A NEW CLASS OF OPTIMAL HIGH-ORDER STRONG-STABILITY-PRESERVING TIME DISCRETIZATION METHODS

RAYMOND J. SPITERI* AND STEVEN J. RUUTH†

Abstract. Strong-stability-preserving (SSP) time discretization methods have a nonlinear stability property that makes them particularly suitable for the integration of hyperbolic conservation laws where discontinuous behaviour is present. Optimal SSP schemes have been previously found for methods of order 1, 2, and 3, where the number of stages s equals the order p. An optimal low-storage SSP scheme with $s = p = 3$ is also known. In this paper, we present a new class of optimal high-order SSP and low-storage SSP Runge-Kutta schemes with $s > p$. We find that these schemes are ultimately more efficient than the known schemes with $s = p$ because the increase in the allowable time step more than offsets the added computational expense per step. We demonstrate these efficiencies on a set of scalar conservation laws.

Key words. strong stability preserving, total variation diminishing, Runge-Kutta methods, high-order accuracy, time discretization

AMS subject classifications. 65L06, 65M20

1. Introduction. The method of lines is a popular semi-discretization method for the solution of time-dependent partial differential equations (PDEs). The idea behind it is to first suitably discretize the spatial variables (e.g., by finite differences, finite volumes, finite elements, or spectral methods) to yield a set of ordinary differential equations (ODEs) in time. Then, this set of ODEs can be integrated using standard time-stepping techniques such as linear multi-step or Runge-Kutta methods.

Standard stability analysis for the solvers of such systems generally focuses on linear stability. Indeed, such analysis is often adequate when the desired solutions are smooth. However, solutions to hyperbolic PDEs may not be smooth: Shock waves or other discontinuous behaviour can develop even from smooth initial data. In such cases, standard discretizations based on linear stability analysis suffer from poor performance due to the presence of spurious oscillations, overshoots, and progressive smearing. The numerical solutions obtained from these discretizations often exhibit a weak form of instability (called *nonlinear instability*) resulting in unphysical behaviour. Accordingly, numerical methods based on a nonlinear stability requirement are very desirable. Such methods were originally referred to as *total variation diminishing* (TVD) [17]; see also the subsequent articles [18, 6]. However, following the more recent article [7], we refer to them in this paper as *strong stability preserving* (SSP) methods.

We are interested in the development, implementation, and analysis of a new class of optimal SSP Runge-Kutta (SSPRK) time-stepping schemes for the system of ODEs

$$\dot{U} = L(U),$$

*Department of Computer Science, Dalhousie University, Halifax, Nova Scotia, B3H 1W5 Canada (`spiteri@cs.dal.ca`). The work of this author was partially supported by grants from NSERC Canada and Imperial Oil.

†Department of Mathematics, Simon Fraser University, Burnaby, British Columbia, V5A 1S6 Canada (`sruuth@sfu.ca`). The work of this author was partially supported by an NSERC Canada grant.

subject to suitable initial conditions, obtained from applying the method of lines to the hyperbolic conservation law

$$u_t + f(u)_x = 0. \tag{1.1}$$

Here, we assume that (1.1) has been suitably discretized in its spatial variables (e.g., using essentially non-oscillatory (ENO) schemes [10], TVD schemes [9], or monotonic upstream-centered schemes for conservation laws (MUSCL) methods [19]) and $U = U(t)$ is a vector of discretized variables; i.e., $[U(t)]_j = U_j(t) = u(x_j, t)$. In particular, if u_j^n is the numerical approximation to $u(x_j, t_n)$, then TVD discretizations have the property that the total variation

$$TV(U^n) = \sum_j |u_j^n - u_{j-1}^n| \tag{1.2}$$

of the numerical solution does not increase with time; i.e.,

$$TV(U^{n+1}) \leq TV(U^n).$$

When combined with a suitable SSP time-stepping scheme, the numerical solution obtained typically does not exhibit nonlinear instabilities. However, nonlinear instabilities can occur in a numerical solution obtained with, e.g., a TVD or MUSCL spatial discretization scheme, but a standard (i.e., linearly stable) time-stepping scheme [6]. Hence, strong-stability-preserving time-stepping schemes are a critical part of the overall solution strategy to (1.1).

It has been known for some time from a result of Goodman and LeVeque [5] that any method that is TVD in two dimensions is at most first-order accurate. However, if we relax the strict requirement of TVD, higher-order methods can be constructed that preserve stability in another suitable norm, such as the maximum norm. These schemes are what we call strong stability preserving, and their favourable properties are derived only from convexity arguments. In particular, if the forward Euler method is strongly stable with a certain CFL number, higher-order SSP Runge-Kutta methods with a modified CFL number can be constructed as convex combinations of forward Euler steps with various step sizes [18].

Optimal SSP schemes based on Runge-Kutta methods have been found for accuracy orders 1, 2, and 3, where the number of stages s is assumed to be equal to the order p. Unfortunately, Gottlieb and Shu [6] recently proved that no such four-stage, fourth-order SSPRK method exists involving just evaluations of $L(\cdot)$. Fourth-order accuracy has only been obtained at the additional expense of introducing two additional evaluations of a related operator $\tilde{L}(\cdot)$, leading to sub-optimal efficiency both in terms of time-step restriction and memory usage (see Section 2). This appears to be where the search for higher-order SSPRK methods has stopped, thus leaving researchers to focus on third-order accurate SSPRK methods.

In this paper, we derive a new class of optimal high-order SSPRK schemes where the restriction $s = p$ is lifted. For s-stage methods of orders 1 and 2, we provide proofs of optimality. The SSPRK scheme (4,3) is also proven to be optimal. The remaining schemes of order 3 and higher and the low-storage schemes are the results of numerical optimization. We investigate the performance of our new schemes on a few test problems designed to capture solution features that pose particular difficulties to numerical methods. These features include contact discontinuities, expansion fans, compressive shocks, and sonic points. The results from these investigations indicate that both the standard and low-storage versions of our schemes offer significant advantages over methods currently available. In particular, our new schemes have significantly better stability restrictions than the best SSPRK schemes currently known. Thus, step-size selection can be based more on accuracy requirements rather than stability requirements, ultimately leading to more efficient integrators. Indeed, the results based on three important test cases indicate that our new fourth-order SSPRK scheme offers between 40% and 80% improvements in the effective time-step restriction over the most popular fourth-order schemes currently in use.

The remainder of this paper unfolds as follows. In Section 2, we describe SSP schemes and motivate their use. In Section 3, we determine optimal families of SSP Runge-Kutta schemes up to five stages and order four. We also give optimal low-storage versions of these schemes. In Section 4, we investigate the performance of our new SSP Runge-Kutta schemes on a set of scalar conservation laws having solutions that commonly cause numerical problems. The success of the new methods is measured relative to the most popular schemes currently in use. Finally, in Section 5, we summarize our findings and offer plans for future work.

2. SSP Schemes. The concept of strong stability is central to our discussion, so we begin with its definition.

DEFINITION 2.1. *A sequence* $\{U^n\}$ *is said to be* strongly stable *in a given norm* $||\cdot||$ *provided that* $||U^{n+1}|| \leq ||U^n||$ *for all* $n \geq 0$.

We tacitly assume that U^n represents a vector of solution values on a mesh obtained from a method-of-lines approach to solving a PDE. The choice of norm is arbitrary[1], with the TV-norm (1.2) and the infinity norm being two natural possibilities. Clearly, strong stability may not be relevant to the solution of an arbitrary PDE. However, the class of PDEs (1.1) forms a notable exception. Exact solutions for this class of problems have a range-diminishing property that forbids existing maxima from increasing, existing minima from decreasing, and new maxima or minima from forming. Although not precisely a discrete analogue to the range-diminishing property, the strong-stability property is a useful property to require of a numerical solution to (1.1): By imposing such a condition on the numerical solution, we can suppress the formation of spurious oscillations under a suitable restriction on the time-step. Such oscillations are termed *nonlinear instabilities*, and are often a precursor for the numerical solution itself to become completely unstable.

The authors in [7] prove the somewhat surprising result that, under rather general assumptions, high-order SSP methods must in fact be explicit. Fortunately, many researchers in fact prefer explicit time discretization methods in order to avoid the expense[2] of solving systems of nonlinear equations at each step. Accordingly, in this paper we will focus on the development of explicit Runge-Kutta methods. Consider an s-stage, explicit Runge-Kutta method written in the form

$$U^{(0)} = U^n \tag{2.1a}$$

$$U^{(i)} = \sum_{k=0}^{i-1}(\alpha_{ik}U^{(k)} + \Delta t\beta_{ik}L(U^{(k)})), \qquad i = 1, 2, \ldots, s, \tag{2.1b}$$

$$U^{n+1} = U^{(s)}, \tag{2.1c}$$

[1]Indeed, the results of this paper still apply if we replace the norm $||\cdot||$ by *any* convex function that maps into the non-negative real line.

