Nonlinear evolution using optimal fourth-order strong-stability-preserving Runge-Kutta methods

Raymond J. Spiteri* Steven J. Ruuth†

October 30, 2001

Abstract

Strong-stability-preserving (SSP) time discretization methods (also known as total-variation-diminishing or TVD methods) are popular and effective algorithms for the simulation of partial differential equations having discontinuous or shock-like solutions. Optimal SSP Runge-Kutta (SSPRK) schemes have been previously found for methods with up to five stages and up to fourth order. In this paper we present new optimal fourth-order SSPRK schemes with mild storage requirements and up to eight stages. We find that these schemes are ultimately more efficient than the known fourth-order SSPRK schemes because the increase in the allowable time step more than offsets the added computational expense per step. We demonstrate these efficiencies on a pair of representative problems from compressible gas flows.

1 Introduction

Popular time-stepping schemes are almost exclusively based on linear stability analysis; i.e., the behaviour of the scheme on the test equation $dy/dt = \lambda y$ for $\Re(\lambda) << 0$. Indeed such analysis is very effective on problems having smooth solutions. However, these schemes often perform poorly on problems having discontinuous or shock-like solutions. This poor performance can manifest itself in the form of spurious oscillations, overshoots, or progressive smearing. On the other hand, strong-stability-preserving (SSP) time discretization methods [13, 14, 5, 15] are based on a *nonlinear* stability property that makes them particularly suitable for the simulation of partial differential equations having nonsmooth solutions. This property can be viewed as a generalization of the more well-known total-variation-diminishing (TVD) property to convex functionals or norms other than total variation [5].

Consider the system of ODEs

$$\dot{U} = F(U), \tag{1}$$

*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 appropriate initial conditions, obtained from the method of lines applied to the hyperbolic conservation law

$$u_t + f(u)_x = 0 \tag{2}$$

subject to appropriate initial and boundary conditions. We assume that (2) has been suitably discretized in its spatial variables. Then when properly combined with a SSP time-stepping scheme and appropriate time-step restriction, the numerical solution obtained does not exhibit nonlinear instabilities. However, nonlinear instabilities can occur in a numerical solution obtained with, e.g., a SSP spatial discretization scheme, but a standard (i.e., linearly stable) time-stepping scheme [4].

The favourable properties of SSPRK methods are derived only from convexity arguments. In particular, if the forward Euler method is strongly stable in a given norm with a certain CFL coefficient, higher-order SSPRK methods with a modified CFL coefficient can be constructed as convex combinations of forward Euler steps with various step sizes [14].

Optimal SSP schemes based on Runge-Kutta methods have been known for some time for accuracy orders 1, 2, and 3, where the number of stages s is assumed to be equal to the order p (see e.g., [13, 5]. Gottlieb and Shu [4] recently proved that no such four-stage, fourth-order SSPRK method exists involving just evaluations of $F(\cdot)$. They obtain fourth-order accuracy at the additional expense of introducing two additional evaluations of a related operator $\tilde{F}(\cdot)$, leading to sub-optimal efficiency both in terms of time-step restriction and memory usage (see also Section 2). In [15], a new class of optimal high-order SSPRK schemes was derived without the restriction $s = p$ and the use of $\tilde{F}(\cdot)$ at fourth order. Optimal SSPRK schemes in this class were obtained for general s-stage schemes of first and second order, and schemes having up to five stages and order three and four. These schemes were shown to be more efficient than the SSPRK schemes that had previously appeared because the increase in the allowable time step more than offset the added computational expense per step. Subsequently it was also shown [12] that it is not possible to construct SSPRK schemes of order five or greater without the use of $\tilde{F}(\cdot)$. This naturally makes a compelling case for the construction of optimal fourth-order schemes.

In this paper, we derive new fourth-order SSPRK schemes with mild storage requirements and up to eight stages. These schemes have significantly better stability restrictions than the best SSPRK schemes currently known: The additional stages allow the stability regions to be increased in size sufficiently to more than offset the added expense per time step. Theoretical estimates of the CFL restriction indicate that the optimal eight-stage fourth-order SSPRK scheme offers more than a 200% improvement in the effective time-step restriction over the most well-known fourth-order SSPRK scheme currently in use [14], and more than a 60% improvement in the effective time-step restriction over the most efficient scheme known [15]. We also 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 preliminary investigations indicate that our schemes can also offer practical advantages over the Runge-Kutta methods currently available.

