

STRONG STABILITY PRESERVING INTEGRATING FACTOR TWO-STEP RUNGE–KUTTA METHODS

LEAH ISHERWOOD*, ZACHARY J. GRANT†, AND SIGAL GOTTLIEB*

Abstract. Problems with components that feature significantly different time scales, where the stiff time-step restriction comes from a linear component, implicit-explicit (IMEX) methods alleviate this restriction if the concern is linear stability. However, when nonlinear non-inner-product stability properties are of interest, such as in the evolution of hyperbolic partial differential equations with shocks or sharp gradients, linear inner-product stability is no longer sufficient for convergence, and so strong stability preserving (SSP) methods are often needed. Where the SSP property is needed, IMEX SSP Runge–Kutta (SSP-IMEX) methods have very restrictive time-steps. An alternative to SSP-IMEX schemes is to adopt an integrating factor approach to handle the linear component exactly and step the transformed problem forward using some time-evolution method. The strong stability properties of integrating factor Runge–Kutta methods were established in [15], where it was shown that it is possible to define explicit integrating factor Runge–Kutta methods that preserve strong stability properties satisfied by each of the two components when coupled with forward Euler time-stepping. It was proved that the solution will be SSP if the transformed problem is stepped forward with an explicit SSP Runge–Kutta method that has non-decreasing abscissas. However, explicit SSP Runge–Kutta methods have an order barrier of $p = 4$, and sometimes higher order is desired. In this work we consider explicit SSP two-step Runge–Kutta integrating factor methods to raise the order. We show that strong stability is ensured if the two-step Runge–Kutta method used to evolve the transformed problem is SSP and has non-decreasing abscissas. We find such methods up to eighth order and present their SSP coefficients. Adding a step allows us to break the fourth order barrier on explicit SSP Runge–Kutta methods; furthermore, our explicit SSP two-step Runge–Kutta methods with non-decreasing abscissas typically have larger SSP coefficients than the corresponding one-step methods. A selection of our methods are tested for convergence and demonstrate the design order. We also show, for selected methods, that the SSP time-step predicted by the theory is a lower bound of the allowable time-step for linear and nonlinear problems that satisfy the total variation diminishing (TVD) condition. We compare some of the non-decreasing abscissa SSP two-step Runge–Kutta methods to previously found methods that do not satisfy this criterion on linear and nonlinear TVD test cases to show that this non-decreasing abscissa condition is indeed necessary in practice as well as theory. We also compare these results to our SSP integrating factor Runge–Kutta methods designed in [15].

This paper is dedicated to the memory of Saul Abarbanel. His wisdom, humor, and kindness were a gift to all who knew him.

1. Introduction. The behavior of the numerical solution of a hyperbolic partial differential equation (PDE) of the form

$$(1.1) \quad U_t + f(U)_x = 0,$$

depends on properties of the spatial discretization and of the time discretization. When the solution is smooth, stability is guaranteed by analyzing the L_2 stability properties of the discretization applied to the linear problem. However, when dealing with a non-smooth solution, the numerical solution may contain non-physical oscillations that prevent the approximation from converging uniformly and thus L_2 linear stability is not sufficient to ensure convergence [26]. To ensure that the numerical solution does not form stability-destroying oscillations, we require that the numerical method satisfies nonlinear non-inner-product stability properties such as stability in the maximum norm or in the total variation (TV) semi-norm.

*Mathematics Department, University of Massachusetts Dartmouth, 285 Old Westport Road, North Dartmouth MA 02747.

†Department of Computational and Applied Mathematics, Oak Ridge National Laboratory, Oak Ridge TN 37830.

For nonlinear hyperbolic problems with discontinuous solutions we must, therefore, analyze the nonlinear non-inner-product stability properties of a highly nonlinear complex spatial discretization combined with a high order time discretization. Instead of this difficult task a method-of-lines formulation is generally followed: we develop a spatial discretization that satisfies nonlinear non-inner-product stability properties when coupled with the forward Euler time stepping method. Next, we use a high order strong stability preserving (SSP) time discretization [29, 30, 28, 34, 14, 19, 18, 22, 9, 23] which preserves the properties of the spatial discretization coupled with forward Euler. Explicit strong stability preserving (SSP) Runge–Kutta methods were developed in [29, 30] to preserve the properties of total variation diminishing (TVD) spatial discretizations for hyperbolic conservation laws (1.1) with discontinuous solutions. They have since been developed extensively and have been widely used to preserve different numerical stability properties needed in a variety of application areas. Furthermore, many classes of SSP time-stepping methods have been studied, including SSP explicit and implicit linear multi-step methods [9], Runge–Kutta methods [29, 30, 18, 19], multi-stage multi-step methods [2, 22, 23], and multi-stage multi-derivative methods [3, 12].

1.1. SSP methods. The key to developing SSP time-stepping methods is ensuring that the methods can be re-written as convex combinations of forward Euler steps. To illustrate this concept, we begin with the PDE (1.1) above, and use a spatial discretization of $f(U)_x$ that ensures that when we evolve the resulting semi-discretized system of ordinary differential equations (ODEs)

$$(1.2) \quad u_t = F(u),$$

using the forward Euler method, the strong stability property in the convex functional $\|\cdot\|$ is satisfied

$$(1.3) \quad \|u^{n+1}\| = \|u^n + \Delta t F(u^n)\| \leq \|u^n\|,$$

under a step size restriction

$$(1.4) \quad 0 \leq \Delta t \leq \Delta t_{\text{FE}}.$$

Next, we use a higher order time integrator that can be written as a convex combination of forward Euler steps, so that we ensure that any convex functional strong stability property that is satisfied by the forward Euler method will still be satisfied by the higher order time discretization, perhaps under a different time-step. For example, an s -stage explicit Runge–Kutta method (denoted eSSPRK(s, p) where p is the order) can be written in Shu–Osher form [9]:

$$(1.5) \quad \begin{aligned} u^{(0)} &= u^n, \\ u^{(i)} &= \sum_{j=0}^{i-1} \left(\alpha_{i,j} u^{(j)} + \Delta t \beta_{i,j} F(u^{(j)}) \right), \quad i = 1, \dots, s \\ u^{n+1} &= u^{(s)}. \end{aligned}$$

(Where $\sum_{j=0}^{i-1} \alpha_{i,j} = 1$ is required for consistency). If $\alpha_{i,j}$ and $\beta_{i,j}$ are non-negative, and any $\alpha_{i,j}$ is zero only if its corresponding $\beta_{i,j}$ is zero, then we can rearrange each stage into a convex combination of forward Euler steps:

$$\|u^{(i)}\| = \left\| \sum_{j=0}^{i-1} \left(\alpha_{i,j} u^{(j)} + \Delta t \beta_{i,j} F(u^{(j)}) \right) \right\|_2 = \left\| \sum_{j=0}^{i-1} \alpha_{i,j} \left(u^{(j)} + \Delta t \frac{\beta_{i,j}}{\alpha_{i,j}} F(u^{(j)}) \right) \right\|.$$

Clearly, then, (1.3) tells us that

$$\|u^{(i)}\| \leq \sum_{j=0}^{i-1} \alpha_{i,j} \left\| u^{(j)} + \Delta t \frac{\beta_{i,j}}{\alpha_{i,j}} F(u^{(j)}) \right\| \leq \|u^n\|,$$

while (1.4) imposes the time-step restriction

$$(1.6) \quad \Delta t \leq \min_{i,j} \frac{\alpha_{i,j}}{\beta_{i,j}} \Delta t_{\text{FE}}.$$

(We use the convention that the ratio is considered infinite in the case where a $\beta_{i,j}$ is equal to zero). This convex combination ensures that the internal stages also satisfy the strong stability property under the same time-step restriction; this can be important when pressure, density, or water height, are simulated, because in these cases a negative value at an intermediate or final stage may prevent the simulation from continuing [13]. Clearly, the convex combination condition is a sufficient condition for strong stability preservation for explicit Runge–Kutta methods; in [9, 24, 33] it was shown to be a necessary condition as well.

This shows that when a higher order time discretization method is written as a convex combination of forward Euler steps, it will *preserve* the forward Euler condition (1.3), under a modified time-step restriction $\Delta t \leq \mathcal{C} \Delta t_{\text{FE}}$. When $\mathcal{C} > 0$, the method is called *strong stability preserving* (SSP) with *SSP coefficient* \mathcal{C} [29]. While in the original papers [29, 30], the term $\|\cdot\|$ was the total variation semi-norm, the strong stability preservation property holds for any convex functional $\|\cdot\|$ as long as the spatial discretization satisfies (1.3) in that convex functional for some Δt_{FE} as in (1.4).

Not all methods can be decomposed into convex combinations of forward Euler steps with $\mathcal{C} > 0$. Explicit SSP Runge–Kutta methods cannot exist for order $p > 4$ [24, 28]. Furthermore, the SSP coefficient \mathcal{C} is restricted too, as all explicit s -stage Runge–Kutta methods have a SSP bound $\mathcal{C} \leq s$ [9]. Unlike for smooth problems, where implicit methods (or implicit treatment of stiff terms) can eliminate the time-step restriction needed for linear or nonlinear inner-product norm stability, a strong stability preserving general linear method (GLM) of order greater than one has a finite SSP time-step [32]. In fact, it has been shown [25, 6, 19, 4] that the SSP time-step restrictions for implicit and implicit-explicit (IMEX) SSP Runge–Kutta methods have a restrictive observed bound of $\mathcal{C} \leq 2s$. The limitation on the SSP coefficient of implicit and IMEX SSP Runge–Kutta methods led to the study of SSP integrating factor Runge–Kutta methods in [15].

1.2. Overview of current paper. The explicit SSP integrating factor Runge–Kutta methods and the associated theory developed in our prior work [15], reviewed in Section 2, allow us to alleviate the time-step restriction while preserving the strong stability property. However, because explicit SSP Runge–Kutta methods have an order barrier of $p \leq 4$, we cannot hope to get higher order explicit SSP integrating factor Runge–Kutta methods. We could try to work with an integrating factor multi-step approach, which is SSP as long as the multi-step method is SSP (as we show in Subsection 4.1) but these methods have small SSP coefficients and require many steps for high order. Instead, we follow the process in [22, 23] and examine the SSP properties of integrating factor methods based on explicit SSP two-step Runge–Kutta (eSSP-TSRK) methods such as those in [22]. When these are used as a basis for integrating factor TSRK methods, the result is SSP provided that the abscissas are

non-decreasing. In Section 3 we discuss explicit SSP two-step Runge–Kutta methods, review the optimization problem, and review the resulting eSSP-TSRK methods presented in [22]. In Section 4 we develop the SSP theory for explicit SSP integrating factor two-step Runge–Kutta (SSPIF-TSRK) methods, which is very similar to that of the explicit SSP integrating factor Runge–Kutta (SSPIFRK) methods in [15], and show that as long as these methods have non-decreasing abscissas the result is SSP. We show that to find appropriate explicit SSP two-step Runge–Kutta for pairing with integrating factor methods, the optimization problem needs to be augmented by a simple condition on the abscissas. In Section 5 we present the optimized methods found using this approach, and discuss their features. In Section 6 we test these methods for convergence and for their SSP performance on numerical test cases.

