *One could formulate a version of Theorem 1 with the hypothesis that the method in form (2.1) is HS-irreducible. This would lead only to the class of Type I methods, and in the resulting formalism the number of stages for Type II methods would be artificially increased by one (see Remark 1). The advantage of explicitly considering the case $y_j^n = u^n$ is that we obtain Type II methods with the number of stages that accurately represents their cost.*

We now introduce a compact, unified notation for Type I and Type II methods. This form is convenient for expressing the order conditions and restrictions on the coefficients. First we rewrite an s -stage Type II method (2.3) by including u^{n-1} as one of the stages:

$$\begin{aligned} y_0^n &= u^{n-1}, \\ y_1^n &= u^n, \\ y_i^n &= d_i u^{n-1} + (1 - d_i)u^n + \Delta t \sum_{j=0}^s a_{ij} F(y_j^n), \quad 2 \leq i \leq s, \\ u^{n+1} &= \theta u^{n-1} + (1 - \theta)u^n + \Delta t \sum_{j=0}^s b_j F(y_j^n). \end{aligned}$$

Then both Type I and Type II methods can be written in the compact form

$$\mathbf{y}^n = \bar{\mathbf{d}}u^{n-1} + (\mathbf{1} - \bar{\mathbf{d}})u^n + \Delta t \bar{\mathbf{A}}\mathbf{f}^n, \quad (2.13a)$$

$$u^{n+1} = \theta u^{n-1} + (1 - \theta)u^n + \Delta t \bar{\mathbf{b}}^T \mathbf{f}^n, \quad (2.13b)$$

where, for Type I methods the coefficients with bars are equal to the corresponding coefficients without bars in (2.2) and

$$\mathbf{y}^n = [y_1^n, \dots, y_s^n]^T, \quad \mathbf{f}^n = [F(y_1^n), \dots, F(y_s^n)]^T.$$

Meanwhile, for Type II methods

$$\begin{aligned} \mathbf{y}^n &= [u^{n-1}, u^n, y_2^n, \dots, y_s^n]^T, \quad \mathbf{f}^n = [F(u^{n-1}), F(u^n), F(y_2^n), \dots, F(y_s^n)]^T, \\ \bar{\mathbf{d}} &= [1, 0, d_2, \dots, d_s]^T, \quad \bar{\mathbf{b}} = [\hat{b}_1 \quad \mathbf{b}^T]^T, \quad \bar{\mathbf{A}} = \begin{pmatrix} 0 & \mathbf{0} \\ \hat{\mathbf{a}} & \mathbf{A} \end{pmatrix}, \end{aligned}$$

where $d_j, \mathbf{b}, \mathbf{A}, \hat{b}_1, \hat{\mathbf{a}}$ refer to the coefficients in (2.3).

It is known that irreducible strong stability preserving Runge–Kutta methods have positive stage coefficients, $a_{ij} \geq 0$ 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 HS-reducible nor DJ-reducible, we say it is *irreducible*.

THEOREM 2. *The coefficients of an HS-irreducible TSRK method of Type I (2.2) or Type II (2.3) with positive SSP coefficient satisfy the following bounds:*

$$\bar{\mathbf{A}} \geq 0, \bar{\mathbf{b}} \geq 0, 0 \leq \bar{\mathbf{d}} \leq 1, \text{ and } 0 \leq \theta \leq 1. \quad (2.14)$$

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

$$\bar{\mathbf{b}} > 0. \quad (2.15)$$

All of these inequalities should be interpreted component-wise.

Proof. Application of Lemma 1 implies that $\mathcal{C}(\mathbf{S}, \mathbf{T}) > 0$. Therefore Lemma 2 applies. The first result (2.14) then follows from conditions (2.10a) and (2.10b) of Lemma 2. To prove the second part, observe that condition (2.10d) of Lemma 2 means that if $b_j = 0$ for some j then

$$\sum_i b_i a_{ij} = 0. \quad (2.16)$$

Since \mathbf{A}, \mathbf{b} are non-negative, (2.16) implies that either b_i or a_{ij} is zero for each value of i . Now partition the set $\mathcal{S} = \{1, 2, \dots, s\}$ 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 $\mathcal{S}_2 = \emptyset$. \square

3. Order Conditions and a Barrier. Order conditions for TSRK methods up to order 6 have previously been derived in [25]. However, two of the sixth-order conditions therein appear to contain errors (they do not make sense dimensionally). 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 TSRK methods of Types I and II. 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 a SSP TSRK scheme, we have $0 \leq \theta \leq 1$ (by Theorem 2) 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]).

