Explicit Strong Stability Preserving Multistage Two-Derivative Time-Stepping Schemes

Andrew J. Christlieb[1], Sigal Gottlieb[2], Zachary Grant[2]*, David C. Seal[3]

[1]Department of Computational Mathematics Science and Engineering, Department of Electrical Engineering, and Department of Mathematics, Michigan State University

[2]Department of Mathematics, University of Massachusetts, Dartmouth

[3]Department of Mathematics, U.S. Naval Academy.

Abstract

High order strong stability preserving (SSP) time discretizations are advantageous for use with spatial discretizations with nonlinear stability properties for the solution of hyperbolic PDEs. The search for high order strong stability time-stepping methods with large allowable strong stability time-step has been an active area of research over the last two decades. Recently, multiderivative time-stepping methods have been implemented with hyperbolic PDEs. In this work we describe sufficient conditions for a two-derivative multistage method to be SSP, and find some optimal SSP multistage two-derivative methods. While explicit SSP Runge–Kutta methods exist only up to fourth order, we show that this order barrier is broken for explicit multi-stage two-derivative methods by designing a three stage fifth order SSP method. These methods are tested on simple scalar PDEs to verify the order of convergence, and demonstrate the need for the SSP condition and the sharpness of the SSP time-step in many cases.

1 Introduction

1.1 SSP methods

When numerically approximating the solution to a hyperbolic conservation law of the form

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

difficulties arise when the exact solution develops sharp gradients or discontinuities. Significant effort has been expended on developing spatial discretizations that can handle discontinuities [7], especially for high-order methods. These discretizations have special nonlinear non-inner-product stability properties, such as total variation stability or positivity, which ensure that when the semi-discretized equation

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

(where u is a vector of approximations to U) is evolved using a forward Euler method, the numerical solution satisfies the desired strong stability property,

$$\|u^n + \Delta t F(u^n)\| \leq \|u^n\|, \quad 0 \leq \Delta t \leq \Delta t_{FE}, \tag{3}$$

where $\|\cdot\|$ is any desired norm, semi-norm, or convex functional.

In place of the first order time discretization (3), we typically require a higher-order time integrator, but we still wish to ensure that the strong stability property $\|u^{n+1}\| \leq \|u^n\|$ is satisfied, perhaps under a modified time-step restriction, where u^n is a discrete approximation to U at time t^n. In [32] it was observed that some Runge–Kutta methods can be decomposed into convex combinations of forward Euler steps, so that any convex functional

*Corresponding author: zgrant@umassd.edu

property satisfied by (3) will be *preserved* by these higher-order time discretizations. For example, the s-stage explicit Runge–Kutta method [33],

$$\begin{aligned} y^{(0)} &= u^n, \\ y^{(i)} &= \sum_{j=0}^{i-1} \left(\alpha_{i,j} y^{(j)} + \Delta t \beta_{i,j} F(y^{(j)}) \right), \quad i = 1, ..., s \\ u^{n+1} &= y^{(s)} \end{aligned} \tag{4}$$

can be rewritten as convex combination of forward Euler steps of the form (3). If all the coefficients $\alpha_{i,j}$ and $\beta_{i,j}$ are non-negative, and provided $\alpha_{i,j}$ is zero only if its corresponding $\beta_{i,j}$ is zero, then each stage is bounded by

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

Noting that each $\|y^{(j)} + \Delta t \frac{\beta_{i,j}}{\alpha_{i,j}} F(y^{(j)})\| \leq \|y^{(j)}\|$ for $\frac{\beta_{i,j}}{\alpha_{i,j}} \Delta t \leq \Delta t_{\text{FE}}$, and by consistency $\sum_{j=0}^{i-1} \alpha_{i,j} = 1$, we have $\|u^{n+1}\| \leq \|u^n\|$ as long as

$$\Delta t \leq \mathcal{C} \Delta t_{\text{FE}} \quad \forall i, j, \tag{5}$$

where $\mathcal{C} = \min \frac{\alpha_{i,j}}{\beta_{i,j}}$. (We employ the convention that if any of the β's are equal to zero, the corresponding ratios are considered infinite.) The resulting time-step restriction is a combination of two distinct factors: (1) the term Δt_{FE} that depends on the spatial discretization, and (2) the SSP coefficient $\mathcal{C}$ that depends only on the time-discretization. Any method that admits such a decomposition with $\mathcal{C} > 0$ is called a *strong stability preserving (SSP)* method.

This convex combination decomposition was used in the development of second and third order explicit Runge–Kutta methods [33] and later of fourth order methods [34, 16] that guarantee the strong stability properties of any spatial discretization, provided only that these properties are satisfied when using the forward Euler (first derivative) condition in (3). Additionally, the convex combination approach also guarantees that the intermediate stages in a Runge–Kutta method satisfy the strong stability property as well.

The convex combination approach clearly provides a sufficient condition for preservation of strong stability. Moreover, it has also be shown that this condition is necessary [4, 5, 11, 12]. Much research on SSP methods focuses on finding high-order time discretizations with the largest allowable time-step $\Delta t \leq \mathcal{C} \Delta t_{\text{FE}}$ by maximizing the *SSP coefficient* $\mathcal{C}$ of the method. It has been shown that explicit Runge–Kutta methods with positive SSP coefficient cannot be more than fourth-order accurate [20, 28]; this led to the study of other classes of explicit SSP methods, such as methods with multiple steps. Explicit multistep SSP methods of order $p > 4$ do exist, but have severely restricted time-step requirements [7]. Explicit multistep multistage methods that are SSP and have order $p > 4$ have been developed as well [17, 2].

Recently, multi-stage multiderivative methods have been proposed for use with hyperbolic PDEs [29, 37]. The question then arises as to whether these methods can be strong stability preserving as well. Nguyen-Ba and colleagues studied the SSP properties of the Hermite-Birkoff-Taylor methods with a set of simplified base conditions in [23]. In this work we consider multistage two-derivative methods and develop sufficient conditions for strong stability preservation for these methods, and we show that explicit SSP methods within this class can break this well-known order barrier for explicit Runge–Kutta methods. Numerical results demonstrate that the SSP condition is useful in preserving the nonlinear stability properties of the underlying spatial discretization and that the allowable time-step predicted by the SSP theory we developed is sharp in many cases.

1.2 Multistage multiderivative methods

To increase the possible order of any method, we can use more steps (e.g. linear multistep methods), more stages (e.g. Runge–Kutta methods), or more derivatives (Taylor series methods). It is also possible to combine these

approaches to obtain methods with multiple steps, stages, and derivatives. Multistage multiderivative integration methods were first considered in [24, 38, 35], and multiderivative time integrators for ordinary differential equations have been developed in [30, 31, 14, 15, 22, 25, 3], but only recently have these methods been explored for use with partial differential equations (PDEs) [29, 37]. In this work, we consider explicit multistage two-derivative time integrators as applied to the numerical solution of hyperbolic conservation laws.

We consider the system of ODEs (2) resulting from the spatial discretization of a hyperbolic PDE of the form (1). We define the one-stage, two-derivative building block method $u^{n+1} = u^n + \alpha\Delta t F(u^n) + \beta\Delta t^2 \dot{F}(u^n)$ where $\alpha \geq 0$ and $\beta \geq 0$ are coefficients chosen to ensure the desired order. This method can be at most second order, with coefficients $\alpha = 1$ and $\beta = \frac{1}{2}$. This is the second-order Taylor series method. To obtain higher order explicit methods, we can add more stages:

$$\begin{aligned} y^{(i)} &= u^n + \Delta t \sum_{j=1}^{i-1} \left(a_{ij} F(y^{(j)}) + \Delta t \hat{a}_{ij} \dot{F}(y^{(j)}) \right), \quad i = 1, ..., s \\ u^{n+1} &= u^n + \Delta t \sum_{j=1}^{s} \left(b_j F(y^{(j)}) + \Delta t \hat{b}_j \dot{F}(y^{(j)}) \right). \end{aligned} \tag{6}$$

We can write the coefficients in matrix vector form, where

$$A = \begin{pmatrix} 0 & 0 & \vdots & 0 \\ a_{21} & 0 & \vdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ a_{s1} & a_{s2} & \vdots & 0 \end{pmatrix}, \quad \hat{A} = \begin{pmatrix} 0 & 0 & \vdots & 0 \\ \hat{a}_{21} & 0 & \vdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \hat{a}_{s1} & \hat{a}_{s2} & \vdots & 0 \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ b_2 \\ \\ \vdots \\ b_s \end{pmatrix}, \quad \hat{b} = \begin{pmatrix} \hat{b}_1 \\ \hat{b}_2 \\ \\ \vdots \\ \hat{b}_s \end{pmatrix}.$$

We let $c = A\mathbf{e}$ and $\hat{c} = \hat{A}\mathbf{e}$, where $\mathbf{e}$ is a vector of ones. These coefficients are then selected to attain the desired order, based on the order conditions written in Table 1 as described in [3, 6].

Remark 1. *In this work, we focus on multistage two-derivative methods as time integrators for use with hyperbolic PDEs. In this setting, the operator F is obtained by a spatial discretization of the term $U_t = -f(U)_x$ to obtain the system $u_t = F(u)$. This is the typical method-of-lines approach, and SSP methods were introduced in the context of this approach. The computation of the second derivative term $\dot{F}$ should follow directly from the definition of F, where we compute $\dot{F} = F(u)_t = F_u u_t = F_u F$. In practice, the calculation of F_u may be computationally prohibitive, as for example in the popular WENO method where F has a highly nonlinear dependence on u.*

Instead, we adopt a Lax-Wendroff type approach, where we use the fact that the system of ODEs arises from the PDE (1) to replace the time derivatives by the spatial derivatives, and discretize these in space. This approach begins with the observation that $F(u) = u_t = U_t + O(\Delta x^m)$ (for some integer m). The term $F(u)$ is typically computed using a conservative spatial discretization D_x applied to the flux:

$$F(u) = D_x(-f(u)).$$

Next we approximate

$$F(u)_t = u_{tt} \approx U_{tt} = -f(U)_{xt} = (-f(U)_t)_x = \left(-f'(U)U_t\right)_x \approx \tilde{D}_x \left(-f'(u)u_t\right),$$

where a (potentially different) spatial approximation $\tilde{D}_x$ is used. This means that

$$F(u)_t = u_{tt} = U_{tt} + O(\Delta x^n) = \dot{F} + O(\Delta x^r)$$

(for some integers n and r).

Since $F_t = \dot{F} + O(\Delta x^r)$, we will not necessarily obtain the time order Δt^p when we satisfy the order conditions in Table 1, as our temporal order is polluted by spatial errors as well. The concern about the Lax-Wendroff approach is that the order conditions in Table 1 are based on the assumption that $\dot{F} = F_t$, which is not exactly correct in our case, thus introducing additional errors. However, these errors are of order r in space, so that in practice, as long as the spatial errors are smaller than the temporal errors, we expect to see the correct order of accuracy in time.

To verify this, in Section 4.2 we perform numerical convergence studies of these temporal methods where $\dot{F}$ is approximated by high order spatial schemes and compare the errors from these methods to those from well-known SSP Runge–Kutta methods. We observe that if the spatial and temporal grids are refined together, the expected order of accuracy is demonstrated. Furthermore, if the spatial grid is held fixed but the spatial discretization is highly accurate, the correct time-order is observed until the time error falls below the spatial error. We conclude that in practice, it is not necessary to compute F_t exactly, and that the use of a Lax-Wendroff type procedure that replaces the temporal derivatives by spatial derivatives and discretizes each of these independently, does not destroy the temporal accuracy.

This observation is not new: many other methods have adopted this type of approach and obtained genuine high-order accuracy. Such methods include the original Lax-Wendroff method [21], ENO methods [9], finite volume ADER methods [36], the finite difference WENO Schemes in [27], and the Lax-Wendroff discontinuous Galerkin schemes [26].

In the next section we will discuss how to ensure that a multistage two-derivative method will preserve these strong stability properties.