1.3. Efficient computation of the matrix exponential. It is important to note that in the current work as well as our prior work [15, 16], the cost of computation of the matrix exponential will be a critical factor in the ability to efficiently implement the proposed methods. If the computation of the matrix exponential is only needed once per simulation (i.e. if L is a constant coefficient operator), the cost may be reasonable, but in some cases the exponential must be computed at every step, and low-storage, matrix-free approaches are needed. New approaches to efficiently compute the matrix exponential have been recently considered in [1, 31, 27, 7], and such approaches, as well as others, will be critical for bringing the integrating factor methods proposed in [15, 16] and the current work into practical use.

2. Review of SSP integrating factor Runge–Kutta methods. In [15, 16] we considered problems of the form

$$(2.1) \quad u_t = Lu + N(u)$$

that result from a semi-discretization of a hyperbolic PDE. We focus on the case where the problem has a linear constant coefficient component Lu and a nonlinear component $N(u)$, such that

$$(2.2) \quad \|u^n + \Delta t N(u^n)\| \leq \|u^n\| \quad \text{for} \quad \Delta t \leq \Delta t_{\text{FE}}$$

and

$$(2.3) \quad \|u^n + \Delta t Lu^n\| \leq \|u^n\| \quad \text{for} \quad \Delta t \leq \tilde{\Delta} t_{\text{FE}}.$$

The notation $\|\cdot\|$ here refers to some convex functional (not usually a norm) needed for nonlinear non-inner-product stability. Of particular interest is the case where L is a linear operator that significantly restricts the allowable time-step due to strong stability concerns, i.e. where $\tilde{\Delta} t_{\text{FE}} \ll \Delta t_{\text{FE}}$. For such cases, where nonlinear non-inner-product stability properties are of concern, an implicit or implicit-explicit (IMEX) SSP scheme doesn't significantly alleviate the allowable time-step [4] and such methods will result in severe constraints on the allowable time-step [25, 19, 6, 9, 4].

This motivated our investigation of integrating factor methods, where the linear component Lu is handled exactly and then the time-step restriction is dependent only upon the step size restriction $\Delta t \leq \Delta t_{\text{FE}}$ coming from the the nonlinear component $N(u)$. The methodology we considered in [15] in order to alleviate the restriction on the allowable time-step begins with an integrating factor approach:

$$\begin{aligned} e^{-Lt}u_t - e^{-Lt}Lu &= e^{-Lt}N(u) \\ (e^{-Lt}u)_t &= e^{-Lt}N(u). \end{aligned}$$

Defining $w = e^{-Lt}u$ gives us the modified ODE system

$$(2.4) \quad w_t = e^{-Lt}N(e^{Lt}w) = G(w),$$

which can then be evolved forward in time. In [15] we used an explicit SSP Runge–Kutta method of the form (1.5). We observed that this approach is not sufficient to ensure strong stability of the system. However, as we showed, if the transformed problem is stepped forward using a SSP Runge–Kutta method that is carefully chosen to have non-decreasing abscissas the resulting method *will* preserve the desired strong stability property.

2.1. Motivating example. Before we repeat the formal results from [15], which explain how non-decreasing abscissas in the Runge–Kutta method are needed to preserve strong stability when using an integrating factor approach, we present the following example, presented in [15], which illustrates this fact:

Consider the partial differential equation

$$(2.5) \quad u_t + au_x + \left(\frac{1}{2}u^2\right)_x = 0 \quad u(0, x) = \begin{cases} 1, & \text{if } 0 \leq x \leq 1/2 \\ 0, & \text{if } x > 1/2 \end{cases}$$

where $x \in [0, 1]$ and the boundary conditions are periodic. In this example we consider $a = 10$. A first-order upwind difference with 400 points in space is used to semi-discretize the linear term $Lu \approx -10u_x$. The nonlinear terms $N(u) \approx -\left(\frac{1}{2}u^2\right)_x$ are approximated using a fifth order WENO finite difference [17].

The transformed problem (2.4) is stepped forward in time by the explicit SSP Runge–Kutta method with non-decreasing abscissas, denoted eSSPRK⁺(3,3) in [15], which has $\mathcal{C} = \frac{3}{4}$. The resulting SSP integrating factor Runge–Kutta method is:

$$(2.6) \quad \begin{aligned} u^{(1)} &= \frac{1}{2}e^{\frac{2}{3}\Delta t L}u^n + \frac{1}{2}e^{\frac{2}{3}\Delta t L} \left(u^n + \frac{4}{3}\Delta t N(u^n) \right) \\ u^{(2)} &= \frac{2}{3}e^{\frac{2}{3}\Delta t L}u^n + \frac{1}{3} \left(u^{(1)} + \frac{4}{3}\Delta t N(u^{(1)}) \right) \\ u^{n+1} &= \frac{59}{128}e^{\Delta t L}u^n + \frac{15}{128}e^{\Delta t L} \left(u^n + \frac{4}{3}\Delta t N(u^n) \right) \\ &\quad + \frac{27}{64}e^{\frac{1}{3}\Delta t L} \left(u^{(2)} + \frac{4}{3}\Delta t N(u^{(2)}) \right). \end{aligned}$$

For comparison, the transformed problem (2.4) is also stepped forward in time by the explicit eSSPRK(3,3) Runge–Kutta method in [29] which has $\mathcal{C} = 1$ (this method is widely known as the Shu–Osher method). The resulting (non-SSP) integrating factor Runge–Kutta method becomes:

$$(2.7) \quad \begin{aligned} u^{(1)} &= e^{L\Delta t}u^n + e^{L\Delta t}\Delta t N(u^n) \\ u^{(2)} &= \frac{3}{4}e^{\frac{1}{2}L\Delta t}u^n + \frac{1}{4}e^{-\frac{1}{2}L\Delta t} \left(u^{(1)} + \Delta t N(u^{(1)}) \right) \\ u^{n+1} &= \frac{1}{3}e^{L\Delta t}u^n + \frac{2}{3}e^{\frac{1}{2}L\Delta t} \left(u^{(2)} + \Delta t N(u^{(2)}) \right). \end{aligned}$$

Exponentials with negative exponents appear in this formulation due to the fact that the SSP Shu–Osher method it is based on has decreasing abscissas. Using different

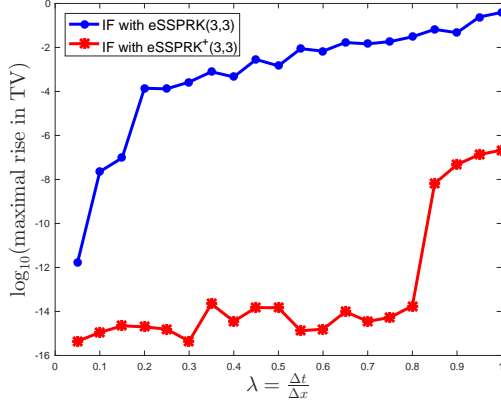


FIG. 2.1. *Motivating example: TV behavior of the evolution over 25 time steps using the integrating factor methods (2.7) (blue dots) and (2.6) (red stars).*

values of Δt we evolved the solution 25 time steps using the integrating factor Runge–Kutta methods (2.6) and (2.7), and calculated the maximal rise in total variation over each stage for all time steps.

In Figure 2.1 we show the \log_{10} of the maximal rise in total variation vs. the value of $\lambda = \frac{\Delta t}{\Delta x}$ of the evolution using (2.7) (in blue) and using (2.6) (in red). The results from (2.6) maintain a small maximal rise in total variation up to $\lambda \approx 0.8$, while in the results from (2.7) we observe that the maximal rise in total variation is large even for very small values of λ . This behavior is due to the fact that basing an integrating factor Runge–Kutta method on an explicit SSP Runge–Kutta method is not enough to ensure the preservation of a strong stability property – the method must also satisfy the non-decreasing abscissa condition to ensure that the strong stability property is preserved.

2.2. SSP analysis of integrating factor Runge–Kutta method. To understand the results in [15] recall that each stage of the explicit Runge–Kutta method (1.5) applied to the transformed problem (2.4) becomes

$$\begin{aligned}
 u^{(i)} &= \sum_{j=0}^{i-1} \left(\alpha_{i,j} e^{L(t_i - t_j)} u^{(j)} + \Delta t \beta_{i,j} e^{L(t_i - t_j)} N(u^{(j)}) \right) \\
 &= \sum_{j=0}^{i-1} \left(\alpha_{i,j} e^{L(c_i - c_j) \Delta t} u^{(j)} + \Delta t \beta_{i,j} e^{L(c_i - c_j) \Delta t} N(u^{(j)}) \right) \\
 &= \sum_{j=0}^{i-1} e^{L(c_i - c_j) \Delta t} \left(\alpha_{i,j} u^{(j)} + \Delta t \beta_{i,j} N(u^{(j)}) \right).
 \end{aligned}$$

In the following, we consider each piece of this formula. Each of the following results is taken directly from [15] and the proofs appear in that work. We first consider the behavior of the exponential operator $e^{\tau L}$. The following theorem ensures that this term is strongly stable as long as $\tau \geq 0$:

THEOREM 2.1. *If a linear operator L satisfies (2.3) for some value of $\tilde{\Delta} t_{\text{FE}} > 0$,*

then

$$(2.8) \quad \|e^{\tau L} u^n\| \leq \|u^n\| \quad \forall \tau \geq 0.$$

Note that in the following results we no longer require the condition (2.3) but rather the weaker condition (2.8). Pairing this exponential with a forward Euler type step still preserves strong stability:

COROLLARY 1. *Given a linear operator L that satisfies (2.8) and a (possibly non-linear) operator $N(u)$ that satisfies (2.2) for some value of $\Delta t_{FE} > 0$, we have*

$$(2.9) \quad \|e^{\tau L}(u^n + \Delta t N(u^n))\| \leq \|u^n\| \quad \forall \Delta t \leq \Delta t_{FE}, \quad \text{provided that } \tau \geq 0.$$

Finally, we put the pieces together:

THEOREM 2.2. *Given a linear operator L that satisfies (2.8) and a (possibly non-linear) operator $N(u)$ that satisfies (2.2) for some value of $\Delta t_{FE} > 0$, and a Runge–Kutta integrating factor method of the form*

$$(2.10) \quad \begin{aligned} u^{(0)} &= u^n, \\ u^{(i)} &= \sum_{j=0}^{i-1} e^{L(c_i - c_j)\Delta t} \left(\alpha_{i,j} u^{(j)} + \Delta t \beta_{i,j} N(u^{(j)}) \right), \quad i = 1, \dots, s \\ u^{n+1} &= u^{(s)} \end{aligned}$$

where $0 = c_1 \leq c_2 \leq \dots \leq c_s$, then u^{n+1} obtained from (2.10) satisfies

$$(2.11) \quad \|u^{n+1}\| \leq \|u^n\| \quad \forall \Delta t \leq C \Delta t_{FE}.$$

where

$$C = \min_{i,j} \frac{\alpha_{i,j}}{\beta_{i,j}}.$$

In [15] we used these results to motivate the search for explicit SSP Runge–Kutta methods with non-decreasing abscissas, that would pair well with an integrating factor approach to produce SSPIFRK methods. We found explicit SSP Runge–Kutta methods of up to $s = 10$ stages and order $p = 4$ and demonstrated the performance of the corresponding explicit SSP integrating factor Runge–Kutta methods on widely used test cases.

