SIAM J. NUMER. ANAL.
Vol. 56, No. 6, pp. 3276–3307

STRONG STABILITY PRESERVING INTEGRATING FACTOR RUNGE–KUTTA METHODS*

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

Abstract. Strong stability preserving (SSP) Runge–Kutta methods are often desired when evolving in time problems that have two components that have very different time scales. Where the SSP property is needed, it has been shown that implicit and implicit-explicit methods have very restrictive time-steps and are therefore not efficient. For this reason, SSP integrating factor methods may offer an attractive alternative to traditional time-stepping methods for problems with a linear component that is stiff and a nonlinear component that is not. However, the strong stability properties of integrating factor Runge–Kutta methods have not been established. In this work we show that it is possible to define explicit integrating factor Runge–Kutta methods that preserve the desired strong stability properties satisfied by each of the two components when coupled with forward Euler time-stepping, or even given weaker conditions. We define sufficient conditions for explicit integrating factor Runge–Kutta methods to be SSP, namely, that they are based on explicit SSP Runge–Kutta methods with nondecreasing abscissas. We find such methods of up to fourth order and up to ten stages, analyze their SSP coefficients, and prove their optimality in a few cases. We test these methods to demonstrate their convergence and to show that the SSP time-step predicted by the theory is generally sharp and that the nondecreasing abscissa condition is needed in our test cases. Finally, we show that on typical total variation diminishing linear and nonlinear test cases our new explicit SSP integrating factor Runge–Kutta methods out-perform the corresponding explicit SSP Runge–Kutta methods, implicit-explicit SSP Runge–Kutta methods, and some well-known exponential time differencing methods.

Key words. strong stability preserving, Runge–Kutta, integrating factor

AMS subject classification. 65

DOI. 10.1137/17M1143290

1. Introduction.

When numerically solving a hyperbolic partial differential equation (PDE) of the form

$$U_t + f(U)_x = 0, \tag{1}$$

the behavior of the numerical solution depends on properties of the spatial discretization combined with the time discretization. For smooth solutions, stability can be determined by analyzing the L_2 stability properties of the discretization applied to the linear problem. However, when dealing with a nonsmooth solution, stability in the L_2 norm is not sufficient to ensure that the numerical solution will converge [51]. This is due to the presence of oscillations that prevent the approximation from converging uniformly. To ensure that the numerical method does not allow stability-destroying oscillations to form, we require that it satisfy stability properties in, e.g., the maximum norm or in the total variation seminorm.

*Received by the editors August 14, 2017; accepted for publication (in revised form) August 17, 2018; published electronically November 27, 2018.
http://www.siam.org/journals/sinum/56-6/M114329.html
Funding: The work of the authors was supported by AFOSR grant FA9550-15-1-0235.
†Mathematics Department, University of Massachusetts Dartmouth, North Dartmouth, MA 02747 (lisherwood@umassd.edu, sgottlieb@umassd.edu).
‡Department of Computational and Applied Mathematics, Oak Ridge National Laboratory, Oak Ridge, TN 37830 (zgrant@umassd.edu).

Thus, to prove stability of numerical methods for nonlinear hyperbolic problems with discontinuous solutions, we need to analyze the nonlinear, non-inner-product stability properties of a highly nonlinear, complex spatial discretization combined with a high order time discretization. This is a difficult, sometimes untenable task. Instead, a method-of-lines formulation is generally followed, and a spatial discretization is developed that satisfies nonlinear, non-inner-product stability properties when coupled with the forward Euler time-stepping method. In practice, higher order time discretizations are needed. Strong stability preserving (SSP) time discretizations were created [67, 68] to allow the nonlinear non-inner-product stability properties of the spatial discretizations coupled with forward Euler to be immediately extended to all SSP higher order time discretizations.

1.1. Background. Linear stability theory is an indispensable tool used to establish the convergence of a numerical method when numerically solving PDEs. Linear stability is necessary and sufficient for convergence of a consistent linear numerical method when the PDE is linear [74]. When the PDE is nonlinear, if a numerical method is consistent and its *linearization* is L_2 stable and adequately dissipative, then convergence can be proved for sufficiently smooth problems [73]. However, discontinuous solutions often arise in the solution of hyperbolic conservation laws of the form (1), and when the solution has discontinuities, linear stability theory is no longer sufficient for convergence.

A famous example of this is that when the linearly L_2 stable second order Lax–Wendroff scheme is applied to Burgers's equation, the method is nonlinearly unstable near stagnation points and does not converge [56]. This example demonstrates that to obtain convergence for a nonlinear PDE with a discontinuous solution, some kind of nonlinear stability is necessary in order to guarantee convergence.

Even in the linear case, L_2 stability is not enough for uniform convergence when dealing with solutions with discontinuities. For example, although a second order Lax–Wendroff scheme is strongly stable in the L_2 norm, when it is applied to a linear advection equation ($f(U) = aU$ in (1) above) with a step function initial condition, the numerical solution will always have an overshoot or undershoot near the discontinuity [49]. Furthermore, this is true not only for the second order Lax–Wendroff but indeed *any* linear, consistent, finite difference scheme of at least second order accuracy will develop an overshoot or undershoot that prevents uniform convergence [49].

These two examples show that L_2 linear stability is not the relevant property when we desire well-behaved numerical solutions of hyperbolic PDEs with discontinuous solutions. However, if we can prevent oscillations from forming by requiring stability in the maximum norm or the total variation seminorm, we can obtain uniform convergence [51]. Consequently, a tremendous amount of effort has been placed on the development of high order spatial discretizations which, when coupled with the forward Euler time-stepping method, have the desired nonlinear stability properties for approximating discontinuous solutions of hyperbolic PDEs (see, e.g., [30, 60, 76, 12, 46, 77, 53]).

However, for actual computation, higher order time discretizations are usually needed. There is no guarantee that a spatial discretization that is strongly stable in some desired norm or seminorm (e.g., L_∞ or total variation) for a nonlinear problem under forward Euler integration will possess the same nonlinear stability property when coupled with a linearly stable higher order time discretization. SSP methods were created to address this need.

1.2. SSP methods. Explicit SSP Runge–Kutta methods were first developed in [67, 68] for use in conjunction with total variation diminishing (TVD) spatial discretizations for hyperbolic conservation laws (1) with discontinuous solutions. These spatial discretizations of $f(U)_x$ ensure that when the resulting semidiscretized system of ordinary differential equations (ODEs),

$$u_t = F(u), \tag{2}$$

is evolved in time using the *forward Euler method*, a strong stability property

$$\|u^{n+1}\| = \|u^n + \Delta t F(u^n)\| \leq \|u^n\| \tag{3}$$

is satisfied, under some step-size restriction

$$0 \leq \Delta t \leq \Delta t_{\mathrm{FE}}. \tag{4}$$

These TVD spatial discretizations are designed to satisfy the strong stability property

$$\|u^{n+1}\| \leq \|u^n\| \tag{5}$$

when coupled with the forward Euler time discretization. However, in actuality a higher order time integrator is desired, for both accuracy and linear stability reasons. If we can rewrite a higher order time discretization as a convex combination of forward Euler steps, we can ensure that any convex functional property (5) 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, we can write an s-stage explicit Runge–Kutta method in Shu–Osher form [27]:

$$\begin{aligned} u^{(0)} &= u^n, \\ u^{(i)} &= \sum_{j=0}^{i-1} \left(\alpha_{i,j} u^{(j)} + \Delta t \beta_{i,j} F(u^{(j)}) \right), \quad i = 1, \ldots, s, \\ u^{n+1} &= u^{(s)}. \end{aligned} \tag{6}$$

Note that for consistency, we must have $\sum_{j=0}^{i-1} \alpha_{i,j} = 1$. If all the coefficients $\alpha_{i,j}$ and $\beta_{i,j}$ are nonnegative, and a given $\alpha_{i,j}$ is zero only if its corresponding $\beta_{i,j}$ is zero, then each stage can be rearranged into a convex combination of forward Euler steps

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

where the final inequality follows from (3) and (4), provided that the time-step satisfies

$$\Delta t \leq \min_{i,j} \frac{\alpha_{i,j}}{\beta_{i,j}} \Delta t_{\mathrm{FE}}. \tag{7}$$

(Note that if any of the β's are equal to zero, the corresponding ratio is considered infinite.)

It is clear from this example that whenever we can rewrite an explicit Runge–Kutta method as a convex combination of forward Euler steps, the forward Euler condition (3) will be *preserved* by the higher order time discretizations, under the

time-step restriction $\Delta t \leq \mathcal{C}\Delta t_{\text{FE}}$ where $\mathcal{C} = \min_{i,j} \frac{\alpha_{i,j}}{\beta_{i,j}}$. As long as $\mathcal{C} > 0$, the method is called SSP with *SSP coefficient* $\mathcal{C}$ [67].

This form also ensures internal stage strong stability, i.e., $\|u^{(i+1)}\| \leq \|u^{(i)}\|$ at each stage i of the time-stepping, under the same time-step restriction. The internal stage monotonicity property is important in simulations that involve pressure, density, or water height, in which a negative value even at the intermediate stage is not acceptable as it may not allow the simulation to proceed [31]. Positivity preserving limiters that prevent this from occurring are typically designed and proved for use with a forward Euler time-stepping and thus naturally extend to SSP time-stepping methods with internal stage monotonicity [35, 86, 88, 89].

We observe that in the original papers [67, 68], the term $\|\cdot\|$ in (3) above represented the total variation seminorm. In general, though, the strong stability preservation property holds for any seminorm, norm, or convex functional, as determined by the design of the spatial discretization. The only requirements are that the forward Euler condition (3) holds and that the time discretization can be decomposed into a convex combination of forward Euler steps with $\mathcal{C} > 0$, as above.

Clearly, this convex combination condition is a sufficient condition for strong stability preservation. In fact, it has been shown that it is also necessary for strong stability preservation [27, 44, 71]. This means that if a method cannot be decomposed into a convex combination of forward Euler steps, then we can always find some ODE with some initial condition such that the forward Euler condition is satisfied, but the method does not satisfy the strong stability condition for any positive time-step [27]. The observation that there are significant connections between SSP theory and contractivity theory [20, 21, 33, 34] has led to many results on SSP methods as well as development of new optimal and efficient SSP methods [22, 40, 39].

It is not always possible to decompose a method into convex combinations of forward Euler steps where $\mathcal{C} > 0$. In fact in [44, 65] it was shown that explicit SSP Runge–Kutta methods cannot exist for order $p > 4$. Furthermore, the value of $\mathcal{C}$ determines in large part what the size of an allowable time-step will be, and so we seek methods that have the largest possible SSP coefficient. Of course, a more important quantity is the total cost of the time evolution, which is related to the allowable time-step relative to the number of function evaluations at each time-step. To allow us to compare the efficiency of explicit methods of a given order, we define the *effective SSP coefficient* $\mathcal{C}_{\text{eff}} = \frac{\mathcal{C}}{s}$ where s is the number of stages (typically the number of function evaluations). Unfortunately, all explicit s-stage Runge–Kutta methods have an SSP bound $\mathcal{C} \leq s$, and therefore $\mathcal{C}_{\text{eff}} \leq 1$ [27]. Even worse, this upper bound is not usually attained. However, many efficient explicit SSP Runge–Kutta methods have been found, as we discuss in section 2.

For smooth problems it is usually the case that implicit methods (or implicit treatment of stiff terms) can remove the time-step restriction needed for stability. In such cases, where the time-step is limited by a linear stability requirement or by an inner-product norm nonlinear stability, there are well-known classes of implicit methods (e.g., A-stable, L-stable, B-stable) that allow the use of arbitrarily large time-steps. This is not the case for time discretizations of order $p > 1$ where the time-step is limited by SSP considerations. For first order ($p = 1$) it can be shown that if the spatial discretization is strongly stable in some norm under forward Euler time integration, then the fully discrete solution will also be strongly stable, in the same norm, for the implicit Euler method, without any time-step restriction [36, 33]. However, any general linear method of order $p > 1$ can only be SSP under some finite

time-step [70]. Worse yet, the time-step restrictions for implicit and implicit-explicit (IMEX) SSP methods are not dramatically larger than those for explicit methods, with the observed bound of $\mathcal{C} \leq 2s$, and therefore $\mathcal{C}_{\text{eff}} \leq 2$ [50, 22, 40, 15].

SSP methods are widely used in the solution of hyperbolic PDEs with discontinuous solutions, where linear L_2 stability is not enough to ensure convergence. They have been paired with many spatial discretizations that were specially designed for hyperbolic PDEs. Some of these spatial discretization approaches involve numerical methods that directly incorporate a nonoscillatory approach, such as ENO [8, 17, 2] and weighted ENO (WENO) [4, 9, 79, 19, 47, 3, 84, 61] methods, in a finite difference or finite element setting. These methods are typically designed and their properties proved with a forward Euler time-stepping, and they rely on the SSP time discretization mechanism to extend to higher order in time. Other approaches include limiters applied to finite difference or discontinuous Galerkin methods [13]. These limiters may enforce a TVD [76] or total variation bounded [66, 12] solution or be used to ensure the numerical solution is maximum principle preserving [85, 87] or positivity preserving [86, 88, 89]. In these cases, SSP methods have proven particularly popular because for each new limiter proposed proofs of the desired property are generated for the method coupled with the first order forward Euler time discretization. Once again, the SSP time discretization mechanism is needed to extend the results on these limiters to higher order in time.