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

$$\begin{aligned} \tilde{\mathbf{y}}^n &= [\tilde{u}(t_n + c_1 \Delta t), \dots, \tilde{u}(t_n + c_s \Delta t)] \\ \tilde{\mathbf{f}}^n &= [F(\tilde{u}(t_n + c_1 \Delta t)), \dots, F(\tilde{u}(t_n + c_s \Delta t))], \end{aligned}$$

where \mathbf{c} identifies the abscissae of the TSRK scheme. These represent the correct stage values and the corresponding correct values of F . Then the *truncation error* τ^n and *stage truncation errors* $\boldsymbol{\tau}^n$ are implicitly defined by

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

$$\tilde{u}(t_{n+1}) = \theta \tilde{u}^{n-1} + (1 - \theta) \tilde{u}^n + \Delta t \bar{\mathbf{b}}^T \tilde{\mathbf{f}}^n + \Delta t \tau^n. \quad (3.1b)$$

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

$$\begin{aligned}\tilde{u}(t_n + c_i \Delta t) &= \sum_{k=0}^{\infty} \frac{1}{k!} \Delta t^k c_i^k \tilde{u}^{(k)}(t_n), \\ F(\tilde{u}(t_n + c_i \Delta t)) &= \tilde{u}'(t_n + c_i \Delta t) = \sum_{k=1}^{\infty} \frac{1}{(k-1)!} \Delta t^{k-1} c_i^{k-1} \tilde{u}^{(k)}(t_n), \\ \tilde{u}(t_{n-1}) &= \sum_{k=0}^{\infty} \frac{(-\Delta t)^k}{k!} \tilde{u}^{(k)}(t_n).\end{aligned}$$

Substitution gives

$$\boldsymbol{\tau}^n = \sum_{k=1}^{\infty} \boldsymbol{\tau}_k \Delta t^{k-1} \tilde{u}^{(k)}(t_n), \quad (3.2a)$$

$$\tau^n = \sum_{k=1}^{\infty} \tau_k \Delta t^{k-1} \tilde{u}^{(k)}(t_n), \quad (3.2b)$$

where

$$\begin{aligned}\boldsymbol{\tau}_k &= \frac{1}{k!} (\mathbf{c}^k - (-1)^k \bar{\mathbf{d}}) - \frac{1}{(k-1)!} \bar{\mathbf{A}} \mathbf{c}^{k-1}, \\ \tau_k &= \frac{1}{k!} (1 - (-1)^k \theta) - \frac{1}{(k-1)!} \bar{\mathbf{b}}^T \mathbf{c}^{k-1}.\end{aligned}$$

Subtracting (3.1) from (2.13) gives

$$\boldsymbol{\epsilon}^n = \bar{\mathbf{d}} \boldsymbol{\epsilon}^{n-1} + (\mathbf{e} - \bar{\mathbf{d}}) \boldsymbol{\epsilon}^n + \Delta t \bar{\mathbf{A}} \boldsymbol{\delta}^n - \Delta t \boldsymbol{\tau}^n, \quad (3.3a)$$

$$\epsilon^{n+1} = \theta \epsilon^{n-1} + (1 - \theta) \epsilon^n + \Delta t \bar{\mathbf{b}}^T \boldsymbol{\delta}^n - \Delta t \tau^n, \quad (3.3b)$$

where $\boldsymbol{\epsilon}^{n+1} = \mathbf{u}^{n+1} - \tilde{\mathbf{u}}(t_{n+1})$ 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

$$\boldsymbol{\delta}^n = \sum_{k=0}^{p-1} \boldsymbol{\delta}_k^n \Delta t^k + \mathcal{O}(\Delta t^p), \quad (3.4)$$

then substituting the expansions (3.4) and (3.2) into the global error formula (3.3) yields

$$\boldsymbol{\epsilon}^n = \bar{\mathbf{d}} \boldsymbol{\epsilon}^{n-1} + (\mathbf{e} - \bar{\mathbf{d}}) \boldsymbol{\epsilon}^n + \sum_{k=0}^{p-1} \bar{\mathbf{A}} \boldsymbol{\delta}_k^n \Delta t^{k+1} - \sum_{k=1}^p \boldsymbol{\tau}_k \tilde{u}^{(k)}(t_n) \Delta t^k + \mathcal{O}(\Delta t^{p+1}), \quad (3.5a)$$

$$\epsilon^{n+1} = \theta \epsilon^{n-1} + (1 - \theta) \epsilon^n + \sum_{k=0}^{p-1} \bar{\mathbf{b}}^T \boldsymbol{\delta}_k^n \Delta t^{k+1} - \sum_{k=1}^p \tau_k \tilde{u}^{(k)}(t_n) \Delta t^k + \mathcal{O}(\Delta t^{p+1}). \quad (3.5b)$$