$p = 1$	$b^T e = 1$
$p = 2$	$b^T c + \hat{b}^T e = \frac{1}{2}$
$p = 3$	$b^T c^2 + 2\hat{b}^T c = \frac{1}{3}$
	$b^T Ac + b^T \hat{c} + \hat{b}^T c = \frac{1}{6}$
$p = 4$	$b^T c^3 + 3\hat{b}^T c^2 = \frac{1}{4}$
	$b^T cAc + b^T c\hat{c} + \hat{b}^T c^2 + \hat{b}^T Ac + \hat{b}^T \hat{c} = \frac{1}{8}$
	$b^T Ac^2 + 2b^T \hat{A}c + \hat{b}^T c^2 = \frac{1}{12}$
	$b^T A^2 c + b^T A\hat{c} + b^T \hat{A}c + \hat{b}^T Ac + \hat{b}^T \hat{c} = \frac{1}{24}$
$p = 5$	$b^T c^4 + 4\hat{b}^T c^3 = \frac{1}{5}$
	$b^T c^2 Ac + b^T c^2 \hat{c} + \hat{b}^T c^3 + 2\hat{b}^T cAc + 2\hat{b}^T c\hat{c} = \frac{1}{10}$
	$b^T cAc^2 + 2b^T c\hat{A}c + \hat{b}^T c^3 + \hat{b}^T Ac^2 + 2\hat{b}^T \hat{A}c = \frac{1}{15}$
	$b^T cA^2 c + b^T cA\hat{c} + b^T c\hat{A}c + \hat{b}^T cAc + \hat{b}^T c\hat{c} + \hat{b}^T A^2 c + \hat{b}^T A\hat{c} + \hat{b}^T \hat{A}c = \frac{1}{30}$
	$b^T (Ac)(Ac) + 2b^T \hat{c}Ac + b^T \hat{c}^2 + 2\hat{b}^T cAc + 2\hat{b}^T c\hat{c} = \frac{1}{20}$
	$b^T Ac^3 + 3b^T \hat{A}c^2 + \hat{b}^T c^3 = \frac{1}{20}$
	$b^T A(cAc) + b^T A(c\hat{c}) + b^T \hat{A}c^2 + b^T \hat{A}Ac + b^T \hat{A}\hat{c} + \hat{b}^T cAc + \hat{b}^T c\hat{c} = \frac{1}{40}$
	$b^T A^2 c^2 + 2b^T A\hat{A}c + b^T \hat{A}c^2 + \hat{b}^T Ac^2 + 2\hat{b}^T \hat{A}c = \frac{1}{60}$
	$b^T A^3 c + b^T A^2 \hat{c} + b^T A\hat{A}c + b^T \hat{A}Ac + b^T \hat{A}\hat{c} + \hat{b}^T A^2 c + \hat{b}^T A\hat{c} + \hat{b}^T \hat{A}c = \frac{1}{120}$

Table 1: Order conditions for multi-stage multiderivative methods of the form (6) as in [3].

2 The SSP condition for multiderivative methods

2.1 Motivating Examples

To understand the strong stability condition for multiderivative methods, we consider the strong stability properties of a multiderivative building block of the form

$$u^{n+1} = u^n + \alpha \Delta t F(u^n) + \beta \Delta t^2 \dot{F}(u^n),$$

and begin with the simple linear one-way wave equation $U_t = U_x$. This equation has the property that its second derivative in time is, with the assumption of sufficient smoothness, also the second derivative in space:

$$U_{tt} = (U_x)_t = (U_t)_x = U_{xx}.$$

We will use this convenient fact in a Lax-Wendroff type approach to define $\dot{F}(u^n)$ by a spatial discretization of U_{xx}.

For this problem, we define F by the original first-order upwind method

$$F(u^n)_j := \frac{1}{\Delta x}\left(u^n_{j+1} - u^n_j\right) \approx U_x(x_j), \tag{7a}$$

and $\dot{F}$ by the second order centered discretization to U_{xx}:

$$\dot{F}(u^n)_j := \frac{1}{\Delta x^2}\left(u^n_{j+1} - 2u^n_j + u^n_{j-1}\right) \approx U_{xx}(x_j). \tag{7b}$$

These spatial discretizations are total variation diminishing (TVD) in the following sense:

$$u^{n+1} = u^n + \Delta t F(u^n) \quad \text{is TVD for} \quad \Delta t \le \Delta x, \tag{8a}$$

$$u^{n+1} = u^n + \Delta t^2 \dot{F}(u^n) \quad \text{is TVD for} \quad \Delta t \le \frac{\sqrt{2}}{2}\Delta x. \tag{8b}$$

Remark 2. *Note that we chose a second derivative $\dot{F}$ in space that is not an* exact *derivative $F_t(u^n)$ of $F(u^n)$ in the method-of-lines formulation. However, as noted in Remark 1 and will be shown in the convergence studies, if the spatial and temporal grids are co-refined, this provides a sufficiently accurate approximation to $\dot{F}(u^n)$. The exact derivative can be obtained by applying the upwind differentiation operator to the solution twice, which produces*

$$F_t := \frac{1}{\Delta x^2}\left(u^n_{j+2} - 2u^n_{j+1} + u^n_j\right).$$

However, computing $\dot{F} = F_t$ using this formulation does not satisfy the condition (8b) for any value of Δt.

To establish the TVD properties of the multiderivative building block we decompose it:

$$\begin{aligned} u^{n+1} &= u^n + \alpha \Delta t F(u^n) + \beta \Delta t^2 \dot{F}(u^n) = a u^n + \alpha \Delta t F(u^n) + (1-a)u^n + \beta \Delta t^2 \dot{F}(u^n) \\ &= a\left(u^n + \frac{\alpha}{a}\Delta t F(u^n)\right) + (1-a)\left(u^n + \frac{\beta}{1-a}\Delta t^2 \dot{F}(u^n)\right). \end{aligned}$$

It follows that for any $0 \le a \le 1$ this is a convex combination of terms of the form (8a) and (8b), and so

$$\begin{aligned} \|u^{n+1}\|_{TV} &\le a\left\|\left(u^n + \frac{\alpha}{a}\Delta t F(u^n)\right)\right\|_{TV} + (1-a)\left\|\left(u^n + \frac{\beta}{1-a}\Delta t^2 \dot{F}(u^n)\right)\right\|_{TV} \\ &\le a\|u^n\|_{TV} + (1-a)\|u^n\|_{TV} \le \|u^n\|_{TV} \end{aligned}$$

for time-steps satisfying $\Delta t \le \frac{a}{\alpha}\Delta x$ and $\Delta t^2 \le \frac{1-a}{2\beta}\Delta x^2$. The first restriction relaxes as a increases while the second becomes tighter as a increases, so that the value of a that maximizes these conditions occurs when these are equal. This is given by

$$a^2 + \frac{\alpha^2}{2\beta}a - \frac{\alpha^2}{2\beta} = 0, \quad \Longrightarrow \quad a = \frac{\alpha\sqrt{\alpha^2+8\beta}-\alpha^2}{4\beta}.$$

Using this SSP analysis, we conclude that

$$\left\| u^n + \alpha\Delta t F(u^n) + \beta\Delta t^2 \dot{F}(u^n) \right\|_{TV} \le \|u^n\|_{TV} \quad \text{for} \quad \Delta t \le \frac{\sqrt{\alpha^2+8\beta}-\alpha}{4\beta}\Delta x. \tag{9}$$

Of course, for this simple example we can directly compute the value of Δt for which the multiderivative building block is TVD. That is, with $\lambda := \frac{\Delta t}{\Delta x} \ge 0$, we observe that

$$\|u^{n+1}\|_{TV} = \left\| \left((1-\alpha\lambda-2\beta\lambda^2)u_j^n + (\alpha\lambda+\beta\lambda^2)u_{j+1}^n + \beta\lambda^2 u_{j-1}^n\right) \right\|_{TV} \le \|u^n\|_{TV},$$

provided that

$$1-\alpha\lambda-2\beta\lambda^2 \ge 0 \iff \lambda \le \frac{\sqrt{\alpha^2+8\beta}-\alpha}{4\beta}.$$

We see that for this case, the SSP bound is sharp: the convex combination approach provides us exactly the same bound as directly computing the requirements for total variation.

We wish to generalize this for cases in which the second derivative condition (8b) holds for $\Delta t \le K\Delta t_{\text{FE}}$ where K can take on any positive value, not just $\frac{\sqrt{2}}{2}$. For the two-derivative building block method this can be done quite easily:

Theorem 1. *Given F and $\dot{F}$ such that*

$$\|u^n + \Delta t F(u^n)\| \le \|u^n\| \quad \textit{for} \quad \Delta t \le \Delta t_{\text{FE}},$$

and

$$\|u^n + \Delta t^2 \dot{F}(u^n)\| \le \|u^n\| \quad \textit{for} \quad \Delta t \le K\Delta t_{\text{FE}},$$

the two-derivative building block

$$u^{n+1} = u^n + \alpha\Delta t F(u^n) + \beta\Delta t^2 \dot{F}(u^n)$$

satisfies the monotonicity condition $\|u^{n+1}\| \le \|u^n\|$ under the time-step restriction

$$\Delta t \le \frac{K}{2\beta}\left(\sqrt{\alpha^2K^2+4\beta}-\alpha K\right)\Delta t_{\text{FE}}.$$

Proof. As above, we rewrite

$$u^{n+1} \quad = \quad a\left(u^n + \frac{\alpha}{a}\Delta t F(u^n)\right) + (1-a)\left(u^n + \frac{\beta}{1-a}\Delta t^2 \dot{F}(u^n)\right),$$

which is a convex combination provided that $0 \le a \le 1$. The time-step restriction that follows from this convex combination must satisfy $\frac{\alpha}{a}\Delta t \le \Delta t_{\text{FE}}$ and $\frac{\beta}{1-a}\Delta t^2 \le K^2\Delta t_{\text{FE}}^2$. The first condition becomes $\Delta t \le \frac{a}{\alpha}\Delta t_{\text{FE}}$ while the second is $\Delta t \le \sqrt{\frac{1-a}{\beta}}K\Delta t_{\text{FE}}$. We observe that on $0 \le a \le 1$ the first term encourages a larger a while the second term is less restrictive with a smaller a. The two conditions are balanced, and thus the allowable time-step is maximized, when we equate the right hand sides:

$$\frac{a}{\alpha} = \sqrt{\frac{1-a}{\beta}}K \quad \rightarrow \quad a = \frac{\alpha K}{2\beta}\left(\sqrt{\alpha^2K^2+4\beta}-\alpha K\right).$$

Now using the first condition, $\Delta t \le \frac{a}{\alpha}\Delta t_{\text{FE}}$ we obtain our result. □

A more realistic motivating example is the unique two-stage fourth order method

$$\begin{aligned} u^* &= u^n + \frac{\Delta t}{2} F(u^n) + \frac{\Delta t^2}{8} \dot{F}(u^n), \\ u^{n+1} &= u^n + \Delta t F(u^n) + \frac{\Delta t^2}{6} (\dot{F}(u^n) + 2\dot{F}(u^*)). \end{aligned} \tag{10}$$

The first stage of the method is a Taylor series method with $\frac{\Delta t}{2}$, while the second stage can be written

$$\begin{aligned} u^{n+1} &= u^n + \Delta t F(u^n) + \frac{\Delta t^2}{6} (\dot{F}(u^n) + 2\dot{F}(u^*)) \\ &= a \left(u^* - \frac{\Delta t}{2} F(u^n) - \frac{\Delta t^2}{8} \dot{F}(u^n) \right) + (1-a) u^n + \Delta t F(u^n) + \frac{\Delta t^2}{6} (\dot{F}(u^n) + 2\dot{F}(u^*)) \\ &= (1-a) \left(u^n + \frac{1 - \frac{a}{2}}{1-a} \Delta t F(u^n) + \frac{\frac{1}{6} - \frac{a}{8}}{1-a} \Delta t^2 \dot{F}(u^n) \right) + a \left(u^* + \frac{\Delta t^2}{3a} \dot{F}(u^*) \right). \end{aligned}$$

For $0 \le a \le 1$ this is a convex combination of two terms. The first term is of the form (9), which gives the time-step restriction

$$\Delta t \le \frac{6}{4 - 3a} \left(\sqrt{\left(\frac{5}{4} a^2 - \frac{10}{3} a + \frac{7}{3} \right)} - 1 + \frac{a}{2} \right) \Delta t_{\text{FE}}.$$

The second is of the form (8b), so we have $\Delta t \le \sqrt{\frac{3a}{2}} \Delta t_{\text{FE}}$. We plot these two in Figure 1, and we observe that the first term is decreasing in a (blue line) while the second term is increasing in a (red line). As a result, we obtain the optimal allowable time-step by setting these two equal, which yields $a \approx 0.3072182638002141$ and the corresponding SSP coefficient, $\mathcal{C} \approx 0.6788426884782078$. A direct computation of the TVD time-step for this case, which takes advantage of the linearity of the problem and the spatial discretization, gives the bound $\Delta t \le (\sqrt{3} - 1) > 0.6788$. This shows that, as we expect, the SSP condition is not always sharp.

The convex combination approach becomes more complicated when dealing with multi-stage methods. It is the most appropriate approach for developing an understanding of the strong stability property of a given method. However, it is not computationally efficient for finding optimal SSP methods. In the following section, we show how to generalize the convex combination decomposition approach and we use this generalization to formulate an SSP optimization problem along the lines of [16, 19].