For spectral and pseudospectral methods, Reddy and Trefethen [64] explored the fact that eigenvalue stability is insufficient to ensure stable simulations and discussed the difficulties in a fully discrete stability analysis. Gottlieb and Tadmor [25] proved stability of spectral approximations for the forward Euler method; this is the approach commonly used in spectral methods simulations [32]. The work of Levy and Tadmor [52] shows the challenges of going from analysis of semidiscrete stability to fully discrete stability, as they analyzed the strong stability of Runge–Kutta schemes (including the first order forward Euler method) for linear problems. Gottlieb, Shu, and Tadmor [29] later showed that the stability of the forward Euler method for any linear coercive approximations [52] could be easily extended using SSP analysis to a much larger class of Runge–Kutta methods. Furthermore, the SSP approach guarantees stability for much larger CFL numbers than the stability analysis in [52]. When using spectral methods on nonlinear problems, regularization using filtering [24, 32] or specially designed viscosity [55, 78, 54, 32] is needed for the simulation to be stable. The stabilization properties of these techniques are proven, usually, first on the semidiscrete form, and only then for the fully discrete form, in conjunction with a forward Euler step; the SSP time-stepping mechanism allows it to be preserved for higher order methods [32, 31].

In addition, SSP time-stepping methods have been paired with other spatial discretizations, including level set methods [63, 8, 18, 10, 14, 37], spectral finite volume methods [75, 11], and spectral difference methods [81, 82]. Examples of application areas where SSP time-stepping methods have been used include compressible flow [81], incompressible flow [62], viscous flow [75], two-phase flow [8, 4], relativistic flow [17, 2, 84], cosmological hydrodynamics [19], magnetohydrodynamics [3], radiation hydrodynamics [57], two-species plasma flow [47], atmospheric transport [11], large-eddy simulation [61], Maxwell's equations [13], semiconductor devices [9], lithotripsy [79], geometrical optics [14], and Schrodinger equations [10, 37].

1.3. SSP integrating factor Runge–Kutta methods. In this work, we are interested in a semidiscretized problem of the form

$$u_t = Lu + N(u),$$

where each component satisfies a forward Euler condition

$$\|u^n + \Delta t L u^n\| \leq \|u^n\| \quad \text{for} \quad \Delta t \leq \Delta \tilde{t}_{\mathrm{FE}}$$

and

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

(where $\|\cdot\|$ is some convex functional needed for nonlinear, non-inner-product stability). In the cases of interest, $\Delta \tilde{t}_{\mathrm{FE}} << \Delta t_{\mathrm{FE}}$, so that L is a linear operator that significantly restricts the allowable time-step. As mentioned above, when the SSP properties are a concern, using an IMEX scheme does not significantly alleviate the allowable time-step [15]. This motivates our investigation of integrating factor methods, where the linear component Lu is handled exactly, and then the time-step restriction is, at worst, $\Delta t \leq \Delta t_{\mathrm{FE}}$ coming from the the nonlinear component $N(u)$. In this work, we discuss the conditions under which this process guarantees that the strong stability property (5) is preserved. In particular, we show that if we step the transformed problem forward using an SSP Runge–Kutta method where the abscissas (i.e., the time-levels approximated by each stage) are nondecreasing, we obtain a method that preserves the desired strong stability property. It is important to note that the efficient use of the proposed methods will depend heavily on the cost of computation of the matrix exponential. This area of research, although outside the scope of this work, will be crucial for the practical implementation of the SSP integrating factor Runge–Kutta methods described in this paper.

The paper is structured as follows. In section 2 we review some known optimal and optimized explicit SSP Runge–Kutta methods of orders $p \leq 4$ and give the SSP coefficients and effective SSP coefficients of the optimal methods in this class. In section 3 we describe explicit integrating factor (also known as Lawson-type) Runge–Kutta (IFRK) methods [48]. We prove that when IFRK methods are based on explicit SSP Runge–Kutta methods with *nondecreasing* abscissas, they preserve the strong stability property. We also show an example that demonstrates that when using an IFRK method based on an explicit SSP Runge–Kutta method that has decreasing abscissas, the SSP property is violated. In section 4 we formulate the optimization problem that will enable us to find optimized explicit SSP Runge–Kutta methods that have nondecreasing abscissas, and in section 5 we present some optimized methods in this class and their SSP coefficients and effective SSP coefficients. We also prove the optimality of one of the methods in this class. In section 6 we present numerical examples that show how our explicit SSP integrating factor Runge–Kutta (eSSPIFRK) methods perform on typical test cases compared to explicit, IMEX, and exponential time differencing (ETD) methods. We also compare the linear stability properties of these methods.

In section 7 we conclude that the newly developed SSP theory for IFRK methods provides a provable bound on the allowable time-step (which is often sharp in practice) for preservation of the nonlinear non-inner-product stability properties. Moreover, the newly developed methods demonstrate a significantly larger allowable SSP step-size than standard methods including the ETD methods of Cox and Matthews [16], SSP IMEX methods [15], standard explicit SSP Runge–Kutta methods [27], and the Runge–Kutta methods of Kinnmark and Gray [43].

Note: The following acronyms and notations are used in this work:

IFRK	integrating factor Runge–Kutta method.
SSP	strong stability preserving.
eSSPRK	explicit SSP Runge–Kutta method.
eSSPRK$^+$	explicit SSP Runge–Kutta method with nondecreasing abscissas.
eSSPKG	explicit SSP Kinnmark and Gray method.
eSSPKG$^+$	explicit SSP Kinnmark and Gray method with nondecreasing abscissas.
eSSPIFRK	explicit SSP integrating factor Runge–Kutta method.
(s,p)	number of stages s and order p.
IMEX	implicit-explicit additive method.
ETD	exponential time differencing methods.

2. A review of explicit SSP Runge–Kutta methods. SSP Runge–Kutta methods guarantee the strong stability (in any norm, seminorm, or convex functional) of the numerical solution of any ODE provided *only* that the forward Euler condition (3) is satisfied under a time-step restriction (4). This requirement leads to severe restrictions on the allowable order of SSP methods, and the allowable time-step $\Delta t \leq \mathcal{C}\Delta t_{\text{FE}}$. These methods have been extensively studied, e.g., in [20, 21, 22, 27, 28, 29, 33, 34, 36, 39, 40, 41, 42, 45, 65]. In this section, we review some popular and efficient explicit SSP Runge–Kutta methods and present the SSP coefficients of optimized methods of up to ten stages and fourth order.

In the original papers on SSP time-stepping methods (there called TVD time-stepping) [67, 68], the authors presented the first explicit SSP Runge–Kutta methods. These methods were second and third order with SSP coefficient $\mathcal{C} = 1$ ($\mathcal{C}_{\text{eff}} = \frac{1}{2}$ and $\mathcal{C}_{\text{eff}} = \frac{1}{3}$, respectively) and were proven optimal [28]. We use the notation eSSPRK(s,p) to denote an explicit SSP Runge–Kutta method with s stages and of order p.

eSSPRK(2,2).

$$\begin{aligned} u^{(1)} &= u^n + \Delta t F(u^n), \\ u^{n+1} &= \frac{1}{2}u^n + \frac{1}{2}\left(u^{(1)} + \Delta t F(u^{(1)})\right). \end{aligned} \tag{8}$$

eSSPRK(3,3).

$$\begin{aligned} u^{(1)} &= u^n + \Delta t F(u^n), \\ u^{(2)} &= \frac{3}{4}u^n + \frac{1}{4}\left(u^{(1)} + \Delta t F(u^{(1)})\right), \\ u^{n+1} &= \frac{1}{3}u^n + \frac{2}{3}\left(u^{(2)} + \Delta t F(u^{(2)})\right). \end{aligned} \tag{9}$$

Method (9) has been extensively used and is known as the Shu–Osher method.

No four-stage fourth order explicit Runge–Kutta methods exist with a positive SSP coefficient [28, 65]. However, fourth order methods with more than four stages ($s > p$) do exist.

eSSPRK(5,4). Found by Spiteri and Ruuth [72].

$$
\begin{aligned}
u^{(1)} &= u^n + 0.391752226571890\Delta t F(u^n),\\
u^{(2)} &= 0.444370493651235u^n + 0.555629506348765u^{(1)} + 0.368410593050371\Delta t F(u^{(1)}),\\
u^{(3)} &= 0.620101851488403u^n + 0.379898148511597u^{(2)} + 0.251891774271694\Delta t F(u^{(2)}),\\
u^{(4)} &= 0.178079954393132u^n + 0.821920045606868u^{(3)} + 0.544974750228521\Delta t F(u^{(3)}),\\
u^{n+1} &= 0.517231671970585u^{(2)} + 0.096059710526147u^{(3)}\\
&\quad + 0.063692468666290\Delta t F(u^{(3)}) + 0.386708617503268u^{(4)}\\
&\quad + 0.226007483236906\Delta t F(u^{(4)});
\end{aligned}
$$

has $\mathcal{C} = 1.508$ ($\mathcal{C}_{\text{eff}} = 0.302$).

eSSPRK(10,4). Found by Ketcheson [39]; has a low-storage formulation:

$$
\begin{aligned}
u^{(1)} &= u^n + \frac{1}{6}\Delta t F(u^n),\\
u^{(i+1)} &= u^{(i)} + \frac{1}{6}\Delta t F(u^{(i)}), \quad i = 1, 2, 3,\\
u^{(5)} &= \frac{3}{5}u^n + \frac{2}{5}\left(u^{(4)} + \frac{1}{6}\Delta t F(u^{(4)})\right),\\
u^{(i+1)} &= u^{(i)} + \frac{1}{6}\Delta t F(u^{(i)}), \quad i = 5, 6, 7, 8,\\
u^{n+1} &= \frac{1}{25}u^n + \frac{9}{25}\left(u^{(4)} + \frac{1}{6}\Delta t F(u^{(4)})\right) + \frac{3}{5}\left(u^{(9)} + \frac{1}{6}\Delta t F(u^{(9)})\right);
\end{aligned}
$$

has $\mathcal{C} = 6$ ($\mathcal{C}_{\text{eff}} = 0.6$).

In Table 1 we present the SSP coefficients of optimized explicit SSP Runge–Kutta methods of up to $s = 10$ stages and order $p = 4$, and in Table 2 we present the corresponding effective SSP coefficients. Unfortunately, no methods of order $p \geq 5$ with positive SSP coefficients can exist [44, 65].

TABLE 1
SSP coefficients of the optimized eSSPRK (s,p) methods [27].

s \ p	2	3	4
1	-	-	-
2	1.0000	-	-
3	2.0000	1.0000	-
4	3.0000	2.0000	-
5	4.0000	2.6506	1.5082
6	5.0000	3.5184	2.2945
7	6.0000	4.2879	3.3209
8	7.0000	5.1071	4.1459
9	8.0000	6.0000	4.9142
10	9.0000	6.7853	6.0000

TABLE 2
Effective SSP coefficients of the optimized eSSPRK(s,p) methods [27].

s \ p	2	3	4
1	-	-	-
2	0.5000	-	-
3	0.6667	0.3333	-
4	0.7500	0.5000	-
5	0.8000	0.5301	0.3016
6	0.8333	0.5864	0.3824
7	0.8571	0.6126	0.4744
8	0.8750	0.6384	0.5182
9	0.8889	0.6667	0.5460
10	0.9000	0.6785	0.6000

3. Explicit SSP Runge–Kutta schemes for use with integrating factor methods. We consider a problem of the form

$$u_t = Lu + N(u) \tag{10}$$

with a linear constant coefficient component Lu and a nonlinear component $N(u)$. The case we are interested in is when some strong stability condition is known for the forward Euler step of the nonlinear component $N(u)$

$$\|u^n + \Delta t N(u^n)\| \leq \|u^n\| \qquad \text{for} \qquad \Delta t \leq \Delta t_{\text{FE}}, \tag{11}$$

while taking a forward Euler step using the linear component Lu results in the strong stability condition

$$\|u^n + \Delta t L u^n\| \leq \|u^n\| \qquad \text{for} \qquad \Delta t \leq \tilde{\Delta} t_{\text{FE}}, \tag{12}$$

where $\tilde{\Delta} t_{\text{FE}} << \Delta t_{\text{FE}}$. In such cases, stepping forward using an explicit SSP Runge–Kutta method, or even an implicit or an IMEX SSP Runge–Kutta method, will result in severe constraints on the allowable time-step [50, 40, 22, 27, 15].

An alternative methodology that may alleviate the restriction on the allowable time-step involves solving the linear part exactly using an integrating factor approach:

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

A transformation of variables $w = e^{-Lt} u$ gives the ODE system

$$w_t = e^{-Lt} N(e^{Lt} w) = G(w), \tag{13}$$

which can then be evolved forward in time using, for example, an explicit Runge–Kutta method of the form (6). For each stage $u^{(i)}$, which corresponds to the solution at time $t_i = t^n + c_i \Delta t$ (where each c_i is the abscissa of the method at the ith stage), the corresponding IFRK method becomes

$$e^{-Lt_i} u^{(i)} = \sum_{j=0}^{i-1} \left(\alpha_{i,j} e^{-Lt_j} u^{(j)} + \Delta t \beta_{i,j} e^{-Lt_j} N(u^{(j)}) \right)$$

or

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

In the following results we establish the SSP properties of this approach.