Hence we find the method is consistent of order p if

$$\bar{\mathbf{b}}^T \boldsymbol{\delta}_k^n = 0 \quad (0 \leq k \leq p-1) \quad \text{and} \quad \tau_k = 0 \quad (1 \leq k \leq p). \quad (3.6)$$

It remains to determine the vectors δ_k^n in the expansion (3.4). In fact, we can relate these recursively to the ϵ_k . First we define

$$\begin{aligned}\mathbf{t}_n &= t_n \mathbf{e} + \mathbf{c} \Delta t, \\ \mathbf{F}(\mathbf{y}, \mathbf{t}) &= [F(y_1(t_1)), \dots, F(y_s(t_s))]^T.\end{aligned}$$

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

$$\begin{aligned}\mathbf{F}^{(j)}(\mathbf{y}, \mathbf{t}) &= [F^{(j)}(y_1(t_1)), \dots, F^{(j)}(y_s(t_s))]^T, \\ \mathbf{g}_j(\mathbf{t}) &= [F^{(j)}(y(t_1)), \dots, F^{(j)}(y(t_s))]^T,\end{aligned}$$

and the dot product denotes component-wise 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 $\mathbf{C} = \text{diag}(\mathbf{c})$, we finally obtain the desired expansion:

$$\boldsymbol{\delta}^n = \sum_{j=1}^{\infty} \sum_{l=0}^{\infty} \frac{\Delta t^l}{j! l!} \mathbf{C}^l (\boldsymbol{\epsilon}^n)^j \cdot \mathbf{g}_j^{(l)}(t_n). \quad (3.7)$$

To determine the coefficients δ_k , we alternate recursively between (3.7) and (3.5a). Typically, the abscissae \mathbf{c} are chosen as $\mathbf{c} = \bar{\mathbf{A}} \mathbf{e}$ so that $\boldsymbol{\tau}_1 = 0$. With these choices, we collect the terms relevant for up to fifth-order accuracy:

Terms appearing in δ_1 :	\emptyset
Terms appearing in ϵ_2 :	$\boldsymbol{\tau}_2$
Terms appearing in δ_2 :	$\boldsymbol{\tau}_2$
Terms appearing in ϵ_3 :	$\bar{\mathbf{A}} \boldsymbol{\tau}_2, \boldsymbol{\tau}_3$
Terms appearing in δ_3 :	$\mathbf{C} \boldsymbol{\tau}_2, \bar{\mathbf{A}} \boldsymbol{\tau}_2, \boldsymbol{\tau}_3$
Terms appearing in ϵ_4 :	$\bar{\mathbf{A}} \mathbf{C} \boldsymbol{\tau}_2, \bar{\mathbf{A}}^2 \boldsymbol{\tau}_2, \bar{\mathbf{A}} \boldsymbol{\tau}_3, \boldsymbol{\tau}_4$
Terms appearing in δ_4 :	$\bar{\mathbf{A}} \mathbf{C} \boldsymbol{\tau}_2, \bar{\mathbf{A}}^2 \boldsymbol{\tau}_2, \bar{\mathbf{A}} \boldsymbol{\tau}_3, \boldsymbol{\tau}_4, \mathbf{C} \bar{\mathbf{A}} \boldsymbol{\tau}_2, \mathbf{C} \boldsymbol{\tau}_3, \mathbf{C}^2 \boldsymbol{\tau}_2, \boldsymbol{\tau}_2^2$

The order conditions are then given by (3.6). 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 τ_2^2 in δ_4 leads to the order condition $\mathbf{b}^T \tau_2^2 = 0$. For SSP methods, since $\mathbf{b} > 0$ (by Theorem 2), this implies that $\tau_2^2 = 0$, i.e. fifth-order SSP TSRK methods must have stage order of at least 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 will include $\mathbf{b}^T \tau_3^2 = 0$, which leads (together with the non-negativity of \mathbf{A}) to the result that implicit SSP RK methods have order at most six. In general, the conditions for order $2k + 1$ will include the condition $\mathbf{b}^T \tau_k^2 = 0$. 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 3. *Any TSRK method (2.1) of order p with positive SSP coefficient has stage order at least $\lfloor \frac{p-1}{2} \rfloor$.*