The remainder of this paper is organized as follows. In Section 2, we briefly describe SSP schemes and their use. In Section 3, we determine optimal fourth-order SSPRK schemes having between six and eight stages. In Section 4, we investigate the performance of our new SSPRK schemes on a set of test problems from compressible gas flows having solutions that commonly cause numerical problems. The success of the new methods is measured relative to methods recently proposed in the literature [5, 15].

2 SSP Schemes

We begin by recalling the definition of strong stability:

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 $n \geq 0$.

Here we 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, with the total-variation norm and the infinity norm being two natural possibilities. The choice of a relevant norm depends on the problem to be solved. For example, strong stability is relevant to the solution of (2) because exact solutions to the scalar problem have a range-diminishing property (see e.g., [9]). Thus the strong-stability property is a useful property to require of a numerical solution to (2), even in the case of systems of conservation laws.

In this paper we take our canonical SSPRK method to be an s-stage, explicit Runge-Kutta method written in the form

$$U^{(0)} = U^n \tag{3a}$$

$$U^{(i)} = \sum_{k=0}^{i-1} (\alpha_{ik} U^{(k)} + \Delta t \beta_{ik} F(U^{(k)})), \qquad i = 1, 2, \ldots, s, \tag{3b}$$

$$U^{n+1} = U^{(s)}, \tag{3c}$$

where all the $\alpha_{ik}, \beta_{ik} \geq 0$ and $\alpha_{ik} = 0$ only if $\beta_{ik} = 0$ [13]. This representation of a Runge-Kutta method corresponds to a unique Butcher array form (see e.g., [6]). However, a given Butcher array form generally corresponds to many algebraically equivalent forms (3) [14, 15].

Throughout this article, we give representations (3) that naturally allow stability restrictions to be read from the coefficients of the scheme.

For consistency, we must have that $\sum_{k=0}^{i-1} \alpha_{ik} = 1$, for all $i = 1, 2, \ldots, s$. Hence, if both sets of coefficients α_{ik}, β_{ik} are positive, then (3) 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 [14, 5]:

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 (3) with $\beta_{ik} \geq 0$ is SSP provided

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

where c is the CFL coefficient

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

A similar result holds with β_{ik} replaced by $|\beta_{ik}|$ in the definition of c provided $F(\cdot)$ is replaced with the related operator $\tilde{F}(\cdot)$ for each $\beta_{ik} < 0$, where $\tilde{F}(\cdot)$ is assumed to be strongly stable for Euler's method solved *backwards* in time for the same time-step restriction.

We will be comparing a new class of fourth-order SSPRK methods with s stages and order 4 with $s > 5$ to methods known in the literature where $s = 5$ or some $\beta_{ik} < 0$. We note that if a method requires n_+ evaluations of $F(\cdot)$ and n_- evaluations of $\tilde{F}(\cdot)$ then the effective number of stages of that method is $n = n_+ + n_-$. 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, in order to make a fair comparison of the computational cost of a step, we introduce the following definition:

Definition 2.3 The *effective CFL coefficient* of an SSPRK method of order p is scheme is c/n where c is the CFL coefficient of the method, and n is the number of function evaluations required for one step of the method.

We shall see that that our proposed SSPRK methods also have greater effective CFL coefficients than any method currently available. This result is intuitive because the additional stages contribute extra degrees of freedom that can be used to enlarge the stability region of a given method. However there is a trade-off between having the enlarged stability region and the additional cost per time step. Fortunately, it turns out that this trade-off can be computationally favourable.