2.2 Formulating the SSP optimization problem

As above, we begin with the hyperbolic conservation law (1) and adopt a spatial discretization so that we have the system of ODEs (2). The spatial discretization F is specially designed so that it satisfies the *forward Euler* (first derivative) condition

$$\textbf{Forward Euler condition} \qquad \|u^n + \Delta t F(u^n)\| \le \|u^n\| \quad \text{for} \quad \Delta t \le \Delta t_{FE}, \tag{11}$$

for the desired stability property indicated by the convex functional $\|\cdot\|$. For multiderivative methods, in addition to the first derivative, we need to appropriately approximate the second derivative in time u_{tt}, to which we represent the discretization as $\dot{F}$. It is not immediately obvious what should be the form of a condition that would account for the effect of the $\Delta t^2 \dot{F}$ term. Motivated by the examples in the previous sections, we choose the

$$\textbf{Second derivative condition} \qquad \|u^n + \Delta t^2 \dot{F}(u^n)\| \le \|u^n\| \quad \text{for} \quad \Delta t \le K \Delta t_{FE}, \tag{12}$$

where K is a scaling factor that compares the stability condition of the second derivative term to that of the forward Euler term. Given conditions (11) and (12), we wish to formulate sufficient conditions so that the multiderivative

method (6) satisfies the desired monotonicity condition under a given time-step. First, we write the method (6) in an equivalent matrix-vector form

$$\mathbf{y} = \mathbf{e}u^n + \Delta t S F(\mathbf{y}) + \Delta t^2 \hat{S}\dot{F}(\mathbf{y}), \tag{13}$$

where

$$S = \begin{bmatrix} A & \mathbf{0} \\ b^T & 0 \end{bmatrix} \quad \text{and} \quad \hat{S} = \begin{bmatrix} \hat{A} & \mathbf{0} \\ \hat{b}^T & 0 \end{bmatrix}$$

and $\mathbf{e}$ is a vector of ones. We are now ready to state our result:

Theorem 2. *Given spatial discretizations F and $\dot{F}$ that satisfy (11) and (12), a two-derivative multistage method of the form (13) preserves the strong stability property $\|u^{n+1}\| \le \|u^n\|$ under the time-step restriction $\Delta t \le r\Delta t_{\text{FE}}$ if satisfies the conditions*

$$\left(I + rS + \frac{r^2}{K^2}\hat{S}\right)^{-1} \mathbf{e} \ge 0 \tag{14a}$$

$$r\left(I + rS + \frac{r^2}{K^2}\hat{S}\right)^{-1} S \ge 0 \tag{14b}$$

$$\frac{r^2}{K^2}\left(I + rS + \frac{r^2}{K^2}\hat{S}\right)^{-1} \hat{S} \ge 0 \tag{14c}$$

for some $r > 0$. In the above conditions, the inequalities are understood component-wise.

Proof. We begin with the method (13), and add the terms $rS\mathbf{y}$ and $\hat{r}\hat{S}\mathbf{y}$ to both sides to obtain

$$\begin{aligned} \left(I + rS + \hat{r}\hat{S}\right)\mathbf{y} &= u^n\mathbf{e} + rS\left(\mathbf{y} + \frac{\Delta t}{r}F(\mathbf{y})\right) + \hat{r}\hat{S}\left(\mathbf{y} + \frac{\Delta t^2}{\hat{r}}\dot{F}(\mathbf{y})\right), \\ \mathbf{y} &= R(\mathbf{e}u^n) + P\left(\mathbf{y} + \frac{\Delta t}{r}F(\mathbf{y})\right) + Q\left(\mathbf{y} + \frac{\Delta t^2}{\hat{r}}\dot{F}(\mathbf{y})\right), \end{aligned}$$

where

$$R = \left(I + rS + \hat{r}\hat{S}\right)^{-1}, \qquad P = rRS, \qquad Q = \hat{r}R\hat{S}.$$

If the elements of P, Q, and $R\mathbf{e}$ are all non-negative, and if $R + P + Q = I$, then these three terms describe a convex combination of terms which are SSP, and the resulting value is SSP as well

$$\|\mathbf{y}\| \le R\|\mathbf{e}u^n\| + P\|\mathbf{y} + \frac{\Delta t}{r}F(\mathbf{y})\| + Q\|\mathbf{y} + \frac{\Delta t^2}{\hat{r}}\dot{F}(\mathbf{y})\|,$$

under the time-step restrictions $\Delta t \le r\Delta t_{\text{FE}}$ and $\Delta t \le K\sqrt{\hat{r}}\Delta t_{\text{FE}}$. As we observed above, the optimal time-step is given when these two are set equal, so we require $r = K\sqrt{\hat{r}}$. Conditions (14a)–(14c) now ensure that $P \ge 0$, $Q \ge 0$, and $R\mathbf{e} \ge 0$ component-wise for $\hat{r} = \frac{r^2}{K^2}$, and our method (13) preserves the strong stability condition $\|u^{n+1}\| \le \|u^n\|$ under the time-step restriction $\Delta t \le r\Delta t_{FE}$. □

This theorem gives us the conditions for the method (13) to be SSP for any the time-step $\Delta t \le r\Delta t_{\text{FE}}$. This allows us to formulate the search for optimal SSP two-derivative methods as an optimization problem, similar to [16, 18, 7], where the aim is to find $\mathcal{C} = \max r$ such that the relevant order conditions (from Section 1.2) and SSP conditions (14a)-(14b) are all satisfied. Based on this, we wrote a MATLAB optimization code for finding optimal two-derivative multistage methods [8], formulated along the lines of David Ketcheson's code [19] for finding optimal SSP multistage multistep methods in [17, 2]. We used this to find optimal SSP multistage two-derivative methods of order up to $p = 5$. However, we also used our observations on the resulting methods to formulate closed form representations of the optimal SSP multistage two-derivative methods. We present both the numerical and closed-form optimal methods in the following section.

Remark 3. *An alternative approach to defining a MSMD SSP method is to begin with spatial discretization that satisfies the "Taylor series" condition, defined by*

$$\|u^n + \Delta t F(u^n) + \frac{1}{2}\Delta t^2 \dot{F}(u^n)\| \leq \|u^n\| \quad \text{for} \quad \Delta t \leq \Delta t_{TS} = K_2 \Delta t_{FE}. \tag{15}$$

This condition replaces (12), and allows us to rewrite (6) as convex combinations of forward Euler and Taylor series steps of the form (15). A similar optimization problem can be defined based on this condition. This approach was adopted in [23].

This condition is more restrictive than what we consider in the present work. Indeed, if a spatial discretization satisfies conditions (11) as well as (12), then it will also satisfy condition (15), with $K_2 = K\sqrt{K^2+2} - K$. However, some methods of interest cannot be written using a Taylor series decomposition, including the two-stage fourth order method in (10). For these reasons, we do not explore this approach further, but we point out that we have experimented with this alternative formulation to generate some optimal SSP methods. Henceforth, we restrict our attention to spatial discretizations that satisfy (11) and (12).

3 Optimal SSP multiderivative methods

3.1 Second order methods

Although we do not wish to use second order methods in our computations, it is interesting to consider the strong stability properties of these methods both as building blocks of higher order methods and as simple example that admit optimal formulations with simple formulas.

One stage methods. The second-order Taylor series method,

$$u^{n+1} = u^n + \Delta t F(u^n) + \frac{1}{2}\Delta t^2 \dot{F}(u^n), \tag{16}$$

is the unique one-stage second-order method. Theorem 1 for the building block (9) with coefficients $\alpha = 1$ and $\beta = \frac{1}{2}$ gives the condition $\Delta t \leq \mathcal{C}\Delta t_{\text{FE}}$ with $\mathcal{C} = K\sqrt{K^2+2} - K^2$. Unlike the SSP single derivative Runge–Kutta methods, the SSP coefficient is not just dependent on the time-stepping method, but also on the value K which comes from the second derivative condition (12). As noted above, the Taylor series method can serve as the basic building block for two-derivative methods, just as the first-order FE can be used to build higher-order RK integrators.

Two stage methods. The optimal SSP two stage second order methods depends on the value of K in (12). A straightforward SSP analysis via convex combinations, and solving the equations for the order conditions allows us to formulate the optimal methods without using the optimization code [8]. However, we used the optimization code to verify our results.

If $K \leq \sqrt{\frac{2}{3}}$ the optimal methods are of the form

$$\begin{aligned} u^* &= u^n + \frac{1}{r}\Delta t F(u^n), \\ u^{n+1} &= u^n + \frac{1}{2}\Delta t \left(F(u^n) + F(u^*)\right) + \frac{r-1}{2r}\Delta t^2 \dot{F}(u^n), \end{aligned} \tag{17}$$

where

$$r = \frac{1}{2}\left(1 - K^2 + \sqrt{1 + 6K^2 + K^4}\right).$$

These methods are SSP for $\mathcal{C} = r$, where $r > 1$ whenever $K > 0$. Notice that if $K = 0$ we have $r = 1$ and our method reduces to the standard two stage second order Runge–Kutta method.

As K increases, the time-step restriction imposed by the condition (12) is alleviated, and consequently the optimal method includes more of the second derivative terms. If $K \geq \sqrt{\frac{2}{3}}$ we get an optimal method

$$\begin{aligned} u^* &= u^n + \frac{1}{2}\Delta t F(u^n) + \frac{1}{8}\Delta t^2 \dot{F}(u^n), \\ u^{n+1} &= u^* + \frac{1}{2}\Delta t F(u^*) + \frac{1}{8}\Delta t^2 \dot{F}(u^*), \end{aligned} \tag{18}$$

which is simply two Taylor series steps with $\frac{1}{2}\Delta t$ in each one, and so has $\mathcal{C} = 2K\sqrt{K^2+2} - 2K^2$. We see in Figure 2 the SSP coefficient $\mathcal{C}$ vs. K for the two methods above (in green and red respectively), and for many numerically optimal methods in blue.

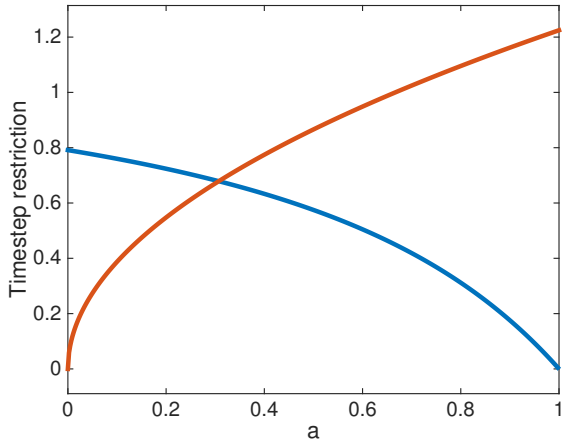


Figure 1: The time-step restriction for the two-stage fourth order method is the minimum of two terms, one of which is decreasing while the other is increasing in a.

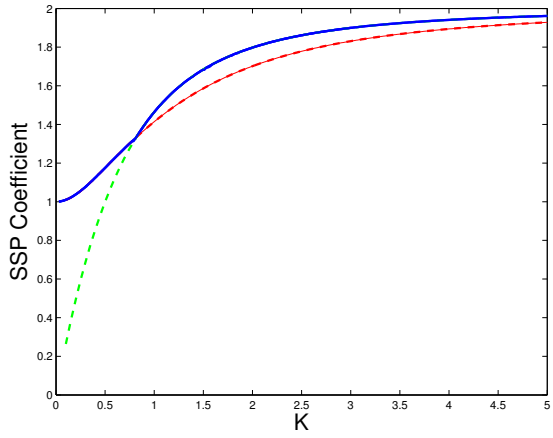


Figure 2: The SSP coefficient $\mathcal{C}$ as a function of the coefficient K in (12) for the optimal two-stage two-derivative second order methods given in Section 3.1.

Note that (18) has four function evaluations compared to the three function evaluations above in (17). If we assume all the function evaluations cost the same, it will never pay off to use the second method, as the first method is always more efficient. However, if one has a special case where the cost of computing $\dot{F}$ is negligible the second method may still be worthwhile.