Proof. Without loss of generality, we consider only irreducible methods; thus we can apply Theorems 1 and 2. Following the procedure outlined above, we find that for order p , the coefficients must satisfy

$$\mathbf{b}^T \tau_k^2 = 0, \quad k = 1, 2, \dots, \left\lfloor \frac{p-1}{2} \right\rfloor.$$

Since $\mathbf{b} > 0$ by Theorem 2, this implies that

$$\tau_k^2 = 0, \quad k = 1, 2, \dots, \left\lfloor \frac{p-1}{2} \right\rfloor.$$

□

Application of Theorem 3 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 3 leads to an upper bound on the order of explicit SSP TSRKs.

THEOREM 4. *The order of an explicit SSP TSRK method is at most eight. Furthermore, if the method has order greater than six, it is of Type II.*

Proof. Without loss of generality, we consider only irreducible methods; thus we can apply Theorems 2 and 3.

To prove the second part, consider an explicit irreducible TSRK method with order greater than six. By Theorem 3, 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 $y_2 = u^{n-1}$, so the method is of type II. 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 (2.13) of order nine. By Theorem 3, 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 2, this implies that the method is not SSP. □

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 2 and 3. We list some examples here, but omit the proofs since these results are not essential to our present purpose.

1. Any SSP TSRK method (implicit or explicit) of order greater than four must have a stage equal to u^{n-1} or u^n .
2. The abscissae c_i of any SSP TSRK method of order greater than four must each be non-negative or equal to -1 .
3. (Implicit) SSP TSRK methods with $p > 8$ must be of Type II.

4. Optimal SSP Two-step Runge–Kutta methods. Our objective in this section is to find SSP 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 TSRK methods with order at least p with no more than s stages.

The methods presented were found via numerical search using MATLAB’s Optimization and Global Optimization toolboxes. We searched over Type I and Type II methods; the optimal methods found are of Type II in every case. For the methods of seventh and eighth order, this is known *a priori* from Theorem 4. Even for the lower order methods, this is not surprising, since explicit s -stage Type II methods (2.3) have an additional $s - 1$ degrees of freedom compared to explicit s -stage Type I methods (2.2). 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. Formulating the Optimization Problem. The optimization problem is formulated using the theory of Section 2:

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

where the inequalities are understood component-wise 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.

In comparing 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*

$$\mathcal{C}_{\text{eff}}(\mathbf{S}, \mathbf{T}) = \frac{\mathcal{C}(\mathbf{S}, \mathbf{T})}{s}.$$

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.

REMARK 4. *By virtue of Theorem 1, optimal SSP methods found in the classes of Type I and Type II TSRK methods are in fact optimal over the larger class of methods (2.1). Also, because they do not use intermediate stages from previous timesteps, special conditions on the starting method (important for methods of the form (2.1) [16, 45, 44]) are unnecessary. Instead, the method can be started with any SSP Runge–Kutta method of the appropriate order.*

REMARK 5. *The optimal Type II methods found here could be rewritten as Type I methods by adding a stage (see Remark 1). However, this would reduce their effective SSP coefficient and render them (apparently) less efficient than some other (Type I) methods. This indicates once more the importance of explicitly accounting for the practical difference between Type I and Type II methods.*

4.2. Low-storage implementation of Type II SSP TSRKs. The form (2.8), with $r = \mathcal{C}(\mathbf{S}, \mathbf{T})$, typically yields very sparse coefficient matrices for optimal Type II SSP TSRK methods. This form is useful for a low-storage implementation. Written out explicitly, this form is:

$$y_i^n = \tilde{d}_i u^{n-1} + \left(1 - \tilde{d}_i - \sum_{j=0}^s q_{ij}\right) u^n + \sum_{j=0}^s q_{ij} \left(y_j^n + \frac{\Delta t}{r} F(y_j^n)\right), \quad (1 \leq i \leq s), \quad (4.1a)$$

$$u^{n+1} = \tilde{\theta} u^{n-1} + \left(1 - \tilde{\theta} - \sum_{j=0}^s \eta_j\right) u^n + \sum_{j=0}^s \eta_j \left(y_j^n + \frac{\Delta t}{r} F(y_j^n)\right), \quad (4.1b)$$

where the coefficients are given by (using the relations (2.9)):

$$\begin{aligned} \mathbf{Q} &= r\bar{\mathbf{A}}(\mathbf{I} + r\bar{\mathbf{A}})^{-1}, & \boldsymbol{\eta} &= r\bar{\mathbf{b}}^T(\mathbf{I} + r\bar{\mathbf{A}})^{-1}, \\ \tilde{\mathbf{d}} &= \bar{\mathbf{d}} - \mathbf{Q}\bar{\mathbf{d}}, & \tilde{\theta} &= \theta - \boldsymbol{\eta}^T \bar{\mathbf{d}}. \end{aligned}$$

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.2, A.3, A.4, and A.5, we give the coefficients in the low-storage form (4.1) for some numerically optimal methods.