In [16] we show that it is not absolutely necessary to have non-decreasing abscissas: even if there are some decreasing abscissas one can preserve the strong stability property by replacing the operator L by an operator \tilde{L} that satisfies

$$\|e^{-\tau \tilde{L}} u^n\| \leq \|u^n\| \quad \forall \tau \geq 0,$$

whenever the difference of abscissas $c_i - c_j$ is negative. For hyperbolic partial differential equations, \tilde{L} is simply the spatial discretization that is stable for the backwards in time version of the equation. This approach is employed in the classical SSP literature, called downwinding. In that case, negative coefficients $\beta_{i,j}$ preserve the SSP property as long as the operator is replaced by a downwinded operator [10, 11, 21].

3. A review of explicit SSP two-step Runge–Kutta methods. The order barrier of $p \leq 4$ on explicit SSP Runge–Kutta methods and the need for higher order explicit methods in simulations led to the investigation of the SSP properties of explicit two-step Runge–Kutta methods (TSRK) [22], and later of explicit multi-step Runge–Kutta methods (MSRK) [23]. In [22] it was shown that explicit SSP TSRK methods do not use values of stages from the previous steps, so that they can be written as:

$$(3.1a) \quad \begin{aligned} y_1^n &= u^n \\ y_i^n &= d_i u^{n-1} + (1 - d_i) u^n + \Delta t \hat{a}_{i1} F(u^{n-1}) + \Delta t \sum_{j=1}^{i-1} a_{ij} F(y_j^n) \quad 2 \leq i \leq s \\ u^{n+1} &= \theta u^n + (1 - \theta) u^{n-1} + \Delta t \hat{b}_1 F(u^{n-1}) + \Delta t \sum_{j=1}^s b_j F(y_j^n). \end{aligned}$$

An extension of this to multiple steps was studied in [23].

We can write the method in the matrix form

$$(3.2) \quad \mathbf{w} = \mathbf{S}\mathbf{x} + \Delta t \mathbf{T}\mathbf{f}$$

where \mathbf{S} and \mathbf{T} are the matrices defined by:

$$(3.3) \quad \mathbf{S} = \begin{pmatrix} 1 & 0 \\ \mathbf{D} \\ \boldsymbol{\theta}^\top \end{pmatrix} \quad \mathbf{T} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & 0 \\ \hat{\mathbf{A}} & \mathbf{A} & \mathbf{0} \\ \hat{\mathbf{b}}^\top & \mathbf{b}^\top & 0 \end{pmatrix}$$

and \mathbf{w} , \mathbf{x} , and \mathbf{f} are defined by:

$$(3.4) \quad \mathbf{w} = \begin{pmatrix} u^{n-1} \\ y_1^n \\ \vdots \\ y_s^n \\ u^{n+1} \end{pmatrix} \quad \mathbf{x} = \begin{pmatrix} u^{n-1} \\ u^n \end{pmatrix} \quad \mathbf{f} = \begin{pmatrix} F(u^{n-1}) \\ F(y_1^n) \\ \vdots \\ F(y_s^n) \\ F(u^{n+1}) \end{pmatrix}$$

The \mathbf{D} is a $2 \times s$ matrix that contains the elements $\mathbf{D}_{i,1} = d_i$ and $\mathbf{D}_{i,2} = 1 - d_i$, for $i = 1, \dots, s$. The vector $\boldsymbol{\theta}$ contains the elements $\theta_1 = \theta$ and $\theta_2 = 1 - \theta$. Note that $\mathbf{S}\mathbf{e} = \mathbf{e}$, where \mathbf{e} is a vector ones of the proper length.

3.1. Formulating the optimization problem. In [22] a SSP optimization problem was defined for TSRK methods. The derivation of the optimization problem begins with the matrix form (3.2) of the TSRK:

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

We then add $r\mathbf{T}\mathbf{w}$ to each side of the method

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

and rearrange to obtain

$$\mathbf{w} = \mathbf{R}\mathbf{x} + \mathbf{P}\left(\mathbf{w} + \frac{\Delta t}{r}\mathbf{f}\right).$$

Where $\mathbf{P} = r(\mathbf{I} + r\mathbf{T})^{-1}\mathbf{T}$ and $\mathbf{R} = (\mathbf{I} + r\mathbf{T})^{-1}\mathbf{S} = (\mathbf{I} - \mathbf{P})\mathbf{S}$. Note that the row sums of $[\mathbf{R} \ \mathbf{P}]$ are each equal to one

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

It is easy to see that if \mathbf{R} and \mathbf{P} have no negative values, then the method is a convex combination of forward Euler steps, and so is SSP with SSP coefficient defined by

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

This leads directly to the optimization problem: We wish to find coefficients \mathbf{T} and \mathbf{S} for explicit SSP two-step Runge–Kutta methods that maximize the value of r while satisfying

$$(3.5a) \quad (\mathbf{I} + r\mathbf{T})^{-1} \mathbf{S} \geq 0$$

$$(3.5b) \quad r(\mathbf{I} + r\mathbf{T})^{-1} \mathbf{T} \geq 0$$

$$(3.5c) \quad \tau_\rho = 0 \quad \text{for } \rho = 1, \dots, p.$$

This requires that $(\mathbf{I} + r\mathbf{T})^{-1}$ exists. The inequalities (3.5a) and (3.5b) are understood componentwise. The τ_ρ in (3.5c) are the order conditions listed in Appendix A.

3.2. Optimized SSP two-step Runge–Kutta methods. In [22] a SSP optimization problem was defined for the TSRK methods, and optimized explicit TSRK methods of up to eighth order were found numerically. The SSP coefficients \mathcal{C} of these methods up to seventh order are in Table 3.1. Of course, a method with more stages will have a higher SSP coefficient, but also require more computations. To account for this we define the *effective SSP coefficient* $\mathcal{C}_{\text{eff}} = \frac{1}{s}\mathcal{C}$, which is the SSP coefficient normalized by the number of stages, in Table 3.2. An eleven stage eighth order method has an SSP coefficient $\mathcal{C} = 0.341$ and an effective SSP Coefficient $\mathcal{C}_{\text{eff}} = 0.031$.

These methods break the fourth-order barrier of (one-step) SSP Runge–Kutta methods and have significantly larger SSP coefficients than the corresponding order multi-step methods. Numerical tests in [22] showed that these high order SSP two-step Runge–Kutta methods are beneficial for preserving the order and strong stability properties when used with high order spatial discretizations for the time integration of a variety of hyperbolic PDEs. It was proved in [22] that explicit SSP TSRK methods have order at most eight. In [23] it was shown that adding more steps overcomes this order barrier as well.

4. Explicit SSP two-step Runge–Kutta schemes for use with integrating factor methods. In this section we consider, as we did in [15, 16], a problem of the form (2.1) with a linear constant coefficient component Lu that satisfies the strong stability condition (2.3) and a nonlinear component $N(u)$ that satisfies the strong stability condition (2.2). When $\tilde{\Delta}t_{\text{FE}} \ll \Delta t_{\text{FE}}$ using an explicit SSP Runge–Kutta method will result in severe constraints on the allowable time-step that is not alleviated by using an implicit or an implicit-explicit (IMEX) SSP Runge–Kutta method [25, 19, 6, 9, 4]. This is in contrast to the case with linear L_2 stability, where using an implicit or an IMEX method may completely alleviate the time-step restriction coming from the stiff component.

To alleviate the restriction on the allowable time-step we can solve the linear part exactly by an integrating factor approach as in (2.4). This new ODE can be

$\begin{smallmatrix} \text{p} \\ \text{s} \end{smallmatrix}$	2	3	4	5	6	7
1	-	-	-	-	-	-
2	1.4142	0.7320	-	-	-	-
3	2.4495	1.6506	0.8588	-	-	-
4	3.4641	2.3027	1.5926	0.8542	-	-
5	4.4721	2.9879	2.3605	1.6481	-	-
6	5.4772	3.7768	3.0559	2.3093	0.5957	-
7	6.4807	4.4836	3.7405	2.9278	1.2719	-
8	7.4833	5.2227	4.4921	3.5794	1.9384	0.5666
9	8.4853	6.0498	5.2705	3.9415	2.5826	1.1199
10	9.4868	6.8274	6.1039	4.2544	3.1992	1.7857

TABLE 3.1

SSP coefficients of the optimized eSSP-TSRK(s, p) methods [22].

$\begin{smallmatrix} \text{p} \\ \text{s} \end{smallmatrix}$	2	3	4	5	6	7
1	-	-	-	-	-	-
2	0.7071	0.3660	-	-	-	-
3	0.8165	0.5502	0.2863	-	-	-
4	0.8660	0.5757	0.3982	0.2135	-	-
5	0.8944	0.5976	0.4721	0.3296	-	-
6	0.9129	0.6295	0.5093	0.3849	0.0993	-
7	0.9258	0.6405	0.5343	0.4183	0.1817	-
8	0.9354	0.6528	0.5615	0.4474	0.2423	0.0708
9	0.9428	0.6722	0.5856	0.4379	0.2869	0.1244
10	0.9487	0.6827	0.6104	0.4254	0.3199	0.1786

TABLE 3.2

Effective SSP coefficients of the optimized eSSP-TSRK(s, p) methods [22].

evolved forward in time using standard methods. As described in Section 2, in [15] we considered stepping the transformed problem (2.4) forward using an explicit SSP Runge–Kutta method of the form (1.5) and found that the result was SSP provided that the method had non-decreasing abscissas. In the following, we consider extending the SSP integrating factor approach by using multi-step and two-step Runge–Kutta integrating factor methods to obtain order $p \geq 4$

4.1. SSP integrating factor linear multi-step methods. It can be easily shown that if we use an explicit SSP multi-step method

$$(4.1) \quad u^{n+1} = \sum_{l=1}^k (\alpha_l u^{n-k+l} + \Delta t \beta_l F(u^{n-k+l})),$$

which satisfies

$$u^{n+1} \leq \max_{l=1, \dots, k} \{\|u^{n-k+l}\|\}.$$

with SSP coefficient $\mathcal{C} = \max_l \frac{\alpha_l}{\beta_l}$, the result is SSP as well, with the same SSP coefficient:

THEOREM 4.1. *Given a linear operator L that satisfies (2.8) and a (possibly non-linear) operator $N(u)$ that satisfies (2.2) for some value of $\Delta t_{FE} > 0$, and a k -step explicit SSP integrating factor multi-step method of the form*

$$(4.2) \quad u^{n+1} = \sum_{l=1}^k e^{L(k-l+1)} (\alpha_l u^{n-k+l} + \Delta t \beta_l N(u^{n-k+l})),$$

then the numerical solution satisfies

$$(4.3) \quad \|u^{n+1}\| \leq \max_{l=1, \dots, k} \{\|u^{n-k+l}\|\}. \quad \forall \Delta t \leq C \Delta t_{FE}.$$

where $C = \max_l \frac{\alpha_l}{\beta_l}$.