THEOREM 1. *If a linear operator L satisfies* (12) *for some value of $\tilde{\Delta} t_{\text{FE}} > 0$, then*

$$\|e^{\tau L} u^n\| \leq \|u^n\| \quad \forall\, \tau \geq 0. \tag{14}$$

Proof. The Taylor series expansion of e^z can be written as

$$e^z = \sum_{j=0}^{\infty} \gamma_j(r) \left(1 + \frac{z}{r}\right)^j \quad \text{where } \gamma_j = \frac{r^j}{j!} e^{-r},$$

where the coefficients γ_j are clearly nonnegative for all values of $r \geq 0$. These coefficients sum to one because

$$\sum_{j=0}^{\infty} \gamma_j = \sum_{j=0}^{\infty} \frac{r^j}{j!} e^{-r} = e^{-r} \sum_{j=0}^{\infty} \frac{r^j}{j!} = e^{-r} e^{r} = 1.$$

Using this we can show that $e^{\tau L} u^n$ can be written as a convex combination of forward Euler steps with a modified time-step $\frac{\tau}{r}$, so that

$$\begin{aligned} \|e^{\tau L} u^n\| &= \left\| \sum_{j=0}^{\infty} \gamma_j(r) \left(1 + \frac{\tau}{r} L\right)^j u^n \right\| \\ &\leq \sum_{j=0}^{\infty} \gamma_j(r) \left\| \left(1 + \frac{\tau}{r} L\right)^j u^n \right\| \\ &\leq \sum_{j=0}^{\infty} \gamma_j(r) \|u^n\| \leq \|u^n\| \quad \text{for any } 0 \leq \tau \leq r\tilde{\Delta} t_{\mathrm{FE}}. \end{aligned}$$

As this is true for any value of $r \geq 0$, we have

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

Note that a negative value of τ is not allowed here. ∎

Remark 1. The theorem above deals with the case that (12) is satisfied for some value of $\tilde{\Delta} t_{\mathrm{FE}} > 0$. However, requiring L to satisfy only condition (14) is sufficient for the IFRK method to be SSP. In the following results, therefore, we only require the condition (14), which is a weaker condition than (12).

COROLLARY 1. *Given a linear operator L that satisfies* (14) *and a (possibly nonlinear) operator $N(u)$ that satisfies* (11) *for some value of $\Delta t_{FE} > 0$, we have*

$$\|e^{\tau L}(u^n + \Delta t N(u^n))\| \leq \|u^n\| \quad \forall \Delta t \leq \Delta t_{\mathrm{FE}}, \quad \textit{provided that } \tau \geq 0. \tag{15}$$

Proof. Separate the term $e^{\tau L}(u^n + \Delta t N(u^n))$ to two steps:

$$\begin{aligned} y^{(1)} &= u^n + \Delta t N(u^n), \\ y^{(2)} &= e^{\tau L} y^{(1)}. \end{aligned}$$

Clearly, from (14) we have

$$\|y^{(2)}\| = \|e^{\tau L} y^{(1)}\| \leq \|y^{(1)}\|$$

for any $\tau \geq 0$. Now, from (11) we also have

$$\|y^{(1)}\| = \|u^n + \Delta t N(u^n)\| \leq \|u^n\| \quad \forall \Delta t \leq \Delta t_{\mathrm{FE}}.$$

Putting these two together we obtain the desired result. ∎

The following theorem describes the conditions under which an IFRK method is SSP.

THEOREM 2. *Given a linear operator L that satisfies* (14) *and a (possibly nonlinear) operator $N(u)$ that satisfies* (11) *for some value of $\Delta t_{FE} > 0$, and a IFRK method of the form*

$$\begin{aligned} u^{(0)} &= u^n, \\ u^{(i)} &= \sum_{j=0}^{i-1} e^{L(c_i-c_j)\Delta t}\left(\alpha_{i,j}u^{(j)} + \Delta t\beta_{i,j}N(u^{(j)})\right), \quad i = 1,\dots,s, \\ u^{n+1} &= u^{(s)}, \end{aligned} \tag{16}$$

where $0 = c_1 \le c_2 \le \ldots \le c_s$, then u^{n+1} obtained from (16) *satisfies*

$$\|u^{n+1}\| \le \|u^n\| \quad \forall \Delta t \le \mathcal{C}\Delta t_{\mathrm{FE}}. \tag{17}$$

Proof. We observe that for each stage of (16)

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

where the last inequality follows from Corollary 1, as long as $c_i - c_j \ge 0$ and $\Delta t\frac{\beta_{i,j}}{\alpha_{i,j}} \le \Delta t_{\mathrm{FE}}$. This establishes the result of the theorem. Furthermore, this proof ensures that these methods have internal stage strong stability as well, i.e., $\|u^{(i+1)}\| \le \|u^{(i)}\|$ at each stage i of the time-stepping, under the same time-step restriction. ∎

Remark 2. It is possible to preserve the strong stability property even with decreasing abscissas, provided that whenever the term $c_i - c_j$ is negative, the operator L is replaced by an operator $\tilde{L}$ that satisfies the condition

$$\|e^{-\tau\tilde{L}}u^n\| \le \|u^n\| \quad \forall\, \tau \ge 0.$$

For hyperbolic PDEs, this is accomplished by using the spatial discretization that is stable for the downwinded analog of the operator. This approach is similar to the one employed in the classical SSP literature, where negative coefficients $\beta_{i,j}$ may be allowed if the corresponding operator is replaced by a downwinded operator [28, 29, 42].

Example. To demonstrate the practical importance of this theorem, consider the PDE

$$u_t + 10u_x + \left(\frac{1}{2}u^2\right)_x = 0, \qquad u(0,x) = \begin{cases} 1 & \text{if } 0 \le x \le 1/2, \\ 0 & \text{if } x > 1/2 \end{cases}$$

on the domain $[0,1]$ with periodic boundary conditions. We discretize the spatial grid with 400 points and use a first order upwind difference to semidiscretize the linear term $Lu \approx -10u_x$ and a fifth order WENO finite difference for the nonlinear terms

$N(u) \approx -\left(\frac{1}{2}u^2\right)_x$. For the time discretization, we use the integrating factor method based on the explicit eSSPRK(3,3) Shu–Osher method (9):

$$
\begin{aligned}
u^{(1)} &= e^{L\Delta t}u^n + e^{L\Delta t}\Delta t N(u^n),\\
u^{(2)} &= \frac{3}{4}e^{\frac{1}{2}L\Delta t}u^n + \frac{1}{4}e^{-\frac{1}{2}L\Delta t}u^{(1)} + \frac{1}{4}e^{-\frac{1}{2}L\Delta t}\Delta t N(u^{(1)}),\\
u^{n+1} &= \frac{1}{3}e^{L\Delta t}u^n + \frac{2}{3}e^{\frac{1}{2}L\Delta t}u^{(2)} + \frac{2}{3}e^{\frac{1}{2}L\Delta t}\Delta t N(u^{(2)}).
\end{aligned} \tag{18}
$$

The appearance of exponentials with negative exponents is due to the fact that (9) has decreasing abscissas. For comparison we also use an IFRK(3,3) method based on an eSSPRK$^+$, denoted eSSPRK$^+(3,3)$ (which will be presented in (22)):

$$
\begin{aligned}
u^{(1)} &= \frac{1}{2}e^{\frac{2}{3}\Delta tL}u^n + \frac{1}{2}e^{\frac{2}{3}\Delta tL}\left(u^n + \frac{4}{3}\Delta t N(u^n)\right),\\
u^{(2)} &= \frac{2}{3}e^{\frac{2}{3}\Delta tL}u^n + \frac{1}{3}\left(u^{(1)} + \frac{4}{3}\Delta t N(u^{(1)})\right),\\
u^{n+1} &= \frac{59}{128}e^{\Delta tL}u^n + \frac{15}{128}e^{\Delta tL}\left(u^n + \frac{4}{3}\Delta t N(u^n)\right)\\
&\quad + \frac{27}{64}e^{\frac{1}{3}\Delta tL}\left(u^{(2)} + \frac{4}{3}\Delta t N(u^{(2)})\right).
\end{aligned} \tag{19}
$$

The eSSPRK$^+(3,3)$ method this integrating factor is based on has SSP coefficient $\mathcal{C} = \frac{3}{4}$, which is smaller than the $\mathcal{C} = 1$ of the Shu–Osher method (9), due to the restriction on the nondecreasing abscissas. Theorem 2 above tells us that the IFRK method (19) will be SSP while the IFRK method (18) based on the Shu–Osher method (9) will not be.

We selected different values of Δt and used each one to evolve the solution 25 time-steps using the IFRK methods (18) and (19). We calculated the maximal rise in total variation over each stage for 25 time-steps. In Figure 1 we show the log_{10} of the maximal rise in total variation versus the value of $\lambda = \frac{\Delta t}{\Delta x}$ of the evolution using (18) (in blue) and using (19) (in red). We observe that the results from method (18)

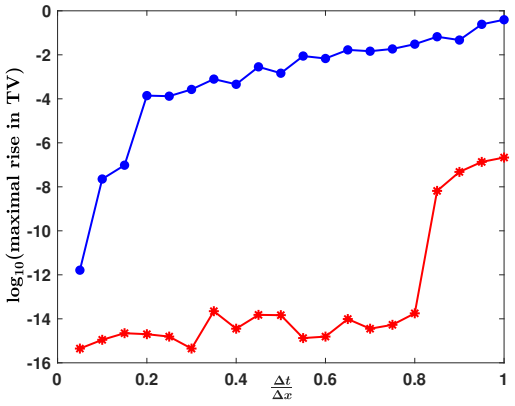

FIG. 1. *Total variation behavior of the evolution over* 25 *time-steps using the integrating factor methods* (18) *(blue dots) and* (19) *(red stars).*

have a large maximal rise in total variation even for very small values of λ, while the results from (19) maintain a small maximal rise in total variation up to $\lambda \approx 0.8$.

This example clearly illustrates that basing an IFRK method on an explicit SSP Runge–Kutta method is not enough to ensure the preservation of a strong stability property. In this case, we *must* use the nondecreasing abscissa condition in Theorem 2 to ensure that the strong stability property is preserved.

4. Formulating the optimization problem. Our aim is to find eSSPRK(s,p) methods to evolve an equation of the form (2) which have nondecreasing abscissas and largest SSP coefficient $\mathcal{C}$. We denote these methods eSSPRK$^+$(s,p). These methods can then be used to produce an integrating factor method (16) that has a guarantee of nonlinear stability, as we showed in section 3 above. Following the approach developed by Ketcheson [39], we formulate an optimization problem similar to the one used for explicit SSP Runge–Kutta methods [27] but with one additional constraint.

Although the SSP coefficient $\mathcal{C}$ is most easily seen in Shu–Osher form, constructing the optimization problem is easier when the method is written in Butcher form:

$$u^{(i)} = u^n + \Delta t \sum_{j=1}^{i-1} a_{ij} F(u^{(j)}) \quad (1 \le i \le s), \tag{20}$$

$$u^{n+1} = u^n + \Delta t \sum_{j=1}^{s} b_j F(u^{(j)}).$$

We can put all the a_{ij} values into a matrix $\mathbf{A}$ and all the b_j into a vector $\mathbf{b}$. Then we define the vector of abscissas $\mathbf{c} = \mathbf{A}\mathbf{e}$, where $\mathbf{e}$ is the vector of ones of the appropriate length. We rewrite (20) in vector form,

$$Y = \mathbf{e}u^n + \Delta t \mathbf{S} F(Y),$$

where $\mathbf{S}$ is the square matrix defined by

$$\mathbf{S} = \begin{pmatrix} \mathbf{A} & 0 \\ \mathbf{b}^T & 0 \end{pmatrix}.$$

We add $r\mathbf{S}Y$ to each side to obtain

$$(I + r\mathbf{S})\, Y = \mathbf{e}u^n + r\mathbf{S}\left(Y + \frac{\Delta t}{r} F(Y)\right) \quad \Rightarrow \quad Y = \mathbf{R}(\mathbf{e}u^n) + \mathbf{P}\left(Y + \frac{\Delta t}{r} F(Y)\right)$$

for $\mathbf{R} = (I + r\mathbf{S})^{-1}$ and $\mathbf{P} = r\mathbf{R}\mathbf{S}$. Clearly, if $\mathbf{R}\mathbf{e}$ and $\mathbf{P}$ have all nonnegative components, we have a convex combination of forward Euler steps. Therefore, the strong stability property (5) will be preserved under the modified time-step restriction $\Delta t \le r \Delta t_{\mathrm{FE}}$.

As discussed in [27], the goal is to maximize the value of r subject to the constraints

$$(I + r\mathbf{S})^{-1} \mathbf{e} \ge 0, \tag{21a}$$

$$r\,(I + r\mathbf{S})^{-1} \mathbf{S} \ge 0, \tag{21b}$$

$$\tau_k(\mathbf{A}, \mathbf{b}) = 0 \quad \text{for} \quad k = 1, \ldots, P, \tag{21c}$$

wherein the inequalities are all componentwise and τ_k in (21c) are the order conditions.

In addition to the constraints (21a)–(21c), we must also add the condition that the abscissas are nondecreasing:

$$c_1 \le c_2 \le \cdots \le c_s \le 1. \tag{21d}$$

Solving this optimization problem will generate an explicit SSP Runge–Kutta method with coefficients $\mathbf{A}$ and $\mathbf{b}$ such that the abscissas are nondecreasing, with an SSP coefficient $\mathcal{C} = r$.

4.1. Order conditions. The equality constraints (21c) for the optimization problem above come from the order conditions, which were derived in [7]. Below are the order conditions for methods up to fourth order. For first order a method must satisfy the consistency condition:

$$\mathbf{b}^T\mathbf{e} = 1.$$

In addition to this condition second order methods must also satisfy

$$\mathbf{b}^T\mathbf{c} = \frac{1}{2}.$$

There are two more order conditions required to obtain third order:

$$\mathbf{b}^T(\mathbf{c}\cdot\mathbf{c}) = \frac{1}{3}, \qquad \mathbf{b}^T\mathbf{A}\mathbf{c} = \frac{1}{6}.$$

For fourth order four additional conditions must be satisfied:

$$\mathbf{b}^T(\mathbf{c}\cdot\mathbf{c}\cdot\mathbf{c}) = \frac{1}{4}, \quad \mathbf{b}^T(\mathbf{c}\cdot\mathbf{A}\mathbf{c}) = \frac{1}{8}, \quad \mathbf{b}^T\mathbf{A}(\mathbf{c}\cdot\mathbf{c}) = \frac{1}{12}, \quad \mathbf{b}^T\mathbf{A}^2\mathbf{c} = \frac{1}{24}.$$

Note that $(\mathbf{a}\cdot\mathbf{b})$ denotes elementwise multiplication. We do not present the order conditions past fourth order since there are no explicit SSP Runge–Kutta methods greater than fourth order.