For this method and all others, once we have the optimal Butcher arrays and the SSP coefficient $\mathcal{C} = r$, we can easily convert them to the Shu-Osher form as follows:

```
% Given A, Ahat, b, bhat, and r, the Shu-Osher matrices are given by
    z=0.0*b;  I=eye(3);e=ones(3,1);
    S=[A,z; b',0]; Shat=[Ahat,z; bhat',0];
    Ri=(I+r*S+(r^2/k^2)*Shat);
    v=Ri\e;
    P = r*(Ri\S);
    Q= r^2/k^2*(Ri\Shat);
```

3.2 Third order methods

Many two-stage two-derivative third order methods exist. As above, the optimal $\mathcal{C}$ depends on the value of K in (12). Through numerical search using our optimization code [8], we found optimal methods for the range $0.1 \le K \le 5$. Figure 3 (blue line) shows the SSP coefficient $\mathcal{C}$ vs. K. We also solved the order conditions explicitly and analyzed the SSP conditions (14a)-(14c) to find the values of the coefficients of these methods as functions of $r = \mathcal{C}$ and K. These methods all have the form

$$\begin{aligned} u^* &= u^n + a\Delta t F(u^n) + \hat{a}\Delta t^2 \dot{F}(u^n), \\ u^{n+1} &= u^n + b_1\Delta t F(u^n) + b_2\Delta t F(u^*) + \hat{b_1}\Delta t^2 \dot{F}(u^n) + \hat{b_2}\Delta t^2 \dot{F}(u^*), \end{aligned} \tag{19}$$

where the coefficients satisfy

$$\begin{aligned} a &= \tfrac{1}{r}\left(K\sqrt{K^2+2} - K^2\right) & \hat{a} &= \frac{1}{2}a^2 \\ b_2 &= \tfrac{2K^2(1-\frac{1}{r})+r}{K\sqrt{K^2+2}+K^2} - \tfrac{r^2}{3K^2} & b_1 &= 1 - b_2 \\ \hat{b}_1 &= \tfrac{1}{2} - \tfrac{1}{2}ab_2 - \tfrac{1}{6a} & \hat{b}_2 &= \frac{1}{6a} - \frac{1}{2}ab_2 \end{aligned} \tag{20}$$

Note that the first stage is always a Taylor series method with Δt replaced by $a\Delta t$, which is underscored by the fact that SSP coefficient is given by $r = \frac{1}{a}\left(K\sqrt{K^2+2} - K^2\right)$. The SSP conditions (14a)-(14c) provide the restrictions on the size of r, and thus on $\mathcal{C} = r$. We observe that the condition that restricts us most here is the non-negativity of $Q_{3,1}$, and so we must select a value of r such that this value is zero. To do this, we select r to be the smallest positive root of $p_0 + p_1 r + p_2 r^2 + p_3 r^3$ where the coefficients p_i are all functions of K

$$\begin{aligned} p_0 &= 2K(\sqrt{K^2+2} - 3K) + 4K^3(\sqrt{K^2+2} - K) \\ p_1 &= -a_0, \quad p_2 = (1-a_0)/(2K^2), \quad p_3 = -\frac{\frac{a_0}{2K} + K}{6K^3}. \end{aligned}$$

Some examples of the value of r as a function of K are given in Table 2. The MATLAB script for this method is found in Appendix B.

K	0.25	0.4	0.5	0.6	0.7	0.8	1.0	1.25	1.5	1.75	2.5	3	3.5	4
r	0.48	0.71	0.84	0.94	1.03	1.11	1.23	1.33	1.39	1.44	1.51	1.54	1.55	1.56

Table 2: The SSP coefficient $\mathcal{C} = r$ for each K in the two-stage third order method (19).

The Shu-Osher representations of the methods can be easily obtained by using r and K to solve for $\hat{r}$, R, P, and Q. For example, for $K = \frac{1}{\sqrt{2}}$ we have $r = 1.04$ and the method in Butcher form has

$$\begin{aligned} a &= 0.594223212099088, \\ \hat{a} &= 0.176550612898679, \end{aligned} \qquad b = \begin{bmatrix} 0.693972512991841 \\ 0.306027487008159 \end{bmatrix}, \quad \hat{b} = \begin{bmatrix} 0.128597465450411 \\ 0.189553898228989 \end{bmatrix}.$$

The optimal Shu Osher formulation for these values is given by $R\mathbf{e} = (1,0,0)^T$,

$$P = \begin{bmatrix} 0 & 0 & 0 \\ 0.618033988749895 & 0 & 0 \\ 0.271611333775367 & 0.318290138472780 & 0 \end{bmatrix}$$

$$Q = \begin{bmatrix} 0 & 0 & 0 \\ 0.381966011250105 & 0 & 0 \\ 0 & 0.410098527751853 & 0 \end{bmatrix}.$$

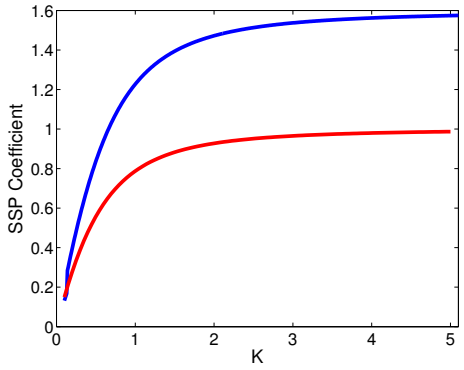


Figure 3: The SSP coefficient vs. K for two stage methods. Third order methods are in blue, the fourth order methods are red.

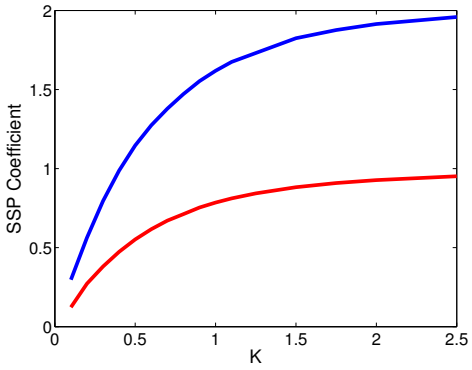


Figure 4: The SSP coefficient vs. K for three stage methods. Fourth order methods are in blue (top line), the fifth order methods in red (bottom line).

3.3 Fourth order methods

3.3.1 Fourth order methods: The two-stage fourth-order method

The two-stage two-derivative fourth order method (10) is unique; there is only one set of coefficients that satisfy the fourth order conditions for this number of stages and derivatives. The method is

$$u^* = u^n + \frac{\Delta t}{2} F(u^n) + \frac{\Delta t^2}{8} \dot{F}(u^n)$$
$$u^{n+1} = u^n + \Delta t F(u^n) + \frac{\Delta t^2}{6} (\dot{F}(u^n) + 2\dot{F}(u^*)).$$

The first stage of the method is a Taylor series method with $\frac{\Delta t}{2}$, while the second stage can be written as a linear combination of a forward Euler and a second-derivative term (but not a Taylor series term). The SSP coefficient of this method is larger as K increases, as can be seen in Figure 3.

To ensure that the SSP conditions (14a)-(14c) are satisfied, we need to select the largest r so that all the terms are non-negative. We observed from numerical optimization that in the case of the two-stage fourth order method the term $(R\mathbf{e})_3$ gives the most restrictive condition: if we choose r to ensure that this term is non-negative, all the other conditions are satisfied. Satisfying this condition, the SSP coefficient $\mathcal{C} = r$ is given by the smallest positive root of the polynomial:

$$(R\mathbf{e})_3 = r^4 + 4K^2 r^3 - 12K^2 r^2 - 24K^4 r + 24K^4.$$

The Shu-Osher decomposition for the optimal method corresponding to this value of K is

$$\begin{aligned} u^* &= \left(1 - \frac{4rK^2 + r^2}{8K^2}\right) u^n + \frac{r}{2}\left(u^n + \frac{\Delta t}{r} F(u^n)\right) + \frac{r^2}{8K^2}\left(u^n + \frac{K^2}{r^2}\Delta t^2 \dot{F}(u^n)\right) \qquad (21) \\ u^{n+1} &= r\left(1 - \frac{r^2}{6K^2}\right)\left(u^n + \frac{\Delta t}{r} F(u^n)\right) + \frac{r^2(4K^2 - r^2)}{24K^4}\left(u^n + \frac{K^2}{r^2}\Delta t^2 \dot{F}(u^n)\right) + \frac{r^2}{3K^2}\left(u^* + \frac{K^2}{r^2}\Delta t^2 \dot{F}(u^*)\right). \end{aligned}$$

3.3.2 Fourth order methods: Three stage methods

If we increase the number of stages to three, we can construct entire families of methods that obtain fourth-order accuracy, and are SSP with a larger allowable time-step. For these methods, we were not able to find closed

form solutions, but our optimization code [8] produced methods for various values of K. The SSP coefficient as a function of K for these methods is given in Figure 3, and we give the coefficients for selected methods in both Butcher array and Shu-Osher form in Appendix A.

3.4 Fifth order methods

As mentioned above, it was shown that explicit SSP Runge–Kutta methods cannot have order $p > 4$ [20, 28]. This order barrier is broken by multiderivative methods. If we allow three stages and two-derivative we can obtain a fifth order SSP method. The explicit three-stage fifth order method has twelve coefficients and sixteen order conditions that need to be satisfied. This is possible if some of the coefficients are set to zero, which allows several of the order conditions to be repetitive and satisfied automatically. The methods resulting from our optimization routine all had the simplified form

$$\begin{aligned} u^* &= u^n + a_{21}\Delta t F(u^n) + \hat{a}_{21}\Delta t^2 \dot{F}(u^n) \\ u^{**} &= u^n + a_{31}\Delta t F(u^n) + \hat{a}_{31}\Delta t^2 \dot{F}(u^n) + \hat{a}_{32}\Delta t^2 \dot{F}(u^*) \\ u^{n+1} &= u^n + \Delta t F(u^n) + \Delta t^2 \left(\hat{b}_1 \dot{F}(u^n) + \hat{b}_2 \dot{F}(u^*) + \hat{b}_3 \dot{F}(u^{**}) \right). \end{aligned} \tag{22}$$

The coefficients of the three-stage fifth order method are then given as a one-parameter system, depending only on a_{21}, that are related through

$$\begin{aligned} \hat{a}_{21} &= \frac{1}{2}a_{21}^2, \quad a_{31} = \frac{3/5 - a_{21}}{1 - 2a_{21}}, \\ \hat{a}_{32} &= \frac{1}{10}\left(\frac{(\frac{3}{5} - a_{21})^2}{a_{21}(1-2a_{21})^3} - \frac{\frac{3}{5} - a_{21}}{(1-2a_{21})^2} \right), \quad \hat{a}_{31} = \frac{1}{2}\frac{(\frac{3}{5} - a_{21})^2}{(1-2a_{21})^2} - \hat{a}_{32}, \\ \hat{b}_2 &= \frac{2a_{31} - 1}{12a_{21}(a_{31} - a_{21})}, \quad \hat{b}_3 = \frac{1 - 2a_{21}}{12a_{31}(a_{31} - a_{21})}, \quad \hat{b}_1 = \frac{1}{2} - \hat{b}_2 - \hat{b}_3. \end{aligned}$$

To satisfy the SSP conditions (14a)-(14c), we must ensure that $(R\mathbf{e})_3$ is non-negative. Based on the optimization code we observed that the extreme case of $(R\mathbf{e})_3 = 0$ gives the optimal methods, and we can obtain a_{21} as a function of K and r through

$$a_{21} = \frac{K^6}{r^6}\left(-\frac{2}{K^4}r^5 + \frac{10}{K^4}r^4 + \frac{40}{K^2}r^3 - \frac{120}{K^2}r^2 - 240r + 240 \right).$$

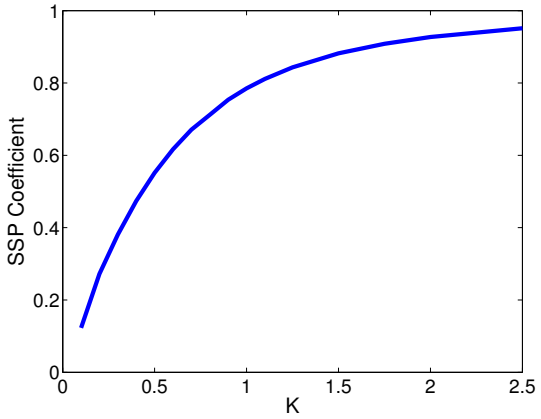
Now, we wish to ensure that $Q_{3,1}$ is nonnegative. The SSP coefficient $\mathcal{C} = r$ is then chosen as the largest positive root of

$$Q_{3,1} = 10r^2 a_{21}^4 - (100K^2 + 10r^2)a_{21}^3 + (130K^2 + 3r^2)a_{21}^2 - 50K^2 a_{21} + 6K^2.$$

The MATLAB script in Appendix C solves for the largest r that satisfies the SSP conditions (14a)-(14c), and then computes the coefficients of the optimal methods both in Butcher array and Shu-Osher form. This approach yields the same optimal methods as those obtained by our optimization code [8]. In Figure 5 we show values of a_{21} and r for given values of K.