Notice that for multi-step methods the notion of SSP includes all the previous steps considered.

Proof. We write the method in the form

$$\begin{aligned} \|u^{n+1}\| &= \left\| \sum_{l=1}^k e^{L(k-l+1)} (\alpha_l u^{n-k+l} + \Delta t \beta_l N(u^{n-k+l})) \right\| \\ &\leq \sum_{l=1}^k \|e^{L(k-l+1)} (\alpha_l u^{n-k+l} + \Delta t \beta_l N(u^{n-k+l}))\| \\ &\leq \sum_{l=1}^k \alpha_l \|e^{L(k-l+1)} \left(u^{n-k+l} + \Delta t \frac{\beta_l}{\alpha_l} N(u^{n-k+l}) \right)\| \\ &\leq \sum_{l=1}^k \alpha_l \|u^{n-k+l}\|. \end{aligned}$$

the last inequality holds provided that $\Delta t \frac{\beta_l}{\alpha_l} \leq \Delta t_{FE}$. Using the fact that by consistency we have $\sum_{l=1}^k \alpha_l = 1$, we can conclude that

$$\|u^{n+1}\| \leq \max_{l=1, \dots, k} \{\|u^{n-k+l}\|\}.$$

□

However, SSP multi-step methods require many steps for high order and have small SSP coefficients as shown in [9]. Now that we see that building explicit SSP multi-step integrating factor methods is trivial, we are ready to approach the main question of this paper: is it possible to extend our results in [15] to two-step Runge–Kutta methods?

4.2. SSP integrating factor two-step Runge–Kutta methods. Any explicit SSP TSRK method of the form (3.1) that is SSP with SSP coefficient $C = r$ can be re-written in the matrix-vector form (3.2), which can also be represented as

$$\begin{aligned} (4.4a) \quad y_1 &= u^n \\ y_i &= v_{i,1} u^{n-1} + v_{i,2} u^n + \hat{\alpha}_i \left(u^{n-1} + \frac{\Delta t}{r} F(u^{n-1}) \right) \\ &\quad + \sum_{j=1}^{i-1} \alpha_{i,j} \left(y_j + \frac{\Delta t}{r} F(y_j) \right) \quad 2 \leq i \leq s+1 \\ u^{n+1} &= y_{s+1}, \end{aligned}$$

where $0 \leq v_i \leq 1$, $\hat{\alpha}_i \geq 0$, and $\alpha_{i,j} \geq 0$. Here, the consistency condition $\mathbf{R}\mathbf{e} + \mathbf{P}\mathbf{e} = \mathbf{e}$ can be written as $v_{i,1} + v_{i,2} + \hat{\alpha}_i + \sum_{j=1}^{i-1} \alpha_{i,j} = 1$ for each row i .

We note that the time-levels for each stage, also called the abscissas,

$$\mathbf{c} = (c_1, c_2, c_3, \dots, c_s, c_{s+1}) = (0, c_2, c_3, \dots, c_s, 1)$$

can be computed (in the matrix form) by \mathbf{D} , \mathbf{A} , and $\hat{\mathbf{A}}$ where $\mathbf{c} = \hat{\mathbf{A}}\mathbf{e} + \mathbf{A}\mathbf{e} - \mathbf{D}_{i,1}$. The abscissa corresponding to u^n is 0, the ones corresponding to all the internal stages $j = 2, \dots, s$ are c_j and the one corresponding to the final stage u^{n+1} is 1.

Applying (4.4) to the transformed equation becomes

$$\begin{aligned} y_1 &= u^n \\ (4.5a) \quad y_i &= v_{i,1}e^{(c_i+1)\Delta t}u^{n-1} + v_{i,2}u^n + \hat{\alpha}_i e^{(c_i+1)\Delta t} \left(u^{n-1} + \frac{\Delta t}{r}N(u^{n-1}) \right) \\ &\quad + \sum_{j=1}^{i-1} \alpha_{i,j} e^{(c_i-c_j)\Delta t} \left(y_j + \frac{\Delta t}{r}N(y_j) \right) \quad 2 \leq i \leq s+1 \\ u^{n+1} &= y_{s+1} \end{aligned}$$

The following theorem tells us what conditions an explicit SSP TSRK method (4.4) has to satisfy to preserve the strong stability of a numerical solution when coupled with an integrating factor approach as in (4.5).

THEOREM 4.2. *Given a linear operator L that satisfies (2.8) and a (possibly non-linear) operator $N(u)$ that satisfies (2.2) for some value of $\Delta t_{FE} > 0$, and an explicit SSP two-step Runge–Kutta integrating factor method of the form (4.5) based on the SSP method (4.4) with non-decreasing abscissas $0 = c_1 \leq c_2 \leq c_3 \dots \leq c_s \leq c_{s+1} = 1$, then the numerical solution satisfies*

$$(4.6) \quad \|u^{n+1}\| \leq \max \{ \|u^{n-1}\|, \|u^n\| \} \quad \forall \Delta t \leq \mathcal{C}\Delta t_{FE}.$$

where the SSP coefficient $\mathcal{C} = r$.

Proof. Consider each stage of (4.5)

$$\begin{aligned} \|y_i\| &= \|v_{i,1}e^{(c_i+1)\Delta t}u^{n-1} + v_{i,2}u^n + \hat{\alpha}_i e^{(c_i+1)\Delta t} \left(u^{n-1} + \frac{\Delta t}{r}N(u^{n-1}) \right) \\ &\quad + \sum_{j=1}^{i-1} \alpha_{i,j} e^{(c_i-c_j)\Delta t} \left(y_j + \frac{\Delta t}{r}N(y_j) \right)\| \\ &\leq v_{i,1}\|e^{(c_i+1)\Delta tL}u^{n-1}\| + v_{i,2}\|u^n\| + \hat{\alpha}_i\|e^{(c_i+1)\Delta t} \left(u^{n-1} + \frac{\Delta t}{r}N(u^{n-1}) \right)\| \\ &\quad + \sum_{j=1}^{i-1} \alpha_{i,j}\|e^{(c_i-c_j)\Delta tL} \left(y_j + \frac{\Delta t}{r}N(y_j) \right)\| \end{aligned}$$

Now noting that $c_i - c_j \geq 0$ and applying the result in Corollary 1, we obtain

$$\begin{aligned} \|y_i\| &\leq v_{i,1}\|u^{n-1}\| + v_{i,2}\|u^n\| + \hat{\alpha}_i\|u^{n-1}\| + \sum_{j=1}^{i-1} \alpha_{i,j}\|u^n\| \\ &\leq \max\{\|u^n\|, \|u^{n-1}\|\} \end{aligned}$$

for any $\Delta t \leq \mathcal{C}\Delta t_{FE}$. □

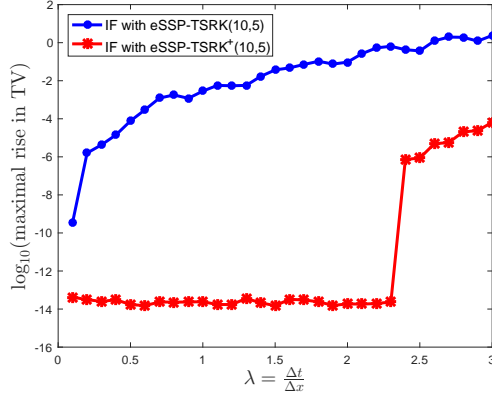


FIG. 4.1. Total variation behavior of the evolution over 25 time-steps for the motivating example in Subsection 2.1, using the integrating factor methods based on $eSSP\text{-}TSRK^+(10, 5)$ (red stars) and $eSSP\text{-}TSRK(10, 5)$ (blue dots).

This theorem shows us that to find optimal SSP two-step Runge–Kutta methods that are suitable for use with the integrating factor approach, we must augment the optimization problem with additional conditions reflecting the need for non-decreasing abscissas.

To see how this theorem matters in practice, consider, once again, the motivating example in Subsection 2.1 where we solve Equation (2.5) using an integrating factor approach. In this case, we use $a = 2$ so $Lu \approx -2u_x$. Otherwise, everything is the same as in Subsection 2.1. We evolve the transformed problem forward 25 time-steps for various time-steps using the $eSSP\text{-}TSRK(10, 5)$ method in [22] and a new $eSSP\text{-}TSRK^+(10, 5)$ method that has non-decreasing abscissas. The \log_{10} of the maximal rise in total variation at each stage vs. the value $\lambda = \frac{\Delta t}{\Delta x}$ is shown in Figure 4.1. We observe that the new method preserves the TVD behavior of the spatial discretization up to a large λ , while the $eSSP\text{-}TSRK(10, 5)$ method in [22] does not preserve the TVD property of the spatial discretization.

5. Optimal and optimized methods with non-decreasing abscissas. We augmented the optimization problem from [22] described in Subsection 3.1 by imposing the non-decreasing abscissa constraint

$$(5.1) \quad 0 = c_1 \leq c_2 \leq c_3 \leq \dots \leq c_{s-1} \leq c_s \leq c_{s+1} = 1$$

in addition to (3.5). We then implemented in MATLAB (as in [18, 20, 19]), and used to find optimized explicit $eSSP\text{-}TSRK^+$ methods of $s \leq 10$ stages and order $p \leq 7$, as well as an eighth order method with $s = 11$. The new methods have non-decreasing abscissas, and are denoted $eSSP\text{-}TSRK^+(s, p)$. As we saw above, these methods preserve the SSP properties of a transformed problem (2.4) and so can be used as a basis for explicit SSP integrating factor two-step Runge–Kutta (SSPIF-TSRK) methods.

The SSP coefficients and effective SSP coefficients of the optimized methods with non-decreasing abscissas ($eSSP\text{-}TSRK^+$) are listed in Tables 5.1 and 5.2. We boldface the coefficients of the $eSSP\text{-}TSRK^+$ methods that are the same as the optimized $eSSP\text{-}TSRK$ methods (previously found in [22, 23]) shown in Tables 3.1 and 3.2. In Figure 5.1 we show the effective SSP coefficients of the $eSSP\text{-}TSRK^+$ methods and compare

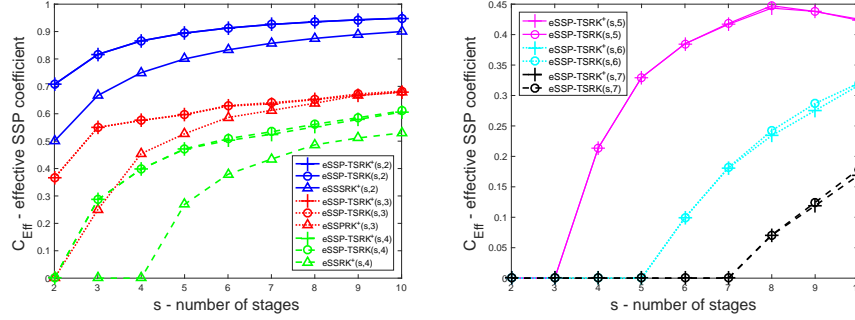


FIG. 5.1. The effective SSP coefficient for methods of order $p = 2$ (blue), $p = 3$ (red) and $p = 4$ (green) on left, and of order $p = 5$ (magenta), $p = 6$ (cyan) and $p = 7$ (black) on right. The circles indicate the SSP coefficient of the optimized eSSP-TSRK methods in [22] while the + signs are the SSP coefficients of the optimized eSSP-TSRK⁺ methods. The triangles show the SSP coefficients of the corresponding eSSPRK⁺ methods (where relevant).

them to those of the corresponding eSSP-TSRK methods. For orders $p \leq 4$ we also show the effective SSP coefficients of the eSSPRK⁺, for comparison. These tables and graphs include all the second order $p = 2$ methods, which already have non-decreasing coefficients.