5. Optimal and optimized methods. The optimization problem above was implemented in MATLAB (as in [39, 41, 40]) and used to find optimized eSSPRK$^+$ methods of up to ten stages and fourth order. These methods have nondecreasing abscissas and so can be used as a basis for eSSPIFRK methods.

The SSP coefficients and effective SSP coefficients of the optimized eSSPRK$^+$ methods are listed in Tables 3 and 4. The SSP coefficients of this family of methods are compared to those of the optimized eSSPRK methods with no constraint on the abscissas in Figure 2, where the circles indicate the SSP coefficient of the optimized explicit SSPRK methods while the lines are the SSP coefficients of the optimized eSSPRK$^+$ methods.

We observe that the optimal second order methods we found have the same SSP coefficients as the previously known SSP Runge–Kutta methods. This is not surprising as the abscissas of those optimal methods are nondecreasing, so our optimization routine found the previously known optimal methods. These eSSPRK$^+$(s,2) methods have SSP coefficient $\mathcal{C} = s - 1$ and effective SSP coefficient $\mathcal{C}_{\text{eff}} = \frac{s-1}{s}$. In section 5.2 we give the coefficients for these methods in both Shu–Osher form and Butcher form and show that the abscissas are indeed nondecreasing.

TABLE 3
SSP coefficients of the optimized eSSPRK$^+$(s,p) methods.

s \ p	2	3	4
1	-	-	-
2	1.0000	-	-
3	2.0000	0.7500	-
4	3.0000	1.8182	-
5	4.0000	2.6351	1.3466
6	5.0000	3.5184	2.2738
7	6.0000	4.2857	3.0404
8	7.0000	5.1071	3.8926
9	8.0000	6.0000	4.6048
10	9.0000	6.7853	5.2997

TABLE 4
Effective SSP coefficients of the optimized eSSPRK$^+$(s,p) methods.

s \ p	2	3	4
1	-	-	-
2	0.5000	-	-
3	0.6667	0.2500	-
4	0.7500	0.4545	-
5	0.8000	0.5270	0.2693
6	0.8333	0.5864	0.3790
7	0.8571	0.6122	0.4343
8	0.8750	0.6384	0.4866
9	0.8889	0.6667	0.5116
10	0.9000	0.6785	0.5300

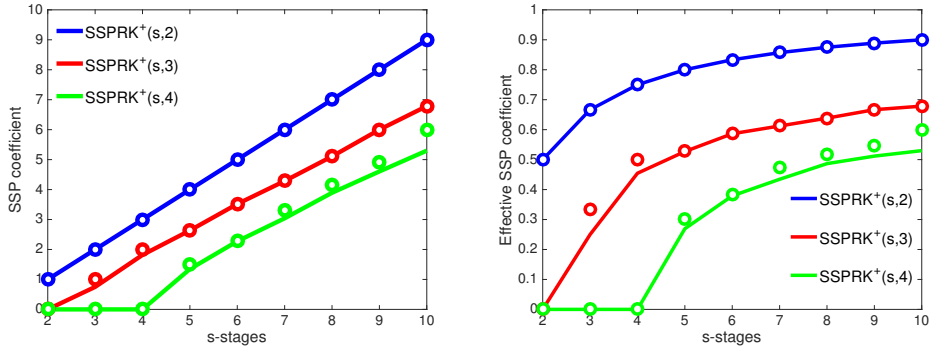

FIG. 2. *SSP coefficient (left) and effective SSP coefficient (right) for orders for $p = 2$ (blue), $p = 3$ (red), and $p = 4$ (green) methods. The circles indicate the SSP coefficient of the optimized eSSPRK methods while the lines are the SSP coefficients of the optimized eSSPRK$^+$ methods.*

In the third order case, the additional requirement that the abscissas be nondecreasing results in smaller SSP coefficients than the typical explicit SSP Runge–Kutta methods. For example, the optimal eSSPRK(3,3) Shu–Osher method (9) has SSP coefficient $\mathcal{C} = 1$ while the optimal eSSPRK$^+$(3,3) method has SSP coefficient $\mathcal{C} = \frac{3}{4}$, as we will prove in section 5.3. This loss in the SSP coefficient is also evident for the eSSPRK$^+$(4,3) method ($\mathcal{C} = \frac{20}{11}$) compared to the eSSPRK(4,3) method ($\mathcal{C} = 2$). However, as we add more stages the impact of the additional requirement of nondecreasing abscissas becomes negligible, and the SSP coefficients of the eSSPRK$^+$(s,3) methods are very close to those of the standard eSSPRK(s,3) methods, as seen in Figure 2.

In the fourth order case, the SSP coefficients of the optimized eSSPRK$^+$ are certainly smaller than those of the corresponding eSSPRK methods. In fact, this does not significantly improve as we increase the number of stages. Notably, the optimal eSSPRK(10,4) method found by Ketcheson [39] has an SSP coefficient of $\mathcal{C} = 6$ while the corresponding eSSPRK$^+$(10,4) method has SSP coefficient $\mathcal{C} = 5.3$, a reduction of over 10%. An exception to this is the optimized eSSPRK$^+$(6,4) method in which the nondescreasing abscissa requirement results in only a 1% reduction of the SSP coefficient compared to the eSSPRK(6,4).

5.1. Suboptimal explicit SSP Kinnmark and Gray Runge–Kutta methods with nondecreasing abscissas. In [43], Kinnmark and Gray presented a set of Runge–Kutta methods for linear problems. More precisely, they presented

TABLE 5

SSP coefficients of the optimized SSP Runge–Kutta methods designed for the Kinnmark and Gray stability regions (eSSPKG methods, in bold) compared to typical SSP Runge–Kutta methods (eSSPRK methods), with (right column) and without (left column) nondecreasing abscissas.

Method	$\mathcal{C}$	Method	$\mathcal{C}$
eSSPRK(5,3)	2.6506	eSSPRK$^+$(5,3)	2.6351
eSSPKG(5,3)	1.0000	**eSSPKG$^+$(5,3)**	0.8750
eSSPRK(6,4)	2.2945	eSSPRK$^+$(6,4)	2.2738
eSSPKG(6,4)	0.9904	**eSSPKG$^+$(6,4)**	0.7851

the linear stability polynomials for these methods. These methods were designed for use with problems that require a linear stability polynomials that include a large area of the imaginary axis. It is interesting to investigate what types of SSP coefficients the methods described in [43] can have. To do so, we modified our code that finds SSP Runge–Kutta methods of s stages and order p to include the linear stability polynomials of Kinnmark and Gray and used this code to find optimized SSP methods with $(s,p) = (3,3),(5,3),(6,4)$. We call these eSSPKG(s,p) methods and present their SSP coefficients and a comparison to the corresponding eSSPRK(s,p) methods in Table 5. All explicit SSP Runge–Kutta methods of $s = p = 3$ have the same stability polynomial, so that the Kinnmark and Gray methods eSSPKG(3,3) have the same linear stability region as eSSPRK(3,3). Although the SSP coefficients of the other SSP Kinnmark and Gray methods are smaller than those of the typical SSP Runge–Kutta methods, they may be used if the linear stability regions of Kinnmark and Gray are of interest. Note that the approach of optimizing an SSP method for a given linear stability region was originally done by [45]. For comparison, we provide the linear stability regions in Figure 3. We observed that, as expected, the Kinnmark Gray methods have larger imaginary axis stability but smaller overall regions.

Next, we added the requirement that the abscissas are nondecreasing to find optimized eSSPKG$^+$(s,p) methods for $(s,p) = (3,3),(5,3),(6,4)$ and compare their SSP coefficients to those of the eSSPRK$^+$(s,p), also in Table 5. These methods are suitable for use with Lawson-type integrating factor methods. In Figure 3 we show the linear stability regions of the $(s,p) = (5,3)$ and $(s,p) = (6,4)$ methods, as well. We test the eSSPKG methods as well as their use within the integrating factor approach in Example 3, where we see that their allowable time-step for preserving the TVD properties of a simple benchmark method are generally smaller than those of the eSSPRK$^+$ methods and the SSPRKIF methods.

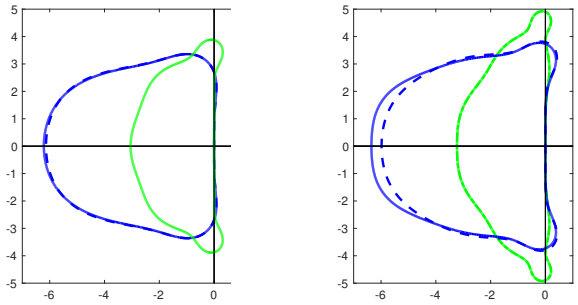

FIG. 3. *Linear stability region of the Kinnmark and Gray methods (in green) compared to the eSSPRK method (blue solid line) and the eSSPRK$^+$ method (blue dashed line). Left: method with $(s,p) = (5,3)$. Right: method with $(s,p) = (6,4)$.*

5.2. Optimal second order explicit SSP Runge–Kutta methods with nondecreasing abscissas. We mentioned above that the optimal eSSPRK(s,2) methods have nondecreasing coefficients. In this section we review these methods, first presented in [29], and show that the abscissas are in fact increasing. These methods can be written in Shu–Osher form (where $u^{(0)} = u^n$):

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

In Butcher form, this becomes

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ \frac{1}{s-1} & 0 & 0 & \dots & 0 & 0 \\ \frac{1}{s-1} & \frac{1}{s-1} & 0 & \dots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \frac{1}{s-1} & \frac{1}{s-1} & \frac{1}{s-1} & \frac{1}{s-1} & 0 & 0 \\ \frac{1}{s-1} & \frac{1}{s-1} & \frac{1}{s-1} & \frac{1}{s-1} & \frac{1}{s-1} & 0 \end{bmatrix}, \quad \mathbf{b}^T = \begin{bmatrix} \frac{1}{s} \\ \frac{1}{s} \\ \frac{1}{s} \\ \vdots \\ \frac{1}{s} \\ \frac{1}{s} \end{bmatrix}$$

with abscissas

$$\mathbf{c}^T = \left[0, \frac{1}{s-1}, \frac{2}{s-1}, \dots, \frac{s-2}{s-1}, 1 \right].$$

Clearly, these optimal explicit SSP Runge–Kutta methods have increasing abscissas and are therefore suitable use with an integrating factor approach to create eSSPIFRK methods.

5.3. Optimized third order explicit SSP Runge–Kutta methods with nondecreasing abscissas.

THEOREM 3. *The eSSPRK$^+$(3,3) method given by*

$$\begin{aligned} u^{(1)} &= \frac{1}{2} u^n + \frac{1}{2} \left(u^n + \frac{4}{3} \Delta t F(u^n) \right), \\ u^{(2)} &= \frac{2}{3} u^n + \frac{1}{3} \left(u^{(1)} + \frac{4}{3} \Delta t F(u^{(1)}) \right), \\ u^{n+1} &= \frac{59}{128} u^n + \frac{15}{128} \left(u^{(1)} + \frac{4}{3} \Delta t F(u^{(1)}) \right) + \frac{27}{64} \left(u^{(2)} + \frac{4}{3} \Delta t F(u^{(2)}) \right) \end{aligned} \tag{22}$$

is strong stability preserving with SSP coefficient $\mathcal{C} = \frac{3}{4}$ and is optimal among all eSSPRK$^+(3,3)$ methods.

Proof. This method is given in its canonical Shu–Osher form. Clearly, we have a convex combination of forward Euler steps with time-step $\frac{4}{3}\Delta t$, and so this method is SSP with $\mathcal{C} = \frac{3}{4}$.

To show that this is optimal among all possible eSSPRK$^+$(3,3) methods, we follow along the lines of the proof in [27]. We assume that $\mathcal{C} > \frac{3}{4}$, which means that $\frac{\alpha_{ij}}{\beta_{ij}} > \frac{3}{4}$ or

$$\alpha_{ij} > \frac{3}{4}\beta_{ij} \quad \text{for any} \quad i, j,$$

and proceed with a proof by contradiction.

First, recall that we can transform between the Shu–Osher coefficients α, β and the Butcher array coefficients $\mathbf{A}$ and $\mathbf{b}$ as follows:

$$\begin{aligned} a_{21} &= \beta_{10}, \;\; a_{31} = \beta_{20} + \alpha_{21}\beta_{10}, \;\; a_{32} = \beta_{21}, \\ b_1 &= \alpha_{32}\alpha_{21}\beta_{10} + \alpha_{31}\beta_{10} + \alpha_{32}\beta_{20} + \beta_{30}, \;\; b_2 = \alpha_{32}\beta_{21} + \beta_{31}, \;\; b_3 = \beta_{32}, \\ c_1 &= 0, \;\; c_2 = a_{21}, \;\; c_3 = a_{31} + a_{32}. \end{aligned}$$

Note that for the method to be SSP, the coefficients must all be nonnegative. If the abscissas are nondecreasing, there are two possible cases: $c_2 = c_3$ and $c_2 < c_3$. We consider each of these cases separately.