Once again, the Shu-Osher decomposition is needed for the method to be SSP, and is easily obtained. For example, for $K = \frac{1}{\sqrt{2}}$ we have $r = 0.6747$ and the method becomes

$$R\mathbf{e} = \begin{bmatrix} 1.0 \\ 0.2369970626512336 \\ 0.7810723816004148 \\ 0.0 \end{bmatrix}, \; P = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.5064804704259125 & 0 & 0 & 0 \\ 0.1862033791874200 & 0 & 0 & 0 \\ 0.5769733539128722 & 0 & 0 & 0 \end{bmatrix} \tag{23}$$



K	a_{21}	$\mathcal{C}$	K	a_{21}	$\mathcal{C}$
0.1	0.7947	0.1452	1.1	0.7393	0.8114
0.2	0.7842	0.2722	1.2	0.7374	0.8335
0.3	0.7751	0.3814	1.3	0.7359	0.8523
0.4	0.7674	0.4741	1.4	0.7346	0.8683
0.5	0.7609	0.5520	1.5	0.7334	0.8819
0.6	0.7555	0.6171	1.6	0.7324	0.8937
0.7	0.7510	0.6712	1.7	0.7316	0.9039
0.8	0.7472	0.7162	1.8	0.7309	0.9127
0.9	0.7441	0.7537	1.9	0.7302	0.9205
1.0	0.7415	0.7851	2.0	0.7296	0.9273

Figure 5: SSP coefficients for three-stage fifth-order methods. Left: the SSP coefficient as a function of K for three stage fifth order methods. Right: a table of a_{21} and the SSP coefficient $\mathcal{C}$, for different values of K as defined in (22). The code to generate the coefficients in Butcher and Shu-Osher form is given in the appendix.

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.2565224669228537 & 0 & 0 & 0 \\ 0 & 0.0327242392121651 & 0 & 0 \\ 0.0615083849004797 & 0.0803574544380432 & 0.2811608067486047 & 0 \end{bmatrix}.$$

These coefficients as well as the coefficients for the optimal method for any value of K can be easily obtained to high precision by the MATLAB code in Appendix C.

4 Numerical Experiments

4.1 Numerical verification of the SSP properties of these methods

4.1.1 Example 1: Linear advection with first order TVD spatial discretization

As a first test case, we consider linear advection, $U_t - U_x = 0$, with a first order finite difference for the first derivative and a second order centered difference for the second derivative defined in (7)

$$F(u^n)_j := \frac{u^n_{j+1} - u^n_j}{\Delta x} \approx U_x(x_j), \qquad \text{and} \qquad \dot{F}(u^n)_j := \frac{u^n_{j+1} - 2u^n_j + u^n_{j-1}}{\Delta x^2} \approx U_{xx}(x_j).$$

Recall from (8) that this first-order spatial discretization satisfies the

Forward Euler condition $u^{n+1}_j = u^n_j + \frac{\Delta t}{\Delta x}\left(u^n_{j+1} - u^n_j\right)$ is TVD for $\Delta t \le \Delta t_{FE} = \Delta x$, and the

Second Derivative condition $u^{n+1}_j = u^n_j + \left(\frac{\Delta t}{\Delta x}\right)^2\left(u^n_{j+1} - 2u^n_j + u^n_{j-1}\right)$ is TVD for $\Delta t \le \frac{1}{\sqrt{2}}\Delta t_{FE}$.

Fo initial conditions, we use a step function

$$u_0(x) = \begin{cases} 1 & \text{if } \frac{1}{4} \le x \le \frac{1}{2}, \\ 0 & \text{otherwise,} \end{cases} \tag{24}$$

with a domain $x \in [0, 1]$ and periodic boundary conditions. This simple example is chosen as our experience has shown [7] that this problem often demonstrates the sharpness of the SSP time-step.

Stages	Order	Predicted $\mathcal{C}$	Observed $\mathcal{C}$	Stages	Order	Predicted $\mathcal{C}$	Observed $\mathcal{C}$
1	2	0.6180	0.6180	2	4	0.6788	0.7320
2	2	1.2807	1.2807	3	4	1.3927	1.3927
2	3	1.0400	1.0400	3	5	0.6746	0.7136

Table 3: Comparison of the theoretical and observed SSP coefficients that preserve the nonlinear stability properties in Example 1.

For all of our simulations, we use a fixed grid of size $\Delta x = \frac{1}{1600}$, and a time-step $\Delta t = \lambda \Delta x$ where we vary $0.05 \leq \lambda \leq 1$. We step each method forward by $N = 50$ time-steps and compare the performance of the various time-stepping methods constructed earlier in this work, for $K = \frac{\sqrt{2}}{2}$. We test this problem using the two stage third order method (19), the two stage fourth order method (10) and the three stage fourth order method in Appendix A, the fifth order method (23). We also consider the non-SSP two stage third order method,

$$\begin{aligned} u^* &= u^n - \Delta t F(u^n) + \frac{1}{2}\Delta t^2 \dot{F}(u^n), \\ u^{n+1} &= u^n - \frac{1}{3}\Delta t F(u^n) + \frac{4}{3}\Delta t F(u^*) + \frac{4}{3}\Delta t^2 \dot{F}(u^n) + \frac{1}{2}\Delta t^2 \dot{F}(u^*), \end{aligned} \tag{25}$$

To measure the effectiveness of these methods, we consider the maximum observed rise in total variation, defined by

$$\max_{0 \leq n \leq N-1} \left(\|u^{n+1}\|_{TV} - \|u^n\|_{TV} \right). \tag{26}$$

We are interested in the time-step in which this rise becomes evident (i.e. well above roundoff error). Another measure that we use is the rise in total variation compared to the total variation of the initial solution:

$$\max_{0 \leq n \leq N-1} \left(\|u^{n+1}\|_{TV} - \|u^0\|_{TV} \right). \tag{27}$$

Figure 6 (top) shows the maximal rise in total variation for each CFL value $\frac{\Delta t}{\Delta x}$. On the left we have the maximal per-step rise in TV (26) and on the right, the maximal rise in TV compared to the TV of the initial solution (27). We clearly see that once the CFL value passes a certain limit, there is a sharp jump in the total variation of the solution. We are interested in the value of $\frac{\Delta t}{\Delta x}$ at which the time-stepping method no longer maintains the nonlinear stability. The fifth order method (black) has the most restrictive value of Δt before the total variation begins to rise. The next most restrictive is the two-stage fourth order method, followed by the two stage third order method. The two-stage second order methods have more freedom than these methods, and so have a much larger allowable Δt, while the three stage fourth order, having the most freedom in the choice of coefficients, outperforms all the other methods. Figure 6 (bottom) compares the performance of the non-SSP method to the SSP method. This graph clearly shows the need for the SSP property of the time-stepping, as the absence of this property results in the loss of the TVD property for any time-step.

We notice that controlling the maximal rise of the total variation compared to the initial condition (27) requires a smaller allowable time-step, so we use this condition as our criterion for maximal allowable time-step. A comparison of the predicted (i.e. theoretical) values of the SSP coefficient and the observed value for the Taylor series method, the two stage methods of order $p = 2, 3, 4$, and the three-stage fourth and fifth order methods are shown in Table 3 We note that for the Taylor series method, the two-stage second order method, the two stage third order method, and three stage fourth order method, the observed SSP coefficient matches exactly the theoretical value. On the other hand, the two-stage fourth order and the three-stage fifth order, both of which have the smallest SSP coefficients (both in theory and practice), have a larger observed SSP coefficient than predicted. For the two-stage fourth order case this is expected, as we noted in Section 2.1 that the TVD time-step for this particular case is (as we observe here) $(\sqrt{3} - 1)$, larger than the more general SSP timestep $\mathcal{C} = 0.6788$.

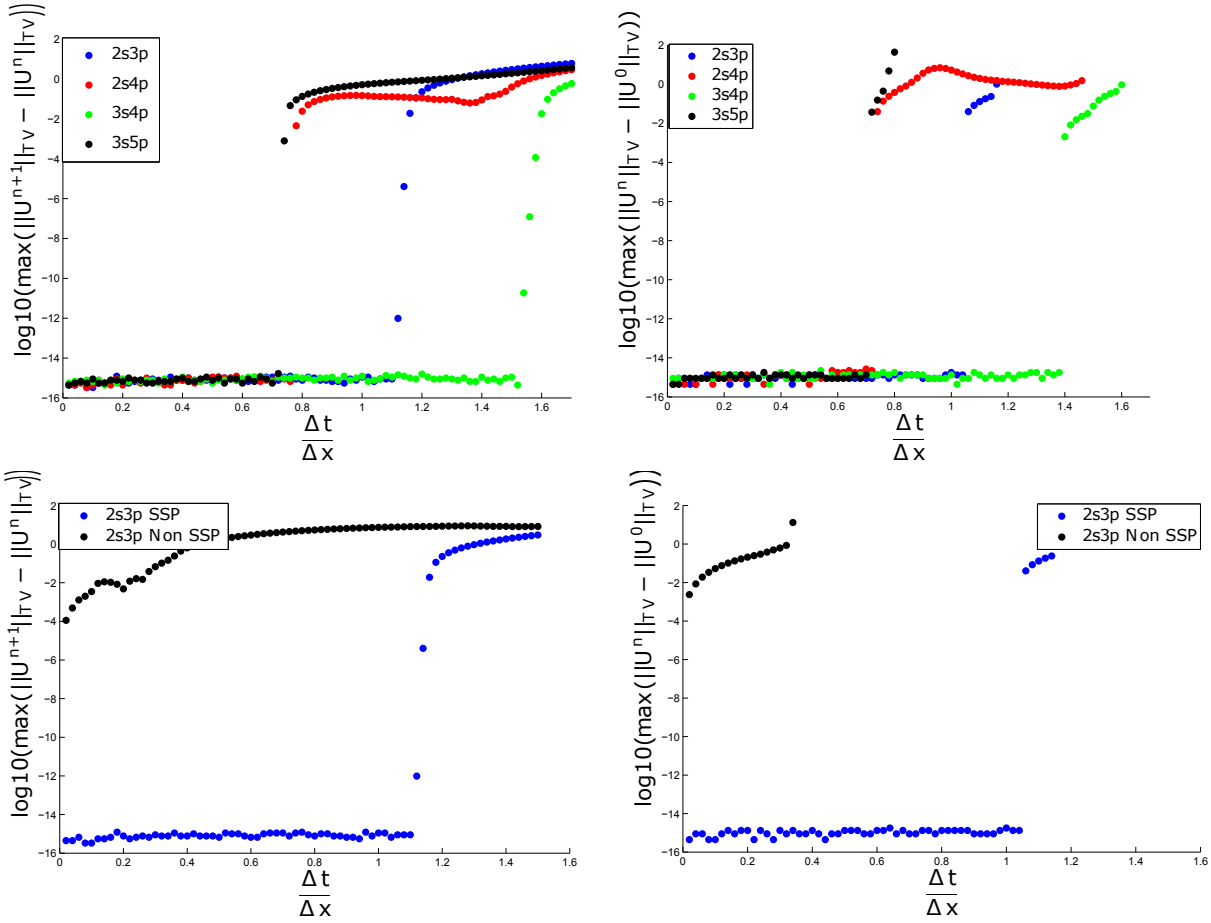


Figure 6: The rise in total variation as a function of the CFL number. On the left is the maximal per time-step rise (26) and on the right the maximal TV rise above the initial TV (27). Top: Comparison of a variety of SSP methods. Bottom: Comparison of two-stage third order SSP and non-SSP methods.

4.1.2 Example 2: MSMD methods with weighted essentially non-oscillatory (WENO) methods

The major use of MSMD time-stepping would be in conjunction with high order methods for problems with shocks. In this section we consider two scalar problems: the linear advection equation

$$U_t + U_x = 0 \tag{28}$$

and the nonlinear Burgers' equation

$$U_t + \left(\frac{1}{2}U^2\right)_x = 0 \tag{29}$$

on $x \in (-1, 1)$. In both cases we use the step function initial conditions (24), and periodic boundaries. We use $N = 201$ points in the domain, so that $\Delta x = \frac{1}{100}$.