[2]both in terms of computation time and software development

where all the $\alpha_{ik} \geq 0$ and $\alpha_{ik} = 0$ only if $\beta_{ik} = 0$ [17]. This representation of a Runge-Kutta method can be converted to the standard Butcher array form (see e.g., [8]) in a straightforward manner; see also [6]. However, the conversion from the Butcher array form to (2.1) is not unique. For example, the modified Euler scheme

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

has a one-parameter family of representations of the form (2.1):

$$\alpha_{10} = 1,\ \alpha_{20} = 1 - \lambda,\ \alpha_{21} = \lambda, \qquad \beta_{10} = 1,\ \beta_{20} = \frac{1}{2} - \lambda,\ \beta_{21} = \frac{1}{2},$$

where $\lambda \in [0, 1]$. All of these representations are algebraically equivalent [18]; i.e., the only differences noticeable between stable implementations of any scheme would be due to round-off errors. However, different choices of λ may lend themselves more easily to implementation, memory management, or determination of stability restrictions. Throughout this article, we give representations that naturally allow stability restrictions to be read from the coefficients of the scheme. Standard Butcher array forms of the schemes presented are given in Appendix B.

For consistency, we must have that $\sum_{k=0}^{i-1} \alpha_{ik} = 1$, $i = 1, 2, \ldots, s$. Hence, if both sets of coefficients α_{ik}, β_{ik} are positive, then (2.1) is a convex combination of forward Euler steps with various step sizes $\frac{\beta_{ik}}{\alpha_{ik}} \Delta t$. The Runge-Kutta scheme written in this form is particularly convenient to make use of the following result [18, 7]:

THEOREM 2.2. *If the forward Euler method is strongly stable under the CFL restriction $\Delta t \leq \Delta t_{FE}$, then the Runge-Kutta method (2.1) with $\beta_{ik} \geq 0$ is SSP provided*

$$\Delta t \leq c \Delta t_{FE},$$

where c is the CFL coefficient

$$c \equiv \min_{i,k} \frac{\alpha_{ik}}{\beta_{ik}}.$$

Thus, we can use the result of this theorem to provide a theoretical criterion according to which we can optimize a given SSPRK method.

SSPRK schemes with negative coefficients β_{ik} are also possible with the appropriate interpretation. Following the procedure first suggested in [17], whenever $\beta_{ik} < 0$, the operator $L(\cdot)$ is replaced with the related operator $\tilde{L}(\cdot)$, where $\tilde{L}(\cdot)$ is assumed to be strongly stable for Euler's method solved *backwards* in time for a suitable time-step restriction. This allows the following generalization Theorem 2.2:

THEOREM 2.3. *Let Euler's method solved forward in time combined with the spatial discretization $L(\cdot)$ be strongly stable under the CFL restriction $\Delta t \leq \Delta t_{FE}$. Let Euler's method solved backward in time combined with the spatial discretization $\tilde{L}(\cdot)$ also be strongly stable under the same CFL restriction $\Delta t \leq \Delta t_{FE}$. Then the Runge-Kutta method (2.1) is SSP provided*

$$\Delta t \leq c \Delta t_{FE},$$

where c is the CFL coefficient

$$c \equiv \min_{i,k} \frac{\alpha_{ik}}{|\beta_{ik}|},$$

where $\beta_{ik}L(\cdot)$ is replaced by $\beta_{ik}\tilde{L}(\cdot)$ whenever β_{ik} is negative.

Note: If both $L(U^{(i)})$ and $\tilde{L}(U^{(i)})$ are required, then the computational cost and storage requirements for that stage are typically doubled. Moreover, there is the added inconvenience of having to code the spatial discretization represented by $\tilde{L}(\cdot)$. These reasons provide the incentive for us to want to avoid negative β_{ik} as much as possible when searching for the most efficient SSPRK methods.

We also note that the quantity

$$\min_{i,k} \frac{\alpha_{ik}}{|\beta_{ik}|} \tag{2.2}$$

obviously depends on the particular representation (2.1) of a given Runge-Kutta scheme. Accordingly, the CFL restriction is determined by the choice of coefficients α_{ij}, β_{ij} that maximizes (2.2); other choices render bounds that are not as sharp.

We will be comparing a new class of SSPRK methods with s stages and order p with $s > p$ to methods known in the literature where $s = p$ or some $\beta_{ik} < 0$. We note that if a method requires n_- extra evaluations of $\tilde{L}(\cdot)$ then the effective number of stages of that method is $m = s + n_-$. We find that the new SSPRK methods can have a significantly greater CFL coefficient (as given in Theorem 2.2) than the methods currently used in practice. However, we must make a fair comparison as to the computational cost of a step. This motivates the following definition:

DEFINITION 2.4. *The* effective CFL coefficient *of an SSPRK method of order p is cs^*/s where c is the CFL coefficient of the method, s^* is the minimum number of stages to theoretically achieve order p, and s is the number of stages required for one step of the method.*

It is well known (see e.g., [8]) that a Runge-Kutta method having s stages can achieve order p for $s = p \leq 4$. For $p > 4$, it is required that $s > p$. Because the cases we consider in this paper involve only $p \leq 4$, we always take $s^* = p$ here.

As conjectured in Shu and Osher [18] and subsequently proven in Gottlieb and Shu [6], the optimal two-stage, order-two SSPRK scheme is the modified Euler scheme,

$$\begin{aligned} U^{(1)} &= U^n + \Delta t L(U^n), \\ U^{n+1} &= \frac{1}{2}U^n + \frac{1}{2}U^{(1)} + \frac{1}{2}\Delta t L(U^{(1)}). \end{aligned}$$

It has a CFL restriction $\Delta t \leq \Delta t_{FE}$, which implies a CFL coefficient of 1. Henceforth, we will refer to this scheme as SSP(2,2). In general, we adopt the convention of referring to an s-stage, order-p SSPRK scheme as SSP(s,p).

Shu and Osher [18] also conjectured that the optimal three-stage, order-three SSPRK scheme is

$$\begin{aligned} U^{(1)} &= U^n + \Delta t L(U^n), \\ U^{(2)} &= \frac{3}{4}U^n + \frac{1}{4}U^{(1)} + \frac{1}{4}\Delta t L(U^{(1)}), \\ U^{n+1} &= \frac{1}{3}U^n + \frac{2}{3}U^{(2)} + \frac{2}{3}\Delta t L(U^{(2)}), \end{aligned}$$

which has a CFL coefficient of 1 as well. The optimality of this scheme was later proved by Gottlieb and Shu [6]. This scheme is commonly called the *Third-Order TVD Runge-Kutta scheme*, but we will simply refer to it as SSP(3,3).

To achieve fourth order, Shu and Osher provide a four-stage method that contains two negative coefficients β_{ik} [18]. A slightly improved scheme (but also containing two negative coefficients β_{ik}) was proposed by Gottlieb and Shu [6]

$$
\begin{aligned}
U^{(1)} &= U^n + \frac{1}{2}\Delta t L(U^n),\\
U^{(2)} &= \frac{649}{1600}U^n - \frac{10890423}{25193600}\Delta t\tilde{L}(U^n) + \frac{951}{1600}U^{(1)} + \frac{5000}{7873}\Delta t L(U^{(1)}),\\
U^{(3)} &= \frac{53989}{2500000}U^n - \frac{102261}{5000000}\Delta t\tilde{L}(U^n) + \frac{4806213}{20000000}U^{(1)}\\
&\quad - \frac{5121}{20000}\Delta t\tilde{L}(U^{(1)}) + \frac{23619}{32000}U^{(2)} + \frac{7873}{10000}\Delta t L(U^{(2)}),\\
U^{n+1} &= \frac{1}{5}U^n + \frac{1}{10}\Delta t L(U^n) + \frac{6127}{30000}U^{(1)} + \frac{1}{6}\Delta t L(U^{(1)})\\
&\quad + \frac{7873}{30000}U^{(2)} + \frac{1}{3}U^{(3)} + \frac{1}{6}\Delta t L(U^{(3)}).
\end{aligned}
$$

This scheme has a CFL coefficient of 0.936 and an effective CFL coefficient of $0.936 \times 4/6 = 0.624$ because 6 function evaluations are required per step. Because this seems to be the best four-stage, order-four SSPRK scheme known, we will refer to it as SSP(4^{**},4), with the two asterisks meant to convey two negative coefficients β_{ik}. Gottlieb and Shu [6] subsequently proved that no four-stage, order-four SSPRK scheme exists with positive coefficients.