The third order eSSP-TSRK⁺($s, 3$) methods with $s = 2, 3, 4$ also have the same effective SSP coefficients as the eSSP-TSRK($s, 3$) methods in [22]. Although the eSSP-TSRK(3, 3) method in [22] has decreasing coefficients, we were able to find an eSSP-TSRK⁺(3, 3) non-decreasing coefficient method with the same SSP coefficient. The eSSP-TSRK⁺($s, 3$) methods with $s > 4$ have smaller SSP coefficients than the corresponding eSSP-TSRK methods, but the loss in size of effective SSP coefficient is under 1%. These methods have a significant advantage over the eSSPRK⁺ methods, but this advantage disappears as we allow a larger number of stages s , as can be seen in Figure 5.1.

The fourth order eSSP-TSRK⁺($s, 4$) methods with $s = 3, 4$ also have the same SSP coefficients as the corresponding eSSP-TSRK in [22]. These two methods are significant because allowing an additional step makes possible fourth order SSP methods with $s = 3, 4$, which cannot be achieved for eSSPRK methods. Once we look at more stages, $s > 4$, we see that the eSSP-TSRK⁺($s, 4$) methods suffer some loss over the eSSP-TSRK($s, 4$) in the magnitude of the effective SSP coefficient, but it is still under 2%, so the differences are not obvious in Figure 5.1. Once again, these eSSP-TSRK⁺($s, 4$) methods offer a significantly larger effective SSP coefficient than the corresponding eSSPRK⁺ methods.

The fifth order eSSP-TSRK⁺($s, 5$) methods with $s = 4, 5, 6, 9$ also have the same SSP coefficients as the corresponding eSSP-TSRK methods in [22]. We note that our optimization code found a slightly better eSSP-TSRK⁺(9, 5) than the eSSP-TSRK(9, 5) previously found in the literature. For $s = 7, 8, 10$ the eSSP-TSRK⁺($s, 5$) cannot match the SSP coefficients of the eSSP-TSRK($s, 5$) methods in [22]. The loss that we see in the effective SSP coefficients is, once again, under 1%, so the difference cannot be seen in Figure 5.1. It is important to note that there are no explicit SSP Runge–Kutta methods of fifth order or above, so the eSSP-TSRK methods are particularly useful if one needs to go to higher order.

The sixth order eSSP-TSRK⁺($s, 6$) methods we found all have smaller SSP coefficients than the eSSP-TSRK($s, 6$) in [22]. For $s \leq 7$ the loss is under 0.5% which

cannot be seen in Figure 5.1. However, once we get to $s = 8, 9, 10$ we see more significant losses in the size of the SSP coefficients, however these are still below 5%. It is interesting to notice that the explicit eSSP-TSRK⁺(10, 6) method has an effective SSP coefficient $C_{\text{eff}} \approx 0.31$ that is about half that of the eSSPRK(10, 4) (which is $C_{\text{eff}} = 0.6$) and about 60% of that of the eSSPRK⁺(10, 4) (which is $C_{\text{eff}} \approx 0.5299$).

The eSSP-TSRK⁺(8, 7) method we found has the same SSP coefficient as the eSSP-TSRK(8, 7) in [22], while the eSSP-TSRK⁺(s , 7) methods with $s = 9, 10$ have a loss in the effective SSP coefficient of 4.4% and 5.55% compared with the corresponding eSSP-TSRK methods in [22]. Although it is possible to find eighth order methods, these require more than ten stages. Using our optimization code, we found an eleven stage eighth order method with non-decreasing abscissas eSSP-TSRK⁺(11, 8) with an effective SSP coefficient $C_{\text{eff}} \approx 0.0249$. The SSP coefficient of the eSSP-TSRK⁺(11, 8) methods is significantly smaller (by close to 20%) than the eleven stage eighth order method eSSP-TSRK(11, 8) in [22] which has $C_{\text{eff}} \approx 0.031$.

In general, we see that for lower number of stages, the SSP coefficients of the methods with non-decreasing abscissas are no smaller than the methods with decreasing abscissas. Perhaps this reflects the fact that the solution space for methods with s close to p is already limited, and so the additional constraint of non-decreasing abscissas poses no major restriction. However, it is interesting to notice that despite the constrained solution space when s is close to p , in every case where an eSSP-TSRK(s , p) method was found it was also possible to find an eSSP-TSRK⁺(s , p) method.

$s \backslash p$	2	3	4	5	6	7
1	-	-	-	-	-	-
2	1.4142	0.7320	-	-	-	-
3	2.4495	1.6506	0.8588	-	-	-
4	3.4641	2.3027	1.5926	0.8542	-	-
5	4.4721	2.9807	2.3523	1.6481	-	-
6	5.4772	3.7672	3.0140	2.3093	0.5958	-
7	6.4807	4.4533	3.6751	2.9173	1.2671	-
8	7.4833	5.2134	4.4178	3.5477	1.8728	0.5666
9	8.4853	6.0012	5.2120	3.9426	2.4784	1.0715
10	9.4868	6.7916	6.0626	4.2362	3.1646	1.6892

TABLE 5.1

SSP coefficients of the optimized eSSP-TSRK⁺(s , p) methods.

6. Numerical Results. In this section we demonstrate the performance of the SSP integrating factor two-step Runge–Kutta (SSPIF-TSRK) methods based on the eSSP-TSRK⁺ methods reported in Section 5. The SSP coefficients of these methods are presented in Table 5.1; the coefficients of these methods can be downloaded from [8].

Our tests focus on convergence and the SSP properties of the methods. In Subsection 6.1 we verify convergence at the design order for a selection of these methods on the van der Pol problem. In Subsections 6.2 and 6.3 we study the behavior of these methods in terms of their allowable TVD time-step on linear and nonlinear problems with spatial discretizations that are TVD. The simple TVD test in these sections have been used extensively because they demonstrate the behavior of the SSP time-step, and provide evidence that there are some cases in which the SSP property is necessary

$\begin{smallmatrix} \text{p} \\ \text{s} \end{smallmatrix}$	2	3	4	5	6	7
1	-	-	-	-	-	-
2	0.7071	0.3660	-	-	-	-
3	0.8165	0.5502	0.2863	-	-	-
4	0.8660	0.5757	0.3982	0.2135	-	-
5	0.8944	0.5961	0.4705	0.3296	-	-
6	0.9129	0.6279	0.5023	0.3849	0.0993	-
7	0.9258	0.6362	0.5250	0.4168	0.1810	-
8	0.9354	0.6517	0.5522	0.4435	0.2341	0.0708
9	0.9428	0.6668	0.5791	0.4381	0.2754	0.1191
10	0.9487	0.6792	0.6063	0.4236	0.3165	0.1689

TABLE 5.2

Effective SSP coefficients of the optimized eSSP-TSRK⁺(s,p) methods.

to preserve the TVD property.

6.1. Example 1: Convergence study. Consider the van der Pol problem, a nonlinear system of ODEs of the form

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

The problem can be written as $\mathbf{u}_t = L\mathbf{u} + N(\mathbf{u})$ where $\mathbf{u} = (u_1; u_2)$ and

$$L = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \quad N(\mathbf{u}) = \begin{pmatrix} 0 \\ (1 - u_1^2)u_2 \end{pmatrix}.$$

We initialize the problem with $\mathbf{u}_0 = (2; 0)$. Using a variety of SSPIF-TSRK methods, we run the problem to final time $T_{final} = 2$, with $\Delta t = 0.01, 0.02, 0.04, 0.05, 0.08, 0.10$. The initial values and the exact solution (for error calculation) was calculated by MATLAB's ODE45 routine with tolerances set to $\text{AbsTol}=2 \times 10^{-14}$ and $\text{RelTol}=2 \times 10^{-14}$. We tested the SSPIF-TSRK methods based on the eSSP-TSRK⁺ methods whose SSP coefficients are presented in Table 5.1 above. We calculated the orders by finding the slopes of the $\log_{10}(\text{errors})$ using MATLAB's `polyfit` function. The results, shown in Figure 6.1 for several selected methods, validate that the methods exhibit the expected order of convergence.

6.2. Example 2: Sharpness of SSP time-step for a linear problem. While the importance of SSP methods is not limited to cases where we need to preserve the TVD property, TVD studies tend to exhibit the sharpness of the SSP time-step and so are traditionally used to test SSP time-stepping methods. We begin with a case where the TVD property of the spatial discretization can be proven, and investigate the TVD behavior of the numerical solutions resulting from evolving the semi-discretized problem using our SSP integrating factor two-step Runge–Kutta methods.

We use our SSPIF-TSRK methods to evolve in time the linear advection equation on the domain $x \in [0, 1]$ with periodic boundary conditions and a step function initial condition

$$(6.1) \quad u_t + au_x + u_x = 0 \quad u(0, x) = \begin{cases} 1, & \text{if } \frac{1}{4} \leq x \leq \frac{3}{4} \\ 0, & \text{else} \end{cases}.$$

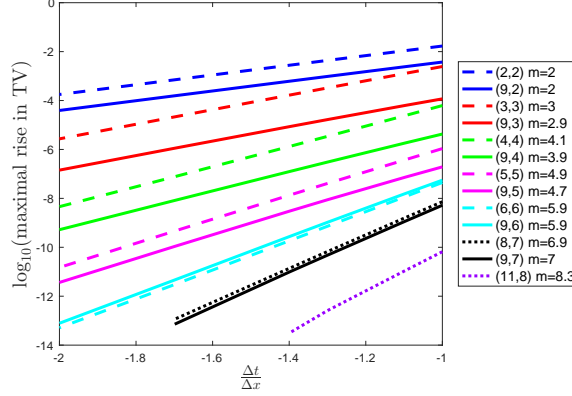


FIG. 6.1. Convergence plot for Example 1. The slope of the line is given by m in the legend.