Case (a). If the abscissas are equal, they must be $c_2 = c_3 = \frac{2}{3}$. The coefficients in this case satisfy

$$a_{21} = \frac{2}{3}, \;\; a_{31} = \frac{2}{3} - \frac{1}{4\omega}, \;\; a_{32} = \frac{1}{4\omega}, \;\; b_1 = \frac{1}{4}, \;\; b_2 = \frac{3}{4} - \omega, \;\; b_3 = \omega$$

for parameter ω. The assumption that $\alpha_{ij} > \frac{3}{4}\beta_{ij}$ and the nonnegativity assumption on the coefficients results in

$$b_2 = \alpha_{32}\beta_{21} + \beta_{31} \geq \alpha_{32}\beta_{21} > \frac{3}{4}\beta_{32}\beta_{21} = \frac{3}{4}b_3 a_{32} = \frac{3}{16} \implies \omega < \frac{9}{16}.$$

On the other hand,

$$a_{31} = \beta_{20} + \alpha_{21}\beta_{10} \geq \alpha_{21}\beta_{10} > \frac{3}{4}\beta_{21}\beta_{10} = \frac{3}{4}a_{32}a_{21} = \frac{3}{4}\frac{1}{4\omega}\frac{2}{3} = \frac{1}{8\omega}$$

so that

$$\frac{2}{3} - \frac{1}{4\omega} > \frac{1}{8\omega} \implies \omega > \frac{9}{16},$$

which contradicts the bound on ω above.

Case (b). If the two abscissas are not equal, and we require nondecreasing abscissas, we must have $c_2 < c_3$. In this case, the coefficients are given by a two parameter system, where the parameters are the abscissas c_2 and c_3.

$$\begin{aligned} a_{21} = c_2, \;\; a_{31} &= \frac{3c_2c_3(1-c_2) - c_3^2}{c_2(2-3c_2)}, \;\; a_{32} = \frac{c_3(c_3-c_2)}{c_2(2-3c_2)}, \\ b_1 = 1 + \frac{2-3(c_2+c_3)}{6c_2c_3}, \;\; b_2 &= \frac{3c_3-2}{6c_2(c_3-c_2)}, \;\; b_3 = \frac{2-3c_2}{6c_3(c_3-c_2)}. \end{aligned}$$

The requirement that $c_2 < c_3$ and that both b_2 and b_3 are nonnegative gives

$$\begin{aligned} b_2 = \frac{3c_3-2}{6c_2(c_3-c_2)} \geq 0 &\implies c_3 \geq \frac{2}{3}, \\ b_3 = \frac{2-3c_2}{6c_3(c_3-c_2)} \geq 0 &\implies c_2 \leq \frac{2}{3}. \end{aligned}$$

Now begin with $a_{31} = \beta_{20} + \alpha_{21}\beta_{10}$, and recall that $\alpha_{21} > \frac{3}{4}\beta_{21}$ and $\beta_{10} = a_{21} = c_2$ and $\beta_{21} = a_{32}$, so that

$$a_{31} > \frac{3}{4}\beta_{21}c_2 = \frac{3}{4}a_{32}c_2 \implies c_3 - a_{32} > \frac{3}{4}a_{32}c_2 \implies c_3 > a_{32}\left(1 + \frac{3}{4}c_2\right),$$

which requires $a_{32} < \frac{c_3}{(1+\frac{3}{4}c_2)}$.

Using the definition of a_{32}, this means

$$\frac{c_3(c_3 - c_2)}{c_2(2 - 3c_2)} < \frac{c_3}{\left(1 + \frac{3}{4}c_2\right)} \implies c_3 < \frac{3c_2 - \frac{9}{4}c_2^2}{\left(1 + \frac{3}{4}c_2\right)}.$$

Next, we look use the fact that $\beta_{31} \geq 0$ to obtain

$$b_2 = \alpha_{32}\beta_{21} + \beta_{31} \geq \alpha_{32}\beta_{21} > \frac{3}{4}b_3 a_{32} \implies a_{32} < \frac{4}{3}\frac{b_2}{b_3} = \frac{4}{3}\frac{c_3(3c_3 - 2)}{c_2(2 - 3c_2)}.$$

Now we use $a_{32} = \frac{c_3(c_3 - c_2)}{c_2(2-3c_2)}$ to conclude that

$$\frac{c_3(c_3 - c_2)}{c_2(2 - 3c_2)} < \frac{4}{3}\frac{c_3(3c_3 - 2)}{c_2(2 - 3c_2)} \implies c_3 > -\frac{1}{3}c_2 + \frac{8}{9}.$$

We now have two statements that need to be simultaneously true,

$$c_3 < \frac{3c_2 - \frac{9}{4}c_2^2}{\left(1 + \frac{3}{4}c_2\right)} \text{ and } c_3 > -\frac{1}{3}c_2 + \frac{8}{9},$$

which means that we must have

$$-\frac{1}{3}c_2 + \frac{8}{9} < \frac{3c_2 - \frac{9}{4}c_2^2}{\left(1 + \frac{3}{4}c_2\right)} \implies (3c_2 - 2)^2 < 0;$$

however, this is a contradiction because $(3c_2 - 2)^2$ is always greater than or equal to zero. This means that our original assumption was not correct, and that if $c_2 < c_3$ we cannot have $\mathcal{C} > \frac{3}{4}$. ∎

5.4. Recommended SSP Runge–Kutta methods for use with integrating factor methods. The optimal second order methods eSSPRK$^+$(s,2) listed above have sparse Shu–Osher representations and a general formula. However, for the optimized third and fourth order methods, we do not have a general formula. In this section we list a few of the optimized third- and fourth-order methods. The coefficients of all the methods we found can be downloaded as `.mat` files from our github repository [26].

eSSPRK$^+$(4,3). This method has rational coefficients and sparse Shu–Osher matrices:

$$\begin{aligned}
u^{(1)} &= u^n + \frac{11}{20}\Delta t F(u^n),\\
u^{(2)} &= \frac{3}{8}u^n + \frac{5}{8}\left(u^{(1)} + \frac{11}{20}\Delta t F(u^{(1)})\right),\\
u^{(3)} &= \frac{4}{9}u^n + \frac{5}{9}\left(u^{(2)} + \frac{11}{20}\Delta t F(u^{(2)})\right),\\
u^{n+1} &= \frac{111}{1331}u^n + \frac{260}{1331}\left(u^n + \frac{11}{20}\Delta t F(u^n)\right) + \frac{960}{1331}\left(u^{(3)} + \frac{11}{20}\Delta t F(u^{(3)})\right).
\end{aligned}$$

The abscissas are $c_1 = 0, c_2 = \frac{11}{20}, c_3 = c_4 = \frac{11}{16}$. This method has $\mathcal{C} = \frac{20}{11}$.

When many stages are required for a high order computation, the amount of storage, particularly for large simulations, may become prohibitive. Low-storage methods are of great interest in such cases. Low-storage Runge–Kutta methods were considered in [83, 38, 80]. More recently, Ketcheson [39] developed many low-storage SSP methods and showed that some of the most efficient methods in terms of the SSP coefficient are also efficient in terms of storage. This method is low-storage in the sense of [39], as many of the storage registers can be overwritten during the implementation, assuming one is willing to recompute $F(u^{(i)})$ when needed [39, 27]. Due to the structure of the Shu–Osher matrices, only two memory registers are required for this method, rather than the full $s+1=5$ that would be needed for a naive implementation.

eSSPRK$^+$(9,3). This method has $\mathcal{C}=6$ and features rational coefficients and sparse Shu–Osher matrices:

$$\begin{aligned}
u^{(0)} &= u^n,\\
u^{(i)} &= u^{(i-1)} + \frac{1}{6}\Delta t F(u^{(i-1)}) \ \text{ for } i=1,\dots,4,\\
u^{(5)} &= \frac{1}{5}u^n + \frac{4}{5}\left(u^{(4)} + \frac{1}{6}\Delta t F(u^{(4)})\right),\\
u^{(6)} &= \frac{1}{4}\left(u^n + \frac{1}{6}\Delta t F(u^n)\right) + \frac{3}{4}\left(u^{(5)} + \frac{1}{6}\Delta t F(u^{(5)})\right),\\
u^{(7)} &= \frac{1}{3}u^{(2)} + \frac{2}{3}\left(u^{(6)} + \frac{1}{6}\Delta t F(u^{(6)})\right),\\
u^{(8)} &= u^{(7)} + \frac{1}{6}\Delta t F(u^{(7)}),\\
u^{n+1} &= u^{(8)} + \frac{1}{6}\Delta t F(u^{(8)}).
\end{aligned}$$

The abscissas are $c_1=0, c_2=\frac{1}{6}, c_3=\frac{2}{6}, c_4=\frac{3}{6}, c_5=c_6=c_7=c_8=\frac{4}{6}, c_9=\frac{5}{6}$, which simplifies the computation of the matrix exponential, as only one needs to be computed.

This method is also efficient in terms of memory as it may be implemented with only three storage registers (assuming one is willing to compute $F(u^{(i)})$ twice). Although this is not a low-storage method in the sense of [39], i.e., it requires more than two registers, we do not require the full $s+1=10$ storage registers naively needed for implementing this method, but only three.

For fourth order methods, we no longer have rational coefficients.

eSSPRK$^+$(5,4). This method has SSP coefficient $\mathcal{C}=r=1.346586417284006$ and nondecreasing abscissas

$$c_1=0, \quad c_2\approx 0.4549, \quad c_3=c_4\approx 0.5165, \quad c_5\approx 0.9903:$$

$$\begin{aligned}
u^{(1)} &= 0.387392167970373\, u^n + 0.612607832029627\left(u^n + \frac{\Delta t}{r}F(u^n)\right),\\
u^{(2)} &= 0.568702484115635\, u^n + 0.431297515884365\left(u^{(1)} + \frac{\Delta t}{r}\Delta t F(u^{(1)})\right),
\end{aligned}$$

$$u^{(3)} = 0.589791736452092\, u^n + 0.410208263547908 \left(u^{(2)} + \frac{\Delta t}{r} F(u^{(2)}) \right),$$
$$u^{(4)} = 0.213474206786188\, u^n + 0.786525793213812 \left(u^{(3)} + \frac{\Delta t}{r} F(u^{(3)}) \right),$$

$$\begin{aligned} u^{n+1} = {} & 0.270147144537063\, u^n + 0.029337521506634 \left(u^n + \frac{\Delta t}{r} F(u^n) \right), \\ & + 0.239419175840559 \left(u^{(1)} + \frac{\Delta t}{r} \Delta t F(u^{(1)}) \right) \\ & + 0.227000995504038 \left(u^{(3)} + \frac{\Delta t}{r} F(u^{(3)}) \right) \\ & + 0.234095162611706 \left(u^{(4)} + \frac{\Delta t}{r} F(u^{(4)}) \right). \end{aligned}$$

eSSPRK$^+$(6,4). This method has SSP coefficient $\mathcal{C} = r = 2.273802749301517$ and nondecreasing abscissas $c_1 = 0$, $c_2 \approx 0.4398$, $c_3 \approx 0.4515$, $c_4 = c_5 \approx 0.5461$, $c_6 \approx .0.9859$:

$$\begin{aligned} u^{(1)} &= u^n + \frac{\Delta t}{r} F(u^n), \\ u^{(2)} &= 0.486695314011133 u^n + 0.513304685988867 \left(u^{(1)} + \frac{\Delta t}{r} \Delta t F(u^{(1)}) \right), \\ u^{(3)} &= 0.387273961537322\, u^n + 0.612726038462678 \left(u^{(2)} + \frac{\Delta t}{r} F(u^{(2)}) \right), \\ u^{(4)} &= 0.419340376206590\, u^n + 0.048271190433595 \left(u^n + \frac{\Delta t}{r} F(u^n) \right) \\ &\quad + 0.532388433359815 \left(u^{(3)} + \frac{\Delta t}{r} F(u^{(3)}) \right), \\ u^{(5)} &= u^{(4)} + \frac{\Delta t}{r} F(u^{(4)}), \\ u^{n+1} &= 0.122021674306995\, u^n + 0.104714614292281 \left(u^{(1)} + \frac{\Delta t}{r} \Delta t F(u^{(1)}) \right) \\ &\quad + 0.316675962670361 \left(u^{(2)} + \frac{\Delta t}{r} \Delta t F(u^{(2)}) \right) \\ &\quad + 0.057551178672633 \left(u^{(4)} + \frac{\Delta t}{r} \Delta t F(u^{(4)}) \right) \\ &\quad + 0.399036570057730 \left(u^{(5)} + \frac{\Delta t}{r} \Delta t F(u^{(5)}) \right). \end{aligned}$$

6. Numerical results. In this section, we test the eSSPIFRK methods based on eSSPRK$^+$ methods presented in section 5 for convergence and SSP properties. First, we test these methods for convergence on a nonlinear system of ODEs to confirm that the new methods exhibit the desired orders. Next, we study the behavior of these methods in terms of their allowable time-step on linear and nonlinear problems with spatial discretizations that are provably TVD. While the utility of SSP methods

goes well beyond its initial purpose of preserving the TVD properties of the spatial discretization coupled with forward Euler, the simple TVD test in this section has been used extensively because it tends to demonstrate the sharpness of the SSP time-step.