Gottlieb and Shu [6] have also carried out an investigation of SSP time discretization methods for generalized Runge-Kutta methods (also known as pseudo-Runge-Kutta methods or hybrid methods [8][3]). They report that they were unable to find effective SSP methods in this wider class of methods. It is from this point that we start our derivations of improved SSPRK schemes where generally $s > p$. The details of these derivations are provided in the next section.

3. Optimal SSP Schemes. We now turn to the task of finding optimal SSPRK schemes. To begin, we seek to optimize an s-stage, order-p SSPRK scheme by maximizing its CFL coefficient according to Theorem 2.2. That is, we seek the global maximum of the nonlinear programming problem

$$\max_{(\alpha_{ik},\beta_{ik})} \min \frac{\alpha_{ik}}{\beta_{ik}}, \tag{3.1}$$

where $\alpha_{ik},\ \beta_{ik}, k = 0, 1, \ldots, i-1,\ i = 1, 2, \ldots, s$ are real and non-negative. The case $\alpha_{ik} = \beta_{ik} = 0$ is defined as `NaN` in the sense that it is not included in the minimization process if it occurs. Besides the non-negativity constraints on the variables $\alpha_{ik},\ \beta_{ik}$, the objective function (3.1) is subject to the constraints

$$\sum_{k=0}^{i-1} \alpha_{ik} = 1, \qquad i = 1, 2, \ldots, s, \tag{3.2}$$

$$\sum_{j=1}^{s} b_j \Phi_j(t) = \frac{1}{\gamma(t)}, \quad t \in T_q, \quad q = 1, 2, \ldots, p. \tag{3.3}$$

Here, the functions $\Phi_j(t)$ are nonlinear constraints that are polynomial in $\alpha_{ik},\ \beta_{ik}$ and that correspond to the order conditions for a Runge-Kutta method to be of order

[3] All of these methods are also special cases of methods known as *general linear methods*.

p (see e.g., [8]); i.e., T_q stands for the set of all rooted trees of order equal to q. The number of constraints represented by the Runge-Kutta order conditions is equal to

$$\sum_{q=1}^{p} \mathrm{card}(T_q),$$

where $\mathrm{card}(T_q)$ is the cardinality of T_q. Also, we use the notation b_j in the usual sense of the Butcher array representation of a Runge-Kutta method; again this would be a polynomial function of the coefficients α_{ik} and β_{ik}. It can be expected that the particular choice of coefficients α_{ik}, β_{ik} that maximizes the quantity (2.2) for a given Runge-Kutta method will be naturally produced by the solution to this nonlinear programming problem; hence the result will be a sharp estimate of the CFL coefficient.

In this form, the optimization problem does not lend itself easily to numerical solution. The difficulty due to the high degree of nonlinearity in the constraints is compounded by the following two considerations. First, the objective function (3.1) is non-smooth and so an optimization strategy that uses gradient information will have difficulty obtaining reliable numerical estimates of the derivatives. Second, the $\min(\cdot)$ function can be quite insensitive to its arguments. This also contributes to the poor performance of optimization software on this problem. We found that even optimizers that do not rely on gradient information were unable to consistently converge to the same optimum with this formulation.

The performance of optimization software on this problem is greatly enhanced through the following standard reformulation. By introducing a dummy variable z, the nonlinear programming problem can be reformulated as

$$\max_{(\alpha_{ik},\beta_{ik})} z, \tag{3.4a}$$

subject to

$$\alpha_{ik} \geq 0, \tag{3.4b}$$

$$\beta_{ik} \geq 0, \tag{3.4c}$$

$$\sum_{k=0}^{i-1} \alpha_{ik} = 1, \qquad i = 1, 2, \ldots, s, \tag{3.4d}$$

$$\sum_{j=1}^{s} b_j \Phi_j(t) = \frac{1}{\gamma(t)}, \quad t \in T_q, \quad q = 1, 2, \ldots, p, \tag{3.4e}$$

$$\alpha_{ik} - z\beta_{ik} \leq 0, \quad k = 0, 1, \ldots, i-1, \; i = 1, 2, \ldots, s. \tag{3.4f}$$

It is easy to see that the dummy variable z corresponds to the CFL coefficient. This reformulation is a standard technique that is widely used in the context of linear programming problems with objective functions of the form $\max(\cdot)$ or $\min(\cdot)$ (see e.g., [2]). It is also a common reformulation of the so-called *feasibility problem*, where any feasible solution to a set of equality or inequality constraints is desired (e.g., as in the first phase of a two-phase simplex algorithm for linear programming [3]).

The reformulated problem (3.4) was solved directly using the `fmincon` function from Matlab's Optimization Toolbox for $s = 1, 2, 3, 4, 5$ and $p = 1, 2, 3, 4$ and the results shown below. Table 3.1 shows the optimal values for the CFL coefficients for given pairs (s, p). The * in the $(4, 4)$ position denotes the fact that no such SSPRK method exists with all coefficients α_{ik}, β_{ik} positive.

	$s=1$	$s=2$	$s=3$	$s=4$	$s=5$
$p=1$	1	2	3	4	5
$p=2$		1	2	3	4
$p=3$			1	2	2.65
$p=4$				*	1.51

TABLE 3.1
Optimal CFL coefficients for s-stage, order-p SSPRK methods.

Table 3.2 gives the theoretical efficiencies of these new schemes relative to the ones where $s = p$. We note that there is no efficiency gain for the first-order methods. For example, although the CFL coefficient of the (2,1) method is twice that of that (1,1) method (forward Euler), it is also twice as much work. The percentages quoted refer to the theoretical increases in allowable step size of the new methods relative to the methods with $s = p$. For example, the (3,2) method has twice the allowable step size compared to the (2,2) method (the modified Euler method), but it requires 3/2 times more work. We thus report that the net effect is a relative increase in step size of $((2/1)/(3/2)-1) \times 100\% = 33\%$. Equivalently, assuming the CFL coefficient to be the exact bound on the time step, the new $(3,2)$ scheme can produce a comparable second-order-accurate answer with only 75% of the computational effort as the $(2,2)$ scheme.

	$s=2$	$s=3$	$s=4$	$s=5$
$p=2$		33%	50%	60%
$p=3$			50%	59%
$p=4$				94%

TABLE 3.2
Theoretical efficiency improvement over standard pth order SSPRK schemes.

We draw particular attention to the efficiency of the $(5,4)$ scheme in Table 3.2. As mentioned earlier, a $(4,4)$ SSPRK scheme does not exist for any positive CFL coefficient. The figure of 94% is measured relative to the $(4^{**},4)$ scheme reported in [6] as the best scheme of order 4 that could be found. Recall that this scheme had a CFL coefficient of 0.936 and effectively used 6 stages because it involved two coefficients β_{ik} that are negative (hence leading to a 50% increase of the storage requirement per step and the overhead of coding $\tilde{L}(\cdot)$). The new $(5,4)$ scheme thus compares very favourably.

The first few optimal SSPRK schemes of orders 1 and 2 are given in Tables 3.3 and 3.4. Here we give the schemes in terms of the coefficients α_{ik}, β_{ik}; the Butcher form of these schemes in given in Appendix B. It is interesting to note that Gerisch and Weiner have independently proposed the SSP(3,2) scheme; see [4] for details.

From Tables 3.1, 3.3, and 3.4, we can conjecture the form of the optimal SSPRK methods with s stages and orders 1 and 2; namely, the optimal SSPRK method with s stages and order 1 has CFL coefficient s; and the optimal SSPRK method with s stages and order 2 has CFL coefficient $s-1$. Shu [17] has given a proof of the first-order result and Gottlieb and Shu [6] have given a proof of the second-order result for $s = 2$. We provide a new proof of the first-order result below as well as a

Stages	α_{ik}			β_{ik}			CFL coefficient
1	1			1			1
2	1			$\frac{1}{2}$			2
	0	1		0	$\frac{1}{2}$		
3	1			$\frac{1}{3}$			3
	0	1		0	$\frac{1}{3}$		
	0	0	1	0	0	$\frac{1}{3}$	

TABLE 3.3
The first few optimal SSPRK schemes of order 1.