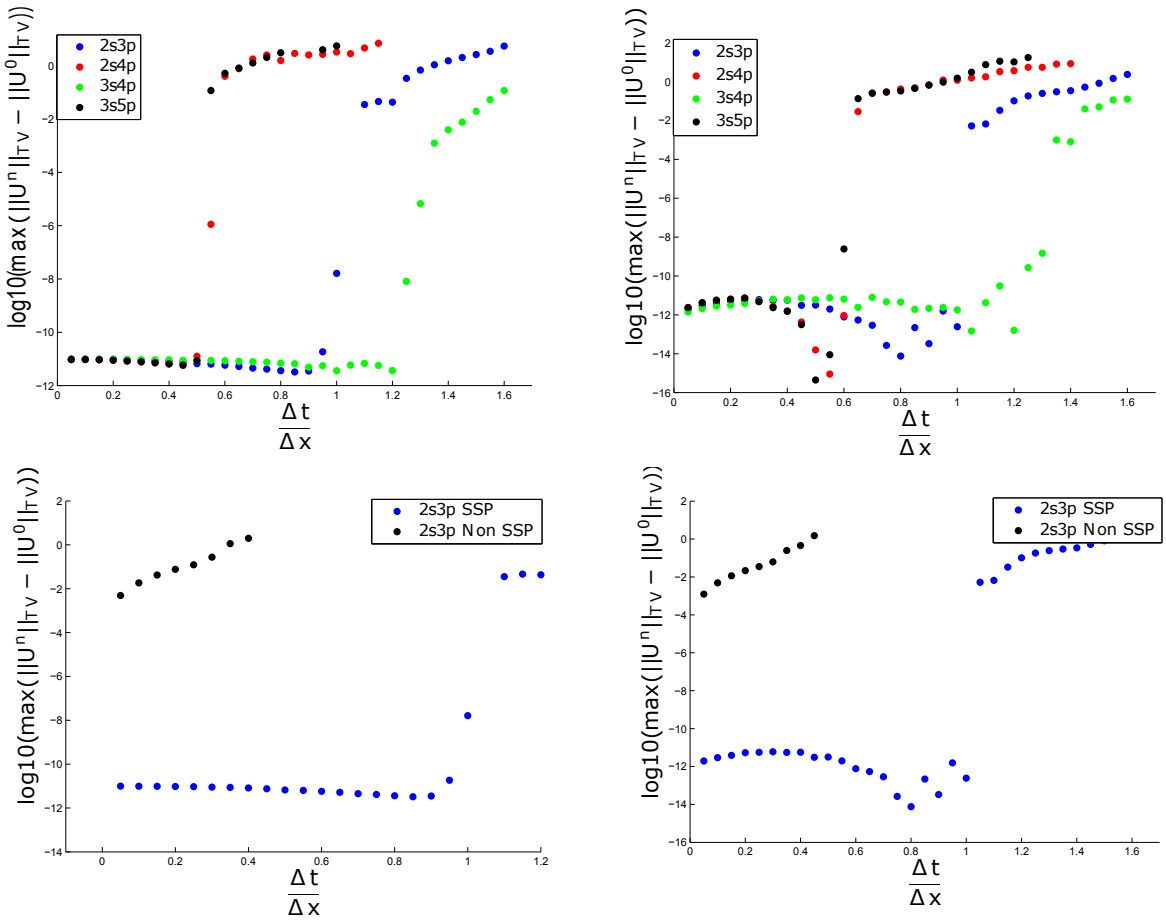


Figure 7: Comparison of the rise in total variation as a function of the CFL number for Example 2. Linear advection on left and Burgers' equation on right. The top graphs compare the performance of different SSP methods, while the bottom graphs compare the two-stage third order SSP and non-SSP methods.

We follow our previous work [29] with a minor modification for the spatial discretization. The spatial discretization is performed as follows: at each iteration we take the known value u^n and compute the flux $f(u^n) = u^n$ in the linear case and $f(u^n) = \frac{1}{2}(u^n)^2$ for Burgers' equation. Now to compute the spatial derivative $f(u^n)_x$ we use the WENO method [13]. In our test cases, we can avoid flux splitting, as $f'(u)$ is strictly non-negative (below, we refer to the WENO method on a flux with $f'(u) \geq 0$ as WENO$^+$ and to to the corresponding method on a flux with $f'(u) \leq 0$ as WENO$^-$).

Now we have the approximation to U_t at time t^n, and wish to compute the approximation to U_{tt}. In previous work we defined the higher order derivative using central differences, but we have found that additional limiting, in the form of the WENO$^-$ differentiation operator, is needed to achieve a pseudo-TVD like property. For the linear flux, this is very straightforward as $U_{tt} = U_{xx}$. To compute this, we take u_x as computed before, and differentiate it using the WENO$^-$ method. Now we can compute the building block method.

For Burgers' equation, we have $U_{tt} = -(UU_t)_x$. We take the approximation to U_t that we obtained above

using $WENO^+$, we multiply it by u^n and differentiate in space using $WENO^-$. The choice of $WENO^+$ followed by $WENO^-$ is made by analogy to the first order finite difference for the linear advection case, where we use a differentiation operator D^+ followed by the downwind differentiation operator D^- to produce a centered difference for the second derivative. The second derivative condition (12) was satisfied by this approach. Now we compute the building block method with the approximations to U_t and U_{tt}. In pseudocode, the building block calculation takes the form:

$$\begin{aligned} f(u^n) &= \frac{1}{2}(u^n)^2; \quad u_t^n = WENO^+(f(u^n)); \\ f'(u^n) &= u^n; \quad f(u^n)_t = f'(u^n)u_t^n \quad u_{tt}^n = WENO^-(f(u^n)_t) \\ u^{n+1} &= u^n + \alpha\Delta t u_t^n + \beta\Delta t^2 u_{tt}^n. \end{aligned}$$

We use the two stage third order SSP method (19), and the non-SSP method (25) the two stage fourth order method (10) and the three stage fourth order method in Appendix A, the fifth order method (23). In these simulations, we use $\Delta t = \lambda\Delta x$ where $0.05 \le \lambda \le 1.6$, and step up to $T_{final} = 1.0$. At each time-step we compute (27), the maximal rise in total variation compared to the total variation of the initial solution. In Figure 7 we observe similar behavior to those of the linear advection with first order time-stepping, and once again see that the SSP method is needed to preserve the nonlinear stability of WENO as well.

4.2 Convergence studies

As a final test case, we investigate the accuracy of the proposed schemes in conjunction with various high order spatial discretization operators. We perform several tests that demonstrate that these methods converge with the correct order for linear and nonlinear problems. In the first study (Example 3), we refine the grid only in time, and show that *if the spatial discretization is sufficiently accurate* the multi-derivative methods exhibit the design-order of convergence. We also compare the performance of the third order multi-derivative method to the three stage third order explicit SSP Runge–Kutta method (SSPRK3,3) [32] and show that the convergence properties are the same, indicating that the additional error in approximating F_t does not affect the accuracy of the method. In the second study (Example 4) we co-refine the spatial and temporal grid by setting $\Delta t = \lambda\Delta x$ for a fixed λ, and shrink Δx. We observed that since the order of the spatial method is higher than the order of the temporal discretization, the time-stepping method achieves the design-order of accuracy both for linear and nonlinear problems.

Example 3a: temporal grid refinement with pseudospectral approximation of the spatial derivative. We begin with a linear advection problem $U_t + U_x = 0$ with periodic boundary conditions and initial conditions $u_0(x) = 0.5 + 0.5\sin(x)$ on the spatial domain $x \in [0, 2\pi]$. We discretize the spatial grid with $N = 41$ equidistant points and use the Fourier pseudospectral differentiation matrix $\mathbf{D}$ [10] to compute $F \approx -U_x \approx -\mathbf{D}u$. We use a Lax-Wendroff approach to approximate $F_t \approx U_{xx} \approx \mathbf{D}^2 u$. In this case, the solution is a sine wave, so that the pseudospectral method is exact. For this reason, the spatial discretization of F is exact and contributes no errors, and the second derivative F_t is also exact $F_t = u_{tt} = -\mathbf{D}u_t = -\mathbf{D}F = \mathbf{D}^2 u = \dot{F}$. We use a range of time steps, $\Delta t = \lambda\Delta x$ where we pick $\lambda = 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1$, and 0.05 to compute the solution to final time $T_f = 2.0$.

In Table 4 we list the errors for the SSPRK3,3, the two derivative two stage third order method in Section 3.2 with $K = \frac{1}{\sqrt{2}}$ (listed as 2s3p in the table), the unique the two derivative two stage fourth order method (2s4p), and the the two derivative three stage fifth order method (22) (3s5p). We observe that the design-order of each method is verified. It is interesting to note that the SSPRK3,3 method has larger errors than the multiderivative method 2s3p, demonstrating that the additional computation of $\dot{F}$ does improve the quality of the solution.

Example 3b: temporal grid refinement with WENO approximations of the spatial derivative. Using the same problem as above we discretize the spatial grid with $N = 101$ equidistant points and use the ninth order weighted essentially non-oscillatory method (WENO9) [1] to differentiate the spatial derivatives. It is interesting to note that although the PDE we solve is linear, the use of the nonlinear method WENO9 results in a non-linear

	SSPRK 3,3		2s3p		2s4p		3s5p	
λ	**error**	**Order**	**error**	**Order**	**error**	**Order**	**error**	**Order**
0.8	7.99×10^{-5}	—	1.86×10^{-5}	—	1.96×10^{-6}	—	6.47×10^{-8}	—
0.7	5.24×10^{-5}	3.16	1.21×10^{-5}	3.17	1.12×10^{-6}	4.17	3.24×10^{-8}	5.17
0.6	3.27×10^{-5}	3.04	7.61×10^{-6}	3.05	6.02×10^{-7}	4.04	1.49×10^{-8}	5.04
0.5	1.93×10^{-5}	2.87	4.50×10^{-6}	2.88	2.97×10^{-7}	3.87	6.12×10^{-9}	4.87
0.4	9.70×10^{-6}	3.10	2.25×10^{-6}	3.10	1.18×10^{-7}	4.10	1.96×10^{-9}	5.09
0.3	4.09×10^{-6}	3.00	9.50×10^{-7}	3.00	3.76×10^{-8}	3.99	4.66×10^{-10}	5.00
0.2	1.21×10^{-6}	2.99	2.81×10^{-7}	3.00	7.43×10^{-9}	4.01	6.13×10^{-11}	5.00
0.1	1.50×10^{-7}	3.01	3.49×10^{-8}	3.01	4.61×10^{-10}	4.00	1.90×10^{-12}	5.01
0.05	1.88×10^{-8}	2.99	4.36×10^{-9}	3.00	2.88×10^{-11}	3.97	5.97×10^{-14}	4.99

Table 4: Convergence study for Example 3a, the linear advection problem with pseudospectral differentiation of the spatial derivatives. Here we use $N = 41$ equidistant points between $(0, 2\pi)$, and $\Delta t = \lambda \Delta x$. The solution is evolved forward to time $T_f = 2.0$ using the explicit SSP Runge–Kutta method (SSPRK3,3), the two-stage third order two derivative method (2s3p), the two-stage fourth order two derivative method (2s4p), and three-stage fifth order two derivative method (3s5p).

ODE. To evolve this ODE in time we use a range of time steps defined by $\Delta t = \lambda \Delta x$ for $\lambda = 0.9, 0.8, 0.7, 0.6, 0.5, 0.4$ to compute the solution to final time $T_f = 2.0$ In Table 5 we list the errors for the SSPRK3,3, the two derivative two stage third order method in Section 3.2 with $K = \frac{1}{\sqrt{2}}$ (listed as 2s3p in the table), the unique the two derivative two stage fourth order method (2s4p), and the the two derivative three stage fifth order method (22) (3s5p). We observe that the design-order of each method is verified and that once again the SSPRK3,3 method has larger errors than the multiderivative method 2s3p. These results indicate that the approximation of the second derivative term F_t via a Lax-Wendroff procedure and discretization in space does not affect the observed order of the time-stepping method, as long as the spatial errors do not dominate.

To see what happens if the spatial discretization errors dominate over the time errors, we compare two different WENO spatial discretizations: the fifth order method WENO5 and the ninth order method WENO9. Here, we use $N = 301$ points in space, and we choose $\Delta t = \lambda \Delta x$ for $\lambda = 0.8, 0.6, 0.4, 0.2, 0.1, 0.05$. We evolve the solution in time to $T_f = 2.0$ using the 2s3p and SSPRK3,3 methods. The log-log graph of the errors vs. Δt is given in Figure 8 We observe that the errors from both time discretizations with the WENO5 spatial discretization (dotted lines) have the correct orders when Δt is larger and the time errors dominate, but as Δt gets smaller the spatial errors dominate convergence is lost. On the other hand, for the range of Δt studied, the time-errors dominate over the spatial errors when using the highly accurate WENO9 (solid line) and so convergence is not lost. We note that to see this behavior, the spatial approximation must be sufficiently accurate. In this case this is attained by using 301 points in space and a high order spatial discretization. We also studied this problem with a co-refinement of the spatial and temporal grids, where we use $\Delta t = 0.8 \Delta x$ for $\Delta x = \frac{2\pi}{N-1}$ and $N = 41, 81, 161, 321$. Figure 9 shows that while for larger Δt the errors for WENO5 are larger than for WENO9, and this is more pronounced for the multi-derivative methods than for the explicit Runge–Kutta, once the grid is sufficiently refined the temporal error dominates and we see the third order convergence in time.

It is interesting to note that in all these studies the explicit SSP Runge–Kutta method SSPRK3,3 and the two-derivative method 2s3p behave similarly, indicating that the approximation of the second derivative does not play a role in the loss of accuracy.