4.3. Optimal 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 $1 \leq s \leq 4$, $2 \leq p \leq 4$) 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 is in fact both a Type I and Type II method, and was found in numerical searches over methods of both forms. It

TABLE 4.1

Effective SSP coefficients C_{eff} of optimal explicit 2-step Runge–Kutta methods of order 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

has SSP coefficient $\mathcal{C} = \sqrt{s(s-1)}$ and nonzero coefficients

$$\begin{aligned}
 q_{i,i-1} &= 1, & (2 \leq i \leq s), \\
 q_{s+1,s} &= 2(\mathcal{C} - s + 1), \\
 \tilde{\mathbf{d}} &= \mathbf{0}, \\
 \tilde{\theta} &= 2(s - \mathcal{C}) - 1.
 \end{aligned}$$

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

4.4. Optimal 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 an 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 eight, it is not possible to achieve a larger effective SSP coefficient than the optimal 8-stage method, even allowing as many as twelve stages. This appears to be accurate, and not simply due to failure of the numerical optimizer, since in the nine-stage case the optimization scheme recovers the apparently optimal method in less than one 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

TABLE 4.2

Effective SSP coefficients C_{eff} of optimal explicit two-step Runge–Kutta methods of order 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

of the same order of accuracy. Furthermore, our twelve-stage TSRK methods have sparse coefficient matrices and can be implemented in the low-storage form (4.1). Specifically, the fifth- through eighth-order methods of twelve 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 sixth through eighth order, the twelve-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 eight-stage method has a larger effective SSP coefficient than the twelve stage method, so the eight stage method seems best in terms of efficiency. However the eight stage method requires more storage registers (6) than the twelve stage method (5). So while the eight stage method might be preferred for efficiency, the twelve stage method is preferred for low storage considerations.

5. Numerical Experiments.

5.1. Start-up procedure. As mentioned in Section 4.1, TSRK are not self-starting and thus require startup procedures, and while in general this is somewhat complicated, for our class of methods it is straightforward. We only require that the starting procedure be of sufficient accuracy and that it also be strong stability preserving.

Figure 5.1 demonstrates one possible start-up procedure that we employed in our convergence studies and our other numerical tests to 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. Specifically, this can be achieved for an TSRK method of order $p = 5, 6, 7$ or 8 by taking

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

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 .

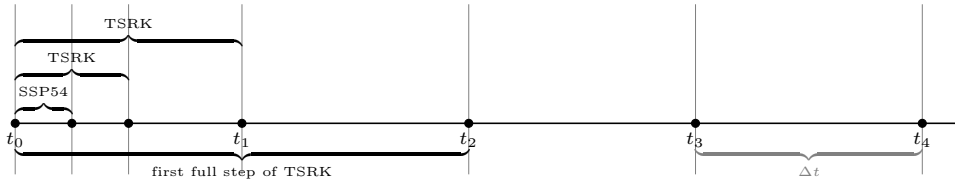


FIG. 5.1. One possible startup procedure for SSP TSRK schemes. The first step from t_0 to t_1 is subdivided into substeps (here there are three substeps of sizes $\frac{h}{4}$, $\frac{h}{4}$, and $\frac{h}{2}$). An SSP Runge–Kutta scheme is used for the first substep. Subsequent substeps are taken with 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.

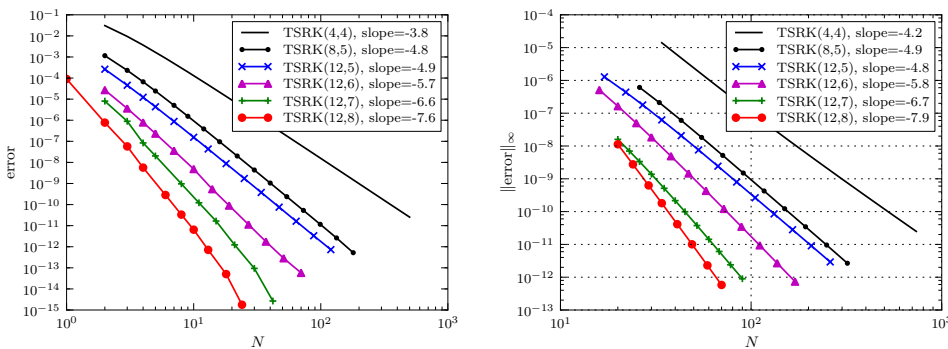


FIG. 5.2. Convergence results for some TSRK schemes on the Dahlquist test problem (left) and van der Pol problem (right). The slopes of the lines confirm the design orders of the TSRK methods.