Stages	α_{ik}				β_{ik}				CFL coefficient
2	1				1				1
	$\frac{1}{2}$	$\frac{1}{2}$			0	$\frac{1}{2}$			
3	1				$\frac{1}{2}$				2
	0	1			0	$\frac{1}{2}$			
	$\frac{1}{3}$	0	$\frac{2}{3}$		0	0	$\frac{1}{3}$		
4	1				$\frac{1}{3}$				3
	0	1			0	$\frac{1}{3}$			
	0	0	1		0	0	$\frac{1}{3}$		
	$\frac{1}{4}$	0	0	$\frac{3}{4}$	0	0	0	$\frac{1}{4}$	

TABLE 3.4
The first few optimal SSPRK schemes of order 2.

proof of the second-order result for arbitrary s. These low-order methods with large CFL coefficients are useful when seeking a time-independent (steady-state) solution of (1.1), given that in such problems the accuracy considerations in time are typically less critical than those in space [17].

THEOREM 3.1. *For $s = 1, 2, 3, \ldots$, the optimal s-stage SSPRK method of order 1 with $\beta_{ik} \geq 0$ has CFL coefficient s and can be represented in the form*

$$\alpha_{ik} = \begin{cases} 1 & k = i-1, \\ 0 & \textit{otherwise}. \end{cases}, \qquad \beta_{ik} = \begin{cases} \frac{1}{s} & k = i-1, \\ 0 & \textit{otherwise}. \end{cases}, \qquad i = 1, 2, \ldots, s.$$

Before giving the proof of Theorem 3.1, we introduce the following notation and give a useful Lemma. We find it convenient to write the general s-stage explicit Runge-Kutta method in the following form (cf. [6]):

$$U^{(0)} = U^n, \tag{3.5a}$$

$$U^{(i)} = U^{(0)} + \Delta t \sum_{k=0}^{i-1} c_{ik} L(U^{(k)}), \quad i = 1, 2, \ldots, s, \tag{3.5b}$$

$$U^{n+1} = U^{(s)}. \tag{3.5c}$$

The coefficients c_{ik} are related to the coefficients $\alpha_{ik},\ \beta_{ik}$ recursively by

$$c_{ik} = \sum_{j=k+1}^{i-1} \alpha_{ij} c_{jk} + \beta_{ik}. \tag{3.6}$$

It is also easy to see that the coefficients c_{ik} are related to the Butcher array quantities $a_{ik},\ b_k$ by

$$\begin{aligned} a_{ik} &= c_{i-1,k-1}, \qquad && k = 1,2,\ldots,i-1, \quad i = 1,2,\ldots,s-1, \\ b_k &= c_{s,k-1}, && k = 1,2,\ldots,s. \end{aligned}$$

LEMMA 3.2. *If a method of the form (2.1) with $\alpha_{ik},\ \beta_{ik} \geq 0$ has a CFL coefficient $c > m > 0$, then $0 \leq c_{ik} < \frac{1}{m}$ for all $k = 0,1,\ldots,i-1$, $i = 1,2,\ldots,s$.*

Proof. From Theorem 2.2, if $c > m > 0$, then $\alpha_{ik} > m\beta_{ik}$, or equivalently $\beta_{ik} < \frac{1}{m}\,\alpha_{ik}$, for all $i,\ k$ such that $\alpha_{ik} \neq 0$.

Now,

$$\alpha_{ik} \geq 0, \quad \sum_{k=0}^{i-1} \alpha_{ik} = 1, \quad i = 1,2,\ldots,s, \quad \Rightarrow \quad \alpha_{ik} \leq 1$$

for all $i,\ k$. Hence, $\beta_{ik} < \frac{1}{m}$ for all $i,\ k$. In particular, $c_{10} = \beta_{10} < \frac{1}{m}$ for any valid SSPRK method.

We now proceed by induction on stage ℓ of an s-stage method. Assume $c_{ij} < \frac{1}{m}$ for $j = 0,1,\ldots,\ell-1$; $i = 1,2,\ldots,\ell$. (We have just shown that this result holds for $\ell = 1$.) Now consider stage $(\ell+1)$ of a valid SSPRK method; i.e., consider coefficients $c_{\ell+1,k}$ for $k = 0,1,\ldots,\ell$ with

$$\sum_{k=0}^{\ell} \alpha_{\ell+1,k} = 1.$$

Then using (3.6),

$$\begin{aligned} c_{\ell+1,0} &= \sum_{k=1}^{\ell} \alpha_{\ell+1,k} c_{k0} + \beta_{\ell+1,0} \\ &< \frac{1}{m} \sum_{k=1}^{\ell} \alpha_{\ell+1,k} + \frac{1}{m}\,\alpha_{\ell+1,0} \\ &= \frac{1}{m}\,. \end{aligned}$$

Similar arguments can be used to show $c_{\ell+1,j} < \frac{1}{m}$ for $j = 1,2,\ldots,\ell$. The Lemma now follows by induction. ∎

Proof of Theorem 3.1. By contradiction, suppose there exists an s-stage, order-1 SSPRK method with CFL coefficient $c > s$. Because the method is order 1, we have

$$\sum_{k=0}^{s-1} c_{sk} = 1. \tag{3.7}$$

But from Lemma 3.2, we have

$$c_{ik} < \frac{1}{s}, \qquad k = 0, 1, \ldots, i-1, \quad i = 1, 2, \ldots, s.$$

Thus,

$$\sum_{k=0}^{s-1} c_{sk} < \sum_{k=0}^{s-1} \frac{1}{s} = 1,$$

contradicting (3.7). Thus, no s-stage, order-1 SSPRK method can exist with CFL coefficient $c > s$. Because the SSPRK methods proposed in Theorem 3.1 have $c = s$, they must be optimal representations. ∎

THEOREM 3.3. *For $s = 2, 3, 4, \ldots$, the optimal s-stage SSPRK method of order 2 with $\beta_{ik} \geq 0$ has CFL coefficient $s-1$ and can be represented in the form*

$$\alpha_{ik} = \begin{cases} 1 & k = i-1, \\ 0 & \textit{otherwise}. \end{cases}, \quad \beta_{ik} = \begin{cases} \frac{1}{s-1} & k = i-1, \\ 0 & \textit{otherwise}. \end{cases}, \quad i = 1, 2, \ldots, s-1.$$

$$\alpha_{ik} = \begin{cases} \frac{1}{s} & k = 0, \\ \frac{s-1}{s} & k = s-1 \\ 0 & \textit{otherwise}. \end{cases}, \qquad \beta_{ik} = \begin{cases} \frac{1}{s} & k = s-1, \\ 0 & \textit{otherwise}. \end{cases}, \qquad i = s.$$

Proof. By contradiction, suppose there exists an s-stage, order-2 SSPRK method with CFL coefficient $c > s-1$. Because it is order 2, the coefficients of the method must satisfy (3.7) and

$$\sum_{i=1}^{s-1} c_{si} \sum_{k=0}^{i-1} c_{ik} = \frac{1}{2}. \tag{3.8}$$

Also, using Lemma 3.2 with $c > s-1$ implies that

$$c_{ik} < \frac{1}{s-1}, \qquad k = 0, 1, \ldots, i-1, \quad i = 1, 2, \ldots, s. \tag{3.9}$$

Using (3.9) in (3.8) for $k = 0, 1, \ldots, i-1,\ i = 1, 2, \ldots, s-1$ leads to

$$\sum_{i=1}^{s-1} \frac{i}{s-1} c_{si} > \frac{1}{2},$$

and using this result in (3.7) yields

$$\sum_{k=0}^{s-2} \frac{s-k-1}{s-1} c_{sk} < \frac{1}{2}.$$

Thus,

$$\begin{aligned} \frac{1}{2} &> \sum_{k=0}^{s-2} \frac{s-k-1}{s-1} c_{sk} \\ &= \sum_{k=0}^{s-2} \frac{s-k-1}{s-1} \left(\sum_{j=k+1}^{s-1} \alpha_{sj} c_{jk} + \beta_{sk} \right). \end{aligned}$$

Now we substitute recursively for c_{jk} using (3.6) in the right-hand side of the above equation and (3.8), and recalling that $\alpha_{ik} > (s-1)\beta_{ik}$ and $\alpha_{ik} \geq 0$ for $k = 0, 1, \ldots, i-1$, $i = 1, 2, \ldots, s$, we can use (3.8) to write

$$\frac{1}{2} > \frac{1}{2} + \sum_{j=1}^{s-2} j \sum_{l=s-j-1}^{s-1} \beta_{sl}\beta_{l,s-2-j} + \sum_{j=0}^{s-2} \frac{s-j-1}{s-1}\beta_{sj}.$$

This now contradicts the fact that $\beta_{ik} \geq 0$ for all $k = 0, 1, \ldots, i-1$, $i = 1, 2, \ldots, s$. Thus, no s-stage, order-2 SSPRK method can have CFL coefficient $c > s-1$. The proof is now completed by noting that because the schemes proposed have $c = s-1$, they must be optimal representations. ■

In Tables A.1–A.2 in Appendix A, we give results for the coefficients of the optimal schemes of order $p = 3,\ 4$ in terms of their numerical values up to double precision.