3 Optimal SSP Schemes

We now turn to the task of finding optimal SSPRK schemes. Our approach follows closely the methodology discussed in [15]. To begin, we seek to optimize an s-stage, order-4 SSPRK scheme by maximizing its CFL coefficient according to Theorem 2.2. That is, we seek the global maximum of the nonlinear programming problem

$$\max_{(\alpha_{ik},\beta_{ik})} \min \frac{\alpha_{ik}}{\beta_{ik}}, \tag{4}$$

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 (4) is subject to the constraints

$$\sum_{j=0}^{i-1} \alpha_{ik} = 1, \qquad i = 1, 2, \ldots, s, \tag{5}$$

$$\sum_{j=1}^{s} b_j \Phi_j(t) = \frac{1}{\gamma(t)}, \quad t \in T_q, \quad q = 1, 2, 3, 4. \tag{6}$$

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 4 (see e.g., [6]); 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 (6) is equal to 8. 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}.

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 (4) 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

$$\max_{(\alpha_{ik},\beta_{ik})} z, \tag{7a}$$

subject to

$$\begin{aligned}
\alpha_{ik} &\geq 0, && && \text{(7b)}\\
\beta_{ik} &\geq 0, && && \text{(7c)}\\
\sum_{j=0}^{i-1} \alpha_{ik} &= 1, && i = 1, 2, \ldots, s, && \text{(7d)}\\
\sum_{j=1}^{s} b_j \Phi_j(t) &= \frac{1}{\gamma(t)}, && t \in T_q, \quad q = 1, 2, \ldots, 4, && \text{(7e)}\\
\alpha_{ik} - z|\beta_{ik}| &\leq 0. && && \text{(7f)}
\end{aligned}$$

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 (7) was solved using Matlab's Optimization Toolbox for $s = 6, 7$, and 8. In Table A.1 in Appendix A, we give results for the coefficients of the optimal seven- and eight-stage schemes in terms of their numerical values up to double precision. We do not offer formal proofs of optimality; however, these are the results of extensive numerical searches. Following the convention of [15], we refer to these six-, seven-, and eight-stage SSP schemes as SSP(6,4), SSP(7,4) and SSP(8,4), respectively.

Table 1 shows the optimal values for the CFL coefficients for both the known fourth-order schemes and our newly proposed schemes. Table 2 gives the theoretical efficiencies of these new schemes relative to the classical SSP(4**,4) [14, 5] and the recent SSP(5,4) scheme [15]. The percentages quoted respectively refer to the theoretical increases in allowable step size of the new methods relative to known methods. For example, SSP(8,4) method has more than three times the effective CFL coefficient compared to the SSP(4**,4) method, leading to a relative increase in step size of more than 200%.

	SSP(4**,4)	SSP(5,4)	SSP(6,4)	SSP(7,4)	SSP(8,4)
CFL coefficient	0.936	1.508	2.295	3.321	3.965
effective CFL	0.156	0.302	0.382	0.474	0.496

Table 1: CFL and effective CFL coefficients for fourth-order methods.

	SSP(6,4)	SSP(7,4)	SSP(8,4)
Efficiency gain over SSP(4**,4)	145%	204%	218%
Efficiency gain over SSP(5,4)	26%	57%	64%

Table 2: Theoretical relative efficiencies for the newly proposed methods.

We draw particular attention to the efficiency of the SSP(8,4) scheme in Table 2. As mentioned earlier, a four-stage, fourth-order SSPRK scheme does not exist with non-negative coefficients. The figure of 218% is measured relative to the SSP(4^{**}, 4) scheme reported in [4] as the best four-stage scheme of order four that could be found. This scheme has a CFL coefficient of 0.936 and effectively used 6 stages because it involved two coefficients β_{ik} that are negative. The new $(8,4)$ scheme thus compares very favourably. Moreover even when compared to the recent improvement, SSP(5,4), a relative theoretical speed-up of 64% is observed. We note as well that the relative efficiencies of SSP(j,4) and SSP($j+1$,4) decreases as j increases, so the j at which a given method is implemented depends entirely on the user. We also note (see Section 4) that the theoretical efficiency gain does not always translate directly to the observed efficiency gain in practice for a given problem.