5.2. 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

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

where we use $\epsilon = 0.01$ with corresponding initial conditions $u^0 = [2; -0.6654321]$ and solve until $t_f = \frac{1}{2}$. The starting procedure used SSPRK(10,4) with constant $A = 1$ in (5.1). Error in the maximum norm is estimated 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.

5.3. High-order WENO. Weighted essentially non-oscillatory schemes (WENO) [18, 17, 26] are finite difference or finite volume schemes that use linear combination of lower order fluxes to obtain a higher order approximations, while ensuring non-oscillatory solutions. This is accomplished by using adaptive stencils which approach

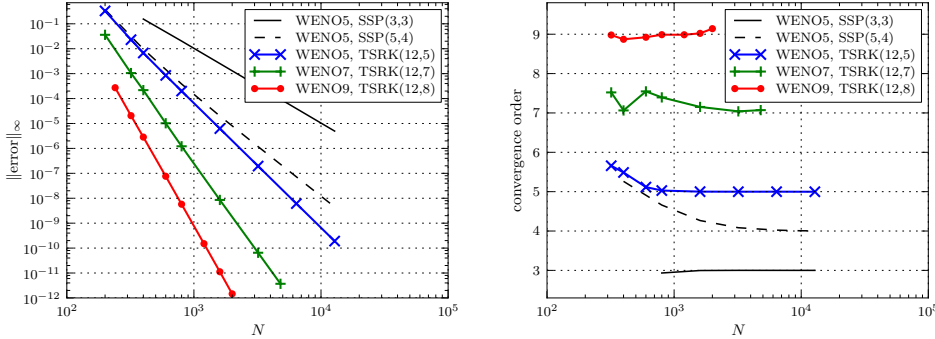


FIG. 5.3. Convergence results for 2D advection using r th-order WENO discretizations and the TSRK integrators (c.f., [13, Fig. 15]). Maximum error versus number of spatial grid points in each direction (left). Observed orders of accuracy calculated from these errors (right). Computed using the same parameters 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 as described in Section 5.1 using the SSPRK(5,4) scheme for the initial substep.

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 of up to 17th-order were constructed and tested numerically. 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 2D 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 5, 7 and 8 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

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

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 $\Delta t \leq \Delta t_{\text{FE}} = 0.0025$ [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 timestep $\Delta t = \sigma \Delta t_{\text{FE}}$. Table 5.1 shows the maximal TVD time-step sizes, expressed as $\Delta t =$

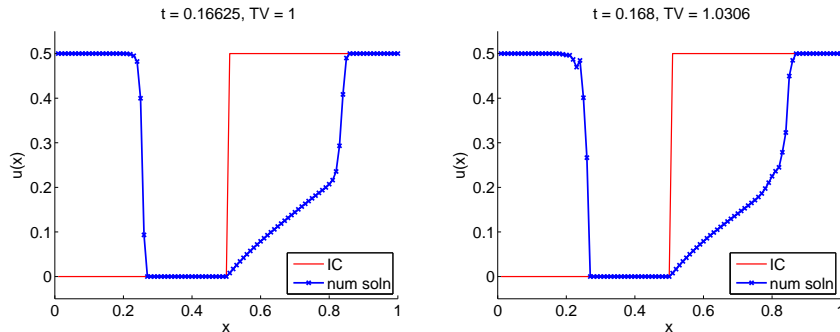


FIG. 5.4. Two numerical solutions of the Buckley–Leverett test problem. Left: time-step satisfies the SSP time-step restriction (TSRK(8,5) using $\Delta t = 3.5\Delta t_{\text{FE}}$). Right: time-step does not satisfy the restriction ($\Delta t = 5.6\Delta t_{\text{FE}}$) and visible oscillations have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

TABLE 5.1

SSP coefficients versus largest time steps exhibiting the TVD property ($\Delta t = \sigma_{\text{BL}}\Delta t_{\text{FE}}$) on the Buckley–Leverett example, for some of the SSP TSRK(s,p) schemes. The effective SSP coefficient \mathcal{C}_{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}_{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)	0.94155	0.0785	4.42	0.368

$\sigma_{\text{BL}}\Delta t_{\text{FE}}$, for the Buckley–Leverett test problem. The results show that the SSP 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 have analyzed the strong stability preserving property of two-step Runge–Kutta (TSRK) methods. We find that SSP TSRK methods have a relatively simple form and that explicit methods are subject to a maximal order of eight. We have presented numerically optimal 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 multi-step 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 up, as they do not use stage values from previous steps.