A proof of optimality for the SSP(4,3) scheme follows easily from a result in [16]. In [16] it is proven that the optimal CFL coefficient of an s-stage SSPRK method of order p applied to a linear, constant-coefficient problem $\dot{U} = LU$ is $s - p + 1$. Thus if a nonlinear scheme can attain this optimal bound, then it must also be optimal. It is easy to see that SSP(4,3) is such a scheme.

We do not offer formal proofs of optimality in the remaining cases; however, these are the results of extensive numerical searches.

Finally, we describe our results for optimal low-storage SSPRK schemes. There are computational problems for which memory management considerations are at least as important as stability considerations when choosing a numerical time discretization method, e.g., direct numerical simulation of Navier-Stokes equations requiring high spatial resolution in three dimensions. In such cases, s-stage explicit Runge-Kutta methods that use less than the usual s units of storage are very desirable (see e.g., [20]). We focus our discussion on SSPRK schemes that require only two units of storage per step[4], although more general methods requiring more storage per step are possible. These schemes take the form

$$dU^{(i)} = A_i dU^{(i-1)} + \Delta t L(U^{(i-1)}), \tag{3.10a}$$

$$U^{(i)} = U^{(i-1)} + B_i dU^{(i-1)}, \qquad i = 1, 2, \ldots, s, \tag{3.10b}$$

where $U^{(0)} = U^n$, $U^{n+1} = U^{(s)}$, and $A_1 \equiv 0$. Again, we note that there is a relation between the coefficients A_i, B_i and the coefficients α_{ik}, β_{ik} or equivalently the usual quantities in the Butcher array. We denote the general s-stage, order-p low-storage SSPRK scheme simply by LS(s,p).

We have solved the corresponding nonlinear programming problems to optimize the CFL coefficient for the low-storage schemes defined by (3.10). The results for the coefficients A_i, B_i are given in Tables 3.5–3.7 for up to 5 stages and order 3. Again, only numerical values of the coefficients are given to double precision. The Butcher array form of these schemes is given in Appendix C. Of course, a traditional implementation of any 2-stage scheme must be low-storage in the sense we are considering, so the optimal low-storage method with $s = p = 2$ corresponds to the optimal SSPRK scheme in Table 3.4. We note that the optimal 3-stage, order-3 low-storage method reported in Table 3.7 agrees with that reported in [6]. We also note that we were not successful in finding a 5-stage, order-4 scheme in this family, and we strongly suspect that such a method does not exist.

[4]We note that if some form of error control is envisaged, perhaps using an embedded [8] SSPRK scheme, then additional storage for the current solution vector is also required.

Stages	A_i	B_i	CFL coefficient
1	0	1	1
2	0	0.25471543653218	1
	0.66323286721269	0.44809394647120	
3	0	0.26237801705341	1
	0.42645094785793	0.20169056000013	
	0.45339958582027	0.27321697994061	
4	0	0.14142439246204	
	0.42623204099143	0.35397016495696	1
	0.38851833123083	0	
	0.01694135866933	0.34465757966021	
5	0	0.03368800719745	1
	0.61573074220688	0.13960527476637	
	0.24191712486786	0.22864919232774	
	0.16549924932085	0.26079330982391	
	-0.04239297405834	0.10750824432183	

TABLE 3.5
The coefficients of the first few optimal low-storage schemes of order 1.

4. Numerical Studies. In this section, we study the numerical behaviour of our schemes and Shu-Osher SSP-schemes for a few test problems designed to capture solution features that pose particular difficulties to numerical methods. Experiments for the classical fourth-order explicit Runge-Kutta method are also included because this method is commonly used in method-of-lines discretizations of hyperbolic conservation laws but is not SSP.

4.1. Test Problems. There are a variety of solution features in computational fluid dynamics that commonly cause numerical problems. For example, many numerical methods produce significant errors near sonic points (points where the wavespeed equals zero). Upwind methods in particular are forced to give sonic points special consideration since the upwind direction changes at sonic points. Shock waves, contact discontinuities, and expansion fans may also lead to a variety of serious problems including oscillations, overshoots, and smearing that can spread discontinuities over several cells. In particular, contact discontinuities do not have any physical compression and thus smearing increases progressively with the number of time steps. Even when approximating smooth solutions, most numerical methods exhibit obvious flaws. For example, many stable numerical methods continuously erode the solution, leading to amplitude and dissipation errors [13].

To investigate the behavior of our time-stepping schemes, we consider three of Laney's five test problems [13]. These three problems involve all of the important flow features identified above: shocks, contacts, expansion fans, sonic points, and smooth solutions. Similar to Laney, we focus on the behaviour of the numerical scheme for interior regions rather than boundaries and impose periodic boundary conditions on the domain $[-1, 1]$. It is known that sometimes a conventional (and intuitive!) treatment of the boundary data (especially in the case of inflow boundary

Stages	A_i	B_i	CFL coefficient
2	0	1	1
	-1	$\frac{1}{2}$	
3	0	0.79609964254616	1
	-0.86514937424574	0.47921739051941	
	-0.01459406292961	0.13955204452449	
4	0	0.08820909208788	
	0.34143758512319	0.62773790223092	1
	-0.80189834090053	0.43908735985479	
	-0.26868602239001	0.10090483677631	
5	0	0.24064789292000	1
	-0.35363900948812	0.28813102587031	
	0.23144682054640	0.15490366543216	
	0.30287923513739	0.33623843526263	
	-0.90122396243589	0.27101878032131	

TABLE 3.6
The coefficients of the first few optimal low-storage schemes of order 2.

Stages	A_i	B_i	CFL coefficient
3	0	0.92457411523577	0.32234930738853
	-2.91549398859489	0.28771294148749	
	0.00000000151682	0.62653829645172	
4	0	1.03216665875130	0.52841816101829
	-4.94661981618529	0.18793881263711	
	0.00000000050902	0.15215751854315	
	-0.15127914578976	0.65675174856653	
5	0	0.67892607116139	1
	-2.60810978953486	0.20654657933371	
	-0.08977353434746	0.27959340290485	
	-0.60081019321053	0.31738259840613	
	-0.72939715170280	0.30319904778284	

TABLE 3.7
The coefficients of the first few optimal low-storage schemes of order 3.

conditions) within the stages of a Runge-Kutta method can lead to a deterioration in the overall accuracy of the integration. We refer to [1] and references therein for a discussion of this problem and a method for its resolution. The spatial discretization and the results of three test cases follow.

4.2. Spatial Discretization. SSPRK schemes are natural candidates for any method-of-lines discretization involving nonsmooth solutions. Similar to the original paper on SSPRK methods [18], we choose finite-difference Shu-Osher methods (ENO) to spatially discretize the equations. These methods are derived using flux reconstruction and have a variety of desirable properties. For example, they naturally extend to an arbitrary order of accuracy in space, and they are independent of the time discretization, thus allowing experimentation with different time discretization methods. Moreover, educational codes are also freely available [13, 12], an attribute which is desirable for standardizing numerical studies. Our simulations are carried out with a discretization that has the same order of accuracy in Δx as the time discretization accuracy p. We further note that flux splitting is carried out according to

$$f^+(U) = \frac{1}{2}(f(U) + \alpha^n_{i+1/2}U),$$
$$f^-(U) = \frac{1}{2}(f(U) - \alpha^n_{i+1/2}U)$$

where $\alpha^n_{i+1/2} = \max\{|f'(U^n_{i+1})|, |f'(U^n_i)|\}$. For full details on the discretization as well as code, see [13, 12].

It is noteworthy that high-order, fully TVD spatial discretization schemes are also available; see Osher and Chakravarthy [15]. In these numerical studies, we choose Shu-Osher spatial discretization schemes rather than TVD schemes since TVD schemes only obtain between first- and second-order accuracy at extrema and they have "been largely superseded by Shu and Osher's class of high-order ENO methods" [13].

It is also noteworthy that recent variations on Shu-Osher methods such as methods based on WENO reconstructions (e.g., [14, 11]) also naturally combine with SSPRK schemes. See [13] for detailed discussions on these and other spatial discretizations appropriate for hyperbolic conservation laws.