Remark 3. The cost of computation of the matrix exponential is a major factor that will determine the efficiency of these methods in practice. There are several approaches that can be taken here. The first and simplest approach is that the approximation of the matrix exponential be done by evolving $u' = Lu$ (where u is a matrix and $u^0 = I$) numerically up to $t = \Delta t$ using an explicit SSP Runge–Kutta method with a sufficiently small step-size as in [58] (a more recent approach combines this idea with a scale and square method [1]). This is not inefficient when performed only once per simulation, which is all that is required when L is a constant coefficient operator. However, when storage is a consideration and matrix-free approaches are desired, there are a number of other efficient approaches that have been proposed and are under active consideration by several research groups working on ETD methods. Many of these methods produce the action of a matrix exponential at a cost less than the cost of an implicit solve. The reader is referred to the work of [1], the EXPOKIT software [69], the *phipm* adaptive method in [59], and the KIOPS method of Gaudreault, Rainwater, and Tokman [23]. Although this active area of research is outside the scope of our paper, it is of great interest as it will be necessary for bringing the eSSPIFRK methods presented in this paper into practical use.

6.1. Example 1: Convergence study. To verify the order of convergence of these methods we test their performance on a nonlinear system of ODEs

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

known as the van der Pol problem. We split the problem in two different ways into a linear part Lu and a nonlinear part $N(u)$ given by

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

and

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

We use initial conditions $\mathbf{u}_0 = (2; 0)$ and run the problem to final time $T_{final} = 0.50$, with $\Delta t = 0.02, 0.04, 0.06, 0.08, 0.10$. The exact solution (for error calculation) was calculated by the ODE45 routine in MATLAB with tolerances set to `AbsTol`=10^{-15} and `RelTol`=10^{-15}. For each splitting, we tested all the methods represented in Table 3 above and calculated the slopes of the orders by the `polyfit` function in MATLAB; we found that they all exhibit the expected order of convergence. Due to space constraints, we show only a representative selection in Figure 4. While the splitting affects the magnitude of the errors, we see that the order of the errors is not affected. As expected, the error constants are smaller for methods with more stages.

Note that we used a van der Pol problem that is not highly oscillatory. This is because we wish to avoid the order reduction that is known to occur with integrating factor methods. This convergence study purposely avoids this issue in order to test the formal convergence of the generated methods.

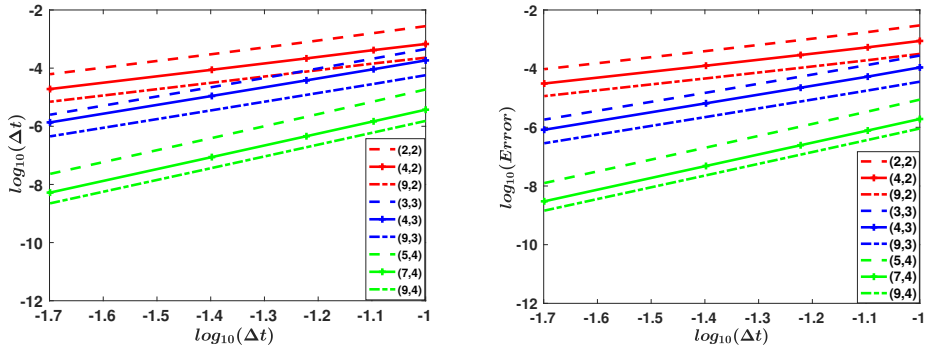

FIG. 4. *Example* 1. *The second order methods are in red, third order in blue, and fourth order in green. The eSSPIFRK methods with* $(s,p) = (2,2),(3,3),(5,4)$ *have a dashed line, those with* $(s,p) = (4,2),(4,3),(7,4)$ *have a solid line, and those with* $s = 9$ *have a dot-dash line. On the left is splitting (a) while on the right is splitting (b).*

6.2. Example 2: Accuracy study. Consider the problem

$$u_t + 10u_x + \left(\frac{1}{2}u^2\right)_x = 0, \qquad u(0,x) = e^{\sin(2\pi x)} \tag{23}$$

on the domain $0 \le x \le 1$. We use a first order upwind finite difference to spatially discretize the linear advection term and the fifth order WENO for the nonlinear term. We use 64 points in space and the `globalorder.m` script in the package EXPINT [6, 5] with its built-in ETD Runge–Kutta methods of orders $p = 2,3,4$ (the schemes by Cox and Matthews called ETDRK2, ETDRK3, and ETDRK4 in EXPINT) and our eSSPIFRK methods with $(s,p) = (2,2),(3,3),(5,4)$. The `globalorder.m` script uses the ODE15s in MATLAB with `AbsTol` $=10^{-15}$ and `RelTol` $= 5 \times 10^{-14}$ to compute the highly accurate reference solution. In Figure 5 (left) we observe that the eSSPIFRK methods are competitive with the ETD methods in terms of accuracy. Despite the fact that we see order reduction in the third and fourth order eSSPIFRK methods, the accuracy of these methods is comparable to that of the corresponding ETD method.

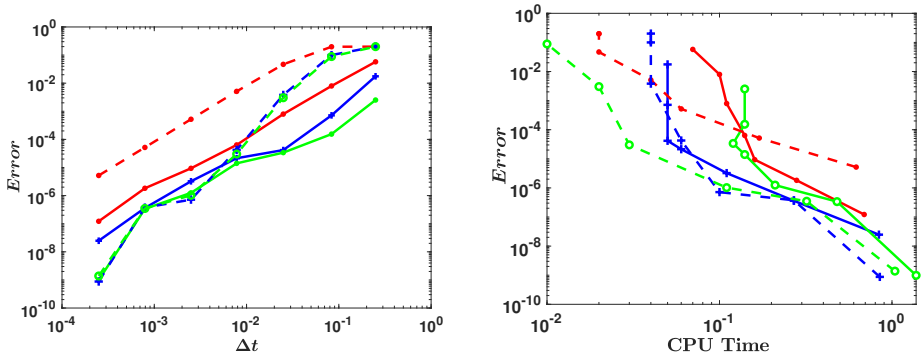

FIG. 5. *Example* 2. *The second order methods are in red, third order in blue, and fourth order in green; dashed lines represent the ETD methods while solid lines are the eSSPIFRK methods.*

A comparison of the CPU times needed for a given level of accuracy in Figure 5 (right) reveals that the ETD methods are generally somewhat more efficient on this

smooth problem. However, we will see below that the ETD methods in fact require inefficiently small time-steps for nonlinear stability in problems with discontinuities that are of interest to us.

6.3. Example 3: Sharpness of SSP time-step for a linear problem. We consider the linear advection equation with a step function initial condition:

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

on the domain $[0,1]$ with periodic boundary conditions. We use a first order forward difference for each of the spatial derivatives to semidiscretize this problem on a grid of $0 \le x \le 1$ with 1000 points in space and evolve it ten time-steps forward.

It is known that this spatial discretization when coupled with forward Euler is TVD, under the time-step restriction $\tilde{\Delta}t_{\text{FE}} = \frac{1}{a}\Delta x$ for the term $Lu \approx au_x$ and the restriction $\Delta t_{\text{FE}} = \Delta x$ for the term $N(u) \approx u_x$.

We measure the total variation of the numerical solution at each stage (to ensure internal stage monotonicity) and compare it to the total variation at the previous stage. We are interested in the size of time-step Δt at which the total variation begins to rise. We refer to this value as the *observed TVD time-step* Δt_{obs}^{TVD}. We compare this value with the expected TVD time-step dictated by the theory. We call the SSP coefficient corresponding to the value of the observed TVD time-step the *observed SSP coefficient* $\mathcal{C}_{obs}$. Note that the expected TVD time-step $\Delta t_{\text{FE}} = \Delta x$, so that

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

6.3.1. Comparison of integrating methods for wavespeed $a = 10$. First we consider this problem with wavespeed $a = 10$. In Figure 6, we show the observed maximal rise in total variation (on the y-axis) when this equation is evolved forward by a variety of methods using different values of the Courant number $\lambda = \frac{\Delta t}{\Delta x}$ (on the x-axis).

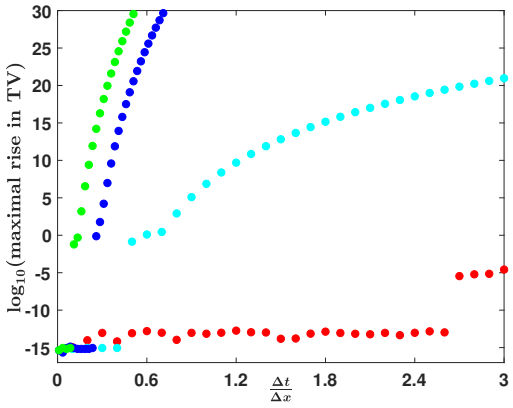

FIG. 6. *Example* 3: *Linear advection with a step function initial condition and* $a = 10$. *The eSSPRK*$(5,3)$ *method is in blue, the eSSPKG*$(5,3)$ *in green, the IMEXSSP*$(5,3,K = 0.1)$ *method in cyan, and the eSSPIFRK*$(5,3)$ *in red.*

Two methods, eSSPRK(5,3) (blue) from [27] and eSSPKG(5,3) (green) generated in this work, treat the two spatial terms in the same manner. In other words, they evolve the linear advection problem with wavespeed $(1 + a) = 11$. As expected, the observed time-step before the total variation begins to rise is given by the SSP coefficient scaled by the wavespeed. Our numerical results (represented in Figure 6) show that the eSSPRK(5,3) method has an allowable Courant number $\lambda_{obs}^{TVD} = \frac{\mathcal{C}}{11} = \frac{2.6506}{11} = 0.2409$ before the maximal total variation begins to rise. On the other hand, the SSP Kinnmark and Gray method eSSPKG(5,3) has a smaller allowable value of $\lambda_{obs}^{TVD} = \frac{\mathcal{C}}{11} = \frac{1}{11} = 0.0909$ before the maximal total variation begins to rise.

Next, we turn our attention to IMEX methods. As shown in [15] both the implicit and explicit terms have to be SSP, and the wavespeed impacts the size of the allowable SSP time-step, even though the fast wave is treated implicitly.[1] We use the SSP IMEX(5,3,$K = 0.1$) method (in cyan) where the slow wave u_x is treated explicitly and the fast wave $10u_x$ is treated implicitly. We use the IMEX(5,3,$K = 0.1$) method found in [15], which is specially optimized in terms of the allowable SSP time-step for the value of $K = \frac{1}{a} = 0.1$. This method has a $\mathcal{C} = 0.407$ for the value $a = 10$. As predicted, the observed time-step before the total variation begin to rise is $\lambda_{obs}^{TVD} = \frac{\mathcal{C}}{10} = 0.407$. We see here that using an IMEX SSP method where the fast moving wave is treated implicitly does not give us much benefit when we are concerned with the TVD behavior of the scheme: it allows us only an increase of 69% in time-step, at the major cost of an implicit solve.

Finally, we use the eSSPIFRK(5,3) method (in red) resulting from using our eSSPRK$^+$(5,3) method in formulation (16) where $N(u) = -Du$ and $Lu = -10Du$. This method has a SSP coefficient $\mathcal{C} = 2.635$ for any value of a, and indeed we observe that the allowable time-step before we see a rise in the total variation is $\Delta t_{obs}^{TVD} = \frac{\mathcal{C}}{\Delta x}$. This means that the largest allowable time-step for the eSSPIFRK(5,3) is more than ten times larger than that for the eSSPRK(5,3) and more than six times larger than for the IMEX method. Additionally, this result is produced without much additional cost, as the few matrix exponentials needed are precomputed once for the entire simulation. Results of the allowable SSP values for more methods are presented in Table 6 below.

6.3.2. Considering different wavespeeds. The main advantage of the SSP integrating factor Runge–Kutta schemes is that the allowable SSP time-step is not impacted by the wavespeed. In this section, we show how the new eSSPIFRK methods perform equally well for different wavespeeds and how other methods (including IMEX methods) do not.

Using $a = 0$, we verify the expected TVD time-step for most of the methods considered (Table 7). However, we observed that for the methods eSSPIFRK(3,3) and eSSPIFRK(5,4), $\mathcal{C}_{obs} > \mathcal{C}$. For the eSSPIFRK(3,3) method, $\mathcal{C} = \frac{3}{4}$, but the observed value is $\mathcal{C}_{obs} = 1$. This is easy to understand because for this linear problem with no exponential component ($a = 0$), all methods with the same number of stages as order ($s = p$) are equivalent. Thus, for this special case, our eSSPIFRK(3,3) method can be rearranged into the eSSPRK(3,3) Shu–Osher method, and we observe the expected TVD time-step for that method. A similar phenomenon occurs for the eSSPIFRK(5,4) method. In this case, we can write the stability polynomial of each stage recursively. If we look at the stability polynomial $P(z)$ of the fourth stage, we

[1] In fact, in Table 6 we observe that an IMEX scheme composed of the Shu–Osher eSSPRK(3,3) for the explicit part and an A-stable implicit part does not preserve the TVD property for any time-step in our numerical tests.

TABLE 6

*The values of the observed CFL number $\lambda^{L_2} = \frac{\Delta t}{\Delta x}$ required for L_2 linear stability for Example 3: $u_t + u_x + 10u_x = 0$, compared to the predicted and observed values λ^{TVD}. An * indicates that these methods were linearly L_2 stable for the largest values tested, $\lambda \leq 27$.*

Method	$\lambda_{obs}^{L_2}$	λ_{pred}^{TVD}	λ_{obs}^{TVD}
eSSPRK(3,3)	0.114	1/11	0.090
eSSPKG(3,3)	0.114	1/11	0.090
eSSPRK$^+$(3,3)	0.114	3/44	0.090
eSSPKG$^+$(3,3)	0.114	1/11	0.090
IMEXSSP(3,3,$K = 0.1$))	0.448	0.149	0.236
IMEXSSP(3,3,$K = \infty$))	1.198	0.000	0.000
eSSPIFKG(3,3)	*	0.750	1.500
eSSPIFRK(3,3)	*	0.750	1.500
eSSPRK(5,3)	0.260	0.240	0.240
eSSPKG(5,3)	0.138	1/11	0.090
eSSPRK$^+$(5,3)	0.261	0.239	0.239
eSSPKG$^+$(5,3)	0.138	1/11	0.090
IMEXSSP(5,3,$K = 0.1$)	0.683	0.407	0.407
eSSPIFKG(5,3)	*	0.875	1.487
eSSPIFRK(5,3)	*	2.635	2.635
eSSPRK(6,4)	0.273	0.208	0.208
eSSPKG(6,4)	0.146	1/11	0.090
eSSPRK$^+$(6,4)	0.270	0.206	0.206
eSSPKG$^+$(6,4)	0.146	0.071	0.090
eSSPIFKG(6,4)	*	0.785	1.805
eSSPIFRK(6,4)	*	2.273	2.273

TABLE 7

The observed SSP coefficient compared to the predicted SSP coefficient for Example 3 with various wavespeeds a. The value of a does not negatively impact the observed SSP coefficient.