Example 4: co-refinement of the spatial and temporal grids on linear advection with WENO7

Example 4a: In this example, we compare the errors and order of convergence of the multiderivative methods on

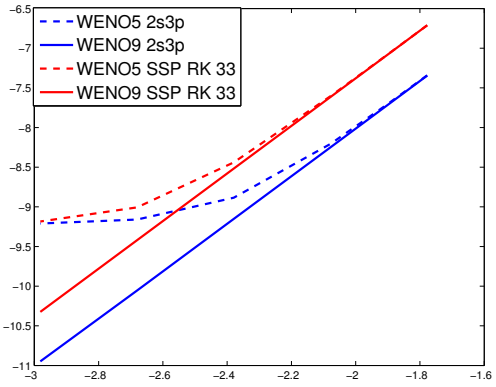


Figure 8: WENO5 vs WENO9 with grid refinement in time but not space Example 3b. On the x-axis are $log_{10}(\Delta t)$ and on the y-axis are $log_{10}(error)$. The errors from both time discretizations with the WENO5 spatial discretization (dotted lines) have the correct orders when Δt is larger and the time errors dominate, but as Δt gets smaller the spatial errors dominate convergence is lost.

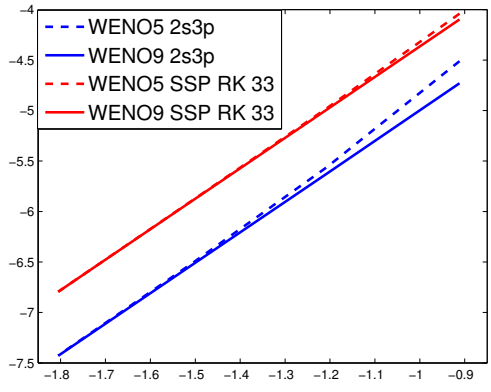


Figure 9: Example 3b: WENO5 vs WENO9 with refinement in both space and time, $\Delta t = 0.8\Delta x$. On the x-axis are $log_{10}(\Delta t)$ and on the y-axis are $log_{10}(error)$. Here we see that for larger Δt the errors for WENO5 are larger than for WENO9 (especially for the multiderivative method) but once the grid is sufficiently refined the temporal order dominates and third order convergence in time is observed.

a linear and nonlinear problem. For the linear problem, we use the linear advection problem

$$U_t + U_x = 0 \quad \text{on } x \in [-1, 1]$$

with the initial condition

$$u_0(x) = 0.5 + 0.5\sin(\pi x), \quad x \in [-1, 1], \tag{30}$$

We compute both F and $\dot{F}$ using a seventh order WENO (WENO7) [1] spatial derivative. We set $\Delta t = 0.8\Delta x$ and evolve the solution to final time of $T_{final} = 2.0$, at which point the exact solution is identical to the initial state. For time-stepping, we use the same three multiderivative schemes 2s3p, 2s4p, and 3s5p and the explicit SSP Runge–Kutta method SSPRK3,3. In Table 6, where we compare the errors and orders of these three methods. These numerical experiments show that once the mesh is sufficiently refined, the design-order of accuracy is reached. The results verify that the proposed schemes are genuinely high-order accurate despite the fact that the second derivative is not an exact second derivative of the method of lines formulation.

Example 4b: Next, we present results for the more difficult non-linear Burgers equation, with initial conditions prescribed by

$$U_0(x) = 1.0 + 0.2\sin(\pi x), \quad x \in [-1, 1], \tag{31}$$

and periodic boundary conditions. Once again, F and $\dot{F}$ are computed via the WENO7 spatial discretization. We use a constant time-step $\Delta t = 0.8\frac{\Delta x}{\max_i |u_0(x_i)|}$ and run this problem with a final time of $T_{final} = 1.4$, before a shock forms. Since the solution at this point the solution remains smooth, we can use the method of characteristics to compute the the exact solution by

$$U(t, x) = U_0\left(x - t \cdot U_0(\xi)\right), \quad \xi = x - t \cdot U_0(\xi). \tag{32}$$

	SSPRK 3,3		2s3p		2s4p		3s5p	
λ	**error**	**Order**	**error**	**Order**	**error**	**Order**	**error**	**Order**
0.9	7.37×10^{-6}	—	1.71×10^{-6}	—	8.25×10^{-8}	—	1.24×10^{-9}	—
0.8	5.24×10^{-6}	2.89	1.22×10^{-6}	2.89	5.21×10^{-8}	3.89	6.99×10^{-10}	4.89
0.7	3.44×10^{-6}	3.14	7.99×10^{-7}	3.14	3.00×10^{-8}	4.14	3.52×10^{-10}	5.14
0.6	2.18×10^{-6}	2.96	5.08×10^{-7}	2.96	1.63×10^{-8}	3.96	1.64×10^{-10}	4.96
0.5	1.27×10^{-6}	2.98	2.94×10^{-7}	2.98	7.89×10^{-9}	3.98	6.61×10^{-11}	4.97
0.4	6.47×10^{-7}	3.01	1.50×10^{-7}	3.01	3.22×10^{-9}	4.01	2.18×10^{-11}	4.97

Table 5: Convergence study for Example 3b, the linear advection problem with WENO9 differentiation of the spatial derivatives. Here we use $N = 101$ equidistant points between $(0, 2\pi)$, and $\Delta t = \lambda \Delta x$.

	SSPRK 3,3		2s3p		2s4p		3s5p	
N	**error**	**order**	**error**	**order**	**error**	**order**	**error**	**order**
41	3.00×10^{-04}	—	5.94×10^{-05}	—	7.54×10^{-06}	− − −	2.59×10^{-06}	—
81	3.75×10^{-05}	3.00	7.39×10^{-06}	3.01	4.71×10^{-07}	4.00	8.03×10^{-08}	5.01
161	4.69×10^{-06}	3.00	9.23×10^{-07}	3.00	2.94×10^{-08}	4.00	2.51×10^{-09}	5.00
321	5.86×10^{-07}	3.00	1.15×10^{-07}	3.00	1.84×10^{-09}	4.00	7.82×10^{-11}	5.00
641	7.32×10^{-08}	3.00	1.44×10^{-08}	3.00	1.15×10^{-10}	4.00	2.45×10^{-12}	5.00
1281	9.15×10^{-09}	3.00	1.80×10^{-09}	3.00	7.19×10^{-12}	4.00	7.75×10^{-14}	4.98

Table 6: Convergence study for Example 4a: linear advection with WENO7 spatial differentiation and refinement of both spatial and temporal grids. Here we use N points in space and $\Delta t = 0.8\Delta x$. The four methods used to evolve the solution to time $T_f = 2.0$ are the explicit SSP Runge–Kutta method SSPRK3,3 and the three multiderivative methods 2s3p, 2s4p, and 3s5p. Despite the fact that the second derivative $\dot{F}$ is not the exact second derivative F_t, we still observe high-order accuracy for all methods once the grids are sufficiently refined.

	SSPRK 3,3		2s3p		2s4p		3s5p	
N	**error**	**order**	**error**	**order**	**error**	**order**	**error**	**order**
161	3.09×10^{-04}	—	1.91×10^{-04}	—	1.58×10^{-04}	—	1.67×10^{-04}	—
321	5.11×10^{-05}	2.60	2.00×10^{-05}	3.25	1.33×10^{-05}	3.57	1.76×10^{-05}	3.25
641	6.96×10^{-06}	2.88	1.70×10^{-06}	3.56	7.12×10^{-07}	4.22	9.34×10^{-07}	4.23
1281	8.80×10^{-07}	2.98	1.81×10^{-07}	3.23	3.38×10^{-08}	4.39	2.73×10^{-08}	5.09
2561	1.10×10^{-07}	3.00	2.22×10^{-08}	3.03	1.99×10^{-09}	4.09	7.49×10^{-10}	5.19
5121	1.38×10^{-08}	3.00	2.76×10^{-09}	3.01	1.24×10^{-10}	4.01	2.21×10^{-11}	5.08
10241	1.72×10^{-09}	3.00	3.44×10^{-10}	3.00	7.75×10^{-12}	4.00	7.04×10^{-13}	4.97

Table 7: Convergence study for Example 4b: Burgers' equation with WENO7 spatial differentiation and refinement of both spatial and temporal grids. Here we use N points in space and $\Delta t = 0.8\Delta x$. The four methods used to evolve the solution to time $T_f = 2.0$ are the explicit SSP Runge–Kutta method SSPRK3,3 and the three multiderivative methods 2s3p, 2s4p, and 3s5p. A more refined grid is needed in this example compared to the linear example, but for a sufficiently refined grid we still observe high-order accuracy for all methods.

We solve for the implicit variable ξ using Netwon iteration with a tolerance of 10^{-14}. The errors and order are presented in Table 7. We see that it takes an even smaller mesh size than the linear problem before the errors reach the asymptotic regime, but that once this happens we achieve the expected order. Again, these numerical experiments further validate the fact that if the spatial error is not allowed to dominate, the high-order design-accuracy of the time discretization attained despite the fact that we do not directly differentiate the method of lines formulation to define the second derivative.

5 Conclusions

With the increasing popularity of multi-stage multiderivative methods for use as time-stepping methods for hyperbolic problems, the question of their strong stability properties needs to be addressed. In this work we presented an SSP formulation for multistage two-derivative methods. We assumed that, in addition to the forward Euler condition, the spatial discretization of interest satisfies a second derivative condition of the form (12). With these assumptions in mind, we formulated an optimization problem which enabled us to find optimal explicit SSP multistage two-derivative methods of up to order five, thus breaking the SSP order barrier for explicit SSP Runge–Kutta methods. Numerical test cases verify the convergence of these methods at the design-order, show that sharpness of the SSP condition in many cases, and demonstrate the need for SSP time-stepping methods in simulations where the spatial discretization is specially designed to satisfy certain nonlinear stability properties. Future work will involve building SSP multiderivative methods while assuming different base conditions (as in Remark 1) and with higher derivatives. Additional work will involve developing new spatial discretizations suited for use with SSP multiderivative time stepping methods. These methods will be based on WENO or discontinuous Galerkin methods and will satisfy pseudo-TVD and similar properties for systems of equations.

Acknowledgements This work was supported by: AFOSR grants FA9550-12-1-0224, FA9550-12-1-0343, FA9550-12-1-0455, FA9550-15-1-0282, and FA9550-15-1-0235; NSF grant DMS-1418804; New Mexico Consortium grant NMC0155-01; and NASA grant NMX15AP39G.

A Coefficients of three stage fourth order methods

1. For $K = \frac{1}{2}$ we obtain an SSP coefficient $\mathcal{C} = r = 1.1464$. The Butcher array coefficients are given by

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0.436148675945340 & 0 & 0 \\ 0.546571371212865 & 0.156647174804152 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 0.528992280543542 \\ 0.105732787708912 \\ 0.365274931747546 \end{bmatrix}$$

$$\hat{A} = \begin{bmatrix} 0 & 0 & 0 \\ 0.095112833764436 & 0 & 0 \\ 0.071032477596813 & 0.107904226252921 & 0 \end{bmatrix}, \quad \hat{b} = \begin{bmatrix} 0.074866026156687 \\ 0.073410341982927 \\ 0.048740310097159 \end{bmatrix}.$$

The Shu-Osher arrays are $R\mathbf{e} = (1, 0, 0, 0)^T$,

$$P = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.5000 & 0 & 0 & 0 \\ 0.253176729307242 & 0.179580018745470 & 0 & 0 \\ 0.181986129712082 & 0.000000001889650 & 0.418750476492704 & 0 \end{bmatrix}$$

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0 \\ 0 & 0.567243251947287 & 0 & 0 \\ 0.140002406985210 & 0.003037359947341 & 0.256223624973014 & 0 \end{bmatrix}$$

2. For $K = \sqrt{\frac{1}{2}}$ we obtain an SSP coefficient $\mathcal{C} = r = 1.3927$ Butcher formulation

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0.443752012194422 & 0 & 0 \\ 0.543193299768317 & 0.149202742858795 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 0.515040964378407 \\ 0.178821699719783 \\ 0.306137335901811 \end{bmatrix}$$

$$\hat{A} = \begin{bmatrix} 0 & 0 & 0 \\ 0.098457924163299 & 0 & 0 \\ 0.062758211639901 & 0.110738910914425 & 0 \end{bmatrix}, \quad \hat{b} = \begin{bmatrix} 0.072864982225864 \\ 0.073840478463180 \\ 0.061973770357455 \end{bmatrix}.$$

The Shu-Osher arrays are $R\mathbf{e} = (1, 0, 0, 0)^T$,

$$P = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.618033988749895 & 0 & 0 & 0 \\ 0.362588515112176 & 0.207801573327953 & 0 & 0 \\ 0.144580879241747 & 0.110491604448675 & 0.426371652664792 & 0 \end{bmatrix}$$