4.3. Test Case 1: Linear advection of a sinusoid. In this test case, the smooth initial conditions

$$u(x,0) = -\sin(\pi x)$$

are evolved to time $t = 30$ according to the linear advection equation

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0$$

using a constant grid spacing of $\Delta x = 1/320$. Since this evolution causes the initial conditions to travel around the periodic domain $[-1, 1]$ exactly 15 times, it is clear that the exact solution is just $u(x, 30) = -\sin(\pi x)$. Test Case 1 effectively illustrates the evolution of a smooth solution with no sonic points and is useful for verifying convergence rates for high-order schemes. Moreover, even on completely smooth solutions most numerical methods designed for hyperbolic conservation laws exhibit obvious flaws [13]. This test case is quite helpful for understanding phase and amplitude errors but should not be used to study dispersion because only one frequency is present in the exact solution. It is also informative to contrast these results with those derived for problems involving shocks and other discontinuities.

To quantify the accuracy of the computed solution, we use the logarithm of the l_1 errors, i.e.,

$$\log_{10}\left(\frac{1}{N}\sum_{i=1}^{N}|U_i - u(x_i, 30)|\right),$$

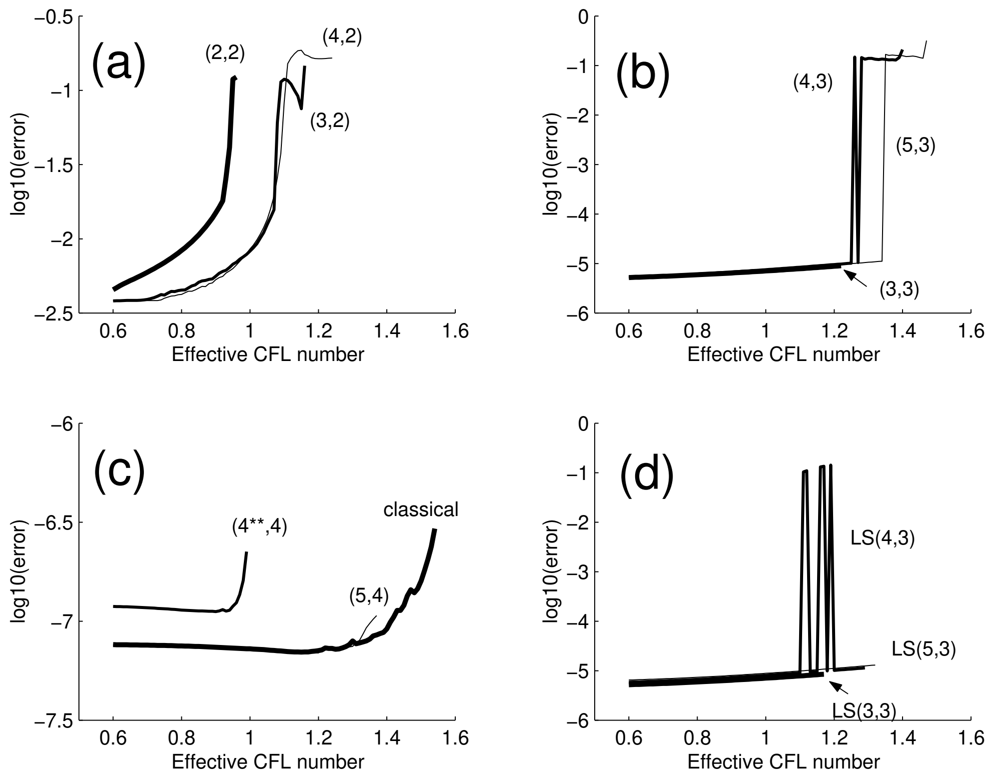

FIG. 4.1. l_1 *errors as a function of the effective CFL number. (a) Second-order schemes (b) Third-order schemes (c) Fourth-order schemes (d) Low-storage schemes.*

where N is the number of grid points and x_i is the i^{th} grid node. A plot of the error is given in Figure 4.1. To ensure a fair comparison for methods with a different number of stages, the error is plotted as a function of the effective CFL number[5] rather than the CFL number itself. This implies that for a particular plot, the total number of function evaluations at a particular abscissa value will be the same for each scheme. We start calculating errors for an effective CFL number of 0.6 and continue until the numerical method is so unstable that a value of `NaN` is returned; i.e., the scheme has become completely unstable.

In this smooth test example, the new second-order schemes give improved stability and accuracy over the original SSP(2,2). Also, SSP(5,3) gives improved stability over SSP(3,3) and SSP(4,3). Calculations for low-storage schemes show that LS(5,3) outperforms both LS(4,3) and LS(3,3). (For clarity, we use arrows to indicate the

[5]Similar to the definition of effective CFL coefficient, the *effective CFL number* of an SSPRK method of order p is $\frac{\Delta t}{\Delta x}\frac{s^*}{s}$ where s^* is the minimum number of stages to theoretically achieve order p, and s is the number of stages required for one step of the method.

exact points at which SSP(3,3) and LS(3,3) go completely unstable.)

Based on these plots, we see that the second-order, third-order, and low-storage schemes all give stability restrictions that are within 20% of one another. This contrasts sharply with the results for fourth-order schemes (plot (c)). Here the new SSP(5,4) scheme gives more than a 40% improvement in the stability time-step restriction over the original SSP(4**,4). Moreover, it produces a marked reduction in the error, signifying a smaller error constant for this problem. It is noteworthy that in this case the classical fourth-order Runge-Kutta scheme outperforms even SSP(5,4): On *smooth* problems, schemes based purely on a linear stability analysis are expected to perform well. SSP schemes are designed to outperform on problems involving discontinuities in the solution or its derivatives, so in this case there is no reason to expect that schemes derived using nonlinear stability analysis will necessarily outperform classical schemes based on linear stability analysis.

4.4. Test Case 2: Linear advection of a square wave. In this test case, the discontinuous initial conditions

$$u(x,0) = \begin{cases} 1 & \text{for } |x| < 1/3, \\ 0 & \text{for } 1/3 < |x| \le 1, \end{cases}$$

are evolved to time $t = 4$ according to the linear advection equation

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0$$

using a constant grid spacing of $\Delta x = 1/320$. Since this evolution causes the initial conditions to travel around the periodic domain $[-1,1]$ exactly 2 times, it is clear that the exact solution at the final time is just $u(x,4) = u(x,0)$. Test Case 2 exhibits two jump discontinuities in the solution that correspond to contact discontinuities. This test case nicely illustrates progressive contact smearing and dispersion.

The log of the l_1 errors as a function of the effective CFL number are plotted in Figure 4.2. Based on these plots, it is immediately clear that a material improvement in both stability and accuracy are obtained using our new schemes.

For example, plot (a) shows that SSP(3,2) and SSP(4,2) allow about a $20-30\%$ improvement in the time-step restriction over the original SSP(2,2). It is also clear that the new schemes also give a substantial improvement in stability and accuracy in the third-order case (b). Here we find that the optimal SSP(5,3) scheme gives about a 40% improvement in the stability time-step restriction over the usual SSP(3,3).

In the fourth-order case (c), even greater improvements are observed. SSP(5,4) gives more than a 60% improvement in the stability time-step restriction and only requires half the number of function evaluations to achieve an error of $10^{-1.5}$. Moreover, SSP(5,4) is clearly superior to the classical fourth-order Runge-Kutta scheme, with more than a 40% improvement in the observed time-step restriction. As conjectured, the best SSP-schemes outperform classical (but non-SSP) schemes when discontinuities in the solution arise.

Out of the low-storage schemes (plot (d)), the new LS(4,3) gives the best performance. It is interesting that the best-performing scheme LS(4,3) requires one-third less storage and is more CPU-efficient than the standard SSP(3,3) in this test example.