Surprisingly, these newly proposed schemes also have very modest storage requirements even though no attempt was made to optimize this property. Consider SSP(7,4) (the SSP(5,4) and SSP(6,4) schemes are similar). From Table A.1 for $s = 7$ we see that it is straightforward to compute a timestep using four memory registers, one for each of $u^{(0)}$ and $u^{(s)}$, and one for each of $u^{(j-1)}$ and $F(u^{(j-1)})$ as we process the jth stage, $j = 1, 2, \ldots, s$. Moreover, if we assume that $F(\cdot)$ can be computed locally (as is commonly the case) relatively little storage is required to evaluate $F(\cdot)$. This leads to an improved count of (little more than) three registers.

4 Numerical Studies

In this section, we study the numerical behaviour of our Runge-Kutta schemes for a few test problems designed to capture solution features that pose particular difficulties to numerical methods. We shall find that our proposed schemes greatly outperform the well-known SSP(4**,4) scheme and offer some useful efficiency gains over the recent SSP(5,4) scheme.

4.1 Test Problems

To investigate the behavior of our time-stepping schemes, we consider two of Laney's five test problems [9]. These two problems involve shocks, contacts, expansion fans, sonic points, and smooth regions. 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 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 solution. The spatial discretization and the results of two 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 [14], we choose fourth-order 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 lend themselves easily to nonuniform meshes, 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 [9, 8], an attribute which is desirable for standardizing numerical studies.

It is noteworthy that high-order, fully TVD spatial discretization schemes are also available; see Osher and Chakravarthy [11]. 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" [9].

It is also noteworthy that recent variations on Shu-Osher methods such as methods based on WENO reconstructions (e.g., [10, 7]) also naturally combine with SSPRK schemes. See [9] for detailed discussions on these and other spatial discretizations appropriate for hyperbolic conservation laws.

4.3 Test Case 1: Linear advection of a square wave

In this test case, the discontinuous initial conditions

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

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 1 exhibits two jump discontinuities in the solution that correspond to contact discontinuities. This test case nicely illustrates progressive contact smearing and dispersion.

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

$$\log_{10}\left(\frac{1}{N}\sum_{i=1}^{N}|U_i - u(x_i,4)|\right),$$

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 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 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 0.1 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 nonsmooth test example, the new fourth-order schemes give a noteworthy improvement in accuracy over SSP(5,4) whereas their respective stabilities are similar. Moreover, a rather dramatic gain in stability (more than 80% improvement) and accuracy can be achieved over the well-known SSP(4**,4) (see Figure 1). For clarity we have not plotted SSP(6,4) and SSP(7,4), but we remark that the performance of these schemes is intermediate between SSP(5,4) and SSP(8,4).

4.4 Test Case 2: 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

$$u(x,t) = \begin{cases} -1 & \text{for } -\infty < x < \xi_1, \\ -1 + 2\frac{x-b_1}{b_2-b_1} & \text{for } \xi_1 < x < \xi_2, \\ 1 & \text{for } \xi_2 < x < \xi_{shock}, \\ -1 & \text{for } \xi_{shock} < x < \infty, \end{cases}$$

where $\xi_1 = -1/3 - t$, $\xi_2 = -1/3 + t$, and $\xi_{shock} = 1/3$ [9]. Test Case 2 is particularly interesting because it illustrates the behaviors near sonic points ($u = 0$) that correspond to an expansion fan and a compressive shock.