Method	$\mathcal{C}$	λ_{obs}^{TVD}			
		for $a = 0$	$a = 1.0$	$a = 10$	$a = 20$
eSSPIFRK(2,2)	1	1	1	1	1
eSSPIFRK(9,2)	8	8	8	8	8
eSSPIFRK(3,3)	3/4	1	3/2	3/2	3/2
eSSPIFRK(4,3)	20/11	20/11	20/11	20/11	20/11
eSSPIFRK(9,3)	6	6	6	6	6
eSSPIFRK(5,4)	1.346	1.5594	2.158	2.158	2.158
eSSPIFRK(6,4)	2.273	2.273	2.273	2.273	2.273
eSSPIFRK(9,4)	4.306	4.306	4.306	4.306	4.306

observe that its third derivative becomes negative at $z = 1.5594$, which is precisely the observed TVD time-step for this method. When we look into the stages to see where the rise in total variation occurs, we see that it first happens in the fourth stage of the first time-step.

Next, we consider various values of $a > 0$. The results in Table 7 confirm that *the value of a does not negatively impact the observed SSP coefficient for this linear example.*[2] In these cases too, we observe that for most methods the SSP condition is sharp, i.e., $\mathcal{C}_{obs} = \mathcal{C}$ for all values of a. The observed TVD time-step for eSSPIFRK(3,3) and eSSPIFRK(5,4) are, once again, larger than expected. For the eSSPIFRK(3,3) method $\mathcal{C}_{obs} = \frac{3}{2}$ is a result of the rise in total variation from the first stage, which has a step-size $\frac{2}{3}\Delta t$.

[2]We note that for values larger than $a = 20$, the significant damping that occurs due to the exponential masks the oscillation and its associated rise in total variation.

TABLE 8

The observed SSP coefficients for an eSSPIFRK method, IMEX method, and explicit Runge–Kutta method for Example 3 with various wavespeeds a. The value of a does not negatively impact the observed SSP coefficient for the eSSPIFRK method, but it does for the IMEX and the explicit Runge–Kutta methods.

Method	λ_{obs}^{TVD}					
	$a = 0$	$a = 1.0$	$a = 2$	$a = 10$	$a = 20$	$a = 100$
eSSPIFRK(4,3)	1.818	1.818	1.818	1.818	1.818	4.200
SSPIMEX(4,3,K)	2.000	1.476	1.192	0.310	0.162	0.033
eSSPRK(4,3)	2.000	1.000	0.666	0.181	0.0952	0.019

Finally, we compare the eSSPIFRK(4,3) method to the eSSPRK method and to IMEX methods. Table 8 shows that the value of a does not negatively impact the observed SSP coefficient for the eSSPIFRK method. When the eSSPRK(4,3) method is applied to the linear advection problem with wavespeed $a + 1$, the observed SSP coefficient matches the predicted

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

as shown in Table 8. We also show the observed SSP coefficient for the SSP IMEX methods IMEXSSP(4,3,K) from [15]. These methods have SSP explicit and implicit parts and were optimized for the SSP step-size for each value of $K = \frac{1}{a}$ in [15]. As we expect from SSP theory, the observed value of λ before a rise in total variation occurs decays linearly as the wavespeed a rises.

6.3.3. Comparison to linear stability properties. In Figure 6 we notice that for each method, as the Courant number $\lambda = \frac{\Delta t}{\Delta x}$ gets large enough, the maximal rise in total variation jumps up. For some methods, this happens for very small values of λ, while for others, it happens when λ is larger. It is interesting to see how this λ^{TVD} allowable for the TVD property compares to the CFL number λ^{L_2} allowed for linear stability for this problem. Once again, we use $a = 10$ and compare the allowable time-step for linear stability and the allowable predicted and observed TVD time-step in Table 6. We evaluate the allowable time-step for linear stability of a given method for this particular problem by calculating the stability polynomial for these operators and determining at which value of λ^{L_2} the L_2 norm of the resulting matrix becomes greater than one.

We notice that linear stability does not provide any prediction on the TVD behavior of these methods. The most striking example of this is when looking at the IMEX methods; in terms of linear stability, these can clearly eliminate the constraint coming from the fast wave (e.g., IMEXSSP(3,3,$K = \infty$)) and provide a large region of linear stability. However, this method fails to be SSP for any positive time-step. Even a specially designed and optimized SSP IMEX method did not even double the allowable time-step for TVD.

These results underscore the fact that when solving problems with discontinuities, the relevant time-step restriction is dictated by λ^{TVD} rather than λ^{L_2}. The most notable result from this table is that all the integrating factor methods have a very large L_2 linear stability region for this problem. In fact, we tested them up to $\lambda = 27$, which is ten times larger than the largest allowable time-step for TVD, and they were still stable at this value. Clearly, the integrating approach has advantages for linear stability as well as for strong stability preservation.

We also note from the results in Table 6 that although the Kinnmark and Gray methods are designed to have larger imaginary axis linear stability regions, for our problem this does not give them an advantage even for linear stability. This is easily understood when we consider that the eigenvalues of our differentiation operator are a circle in the complex plane, and so the linear stability regions of the regular eSSPRK methods are better suited than those of the eSSPKG methods of Kinnmark and Gray (as seen in Figure 3).

6.4. Example 4: Sharpness of SSP time-step for a nonlinear problem. Consider the equation

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

on the domain $[0,1]$ with periodic boundary conditions. We used a first order upwind difference to semidiscretize this linear term and a fifth order WENO finite difference for the nonlinear terms. We solved this problem on a spatial grid with 400 points and evolved it forward 25 time-steps using $\Delta t = \lambda \Delta x$. We measured the total variation at each stage and calculated the maximal rise in total variation over each stage for these 25 time-steps.

In Figure 7 (left) we use the value of $a = 5$ and graph the log_{10} of the maximal rise in total variation versus the ratio $\lambda = \frac{\Delta t}{\Delta x}$. We observe that our eSSPIFRK(3,3) method (19) maintains a very small maximal rise in total variation until close to $\lambda = 0.8$, while the fully explicit third order Shu–Osher method begins to feature a large rise in total variation for a much smaller value of $\lambda = .15 \approx \frac{1}{1+a}$. In contrast, the three-stage third order ETD Runge–Kutta [16] and the integrating factor method based on the Shu–Osher method (18) both have a maximal rise in total variation that increases rapidly with λ.

In Figure 7 (right) we show a similar study using wavespeed $a = 10$ and fourth order methods versus the ratio $\lambda = \frac{\Delta t}{\Delta x}$. We observe that our eSSPIFRK(5,4),

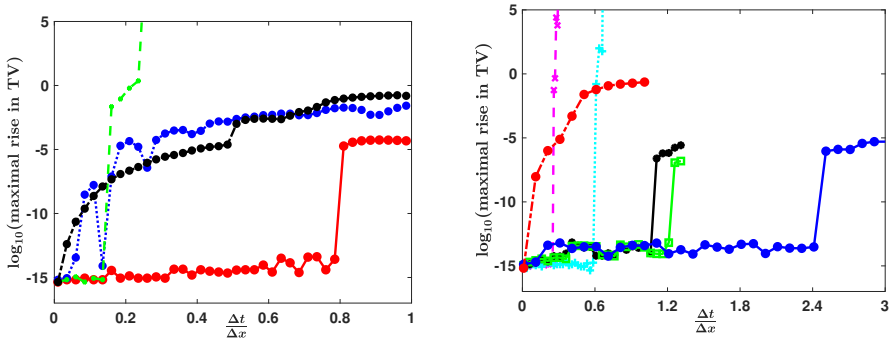

FIG. 7. *Example* 4. *Left:* $a = 5$. *The red solid line is the eSSPIFRK*(3,3) *method, the blue dotted line is the IFRK method based on the eSSPRK*(3,3) *Shu–Osher method, the black dash-dot line is the ETDRK*3 *method, and the green dashed line is the third order explicit eSSPRK*(3,3) *Shu–Osher method. Right:* $a = 10$. *The dashed magenta line with x markers is the SSP IMEX* (5,4) *method. The dash-dot red line with filled circle markers is the ETDRK*4 *method. The dotted cyan line with + markers is the SSPRK*(10,4) *method by Ketcheson. The solid lines are the eSSPIFRK methods: the black line with star markers for* (5,4)*, green line with square markers for* (6,4)*, and blue line with circle markers for* (9,4).

eSSPIFRK(6,4), and eSSPIFRK(9,4) methods maintain a very small maximal rise in total variation until close to $\lambda = 1.06, 1.21, 2.41$, respectively. In comparison, the SSP IMEX (5,4) method features an observed λ value of $\lambda_{obs} = 0.25$, the fully explicit SSPRK(10,4) has $\lambda_{obs} = 0.58$, and the maximal total variation from the simulation using the ETDRK4 method starts rising rapidly from the smallest value of λ.

7. Conclusions. This is the first work to consider SSP IFRK methods. In this work we presented sufficient conditions for preservation of strong stability for IFRK methods. These eSSPIFRK methods are based on eSSPRK methods with nondecreasing abscissas. We used these conditions to develop an optimization problem which we used to find such eSSPRK$^+$ methods. We then showed that these eSSPIFRK methods perform in practice as expected, significantly out-performing the IMEX SSP Runge–Kutta methods and the ETD methods of Cox and Matthews on problems that require the SSP property.

Acknowledgments. The authors thank David Ketcheson and the anonymous referees for their helpful remarks on this work.

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 J. Sci. Comput., 33 (2011), pp. 488–511.

[2] L. Baiotti, I. Hawke, P.J. Montero, F. Loffler, L. Rezzolla, N. Stergioulas, J.A. Font, and E. Seidel. *Three-dimensional relativistic simulations of rotating neutron-star collapse to a Kerr black hole*, Phys. Rev. D, 71 (2005), pp. 24–35.

[3] J. Balbás and E. Tadmor. *A central differencing simulation of the Orszag-Tang vortex system*, IEEE Trans. Plasma Sci., 33 (2005), pp. 470–471.

[4] E. Bassano, *Numerical simulation of thermo-solutal-capillary migration of a dissolving drop in a cavity*, Int. J. Numer. Methods Fluids, 41 (2003), pp. 765–788.

[5] H. Berland, B. Skaflestad, and W. M. Wright, *EXPINT – A MATLAB Package for Exponential Integrators*, 2005.

[6] H. Berland, B. Skaflestad, and W. M. Wright, *EXPINT – A MATLAB Package for Exponential Integrators*, ACM Trans. Math. Software, 33 (2007), 4.

[7] J.C. Butcher, *The Numerical Analysis of Ordinary Differential Equations: Runge-Kutta and General Linear Methods*, Wiley, New York, 1987.

[8] R. Caiden, R.P. Fedkiw, and C. Anderson, *A numerical method for two-phase flow consisting of separate compressible and incompressible regions*, J. Comput. Phys., 166 (2001), pp. 1–27.

[9] J. Carrillo, I.M. Gamba, A. Majorana, and C.-W. Shu, *A WENO-solver for the transients of Boltzmann-Poisson system for semiconductor devices: Performance and comparisons with Monte Carlo methods*, J. Comput. Phys., 184 (2003), pp. 498–525.

[10] 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 Schrodinger equations*, Commun. Math. Sci., 1 (2003), pp. 593–621.

[11] V. Cheruvu, R.D. Nair, and H.M. Tufo, *A spectral finite volume transport scheme on the cubed-sphere*, Appl. Numer. Math., 57 (2007), pp. 1021–1032.

[12] B. Cockburn and C.-W. Shu, *TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws* II*: General framework*, Math. Comp., 52 (1989), pp. 411–435.

[13] B. Cockburn, F. Li, and C.-W. Shu, *Locally divergence-free discontinuous Galerkin methods for the Maxwell equations*, J. Comput. Phys., 194 (2004), pp. 588–610.

[14] B. Cockburn, J. Qian, F. Reitich, and J. Wang, *An accurate spectral/discontinuous finite-element formulation of a phase-space-based level set approach to geometrical optics*, J. Comput. Phys., 208 (2005), pp. 175–195.

[15] S. Conde, S. Gottlieb, Z. Grant, and J.N. Shadid, *Implicit and implicit-explicit strong stability preserving RungeKutta methods with high linear order*, J. Sci. Comput., 73 (2017), pp. 667–690.

[16] S. Cox and P. Matthews, *Exponential time differencing for stiff systems*, J. Comput. Phys., 176 (2002), pp. 430–455.