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.381966011250105 & 0 & 0 & 0 \\ 0 & 0.429609911559871 & 0 & 0 \\ 0.078129569197367 & 0 & 0.240426294447419 & 0 \end{bmatrix}.$$

3. For $K = 1$ we obtain an SSP coefficient $\mathcal{C} = r = 1.6185$. The Butcher array coefficients are given by

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0.452297224196082 & 0 & 0 \\ 0.528050722182308 & 0.159236998008155 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 0.502519798444212 \\ 0.210741084344740 \\ 0.286739117211047 \end{bmatrix}$$

$$\hat{A} = \begin{bmatrix} 0 & 0 & 0 \\ 0.102286389507741 & 0 & 0 \\ 0.055482128781494 & 0.108677624192402 & 0 \end{bmatrix}, \quad \hat{b} = \begin{bmatrix} 0.071256397204544 \\ 0.069475972085130 \\ 0.066877749079721 \end{bmatrix}.$$

The Shu-Osher arrays are $R\mathbf{e} = (1, 0, 0, 0)^T$,

$$P = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.732050807568877 & 0 & 0 & 0 \\ 0.457580533944888 & 0.257727809835459 & 0 & 0 \\ 0.137886171629970 & 0.176326540063367 & 0.464092174540814 & 0 \end{bmatrix},$$

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.267949192431123 & 0 & 0 & 0 \\ 0 & 0.284691656219654 & 0 & 0 \\ 0.046502314960818 & 0 & 0.175192798805030 & 0 \end{bmatrix}.$$

B Two stage third order method

This code gives the SSP coefficient and the Butcher and Shu Osher arrays for the optimal explicit SSP two stage third order method given the value K.

```
clear all
k=sqrt(0.5); % Choose K
tab=[];
% Set up the polynomial for Q(3,1) to solve for SSP coefficient r
AA=sqrt(k^2+2) - k;
p0=2*k*(AA-2*k) + 4*k^3*AA;
p1=-p0;
p2=(1-p0)/(2*k^2);
p3= -(p0/(2*k) + k)/(6*k^3);
CC=[p3,p2,p1,p0]; % polynomial coefficients
RC=roots(CC);
r=RC(find(abs(imag(RC))<10^-15)); % SSP coefficient is the only real root.
%-------------------------------------------------------------------------
% Once we have the k and r we want we define the method following (21)
a= (k*sqrt(k^2+2)-k^2)/r;
b2 = ((k^2*(1-1/r)) + r*(.5-1/(6*a)))/(k^2+.5*r*a);
b1=1-b2;
ahat=.5*a^2;
bhat1=.5*(1-b2*a)-1/(6*a);
bhat2=1/(6*a)-.5*b2*a;
% The Butcher arrays are given by
A=zeros(2,2); Ahat=A;
A(2,1)=a;
b=[b1,b2];
Ahat(2,1)=ahat;
bhat=[bhat1,bhat2];
S=[0 0 0 ; a 0 0; b1 b2 0];
Shat=[0 0 0 ; ahat 0 0 ; bhat1 bhat2 0];
% The Shu-Osher matrices are given by
I=eye(3);e=ones(3,1);
Ri=I+r*S+(r^2/k^2)*Shat;
v=Ri\e;
P = r*(Ri\S);
Q= r^2/k^2*(Ri\Shat);
violation=min(min([v, S, Shat, P, Q])) % use this to check that all these are positive
tab=[tab;[k,r,violation]]; % build the table of values
```

C Three stage fifth order method

This code gives the SSP coefficient and the Butcher and Shu Osher arrays for the optimal explicit SSP three stage fifth order method given the value K.

```
clear all
format long
syms r
tab=[];
k=sqrt(0.5) %The second derivative condition coefficient K
```

```
% Find the SSP coefficient C given K
a21= 240*k^6*(1 -r - r^2/(2*k^2) + r^3/(6*k^2) + r^4/(24*k^4) - r^5/(120*k^4))/r^6;
Q31=10*r^2*a21^4 - 100*k^2*a21^3 - 10*r^2*a21^3 + 130*k^2*a21^2  + 3*r^2*a21^2 - 50*k^2*a21 +6*k^2;
RC=vpasolve(simplify(r^22*Q31)==0);
rr=RC(find(abs(imag(RC))<10^-15));
C= max(rr) %The SSP coefficient
% ------------------------------------------------------------------------
% The Butcher array coefficients given K and C
a21= 240*k^6*(1 -C - C^2/(2*k^2) + C^3/(6*k^2) + C^4/(24*k^4) - C^5/(120*k^4))/C^6;
ah32= ( (3/5 -a21)^2/(a21*(1-2*a21)^3) - (3/5 -a21)/(1-2*a21)^2 )/10;
ah31= ( (3/5 -a21)^2/(1-2*a21)^2)/2  -ah32;
a31= (3/5 -a21)/(1-2*a21);
bh2=(2*a31-1)/(12*a21*(a31-a21));
bh3=(1-2*a21)/(12*a31*(a31-a21));
bh1=1/2-bh2-bh3;
ah21=(1/24 - bh3*(ah31+ah32) )/bh2;
% Build the Butcher matrices
a= C*a21+ah21*C^2/k^2;
b= C*a31+ah31*C^2/k^2;
c= ah32*C^2/k^2;
d= C+ bh1*C^2/k^2;
e= bh2*C^2/k^2;
f = bh3*C^2/k^2;
Ri=([1 0 0 0; a 1 0 0 ; b c 1 0; d e f 1]);
S=[0 0 0 0; a21 0 0 0; a31 0 0 0 ;1 0 0 0];
Shat=[0 0 0 0; ah21 0 0 0; ah31 ah32 0 0 ;bh1 bh2 bh3 0];
% The Shu Osher matrices given C
eone=ones(4,1)
v=Ri\eone;
P = C*(Ri\S);
Q= C^2/k^2*(Ri\Shat);
% Double check that there are no violations of the SSP conditions:
violation=min(min([v, S, Shat, P, Q])) % use this to check that all these are positive
tab=[tab;[k,a21,C,violation]]; % build the table of values
```

References

[1] D. S. Balsara and C.-W. Shu, *Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy*, Journal of Computational Physics, 160 (2000), pp. 405–452.

[2] C. Bresten, S. Gottlieb, Z. Grant, D. Higgs, D. I. Ketcheson, and A. Németh, *Strong stability preserving multistep Runge-Kutta methods.* Accepted for publication in Mathematics of Computation.

[3] R. P. K. Chan and A. Y. J. Tsai, *On explicit two-derivative Runge-Kutta methods*, Numerical Algorithms, 53 (2010), pp. 171–194.

[4] L. Ferracina and M. N. Spijker, *Stepsize restrictions for the total-variation-diminishing property in general Runge–Kutta methods*, SIAM Journal of Numerical Analysis, 42 (2004), pp. 1073–1093.

[5] ———, *An extension and analysis of the Shu–Osher representation of Runge–Kutta methods*, Mathematics of Computation, 249 (2005), pp. 201–219.

[6] E. Gekeler and R. Widmann, *On the order conditions of Runge-Kutta methods with higher derivatives*, Numer. Math., 50 (1986), pp. 183–203.

[7] S. Gottlieb, D. I. Ketcheson, and C.-W. Shu, *Strong Stability Preserving Runge–Kutta and Multistep Time Discretizations*, World Scientific Press, 2011.

[8] Z. J. Grant, *Explicit SSP multistage two-derivative SSP optimization code.* https://github.com/SSPmethods/SSPMultiStageTwoDerivativeMethods, February 2015.

[9] A. Harten, B. Engquist, S. Osher, and S. R. Chakravarthy, *Uniformly high-order accurate essentially nonoscillatory schemes. III*, J. Comput. Phys., 71 (1987), pp. 231–303.

[10] J. Hesthaven, S. Gottlieb, and D. Gottlieb, *Spectral methods for time dependent problems*, Cambridge Monographs of Applied and Computational Mathematics, Cambridge University Press, 2007.

[11] I. Higueras, *On strong stability preserving time discretization methods*, Journal of Scientific Computing, 21 (2004), pp. 193–223.

[12] ———, *Representations of Runge–Kutta methods and strong stability preserving methods*, SIAM Journal On Numerical Analysis, 43 (2005), pp. 924–948.

[13] G.-S. Jiang and C.-W. Shu, *Efficient implementation of weighted ENO schemes*, J. Comput. Phys., 126 (1996), pp. 202–228.

[14] K. Kastlunger and G. Wanner, *On Turan type implicit Runge-Kutta methods*, Computing (Arch. Elektron. Rechnen), 9 (1972), pp. 317–325. 10.1007/BF02241605.

[15] K. H. Kastlunger and G. Wanner, *Runge Kutta processes with multiple nodes*, Computing (Arch. Elektron. Rechnen), 9 (1972), pp. 9–24.

[16] D. I. Ketcheson, *Highly efficient strong stability preserving Runge–Kutta methods with low-storage implementations*, SIAM Journal on Scientific Computing, 30 (2008), pp. 2113–2136.

[17] D. I. Ketcheson, S. Gottlieb, and C. B. Macdonald, *Strong stability preserving two-step Runge-Kutta methods*, SIAM Journal on Numerical Analysis, (2012), pp. 2618–2639.

[18] D. I. Ketcheson, C. B. Macdonald, and S. Gottlieb, *Optimal implicit strong stability preserving Runge–Kutta methods*, Applied Numerical Mathematics, 52 (2009), p. 373.

[19] D. I. Ketcheson, M. Parsani, and A. J. Ahmadia, *Rk-opt: Software for the design of Runge–Kutta meththods, version 0.2.* https://github.com/ketch/RK-opt.

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

[21] P. Lax and B. Wendroff, *Systems of conservation laws*, Communications in Pure and Applied Mathematics, 13 (1960), pp. 217–237.

[22] T. Mitsui, *Runge-Kutta type integration formulas including the evaluation of the second derivative. i.*, Publ. Res. Inst. Math. Sci., 18 (1982), pp. 325–364.

[23] T. Nguyen-Ba, H. Nguyen-Thu, T. Giordano, and R. Vaillancourt, *One-step strong-stability-preserving Hermite-Birkhoff-Taylor methods*, Scientific Journal of Riga Technical University, 45 (2010), pp. 95–104.

[24] N. Obreschkoff, *Neue Quadraturformeln*, Abh. Preuss. Akad. Wiss. Math.-Nat. Kl., 4 (1940).

[25] H. Ono and T. Yoshida, *Two-stage explicit Runge-Kutta type methods using derivatives.*, Japan J. Indust. Appl. Math., 21 (2004), pp. 361–374.

[26] J. Qiu, M. Dumbser, and C.-W. Shu, *The discontinuous Galerkin method with Lax–Wendroff type time discretizations*, Computer Methods in Applied Mechanics and Engineering, 194 (2005), pp. 4528–4543.

[27] J. Qiu and C.-W. Shu, *Finite Difference WENO schemes with Lax–Wendroff-type time discretizations*, SIAM Journal on Scientific Computing, 24 (2003), pp. 2185–2198.

[28] S. J. Ruuth and R. J. Spiteri, *Two barriers on strong-stability-preserving time discretization methods*, Journal of Scientific Computation, 17 (2002), pp. 211–220.

[29] D. C. Seal, Y. Guclu, and A. J. Christlieb, *High-order multiderivative time integrators for hyperbolic conservation laws*, Journal of Scientific Computing, 60 (2014), pp. 101–140.

[30] H. Shintani, *On one-step methods utilizing the second derivative*, Hiroshima Mathematical Journal, 1 (1971), pp. 349–372.

[31] ——, *On explicit one-step methods utilizing the second derivative*, Hiroshima Mathematical Journali, 2 (1972), pp. 353–368.

[32] C.-W. Shu, *Total-variation diminishing time discretizations*, SIAM J. Sci. Stat. Comp., 9 (1988), pp. 1073–1084.

[33] C.-W. Shu and S. Osher, *Efficient implementation of essentially non-oscillatory shock-capturing schemes*, Journal of Computational Physics, 77 (1988), pp. 439–471.

[34] 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.

[35] D. D. Stancu and A. H. Stroud, *Quadrature formulas with simple Gaussian nodes and multiple fixed nodes*, Math. Comp., 17 (1963), pp. 384–394.

[36] E. Toro and V. Titarev, *Solution of the generalized Riemann problem for advection–reaction equations*, Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, 458 (2002), pp. 271–281.

[37] A. Y. J. Tsai, R. P. K. Chan, and S. Wang, *Two-derivative Runge–Kutta methods for PDEs using a novel discretization approach*, Numerical Algorithms, 65 (2014), pp. 687–703.

[38] P. Turán, *On the theory of the mechanical quadrature*, Acta Sci. Math. Szeged, 12 (1950), pp. 30–37.