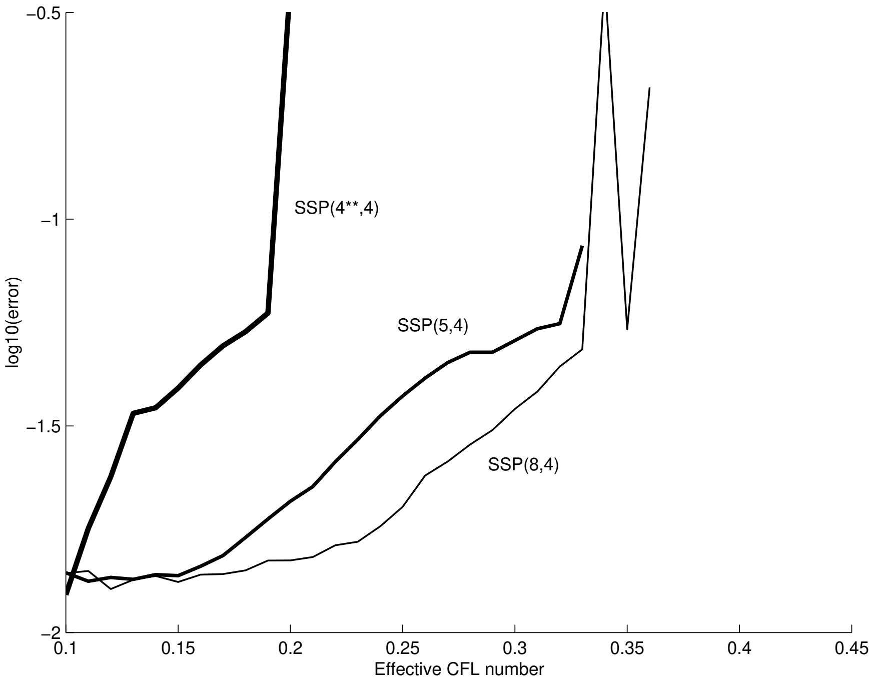

Figure 1: l_1 errors as a function of the effective CFL number for Test Case 1.

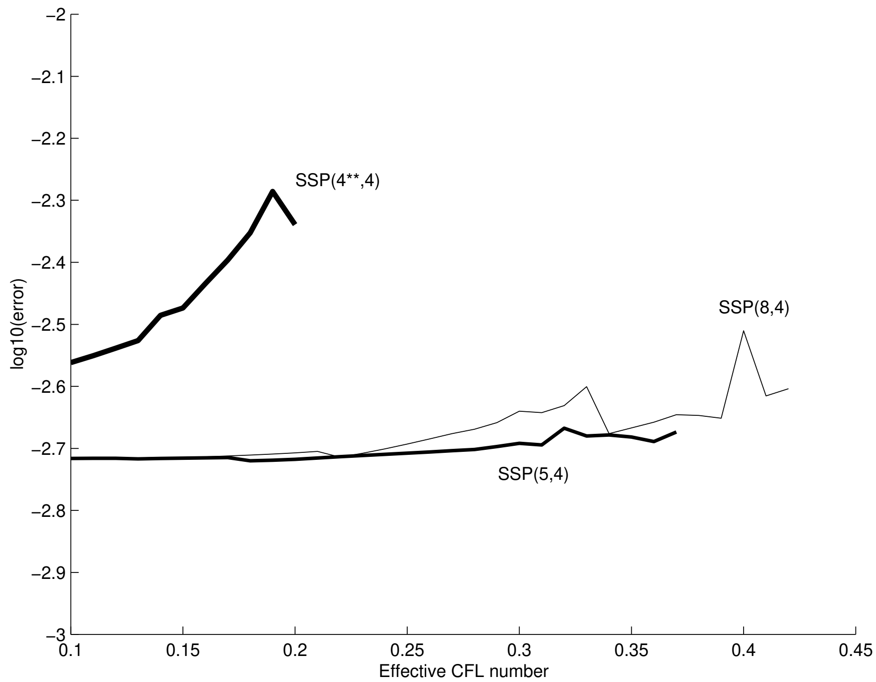

Figure 2: l_1 errors as a function of the effective CFL number for Test Case 2.

The log of the l_1 errors as a function of the effective CFL number are plotted in Figure 2. Based on these plots, it is clear that SSP(8,4) gives approximately a 14% improvement in stability over the carefully optimized SSP(5,4) and a 110% improvement over the well-known SSP(4**,4) scheme. We conclude by noting that SSP(6,4) and SSP(7,4) become unstable at effective CFL numbers of 0.39 and 0.41, respectively.

Appendix A Optimal $(\alpha_{ik},\ \beta_{ik})$ for $s = 7,\ 8$

Table A.1 gives the optimal SSPRK methods of order 4 and 7 or 8 stages in the representation (3).

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] S. Gottlieb and C. Shu, *Total variation diminishing Runge-Kutta schemes*, Math. Comput., 67 (1998), pp. 73–85.

[5] S. Gottlieb, C.-W. Shu, and E. Tadmor, *Strong-stability-preserving high-order time discretization methods*, SIAM Review, 43 (2001), pp. 89–112.