[17] L. Del Zanna and N. Bucciantini, *An efficient shock-capturing central-type scheme for multidimensional relativistic flows:* I. *Hydrodynamics*, Astronom. Astrophys., 390 (2002), pp. 1177–1186.
[18] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell, *A hybrid particle level set method for improved interface capturing*, J. Comput. Phys., 183 (2002), pp. 83–116.
[19] L.-L. Feng, C.-W. Shu, and M. Zhang, *A hybrid cosmological hydrodynamic/n-body code based on a weighted essentially nonoscillatory scheme*, Astrophys. J., 612 (2004), pp. 1–13.
[20] L. Ferracina and M.N. Spijker, *Stepsize restrictions for the total-variation-diminishing property in general Runge-Kutta methods*, SIAM J. Numer. Anal., 42 (2004), pp. 1073–1093.
[21] L. Ferracina and M.N. Spijker, *An extension and analysis of the Shu-Osher representation of Runge-Kutta methods*, Math. Comp., 249 (2005), pp. 201–219.
[22] L. Ferracina and M. N. Spijker, *Strong stability of singly-diagonally-implicit Runge-Kutta methods*, Appl. Numer. Math., 58 (2008), pp. 1675–1686.
[23] S. Gaudreault, G. Rainwater, and M. Tokman, *KIOPS: A fast adaptive Krylov subspace solver for exponential integrators*, J. Comput. Phys., 372 (2018), pp. 236–255.
[24] D. Gottlieb and J. S. Hesthaven, *Spectral methods for hyperbolic problems*, J. Comput. Appl. Math., 128 (2001), pp. 83–131.
[25] D. Gottlieb and E. Tadmor *The CFL condition for spectral approximations to hyperbolic initial-boundary value problems*, Math. Comp., 56 (1991), pp. 565–588.
[26] S. Gottlieb, Z. Grant, and L. Isherwood, *Optimized Strong Stability Preserving Integrating Factor Runge–Kutta Methods*, 2017, https://github.com/SSPmethods/SSPIFRK-methods.
[27] S. Gottlieb, D. I. Ketcheson, and C.-W. Shu, *Strong Stability Preserving Runge–Kutta and Multistep Time Discretizations*, World Scientific Singapore, 2011.
[28] S. Gottlieb and C.-W. Shu, *Total variation diminishing Runge–Kutta methods*, Math. Comp., 67 (1998), pp. 73–85.
[29] S. Gottlieb, C.-W. Shu, and E. Tadmor, *Strong stability preserving high-order time discretization methods*, SIAM Rev., 43 (2001), pp. 89–112.
[30] A. Harten, *High resolution schemes for hyperbolic conservation laws*, J. Comput. Phys., 49 (1983), pp. 357–393.
[31] J.S. Hesthaven, *Numerical Methods for Conservation Laws: From Analysis to Algorithms*, SIAM, Philadelphia, 2017.
[32] J.S. Hesthaven, S. Gottlieb, and D. Gottlieb, *Spectral Methods for Time Dependent Problems*, Cambridge Monogr. Appl. Comput. Math. 21, Cambridge University Press, Cambridge, UK, 2006.
[33] I. Higueras, *On strong stability preserving time discretization methods*, J. Sci. Comput., 21 (2004), pp. 193–223.
[34] I. Higueras, *Representations of Runge-Kutta methods and strong stability preserving methods*, SIAM J. Numer. Anal., 43 (2005), pp. 924–948.
[35] X. Y. Hu, N. A. Adams, and C.-W. Shu, *Positivity-preserving method for high-order conservative schemes solving compressible Euler equations*, J. Comput. Phys., 242 (2013), pp. 169–180.
[36] W. Hundsdorfer, S.J. Ruuth, and R.J. Spiteri, *Monotonicity-preserving linear multistep methods*, SIAM J. Numer. Anal., 41 (2003), pp. 605–623.
[37] S. Jin, H. Liu, S. Osher, and Y.-H.R. Tsai, *Computing multivalued physical observables for the semiclassical limit of the Schrodinger equation*, J. Comput. Phys., 205 (2005), pp. 222–241.
[38] C.A. Kennedy, M.H. Carpenter, and R.M. Lewis, *Low-storage explicit Runge–Kutta schemes for the compressible Navier-Stokes equations*, Appl. Numer. Math., 35 (2000), pp. 177–219.
[39] D. I. Ketcheson, *Highly efficient strong stability preserving Runge–Kutta methods with low-storage implementations*, SIAM J. Sci. Comput., 30 (2008), pp. 2113–2136.
[40] D.I. Ketcheson, C.B. Macdonald, and S. Gottlieb, *Optimal implicit strong stability preserving Runge-Kutta methods*, Appl. Numer. Math., 52 (2009), pp. 373–392.
[41] D. I. Ketcheson, *Computation of optimal monotonicity preserving general linear methods*, Math. Comp., 78 (2009), pp. 1497–1513.
[42] D. I. Ketcheson, *Step sizes for strong stability preservation with downwind-biased operators*, SIAM J. Numer. Anal., 49 (2011), pp. 1649–1660.
[43] Kinnmark and Gray, *One step integration methods with maximum stability regions*, Math. Comput. Simulation, 26 (2) (1984), pp. 87–92.

[44] J. F. B. M. Kraaijevanger, *Contractivity of Runge–Kutta methods*, BIT, 31 (1991), pp. 482–528.
[45] E.J. Kubatko, B. A. Yeager, and D. I. Ketcheson, *Optimal strong-stability-preserving RungeKutta time discretizations for discontinuous Galerkin methods*, J. Sci. Comput., 60 (2014), pp. 313–344.
[46] A. Kurganov and E. Tadmor, *New high-resolution schemes for nonlinear conservation laws and convection-diffusion equations*, J. Comput. Phys., 160 (2000), pp. 241–282.
[47] S. Labrunie, J. Carrillo, and P. Bertrand, *Numerical study on hydrodynamic and quasi-neutral approximations for collisionless two-species plasmas*, J. Comput. Phys., 200 (2004), pp. 267–298.
[48] J. D. Lawson, *Generalized Runge-Kutta processes for stable systems with large Lipschitz constants*, SIAM J. Numer. Anal., 4 (1967), pp. 372–380.
[49] P.D. Lax, *Gibbs phenomena*, J. Sci. Comput., 28 (2006), pp. 445–449.
[50] H.W.J. Lenferink, *Contractivity-preserving implicit linear multistep methods*, Math. Comp., 56 (1991), pp. 177–199.
[51] R. J. LeVeque, *Numerical Methods for Conservation Laws*, ETH Lectures in Mathematics Series, Birkhauser-Verlag, Basel, Switzerland, 1990.
[52] D. Levy and E. Tadmor, *From semi-discrete to fully-discrete: The stability of Runge-Kutta schemes by the energy method*, SIAM Rev., 40 (1998), pp. 40–73.
[53] X.-D. Liu, S. Osher, and T. Chan, *Weighted essentially non-oscillatory schemes*, J. Comput. Phys., 115 (1994), pp. 200–212.
[54] H. Ma, *Chebyshev–Legendre super spectral viscosity method for nonlinear conservation laws*, SIAM J. Numer. Anal., 35 (1998), pp. 893–908.
[55] Y. Maday and E. Tadmor, *Analysis of the spectral vanishing viscosity method for periodic conservation laws*, SIAM J. Numer. Anal., 26 (1989), pp. 854–870.
[56] A. Majda and S. Osher, *A systematic approach for correcting nonlinear instabilities. The Lax-Wendroff scheme for scalar conservation laws*, Numer. Math., 30 (1978), pp. 429–452.
[57] A. Mignone, *The dynamics of radiative shock waves: Linear and nonlinear evolution*, Astrophys. J., 626 (2005), pp. 373–388.
[58] G. Golub and C. Van Loan, *Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later*, SIAM Rev., 45 (2003), pp. 801–836.
[59] J. Niesen and W.M. Wright, *Algorithm* 919*: A Krylov subspace algorithm for evaluating the ϕ-functions appearing in exponential integrators*, ACM Trans. Math. Softw., 38 (2012), pp. 22.
[60] S. Osher and S. Chakravarthy, *High resolution schemes and the entropy condition*, SIAM J. Numer. Anal., 21 (1984), pp. 955–984.
[61] C. Pantano, R. Deiterding, D.J. Hill, and D.I. Pullin, *A low numerical dissipation patch-based adaptive mesh refinement method for large-eddy simulation of compressible flows*, J. Comput. Phys., 221 (2007), pp. 63–87.
[62] S. Patel and D. Drikakis, *Effects of preconditioning on the accuracy and efficiency of incompressible flows*, Int. J. Numer. Methods Fluids, 47 (2005), pp. 963–970.
[63] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang, *A PDE-based fast local level set method*, J. Comput. Phys., 155 (1999), pp. 410–438.
[64] S. C.Reddy and L. N.Trefethen, *Lax stability of fully discrete spectral methods via stability regions and pseudo-eigenvalues*, Comput. Methods Appl. Mech. Engrg., 80 (1990), pp. 147–164.
[65] S. J. Ruuth and R. J. Spiteri, *Two barriers on strong-stability-preserving time discretization methods*, J. Sci. Comput., 17 (2002), pp. 211–220.
[66] C.-W. Shu, *TVB uniformly high-order schemes for conservation laws*, Math. Comp., 49 (1987), pp. 105–121.
[67] C.-W. Shu, *Total-variation diminishing time discretizations*, SIAM J. Statist. Sci. Comput., 9 (1988), pp. 1073–1084.
[68] C.-W. Shu and S. Osher, *Efficient implementation of essentially non-oscillatory shock-capturing schemes*, J. Comput. Phys., 77 (1988), pp. 439–471.
[69] R.B. Sidje, *EXPOKIT: A software package for computing matrix exponentials*, ACM Trans. Math. Softw., 24 (1998), pp. 130–156.
[70] M.N. Spijker, *Contractivity in the numerical solution of initial value problems*, Numer. Math., 42 (1983), pp. 271–290.
[71] M. Spijker, *Stepsize conditions for general monotonicity in numerical initial value problems*, SIAM J. Numer. Anal., 45 (2008), pp. 1226–1245.

[72] 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.
[73] G. Strang, *Accurate partial difference methods* II*: Nonlinear problems*, Numer. Math., 6 (1964), pp. 37–46.
[74] J.C. Strikwerda, *Finite Difference Schemes and Partial Differential Equations*, Cole Mathematics Series, Wadsworth and Brooks, Belmont, CA, 1989.
[75] Y. Sun, Z.J. Wang, and Y. Liu, *Spectral (finite) volume method for conservation laws on unstructured grids* VI*: Extension to viscous flow*, J. Comput. Phys., 215 (2006), pp. 41–58.
[76] P.K. Sweby, *High resolution schemes using flux limiters for hyperbolic conservation laws*, SIAM J. Numer. Anal., 21 (1984), pp. 995–1011.
[77] E. Tadmor, *Approximate solutions of nonlinear conservation laws*, in Advanced Numerical Approximation of Nonlinear Hyperbolic Equations: Lectures Given at the 2nd Session of the Centro Internazionale Matematic Estivo (C.I.M.E.) Held in Centraro, Italy, June 23–28, 1997, A. Quarteroni, ed., Lect. Notes Math. 1697, Springer-Verlag, Berlin, 1998, pp. 1–150.
in Advanced Numerical Approximation of Nonlinear Hyperbolic Equations," Lectures Notes from CIME Course Cetraro, Italy, 1997, pp. 1–150 in Approximate solutions of nonlinear conservation laws, Number 1697 in Lecture Notes in Mathematics. Springer-Verlag, 1998,
[78] E. Tadmor, *Shock capturing by the spectral viscosity method*, Comput. Methods Appl. Mech. Engrg., 80 (1990), pp. 197-208.
[79] M. Tanguay and T. Colonius, *Progress in modeling and simulation of shock wave lithotripsy*, in Proceedings of the Fifth International Symposium on Cavitation, 2003.
[80] P.J. van der Houwen, *Explicit Runge–Kutta formulas with increased stability boundaries*, Numer. Math., 20 (1972), pp. 149–164.
[81] Z.J. Wang and Y. Liu, *The spectral difference method for the* 2*D Euler equations on unstructured grids*, in Proceedings of the 17th AIAA Computational Fluid Dynamics Conference, AIAA, 2005.
[82] Z.J. Wang, Y. Liu, G. May, and A. Jameson, *Spectral difference method for unstructured grids* II*: Extension to the Euler equations*, J. Sci. Comput., 32 (2007), pp. 45–71.
[83] J. H. Williamson, *Low storage Runge–Kutta schemes*, J. Comput. Phys., 35 (1980), pp. 48-56.
[84] W. Zhang and A.I. MacFayden, *RAM: A relativistic adaptive mesh refinement hydrodynamics code*, Astrophys. J. Suppl. Ser., 164 (2006), pp. 255–279.
[85] X. Zhang and C.-W. Shu, *On maximum-principle-satisfying high order schemes for scalar conservation laws*, J. Comput. Phys., 229 (2010), pp. 3091–3120.
[86] X. Zhang and C.-W. Shu, *On positivity-preserving high order discontinuous Galerkin schemes for compressible Euler equations on rectangular meshes,* J. Comput. Phys., 229 (2010), pp. 8918–8934.
[87] X. Zhang and C.-W. Shu, *Maximum-principle-satisfying and positivity-preserving high-order schemes for conservation laws: survey and new developments*, Proc. Roy. Soc. Edinburgh Sect. A, 467 (2011), pp. 2752–2776.
[88] X. Zhang and C.-W. Shu, *Positivity-preserving high order discontinuous Galerkin schemes for compressible Euler equations with source terms*, J. Comput. Phys., 230 (2011), pp. 1238–1248.
[89] X. Zhang and C.-W. Shu, *Positivity-preserving high order finite difference WENO schemes for compressible Euler equations*, J. Comput. Phys., 231 (2012), pp. 2245–2258.