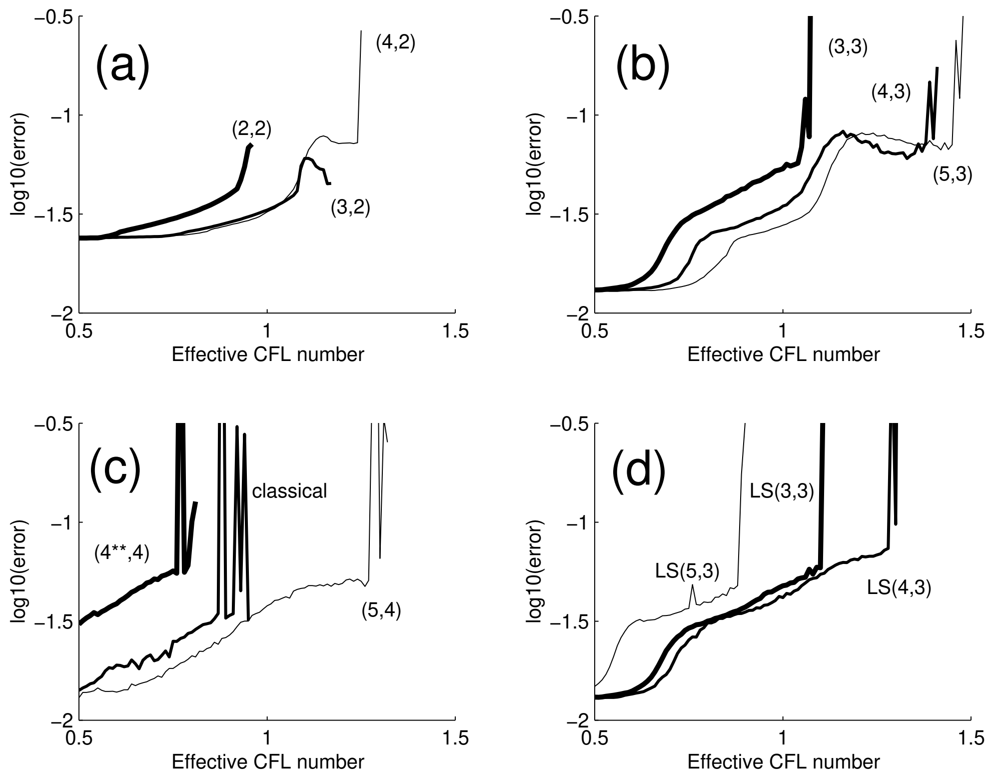

FIG. 4.2. l_1 *errors as a function of the effective CFL number. (a) Second-order schemes (b) Third-order schemes (c) Fourth-order schemes (d) Low-storage schemes.*

4.5. Test Case 3: Evolution of a square wave by Burgers' equation. In this test case, the discontinuous initial conditions

$$u(x,0) = \begin{cases} 1 & \text{for } |x| < 1/3, \\ -1 & \text{for } 1/3 < |x| \le 1, \end{cases}$$

are evolved to time $t = 0.3$ according to Burgers' equation

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(\frac{1}{2}u^2\right) = 0$$

using a constant grid spacing of $\Delta x = 1/320$. In this example, the jump at $x = -1/3$ creates a simple centered expansion fan and the jump at $x = 1/3$ creates a steady shock. Until the shock and expansion fan intersect (at time $t = 2/3$), the exact solution is

$$u(x,t) = \begin{cases} -1 & \text{for } -\infty < x < b_1, \\ -1 + 2\frac{x-b_1}{b_2-b_1} & \text{for } b_1 < x < b_2, \\ 1 & \text{for } b_2 < x < b_{shock}, \\ -1 & \text{for } b_{shock} < x < \infty, \end{cases}$$

where $b_1 = -1/3 - t$, $b_2 = -1/3 + t$ and $b_{shock} = 1/3$ [13]. Test Case 3 is particularly interesting because it illustrates the behaviors near sonic points ($u = 0$) that correspond to an expansion fan and a compressive shock.

The log of the l_1 errors as a function of the effective CFL number are plotted in Figure 4.3. Based on these plots, it is clear that a marked improvement in both stability and accuracy are obtained in the second-, third-, and fourth-order cases using our new schemes.

Once again, plot (a) shows that SSP(3,2) and SSP(4,2) show about a 20–30% improvement in the time-step restriction over the original SSP(2,2). It is also clear that the new schemes also give a substantial improvement in stability and accuracy in the third-order case (b). Here we find that the optimal SSP(5,3) scheme gives about a 20% improvement in the stability time-step restriction over the usual SSP(3,3).

In the fourth-order case (c), even greater improvements are observed than in Test Case 2. SSP(5,4) gives an 80% improvement in the stability time-step restriction and only requires one-third the number of function evaluations to achieve an error of $10^{-2.6}$. Moreover, SSP(5,4) is clearly superior to the classical fourth-order Runge-Kutta scheme, with more than a 60% improvement in the observed time-step restriction. Similar to the previous example, SSP(5,4) outperforms classical (but non-SSP) schemes when discontinuities in the solution arise.

Out of the low-storage schemes (plot (d)), LS(3,3) and the new LS(4,3) give the best performance. In this test case, the best low-storage schemes are nearly as CPU-efficient as SSP(3,3) but require one-third less storage.

5. Summary and Future Work. We have presented new optimal SSPRK time discretization methods of orders 1 through 4 and stages 1 through 5. We find that, by allowing the number of stages to differ from the order of the method, it is possible to derive schemes with better effective CFL coefficients than those that are most commonly used. We have also performed a comparison of the new methods with Runge-Kutta methods (both SSP and non-SSP) most commonly used in practice on three problems involving scalar conservation laws. Our new methods compare favourably in terms of computational efficiency per time step, especially when the solution exhibits discontinuous behaviour. The improvements are the greatest for the new fourth-order scheme with 5 stages (SSP(5,4)), where the allowable time step is significantly greater than the Shu-Osher fourth-order scheme and the classical fourth-order explicit Runge-Kutta scheme.

We also give results of a similar treatment of low-storage SSPRK schemes, where again we find significant improvements over the schemes most commonly used. The results are for orders 1 through 3 and stages 1 through 5. We were unable to find a low-storage scheme of order 4 having only 5 stages.

We have already examined the possibility of finding even more efficient SSPRK schemes by lifting the positivity constraint on the coefficients β_{ij}. Not surprisingly, improvements in the raw CFL coefficient are possible; however, the reduction in the

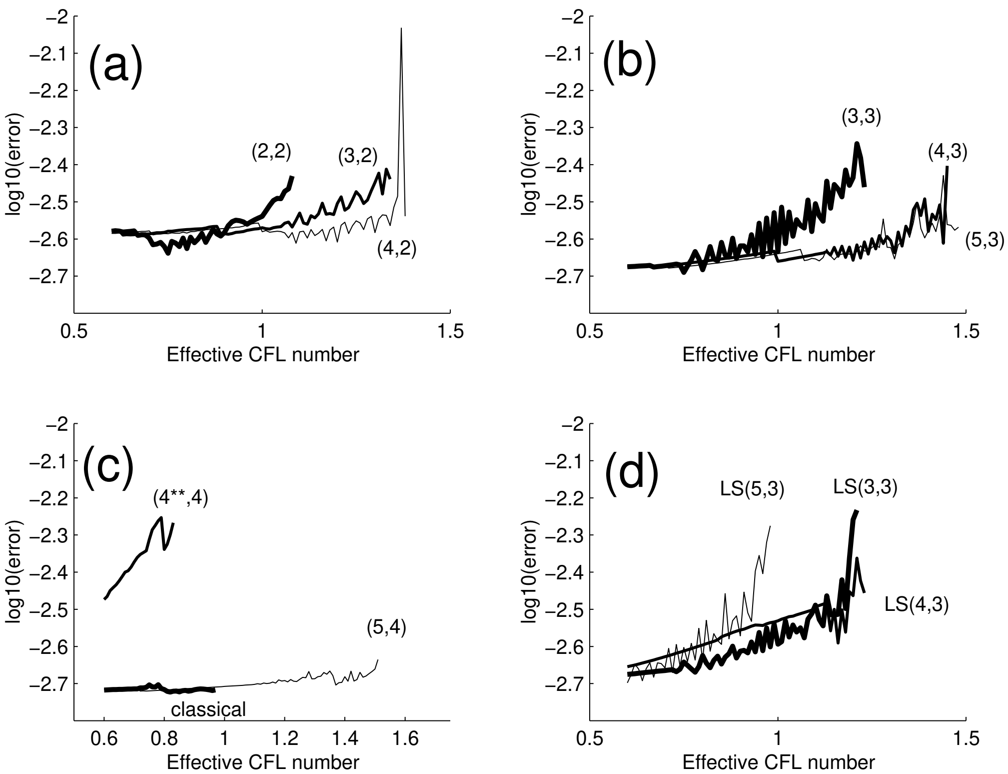

FIG. 4.3. *l_1 errors as a function of the effective CFL number. (a) Second-order schemes (b) Third-order schemes (c) Fourth-order schemes (d) Low-storage schemes.*

effective CFL coefficient necessitated by the introduction of $\tilde{L}(\cdot)$ whenever $\beta_{ij} < 0$ causes these methods to be uncompetitive.

We are currently extending our investigation of optimal SSPRK methods to methods having more than 5 stages and to orders 4 and 5. This work includes the study of low-storage SSPRK methods of order 4. We have also derived families of embedded SSPRK schemes for local error estimation and step-size control. We report on these findings elsewhere.