Using a simple first-order forward difference to discretize the spatial derivatives with a spatial grid of 1000 points. We split the problem as with $Lu \approx -au_x$ and $N(u) \approx -u_x$. The spatial discretization is TVD when coupled with forward Euler in the sense that

$$\|u^n + \Delta t Lu^n\|_{TV} \leq \|u^n\|_{TV} \quad \text{for} \quad \tilde{\Delta t}_{FE} = \frac{1}{a} \Delta x$$

and

$$\|u^n + \Delta t N(u^n)\|_{TV} \leq \|u^n\|_{TV} \quad \text{for} \quad \Delta t_{FE} = \Delta x.$$

We evolve the numerical solution forward ten time-steps for different values of $\lambda = \frac{\Delta t}{\Delta x}$, and measure the total variation at each stage. To ensure internal stage monotonicity we compare the total variation at any stage to the total variation at the previous stage. We are interested in the *observed TVD time-step* Δt_{obs}^{TVD} , which is the size of time-step Δt at which the maximal rise in total variation between any two stages is greater than 10^{-12} . We compare the observed TVD time-step with the theoretical TVD time-step condition. The SSP coefficient that corresponds to the value of the observed TVD time-step is called the *observed SSP coefficient* \mathcal{C}_{obs} , defined by

$$\mathcal{C}_{obs} = \frac{\Delta t_{obs}^{TVD}}{\Delta t_{FE}}.$$

Since the forward Euler TVD time-step for this problem is $\Delta t_{FE} = \Delta x$, we have

$$\mathcal{C}_{obs} = \frac{\Delta t_{obs}^{TVD}}{\Delta x} = \lambda_{obs}^{TVD}.$$

6.2.1. Example 2a: Comparison of integrating methods for wavespeeds $a = 5$ and $a = 2$. In this example we consider a variety of SSPIF-TSRK methods for the time-evolution of (6.1) with wavespeeds $a = 5$ and $a = 2$.

In Figure 6.2 (left) we show the observed maximal rise in total variation when this equation is evolved forward by several SSPIF-TSRK methods using different values of the CFL number $\lambda = \frac{\Delta t}{\Delta x}$. We consider a selection of fourth order methods. First, we explore the performance of the SSPIF-TSRK(s, p) methods with $(s, p) = (3, 4), (4, 4)$ (the corresponding SSPIFRK methods do not exist). If we consider the effective observed time-step

$$\mathcal{C}_{obs}^{eff} = \frac{1}{s} \mathcal{C}_{obs}$$

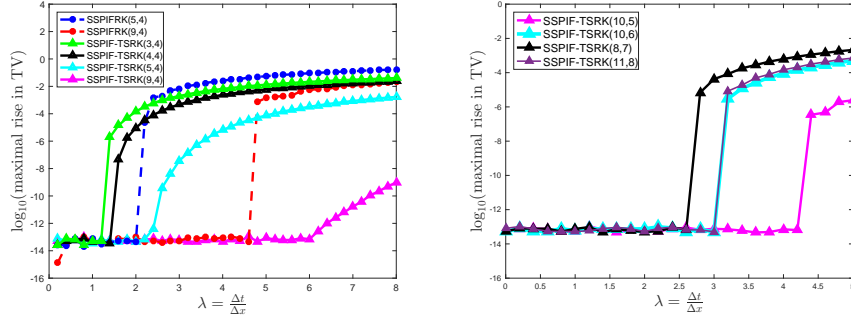


FIG. 6.2. *Example 2a: Linear advection with a step function initial condition and wavespeeds $a = 5$ (left) and $a = 2$ (right).*

(i.e. the observed time-step normalized by the number of stages s) we see that the SSPIFRK(9, 4) has $\mathcal{C}_{obs}^{eff} = 0.47$ while the SSPIF-TSRK(3, 4) has a smaller $\mathcal{C}_{obs}^{eff} = 0.42$ and SSPIF-TSRK(4, 4) has an even smaller $\mathcal{C}_{obs}^{eff} = 0.4$. However, we observe that the allowable TVD time-step of the SSPIFRK methods with $(s, p) = (5, 4), (9, 4)$ is smaller than that of the corresponding SSPIF-TSRK methods. The SSPIF-TSRK methods allow us to obtain higher order for fewer stages than the SSPIFRK methods, and larger observed time-steps for the same (s, p) .

Figure 6.2 (right) shows us the performance of the higher order SSPIF-TSRK methods, with orders $p = 5, 6, 7, 8$. In this case we use $a = 2$ in (6.1). We selected the best performing methods in terms of observed SSP time-step, the SSPIF-TSRK(s, p) methods for $(s, p) = (10, 5), (10, 6), (8, 7), (11, 8)$. We note that the $(s, p) = (11, 8)$ method has the same SSP time-step performance as the $(s, p) = (10, 6)$, showing the eighth order method to be a viable and efficient method.

6.2.2. Example 2b: Considering different wavespeeds. In [15] we showed that unlike fully explicit SSP Runge–Kutta methods or implicit-explicit SSP Runge–Kutta (SSP-IMEX) methods, the SSP integrating factor Runge–Kutta schemes have an allowable SSP time-step that is not impacted by the restriction on the time-step resulting from the linear component L . This is also true for the SSP integrating factor two-step Runge–Kutta methods. In this section, we investigate how different wavespeeds in L impact the allowable SSP time-step of different methods.

In Table 6.1 we show the observed SSP coefficient $\lambda_{obs}^{TV D}$ for values of $a = 0, 1, 5$ in (6.1) for a variety of SSPIF-TSRK methods. We notice that for the methods with $(s, p) = (5, 4), (9, 4), (6, 5), (9, 5)$ the observed SSP coefficient for $a = 0, 1$ was sharp, i.e. exactly the SSP coefficient \mathcal{C} predicted by the theory. As a gets larger ($a = 5$) the observed SSP coefficient becomes larger than predicted by the theory, which only provides a guaranteed lower bound. This occurs because the damping effect of the exponential increases as a gets larger and so rises in total variation are damped out and fall below the threshold. For the other methods, the observed SSP coefficient for $a = 0$ is larger than predicted by the theory. For $(s, p) = (7, 6), (9, 6)$ the observed SSP coefficient for $a = 0, 1$ was the same, while for $a = 5$ the observed SSP coefficient is larger. For the other methods, we see an increase in the observed SSP coefficient as a increases.

These results are in sharp distinction to explicit SSP Runge–Kutta methods and SSP-IMEX methods. In Table 6.2 we compare the observed SSP coefficient of some

Method	\mathcal{C}	λ_{obs}^{TVD}		
		for $a = 0$	$a = 1.0$	$a = 5$
SSPIF-TSRK(3,4)	0.8588	1.0454	1.2550	1.2621
SSPIF-TSRK(5,4)	2.3523	2.3523	2.3523	2.4123
SSPIF-TSRK(9,4)	5.2120	5.2120	5.2120	6.4010
SSPIF-TSRK(4,5)	0.8542	1.1852	1.3388	1.3389
SSPIF-TSRK(6,5)	2.3093	2.3093	2.3093	2.3094
SSPIF-TSRK(9,5)	3.9426	3.9426	3.9426	4.1173
SSPIF-TSRK(6,6)	0.5958	1.7771	1.7891	1.7893
SSPIF-TSRK(7,6)	1.2671	2.0239	2.0239	2.0261
SSPIF-TSRK(9,6)	2.4784	2.8038	2.8038	2.8204
SSPIF-TSRK(8,7)	0.5666	1.6624	2.6737	2.7788
SSPIF-TSRK(9,7)	1.0715	2.1626	2.4053	2.4053
SSPIF-TSRK(11,8)	0.2743	2.3871	3.1137	3.1271

TABLE 6.1

The observed SSP coefficient of a variety of SSPIF-TSRK(s, p) methods compared to their predicted SSP coefficient, shown for various wavespeeds a . The value of a does not negatively impact the observed SSP coefficient.

four stage explicit SSP integrating factor methods to those of the explicit SSP Runge–Kutta method eSSPRK(4,3) and the SSP-IMEX(4,3,K) methods.

When the eSSPRK(4,3) method is applied to (6.1) with wavespeed $a + 1$, where $a = 0, 1, 2, 10$, Table 6.2 shows that the observed SSP coefficient exactly matches the predicted

$$\lambda_{obs}^{TVD} = \frac{\mathcal{C}}{a+1} = \frac{2}{a+1}.$$

This means that as the wavespeed a increases, the observed SSP coefficient goes down as expected.

The SSP-IMEX(4,3,K) we use are from [4], and have SSP explicit and implicit parts that were optimized for the SSP step size for each value of $K = \frac{1}{a}$. The results presented in Table 6.2 show that, as expected from SSP theory, the observed SSP coefficient decays linearly as the wavespeed a rises. This matches with the results in [4] that, in contrast to the fact that IMEX methods may be A-stable, they cannot be unconditionally SSP past first order. This limitation highlights the need for SSP integrating factor methods.

Finally, we see that the observed SSP coefficients for the explicit SSP integrating factor methods do not decay as the wavespeed a goes up. For the SSPIFRK(4,3) method the observed SSP coefficient is exactly the same for $a = 0, 1, 2, 10$, while the SSPIF-TSRK(4,3) and SSPIF-TSRK(4,4) methods have the same observed SSP coefficient for $a = 0, 1, 2$ and a *larger* observed SSP coefficient for $a = 10$.

6.3. Example 3: Sharpness of SSP time-step for a nonlinear problem.

Once again we consider the motivating problem (2.5) in Subsection 2.1:

$$u_t + au_x + \left(\frac{1}{2}u^2\right)_x = 0 \quad u(0, x) = \begin{cases} 1, & \text{if } 0 \leq x \leq 1/2 \\ 0, & \text{if } x > 1/2 \end{cases}$$

on the domain $[0, 1]$ with periodic boundary conditions.

We used a first-order upwind difference to semi-discretize, with 400 spatial points, the linear term $Lu \approx -au_x$, and a fifth order WENO finite difference for the nonlinear

Method	λ_{obs}^{TVD}			
	$a = 0$	$a = 1.0$	$a = 2$	$a = 10$
SSPIF-TSRK(4,3)	2.303	2.303	2.303	2.775
SSPIF-TSRK(4,4)	1.593	1.593	1.593	1.639
SSPIFRK(4,3)	1.818	1.818	1.818	1.818
SSP-IMEX(4,3,K)	2.000	1.476	1.192	0.310
eSSPRK(4,3)	2.000	1.000	0.666	0.181

TABLE 6.2

The observed SSP coefficients for integrating factor methods, an SSP-IMEX method, and an explicit Runge–Kutta method for Example 2b with various wavespeeds a . The value of a does not negatively impact the observed SSP coefficient for the integrating factor methods, but it does for the IMEX and the explicit Runge–Kutta methods.

terms $N(u) \approx -(\frac{1}{2}u^2)_x$. We note that the WENO scheme is not TVD. However, its total variation is generally well-controlled in typical simulations.

We evolve the semi-discretized problem forward 25 time steps using $\Delta t = \lambda \Delta x$, and measure the total variation at each stage. We then calculate the maximal rise in total variation over each stage. Once again, the quantity of interest is the observed TVD time-step, but in the absence of a forward Euler condition (as WENO is not TVD) we cannot determine the observed SSP coefficient, only the observed TVD time-step.