We show that high-order SSP two-step Runge–Kutta 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 multi-step Runge–Kutta methods.

Acknowledgment. The authors are grateful to an anonymous referee, whose careful reading and detailed comments improved several technical details of the paper.

Appendix A. Coefficients of Numerically Optimal Methods.

TABLE A.1

Coefficients of the optimal explicit 8-stage 5th-order SSP TSRK method (Type II)

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

TABLE A.2

Coefficients of the optimal explicit 12-stage 5th-order SSP TSRK method (Type II)

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

TABLE A.3

Coefficients of the optimal explicit 12-stage 6th-order SSP TSRK method (Type II)

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

REFERENCES

TABLE A.4
Coefficients of the optimal explicit 12-stage 7th-order SSP TSRK method (Type II)

$\tilde{\theta} = 1.040248277612947e - 04$	$q_{2,1} = 0.849449065363225$	$q_{4,3} = 0.166320497215237$
$\tilde{d}_0 = 1.000000000000000$	$q_{3,1} = 0.120943274105256$	$q_{10,3} = 0.032851385162085$
$\tilde{d}_2 = 0.003229110378701$	$q_{4,1} = 0.368587879161520$	$q_{5,4} = 0.343703780759466$
$\tilde{d}_4 = 0.006337974349692$	$q_{5,1} = 0.222052624372191$	$q_{6,5} = 0.519758489994316$
$\tilde{d}_5 = 0.002497954201566$	$q_{6,1} = 0.137403913798966$	$q_{7,6} = 0.598177722195673$
$\tilde{d}_8 = 0.017328228771149$	$q_{7,1} = 0.146278214690851$	$q_{8,7} = 0.488244475584515$
$\tilde{d}_{12} = 0.000520256250682$	$q_{8,1} = 0.444640119039330$	$q_{10,7} = 0.356898323452469$
$\eta_0 = 0.000515717568412$	$q_{9,1} = 0.143808624107155$	$q_{11,7} = 0.508453150788232$
$\eta_1 = 0.040472655980253$	$q_{10,1} = 0.102844296820036$	$q_{12,7} = 0.496859299069734$
$\eta_6 = 0.081167924336040$	$q_{11,1} = 0.071911085489036$	$q_{9,8} = 0.704865150213419$
$\eta_7 = 0.238308176460039$	$q_{12,1} = 0.057306282668522$	$q_{10,9} = 0.409241038172241$
$\eta_8 = 0.032690786323542$	$q_{3,2} = 0.433019948758255$	$q_{11,10} = 0.32700595932695$
$\eta_{12} = 0.547467490509490$	$q_{7,2} = 0.014863996841828$	$q_{12,11} = 0.364647377606582$
$q_{2,0} = 0.147321824258074$	$q_{9,2} = 0.026942009774408$	

TABLE A.5
Coefficients of the optimal explicit 12-stage 8th-order SSP TSRK method (Type II)