[6] E. Hairer, S. Norsett, and G. Wanner, *Solving Ordinary Differential Equations I*, Springer-Verlag, 1987.

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

[8] C. Laney, *CFD Recipes: Software for Computational Gasdynamics.* Web Address: http://capella.colorado.edu/~laney/booksoft.htm.

[9] ——, *Computational Gasdynamics*, Cambridge University Press, 1998.

[10] X.-D. Liu, S. Osher, and T. Chan, *Weighted essentially nonoscillatory schemes*, J. Comput. Phys., 115 (1994), pp. 200–212.

[11] 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.

[12] S. Ruuth and R. Spiteri, *Two barriers on strong-stability-preserving time discretization methods*, J. Scientific Computation, To appear.

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

[14] C.-W. Shu and S. Osher, *Efficient implementation of essentially nonoscillatory shock-capturing schemes*, J. Comput. Phys., 77 (1988), pp. 439–471.

[15] R. J. Spiteri and S. J. Ruuth, *A new class of optimal high-order strong-stability-preserving time-stepping schemes.* Unpublished manuscript (under review), April 2001.

Table A.1: The coefficients of the optimal SSPRK (7,4) and SSPRK (8,4) schemes.

Stages $s = 7$							
1.00000000000000							α_{ik}
0.20161507213829	0.79838492786171						
0.19469598207921	0.00000000000000	0.80530401792079					
0.58143386885601	0.00000000000000	0.00000000000000	0.41856613114399				
0.01934367892154	0.00000000000000	0.00000000000000	0.00000000000000	0.98065632107846			
0.00000000000000	0.00000000000000	0.00000000000000	0.00000000000000	0.00000000000000	1.00000000000000		
0.06006304558847	0.00000000000000	0.30152730794242	0.10518998496676	0.01483791154585	0.00000000000000	0.51838174995650	
0.30111872706068							β_{ik}
0.00000000000000	0.24040865318216						
0.00000000000000	0.00000000000000	0.24249212077315					
0.00000000000000	0.00000000000000	0.00000000000000	0.12603810060080				
0.00000000000000	0.00000000000000	0.00000000000000	0.00000000000000	0.29529398308716			
0.00000000000000	0.00000000000000	0.00000000000000	0.00000000000000	0.00000000000000	0.30111872706068		
0.00000000000000	0.00000000000000	0.09079551914158	0.02888359354880	0.00000000000000	0.00000000000000	0.15609445267839	
CFL number $c = 3.32094921415661$							

Stages $s = 8$								
1.00000000000000								α_{ik}
0.00567928750917	0.99432071249083							
0.23497715494190	0.00000000000000	0.76502284505810						
0.59375456337731	0.00000000000000	0.00000000000000	0.40624543662269					
0.02420769361812	0.00000000000000	0.00000000000000	0.00000000000000	0.97579230638188				
0.06350784267270	0.00960092568116	0.00000000000000	0.00000000000000	0.00740891015485	0.91948232149129			
0.00000000000000	0.00000000000000	0.02266881686153	0.00356332290044	0.00000000000000	0.00291914409966	0.97084871613837		
0.02227429049794	0.00000000000000	0.10124098564931	0.26739070665415	0.02066582656377	0.04170345042967	0.00295617243505	0.54376856777011	
0.25222876436176								β_{ik}
0.00000000000000	0.25079628469157							
0.00000000000000	0.00000000000000	0.19296076691950						
0.00000000000000	0.00000000000000	0.00000000000000	0.10246678450575					
0.00000000000000	0.00000000000000	0.00000000000000	0.00000000000000	0.24612288770977				
0.00046891860383	0.00000000000000	0.00000000000000	0.00000000000000	0.00186874025601	0.23191988980488			
0.00000000000000	0.00000000000000	0.00571772767067	0.00000000000000	0.00000000000000	0.00073629210376	0.23503283043482		
0.00000000000000	0.00000000000000	0.02553588871387	0.06710254950598	0.00201685177128	0.00060607207992	0.00000000000000	0.12179666277671	
CFL number $c = 3.96465487403998$								