We first show the results for a selection of third order methods on problem (2.5) with the value $a = 5$. In Figure 6.3 (left) we plot \log_{10} of the maximal rise in total variation versus the ratio $\lambda = \frac{\Delta t}{\Delta x}$. Our SSPIF-TSRK(3,3) method out-performs the SSPIFRK(3,3) method in [15] by more than 10%. In contrast, the fully explicit three stage third order Shu–Osher method applied begins to feature a large rise in total variation for a much smaller value of $\lambda = .15 \approx \frac{1}{1+a}$, as expected. Evolving the transformed problem with the Shu–Osher method (2.7) (denoted IF with eSSPRK(3,3)) and with the three-stage third order Cox and Matthews exponential time differencing Runge–Kutta method (ETDRK3) [5] results in a maximal rise in total variation that increases rapidly with λ .

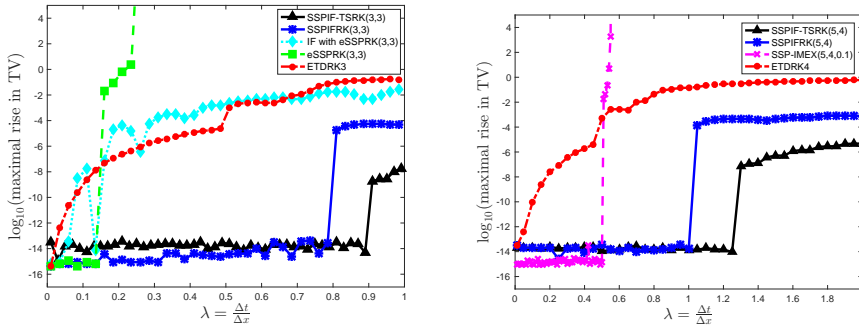


FIG. 6.3. A comparison of the TVD time-step for a variety of integrating factor, exponential time differencing, explicit and IMEX SSP Runge–Kutta methods applied to Example 3. Left: Third order methods. Right: Fourth order methods.

In Figure 6.3 (right) we show a similar study using fourth order methods. Our SSPIF-TSRK(5,4) method allows a TVD time-step of $\lambda_{obs} \approx 1.3$, while the corre-

sponding integrating factor Runge–Kutta method SSPIFRK(5,4) method maintains a small maximal rise in total variation until $\lambda_{obs} \approx 1$. The fourth order ETDRK4 method of [5] does not have good TVD performance: the maximal rise in total variation when using the ETDRK4 rises rapidly from the smallest value of λ . The SSP-IMEX(5,4,0.1) method features an observed value of $\lambda_{obs} \approx 0.5$, pointing once again to the fact that IMEX methods are not as suitable when the SSP property is desired.

Finally, we compare TVD time-step of the SSPIF-TSRK(s, p) methods with $(s, p) = (4, 3), (5, 4), (6, 5), (7, 6), (8, 7)$. In Figure 6.4 we show the \log_{10} of the maximal rise in total variation at each stage compared to the previous stage, plotted against the ratio $\lambda = \frac{\Delta t}{\Delta x}$. These integrating factor methods all perform well for this problem. Notably, the seventh order method has the largest observed TVD time-step and the sixth order method has the smallest observed TVD time-step for both $a = 1$ and $a = 5$. The fourth order method behaves consistently for both $a = 1$ and $a = 5$. These results are all consistent with those of the linear test case. Once again, we observe that as the exponent increases we generally have a larger allowable TVD time-step, as we noted above.

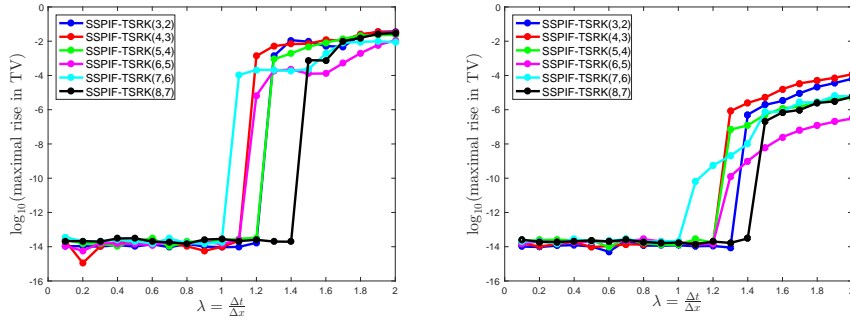


FIG. 6.4. Performance of integrating factor methods for Example 3 for $a = 1$ (left) and $a = 5$ (right).

7. Conclusions. In this work we extended the SSP theory for integrating factor Runge–Kutta methods established in [15] to integrating factor two step Runge–Kutta methods. SSP integrating factor methods are needed for problems that require a nonlinear non-inner product stability property, and where the linear component is severely restricting the time-step. Unlike the case where linear L_2 stability is of interest, implicit or IMEX methods do not alleviate this restriction. However, integrating factor Runge–Kutta methods based on applying a SSP Runge–Kutta method with nondecreasing abscissas to the transformed equation completely alleviate the time-step restriction. SSP Runge–Kutta methods only exist up to fourth order. To extend these results to higher order methods, we considered herein SSP two-step Runge–Kutta methods applied in conjunction with an integrating factor approach. We showed that, just as for the Runge–Kutta case, if the two-step Runge–Kutta methods are SSP and have non-decreasing abscissas then they can be used to time-step the transformed problem. We enhanced the optimization problem in [22] with the condition (5.1) and found SSP two-step Runge–Kutta methods with non-decreasing abscissas that have optimized SSP coefficients (presented in Table 5.1). We tested the resulting SSPIF-TSRK methods on a variety of problems and showed that the new high order methods perform as expected in terms of convergence and the strong

stability property.

Appendix A. Order Conditions. The order conditions, (3.5c) for a k -step SSP Runge-Kutta were derived by Ketcheson and used in [23]. We can write the generalized form for a multi-step Runge-Kutta method as

$$\begin{aligned} y_1^n &= u^n \\ y_i^n &= \sum_{l=1}^k d_{il} u^{n-k+l} + \Delta t \sum_{l=1}^{k-1} \hat{a}_{il} F(u^{n-k+l}) + \Delta t \sum_{j=1}^{i-1} a_{ij} F(y_j^n) \quad 2 \leq i \leq s \\ u^{n+1} &= \sum_{l=1}^k \theta_l u^{n-k+l} + \Delta t \sum_{l=1}^{k-1} \hat{b}_l F(u^{n-k+l}) + \Delta t \sum_{j=1}^s b_j F(y_j^n). \end{aligned}$$

and rewrite it into a matrix form by defining

$$\tilde{\mathbf{D}} = \begin{pmatrix} \mathbf{I}_{(k-1) \times (k-1)} & \mathbf{0}_{1 \times (k-1)} \\ \mathbf{D} & \end{pmatrix} \quad \tilde{\mathbf{A}} = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \hat{\mathbf{A}} & \mathbf{A} \end{pmatrix} \quad \text{and} \quad \tilde{\mathbf{b}} = \begin{pmatrix} \hat{\mathbf{b}} & \mathbf{b} \end{pmatrix}.$$

The method then becomes

$$\begin{aligned} \mathbf{y}^n &= \tilde{\mathbf{D}} \mathbf{u}^n + \Delta t \tilde{\mathbf{A}} \mathbf{f}^n \\ u^{n+1} &= \boldsymbol{\theta}^T \mathbf{u}^n + \Delta t \tilde{\mathbf{b}}^T \mathbf{f}^n \end{aligned}$$

and by letting \mathbf{l} be the vector $\mathbf{l} = (k-1, k-2, \dots, 1, 0)^T$ to compute the abscissas $\mathbf{c} = \tilde{\mathbf{A}} \mathbf{e} - \tilde{\mathbf{D}} \mathbf{l}$ to get the the following expressions

$$\begin{aligned} \tau_\rho &= \frac{1}{\rho!} \left(\mathbf{c}^\rho - \tilde{\mathbf{D}}(-\mathbf{l})^\rho \right) - \frac{1}{(\rho-1)!} \tilde{\mathbf{A}} \mathbf{c}^{\rho-1} \\ \tau_\rho &= \frac{1}{\rho!} (1 - \boldsymbol{\theta}(-\mathbf{l})^\rho) - \frac{1}{(\rho-1)!} \tilde{\mathbf{b}}^T \mathbf{c}^{\rho-1} \end{aligned}$$

Below are the order conditions for methods from first to eighth order. As the order increases the methods must satisfy all previous order conditions as well as the additional ones for each order.

First:

$$\tilde{\mathbf{b}}^T \mathbf{e} = 1 + \boldsymbol{\theta}^T \mathbf{l}$$

Second:

$$\tilde{\mathbf{b}}^T \mathbf{c} = \frac{1 - \boldsymbol{\theta}^T \mathbf{l}^2}{2}.$$

Third:

$$\tilde{\mathbf{b}}^T (\mathbf{c}^2) = \frac{1 + \boldsymbol{\theta}^T \mathbf{l}^3}{3}, \quad \tilde{\mathbf{b}}^T \boldsymbol{\tau}_2 = 0.$$

Fourth:

$$\tilde{\mathbf{b}}^T (\mathbf{c}^3) = \frac{1 - \boldsymbol{\theta}^T \mathbf{l}^4}{4}, \quad \tilde{\mathbf{b}}^T \tilde{\mathbf{A}} \boldsymbol{\tau}_2 = 0, \quad \tilde{\mathbf{b}}^T \mathbf{C} \boldsymbol{\tau}_2 = 0, \quad \tilde{\mathbf{b}}^T \boldsymbol{\tau}_3 = 0$$

Fifth:

$$\tilde{\mathbf{b}}^T(\mathbf{c}^4) = \frac{1 + \boldsymbol{\theta}^T \mathbf{l}^5}{5}, \quad \tilde{\mathbf{b}}^T \tilde{\mathbf{A}} \boldsymbol{\tau}_3 = 0, \quad \tilde{\mathbf{b}}^T \mathbf{C} \boldsymbol{\tau}_3 = 0, \quad \tilde{\mathbf{b}}^T \boldsymbol{\tau}_4 = 0, \quad \boldsymbol{\tau}_2 = 0$$

Sixth:

$$\begin{aligned} \tilde{\mathbf{b}}^T(\mathbf{c}^5) &= \frac{1 - \boldsymbol{\theta}^T \mathbf{l}^6}{6}, & \tilde{\mathbf{b}}^T \tilde{\mathbf{A}} \boldsymbol{\tau}_4 &= 0, & \tilde{\mathbf{b}}^T \mathbf{C} \boldsymbol{\tau}_4 &= 0, & \tilde{\mathbf{b}}^T \boldsymbol{\tau}_5 &= 0, \\ \tilde{\mathbf{b}}^T \tilde{\mathbf{A}}^2 \boldsymbol{\tau}_3 &= 0, & \tilde{\mathbf{b}}^T \tilde{\mathbf{A}} \mathbf{C} \boldsymbol{\tau}_3 &= 0, & \tilde{\mathbf{b}}^T \mathbf{C} \tilde{\mathbf{A}} \boldsymbol{\tau}_3 &= 0, & \tilde{\mathbf{b}}^T \mathbf{C}^2 \boldsymbol{\tau}_3 &= 0 \end{aligned}$$