6. Acknowledgements. The authors would like to express their thanks to J. Borwein and W. Sutherland for helpful discussions.

Appendix A. Optimal $(\alpha_{ik},\ \beta_{ik})$ for $p = 3,\ 4$. Tables A.1–A.2 respectively give the optimal SSPRK methods of orders 3 and 4 and up to 5 stages in the representation (2.1).

Appendix B. Butcher array forms of SSPRK schemes. The following are the Butcher array representations of the optimal SSPRK schemes given in this paper.

TABLE A.1
The first few optimal SSPRK schemes of order 3.

Stages	α_{ik}				β_{ik}				CFL coefficient
3	1				1				1
	$\frac{3}{4}$	$\frac{1}{4}$			0	$\frac{1}{4}$			
	$\frac{1}{3}$	0	$\frac{2}{3}$		0	0	$\frac{2}{3}$		
4	1				$\frac{1}{2}$				2
	0	1			0	$\frac{1}{2}$			
	$\frac{2}{3}$	0	$\frac{1}{3}$		0	0	$\frac{1}{6}$		
	0	0	0	1	0	0	0	$\frac{1}{2}$	

Stages	5				
α_{ik}	1				
	0	1			
	0.56656131914033	0	0.43343868085967		
	0.09299483444413	0.00002090369620	0	0.90698426185967	
	0.00736132260920	0.20127980325145	0.00182955389682	0	0.78952932024253
β_{ik}	0.37726891511710				
	0	0.37726891511710			
	0	0	0.16352294089771		
	0.00071997378654	0	0	0.34217696850008	
	0.00277719819460	0.00001567934613	0	0	0.29786487010104
CFL coefficient			2.65062919294483		

Order 1:

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array} \qquad \begin{array}{c|cc} 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \qquad \begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ \frac{2}{3} & \frac{1}{3} & \frac{1}{3} & 0 \\ \hline & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{array}$$

TABLE A.2
The coefficients of the optimal SSPRK (5,4) scheme.

Stages	5				
	1				
	0.44437049406734	0.55562950593266			
α_{ik}	0.62010185138540	0	0.37989814861460		
	0.17807995410773	0	0	0.82192004589227	
	0.00683325884039	0	0.51723167208978	0.12759831133288	0.34833675773694
	0.39175222700392				
	0	0.36841059262959			
β_{ik}	0	0	0.25189177424738		
	0	0	0	0.54497475021237	
	0	0	0	0.08460416338212	0.22600748319395
CFL coefficient	1.50818004975927				

Order 2:

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \qquad \begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 1 & \frac{1}{2} & \frac{1}{2} & 0 \\ \hline & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{array} \qquad \begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 \\ \frac{2}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ 1 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \\ \hline & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{array}$$

Order 3:

$$
\begin{array}{c|ccc}
0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 \\
\frac{1}{2} & \frac{1}{4} & \frac{1}{4} & 0 \\
\hline
 & \frac{1}{6} & \frac{1}{6} & \frac{2}{3}
\end{array}
\qquad
\begin{array}{c|cccc}
0 & 0 & 0 & 0 & 0 \\
\frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\
1 & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\
\frac{1}{2} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & 0 \\
\hline
 & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{2}
\end{array}
$$

0	0	0	0	0	0
0.37726891511710	0.37726891511710	0	0	0	0
0.75453783023419	0.37726891511710	0.37726891511710	0	0	0
0.49056882269314	0.16352294089771	0.16352294089771	0.16352294089771	0	0
0.78784303014311	0.14904059394856	0.14831273384724	0.14831273384724	0.34217696850008	0
	0.19707596384481	0.11780316509765	0.11709725193772	0.27015874934251	0.29786487010104

Order 4:

0	0	0	0	0	0
0.39175222700392	0.39175222700392	0	0	0	0
0.58607968896779	0.21766909633821	0.36841059262959	0	0	0
0.47454236302687	0.08269208670950	0.13995850206999	0.25189177424738	0	0
0.93501063100924	0.06796628370320	0.11503469844438	0.20703489864929	0.54497475021237	0
	0.14681187618661	0.24848290924556	0.10425883036650	0.27443890091960	0.22600748319395

Appendix C. Butcher array forms of LS schemes. The optimal low-storage SSPRK schemes of order 1 and order 2 occur when $s = p$ and have already been given both in terms of representation (2.1) and Butcher arrays. Here we provide the Butcher array representation of the third-order schemes presented in Table 3.7.

Order 3:

0	0	0	0
0.92457411523577	0.92457411523577	0	0
0.37346170537554	0.08574876388805	0.28771294148749	0
	0.08574876111733	0.28771294243783	0.62653829645172

0	0	0	0	0
1.03216665875130	1.03216665875130	0	0	0
0.29044361656735	0.10250480393024	0.18793881263711	0	0
0.44260113480482	0.10250480354712	0.18793881271456	0.15215751854315	0
	0.10250480379728	0.18793881266399	0.05280467502407	0.65675174856653

0	0	0	0	0	0
0.67892607116139	0.67892607116139	0	0	0	0
0.34677649493991	0.14022991560621	0.20654657933371	0	0	0
0.66673359500982	0.20569370073026	0.18144649137471	0.27959340290485	0	0
0.76590087429032	0.16104646283838	0.19856511041100	0.08890670263481	0.31738259840613	0
	0.19215670424132	0.18663683901393	0.22177739201759	0.09623007655432	0.30319904778284

REFERENCES

[1] S. Abarbanel, D. Gottlieb, and M. H. Carpenter, *On the removal of boundary errors caused by Runge-Kutta integration of nonlinear partial differential equations*, SIAM J. Sci. Comput., 17 (1996), pp. 777–782.

[2] V. Chvátal, *Linear programming*, W. H. Freeman and Company, New York, 1983.

[3] R. Fletcher, *Practical methods of optimization*, John Wiley & Sons Ltd., Chichester, second ed., 1987.

[4] A. Gerisch and R. Weiner, *On the positivity of low order explicit Runge-Kutta schemes applied in splitting methods*, Computer and Mathematics with Applications, (to appear).
[5] J. B. Goodman and R. J. LeVeque, *On the accuracy of stable schemes for 2D scalar conservation laws*, Math. Comp., 45 (1985), pp. 15–21.
[6] S. Gottlieb and C.-W. Shu, *Total variation diminishing Runge-Kutta schemes*, Math. Comput., 67 (1998), pp. 73–85.
[7] S. Gottlieb, C.-W. Shu, and E. Tadmor, *Strong-stability-preserving high-order time discretization methods*, SIAM Review, 43 (2001), pp. 89–112.
[8] E. Hairer, S. Norsett, and G. Wanner, *Solving Ordinary Differential Equations I*, Springer-Verlag, 1987.
[9] A. Harten, *High resolution schemes for hyperbolic conservation laws*, J. Comput. Phys., 49 (1983), pp. 357–393.
[10] 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.
[11] G.-S. Jiang and C.-W. Shu, *Efficient implementation of weighted ENO schemes*, J. Comput. Phys., 126 (1996), pp. 202–228.
[12] C. Laney, *CFD Recipes: Software for Computational Gasdynamics.* Web Address: http://capella.colorado.edu/~laney/booksoft.htm.
[13] ———, *Computational Gasdynamics*, Cambridge University Press, 1998.
[14] X.-D. Liu, S. Osher, and T. Chan, *Weighted essentially nonoscillatory schemes*, J. Comput. Phys., 115 (1994), pp. 200–212.
[15] S. Osher and S. Chakravarthy, *Very high order accurate TVD schemes*, in Oscillation Theory, Computation, and methods of Compensated Compactness. The IMA Volumes in Mathematics and Its Applications, C. Dafermos, J. Erikson, D. Kinderlehrer, and M. Slemrod, eds., vol. 2, Springer-Verlag, New York, 1986, pp. 229–271.
[16] S. Ruuth and R. Spiteri, *Two barriers on strong-stability-preserving time discretization methods*, J. Scientific Computation, (to appear).
[17] C.-W. Shu, *Total-variation-diminishing time discretizations*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 1073–1084.
[18] C.-W. Shu and S. Osher, *Efficient implementation of essentially nonoscillatory shock-capturing schemes*, J. Comput. Phys., 77 (1988), pp. 439–471.
[19] B. van Leer, *Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method*, J. Comput. Phys., 32 (1979), pp. 101–136.
[20] J. H. Williamson, *Low-storage Runge-Kutta schemes*, J. Comput. Phys., 35 (1980), pp. 48–56.