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

- [1] P. ALBRECHT, *A new theoretical approach to Runge–Kutta methods*, SIAM Journal on Numerical Analysis, 24 (1987), pp. 391–406.
- [2] P. ALBRECHT, *The Runge–Kutta theory in a nutshell*, SIAM Journal on Numerical Analysis, 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*, Journal of Computational Physics, 160 (2000), pp. 405–452.
- [4] J. C. BUTCHER AND S. TRACOGNA, *Order conditions for two-step Runge–Kutta methods*, Applied Numerical Mathematics, 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*, Journal of Computational Physics, 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*, Applied Numerical Mathematics, 57 (2007), pp. 1021–1032.
- [8] E. CONSTANTINESCU AND A. SANDU, *Optimal explicit strong-stability-preserving general linear methods*, SIAM Journal of Scientific Computing, 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*, Journal of Computational Physics, 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*, The Astrophysical Journal, 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 Journal of Numerical Analysis, 42 (2004), pp. 1073–1093.
- [12] ———, *An extension and analysis of the Shu–Osher representation of Runge–Kutta methods*, Mathematics of Computation, 249 (2005), pp. 201–219.
- [13] G. GEROLYMOS, D. SÉNÉCHAL, AND I. VALLET, *Very-high-order WENO schemes*, Journal of Computational Physics, 228 (2009), pp. 8481–8524.
- [14] S. GOTTLIEB, C.-W. SHU, AND E. TADMOR, *Strong Stability Preserving High-Order Time Discretization Methods*, SIAM Review, 43 (2001), pp. 89–112.
- [15] E. HAIRER AND G. WANNER, *Solving ordinary differential equations II: Stiff and differential-algebraic problems*, vol. 14 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 1991.
- [16] ———, *Order conditions for general two-step Runge–Kutta methods*, SIAM Journal on Numerical Analysis, 34 (1997), pp. 2087–2089.
- [17] A. HARTEN, B. ENGGUIST, S. OSHER, AND S. R. CHAKRAVARTHY, *Uniformly high-order accurate essentially nonoscillatory schemes. III*, J. Comput. Phys., 71 (1987), pp. 231–303.
- [18] ———, *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*, Journal of Scientific Computing, 21 (2004), pp. 193–223.
- [20] ———, *Representations of Runge–Kutta methods and strong stability preserving methods*, Siam Journal On Numerical Analysis, 43 (2005), pp. 924–948.
- [21] C. HUANG, *Strong stability preserving hybrid methods*, Applied Numerical Mathematics, 59 (2009), pp. 891–904.
- [22] W. HUNSDORFER AND S. J. RUUTH, *On monotonicity and boundedness properties of linear multistep methods*, Mathematics of Computation, 75 (2005), pp. 655–672.
- [23] W. H. HUNSDORFER AND J. G. VERWER, *Numerical solution of time-dependent advection-diffusion-reaction equations*, vol. 33 of Springer Series in Computational Mathematics, Springer, 2003.
- [24] Z. JACKIEWICZ, *General Linear Methods for Ordinary Differential Equations*, Wiley, 2009.
- [25] Z. JACKIEWICZ AND S. TRACOGNA, *A general class of two-step Runge–Kutta methods for ordinary differential equations*, SIAM Journal of Numerical Analysis, 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*, Journal of Computational Physics, 205 (2005), pp. 222–241.
- [28] D. I. KETCHESON, *Highly efficient strong stability preserving Runge–Kutta methods with low-storage implementations*, SIAM Journal on Scientific Computing, 30 (2008), pp. 2113–2136.
- [29] ———, *Computation of optimal monotonicity preserving general linear methods*, Mathematics of Computation, (2009).
- [30] D. I. KETCHESON, *Runge–Kutta methods with minimum storage implementations*, Journal of Computational Physics, 229 (2010), pp. 1763–1773.
- [31] D. I. KETCHESON, C. B. MACDONALD, AND S. GOTTLIEB, *Optimal implicit strong stability preserving Runge–Kutta methods*, Applied Numerical Mathematics, 52 (2009), p. 373.
- [32] J. F. B. M. KRAALJEVANGER, *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*, Journal of Computational Physics, 200 (2004), pp. 267–298.
- [34] H. W. J. LENFERINK, *Contractivity-preserving explicit linear multistep methods*, Numerische Mathematik, 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. doi:10.1007/s10915-007-9180-6.
- [36] D. PENG, B. MERRIMAN, S. OSHER, H. ZHAO, AND M. KANG, *A PDE-based fast local level set method*, Journal of Computational Physics, 155 (1999), pp. 410–438.

- [37] S. J. RUUTH AND W. HUNSDORFER, *High-order linear multistep methods with general monotonicity and boundedness properties*, Journal of Computational Physics, 209 (2005), pp. 226–248.
- [38] S. J. RUUTH AND R. J. SPITERI, *Two barriers on strong-stability-preserving time discretization methods*, Journal of Scientific Computation, 17 (2002), pp. 211–220.
- [39] C.-W. SHU, *Total-variation diminishing time discretizations*, SIAM J. Sci. Stat. Comp., 9 (1988), pp. 1073–1084.
- [40] C.-W. SHU AND S. OSHER, *Efficient implementation of essentially non-oscillatory shock-capturing schemes*, Journal of Computational Physics, 77 (1988), pp. 439–471.
- [41] M. SPIJKER, *Stepsize conditions for general monotonicity in numerical initial value problems*, SIAM Journal on Numerical Analysis, 45 (2008), 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 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*, Journal of Computational and Applied Mathematics, 185 (2006), pp. 292–307.