Seventh:

$$\begin{aligned} \tilde{\mathbf{b}}^T(\mathbf{c}^6) &= \frac{1 + \boldsymbol{\theta}^T \mathbf{l}^7}{7}, & \tilde{\mathbf{b}}^T \tilde{\mathbf{A}} \boldsymbol{\tau}_5 &= 0, & \tilde{\mathbf{b}}^T \mathbf{C} \boldsymbol{\tau}_5 &= 0, \\ \tilde{\mathbf{b}}^T \boldsymbol{\tau}_6 &= 0, & \tilde{\mathbf{b}}^T \tilde{\mathbf{A}}^2 \boldsymbol{\tau}_4 &= 0, & \tilde{\mathbf{b}}^T \tilde{\mathbf{A}} \mathbf{C} \boldsymbol{\tau}_4 &= 0, \\ \tilde{\mathbf{b}}^T \mathbf{C} \tilde{\mathbf{A}} \boldsymbol{\tau}_4 &= 0, & \tilde{\mathbf{b}}^T \mathbf{C}^2 \boldsymbol{\tau}_4 &= 0, & \boldsymbol{\tau}_3 &= 0 \end{aligned}$$

Eighth:

$$\begin{aligned} \tilde{\mathbf{b}}^T(\mathbf{c}^7) &= \frac{1 - \boldsymbol{\theta}^T \mathbf{l}^8}{8}, & \tilde{\mathbf{b}}^T \tilde{\mathbf{A}} \boldsymbol{\tau}_6 &= 0, & \tilde{\mathbf{b}}^T \mathbf{C} \boldsymbol{\tau}_6 &= 0, & \tilde{\mathbf{b}}^T \boldsymbol{\tau}_7 &= 0, \\ \tilde{\mathbf{b}}^T \tilde{\mathbf{A}}^3 \boldsymbol{\tau}_4 &= 0, & \tilde{\mathbf{b}}^T \tilde{\mathbf{A}}^2 \boldsymbol{\tau}_5 &= 0, & \tilde{\mathbf{b}}^T \tilde{\mathbf{A}}^2 \mathbf{C} \boldsymbol{\tau}_4 &= 0, & \tilde{\mathbf{b}}^T \tilde{\mathbf{A}} \mathbf{C} \tilde{\mathbf{A}} \boldsymbol{\tau}_4 &= 0, \\ \tilde{\mathbf{b}}^T \tilde{\mathbf{A}} \mathbf{C} \boldsymbol{\tau}_5 &= 0, & \tilde{\mathbf{b}}^T \tilde{\mathbf{A}} \mathbf{C}^2 \boldsymbol{\tau}_4 &= 0, & \tilde{\mathbf{b}}^T \mathbf{C} \tilde{\mathbf{A}}^2 \boldsymbol{\tau}_4 &= 0, & \tilde{\mathbf{b}}^T \mathbf{C} \tilde{\mathbf{A}} \boldsymbol{\tau}_5 &= 0, \\ \tilde{\mathbf{b}}^T \mathbf{C} \tilde{\mathbf{A}} \mathbf{C} \boldsymbol{\tau}_4 &= 0, & \tilde{\mathbf{b}}^T \mathbf{C}^2 \tilde{\mathbf{A}} \boldsymbol{\tau}_4 &= 0, & \tilde{\mathbf{b}}^T \mathbf{C}^2 \boldsymbol{\tau}_5 &= 0, & \tilde{\mathbf{b}}^T \mathbf{C}^3 \boldsymbol{\tau}_4 &= 0 \end{aligned}$$

Where \mathbf{C} is a diagonal matrix of the abscissas $\mathbf{C} = \text{diag}(\mathbf{c})$ and the exponentiation is considered to be element wise i.e. $\mathbf{c}^3 = (\mathbf{c} \cdot \mathbf{c} \cdot \mathbf{c})$.

Acknowledgment. This publication is based on work supported by AFOSR grant FA9550-18-1-0383. A part of this research is sponsored by the Office of Advanced Scientific Computing Research; US Department of Energy, and was performed at the Oak Ridge National Laboratory, which is managed by UT-Battelle, LLC under Contract no. De-AC05-00OR22725. This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

REFERENCES

- [1] A.H. AL MOHY AND N.J. HIGHAM, *Computing the action of the matrix exponential of a matrix, with an application to exponential integrators*, SIAM Journal on Scientific Computing 33(2) (2011), pp. 488–511.
- [2] E. M. CONSTANTINESCU AND A. SANDU, *Optimal strong-stability-preserving general linear methods*, SIAM Journal of Scientific Computing 32(5) (2010), pp. 3130–3150.
- [3] A.J. CHRISTLIEB, S. GOTTLIEB, Z. GRANT, AND D. C. SEAL, *Explicit Strong Stability Preserving Multistage Two-Derivative Time-Stepping Schemes*, Journal of Scientific Computing 68(3) (2016), pp. 914–942.
- [4] S. CONDE, S. GOTTLIEB, Z. GRANT, J.N. SHADID, *Implicit and Implicit-Explicit Strong Stability Preserving RungeKutta Methods with High Linear Order*, Journal of Scientific Computing 73(2) (2017), pp. 667–690.
- [5] S. COX AND P. MATTHEWS, *Exponential time differencing for stiff systems*, Journal of Computational Physics, 176 (2002), pp. 430–455.
- [6] L. FERRACINA AND M.N. SPIJKER, *Strong stability of singly-diagonally-implicit Runge-Kutta methods*, Applied Numerical Mathematics 58 (2008), pp. 1675–1686.
- [7] S. GAUDREAU, G. RAINWATER, M. TOKMAN, *KIOPS: A fast adaptive Krylov subspace solver for exponential integrators*, Journal of Computational Physics 372 (2018), pp. 236–255.
- [8] S. GOTTLIEB, Z. GRANT, AND L. ISHERWOOD, *Optimized strong stability preserving integrating factor two-step Runge–Kutta methods*. <https://github.com/SSPmethods/SSPIF-TSRK-methods>.
- [9] S. GOTTLIEB, D. I. KETCHESON, AND C.-W. SHU, *Strong Stability Preserving Runge–Kutta and Multistep Time Discretizations*, World Scientific Press, 2011.
- [10] S. GOTTLIEB AND C.-W. SHU, *Total variation diminishing Runge–Kutta methods*, Mathematics of Computation, 67 (1998), pp. 73–85.
- [11] S. GOTTLIEB, C.-W. SHU, AND E. TADMOR, *Strong Stability Preserving High-Order Time Discretization Methods*, SIAM Review, 43 (2001), pp. 89–112.
- [12] Z. GRANT, S. GOTTLIEB, D.C. SEAL, *A Strong Stability Preserving Analysis for Explicit Multistage Two-Derivative Time-Stepping Schemes Based on Taylor Series Conditions*, To appear in Communication on Applied Mathematics and Computation 1(1) (2019), pp. 21–59.
- [13] J.S. HESTHAVEN, *Numerical methods for conservation laws: From analysis to algorithms*, SIAM Publishing, Philadelphia (2017).
- [14] W. HUNSDORFER, S.J. RUUTH AND R.J. SPITERI, *Monotonicity-preserving linear multistep methods*, SIAM Journal on Numerical Analysis 41 (2003), pp. 605–623.
- [15] L. ISHERWOOD, S. GOTTLIEB, Z. GRANT, *Strong Stability Preserving Integrating Factor Runge–Kutta Methods*, SIAM Journal on Numerical Analysis 56(6) (2018), pp. 3276–3307.
- [16] L. ISHERWOOD, S. GOTTLIEB, Z. GRANT, *Downwinding for Preserving Strong Stability in Explicit Integrating Factor Runge–Kutta Methods*, Pure and Applied Mathematics Quarterly 14(1) (2019), pp. 3–25.
- [17] G.-S. JIANG AND C.-W. SHU, *Efficient Implementation of Weighted ENO Schemes*, Journal of Computational Physics 126(1) (1996), pp. 202–228.
- [18] 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.
- [19] D.I. KETCHESON, C.B. MACDONALD AND S. GOTTLIEB, *Optimal implicit strong stability preserving Runge-Kutta methods*, Applied Numerical Mathematics 52 (2009), pp. 373–392.
- [20] D. I. KETCHESON, *Computation of optimal monotonicity preserving general linear methods*, Mathematics of Computation 78 (2009), pp. 1497–1513.
- [21] D. I. KETCHESON, *Step sizes for strong stability preservation with downwind-biased operators*, SIAM Journal on Numerical Analysis 49 (4) (2011), pp. 1649–1660.
- [22] D.I. KETCHESON, S. GOTTLIEB, AND C. B. MACDONALD, *Strong stability preserving two- step Runge-Kutta methods*, SIAM Journal on Numerical Analysis 49 (2012), pp. 2618–2639.
- [23] C. BRESTEN, S. GOTTLIEB, Z. GRANT, D. HIGGS, D.I. KETCHESON, AND A. NEMETH, *Explicit strong stability preserving multistep Runge-Kutta methods*, Mathematics of Computation 86 (2017), pp. 747–769.
- [24] J. F. B. M. KRAAIJEVANGER, *Contractivity of Runge–Kutta methods*, BIT, 31 (1991), pp. 482–528.
- [25] H.W.J. LENFERINK, *Contractivity-preserving implicit linear multistep methods*, Mathematics of Computation 56 (1991), pp. 177–199.
- [26] R. J. LEVEQUE, *Numerical Methods for Conservation Laws*, ETH Lectures in Mathematics Series, Birkhauser-Verlag, (1990).

- [27] J. NIESEN AND W.M. WRIGHT, *Algorithm 919: A Krylov subspace algorithm for evaluating the ϕ -functions appearing in exponential integrators*, ACM Transactions on Mathematical Software 38(3) (2012), pp. 22.
- [28] 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.
- [29] C.-W. SHU, *Total-variation diminishing time discretizations*, SIAM Journal on Scientific Statistical Computing 9 (1988), pp. 1073–1084.
- [30] C.-W. SHU AND S. OSHER, *Efficient implementation of essentially non-oscillatory shock-capturing schemes*, Journal of Computational Physics 77 (1988), pp. 439–471.
- [31] R.B. SIDJE, *EXPOKIT: A software package for computing matrix exponentials*, ACM Transactions on Mathematical Software 24(1) (1998), pp. 130–156.
- [32] M.N. SPIJKER, *Contractivity in the numerical solution of initial value problems*, Numerische Mathematik 42 (1983), pp. 271–290.
- [33] M. SPIJKER, *Stepsize conditions for general monotonicity in numerical initial value problems*, SIAM Journal on Numerical Analysis, 45 (2008), pp. 1226–1245.
- [34] R. J. SPITERI AND S. J. RUUTH, *A new class of optimal high-order strong-stability-preserving time discretization methods*, SIAM Journal on Numerical Analysis, 40 (2002), pp. 469–491.