MATHEMATICS OF COMPUTATION
Volume 75, Number 253, Pages 183–207
S 0025-5718(05)01772-2
Article electronically published on September 16, 2005

GLOBAL OPTIMIZATION OF EXPLICIT STRONG-STABILITY-PRESERVING RUNGE-KUTTA METHODS

STEVEN J. RUUTH

Abstract. Strong-stability-preserving Runge-Kutta (SSPRK) methods are a type of time discretization method that are widely used, especially for the time evolution of hyperbolic partial differential equations (PDEs). Under a suitable stepsize restriction, these methods share a desirable nonlinear stability property with the underlying PDE; e.g., positivity or stability with respect to total variation. This is of particular interest when the solution exhibits shock-like or other nonsmooth behaviour. A variety of optimality results have been proven for simple SSPRK methods. However, the scope of these results has been limited to low-order methods due to the detailed nature of the proofs. In this article, global optimization software, BARON, is applied to an appropriate mathematical formulation to obtain optimality results for general explicit SSPRK methods up to fifth-order and explicit low-storage SSPRK methods up to fourth-order. Throughout, our studies allow for the possibility of negative coefficients which correspond to downwind-biased spatial discretizations. Guarantees of optimality are obtained for a variety of third- and fourth-order schemes. Where optimality is impractical to guarantee (specifically, for fifth-order methods and certain low-storage methods), extensive numerical optimizations are carried out to derive numerically optimal schemes. As a part of these studies, several new schemes arise which have theoretically improved time-stepping restrictions over schemes appearing in the recent literature.

1. Introduction

Popular time-stepping schemes are typically based on linear stability analysis. Such analysis is often 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 loss of positivity. On the other hand, strong-stability-preserving (SSP) time discretization methods [21, 22, 8, 24] are designed to preserve the nonlinear stability properties that arise when the forward Euler time-stepping scheme is applied to the spatially discretized system. The ability to preserve this underlying nonlinear stability makes SSP time-stepping methods particularly suitable for the simulation of partial differential equations with nonsmooth solutions.

Received by the editor July 2, 2003 and, in revised form, August 31, 2004.
2000 *Mathematics Subject Classification.* Primary 65L06, 65M20.
Key words and phrases. Strong-stability-preserving, total-variation-diminishing, Runge-Kutta methods, high-order accuracy, time discretization, downwinding, low-storage.
This work was partially supported by a grant from NSERC Canada.

In this paper, particular attention will be paid to the development of guaranteed optimal[1] explicit SSP Runge-Kutta (SSPRK) time-stepping methods for hyperbolic PDEs; e.g.,

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

subject to appropriate initial conditions. Solutions to these and other PDEs are often approximated by sequentially discretizing the temporal and spatial derivatives. For example, in the method of lines, a discretization of the spatial derivatives of the PDE is carried out to produce a large set of coupled time-dependent ordinary differential equations (ODEs)

$$\dot{U} = F(U). \tag{1.2}$$

These ODEs can then be treated by suitable time-stepping techniques such as linear multi-step or Runge-Kutta methods.

For hyperbolic conservation laws, papers by Shu [21], Shu and Osher [22], and subsequent work [7, 8, 20] have shown that improved nonlinear stability can sometimes be found by appropriately selecting nonlinearly stable upwind-biased ($F(U)$) and downwind-biased ($\tilde{F}(U)$) spatial discretizations according to the coefficients of the time-stepping method. Thus, this paper considers optimality over the broad class of explicit SSPRK schemes that includes both upwind-and downwind-biased spatial discretizations. We remark, however, that most of the optimal schemes that we present involve only nonnegative coefficients. Nonnegative coefficient schemes are appropriate to apply to general problems (including dissipative PDEs) since they do not involve $\tilde{F}(\cdot)$

Optimal explicit SSPRK schemes with nonnegative coefficients and where the number of stages s is equal to the order p for $s = p = 1, 2$, and 3 have been known for some time [21, 22, 7]. Gottlieb and Shu [7] showed that no such method exists with nonnegative coefficients when $s = p = 4$. In [24, 25], Spiteri and Ruuth studied the general class of explicit nonnegative coefficient SSPRK methods with $s > p$. They gave optimal SSPRK schemes with s stages and orders 1 and 2 (see also [21, 7]), as well as specific schemes for $p = 3$, $s = 4, 5$ and $p = 4$, $s = 5, 6, 7, 8$. The advantage afforded by these high-stage schemes is that the increase in the CFL coefficient allows for a large enough increase in the stable time step to more than offset the increase in computational cost per step. However, in [19] they showed that it was impossible to have an explicit SSPRK method with order greater than 4 with nonnegative coefficients. Ruuth and Spiteri [20] later gave a unified treatment of explicit SSPRK schemes with unrestricted coefficients and found that many of the optimal explicit nonnegative coefficient SSPRK methods are also optimal in terms of effective CFL coefficient when the sign of the coefficients is not restricted. However, optimality proofs for third-order schemes were limited to the simplest cases ($s = 3, 4$), and proofs were not attempted for higher orders. Indeed, even *finding* efficient fifth-order SSPRK schemes proved challenging, and motivated the development of this study.

To date, most of the practical interest in SSPRK schemes has been for explicit schemes, and, unless otherwise stated, all studies appearing in this report are for *explicit* SSPRK schemes. Implicit SSPRK schemes have been investigated by a number of authors [8, 4, 10, 9]. It is noteworthy that implicit SSPRK schemes with

[1]Optimality will be determined with respect to the *effective CFL coefficient* defined in Section 2.

nonnegative coefficients cannot be unconditionally SSP [4, 10, 9] and that diagonally implicit schemes with unrestricted coefficients cannot be unconditionally SSP [8].

The remainder of the paper is organized as follows. In Section 2 we review some relevant results on SSP schemes and define key concepts such as the effective CFL coefficient. In Section 3 we develop a mathematical formulation of the problem suitable for global optimization and provide details on our optimization procedures. Section 4 uses these techniques to achieve the first guarantees of optimality of several third- and fourth-order SSPRK schemes. New fifth-order methods are also derived which exceed the theoretical efficiency of previously known methods. In Section 5 our approach is used to guarantee the optimality of a variety of low-storage SSPRK schemes appearing in the literature. Some improvements for recently derived low-storage schemes are also found. Finally Section 6 concludes by summarizing the main findings of the paper.

2. Background on SSP schemes

In this section we give some theoretical background on SSPRK schemes. We begin by recalling the definition of strong stability:

Definition 1. A sequence $\{U^n\}$ is said to be *strongly stable* in a given semi-norm $||\cdot||$ if $||U^{n+1}|| \le ||U^n||$ for all $n \ge 0$.

Now assume that upwind-biased spatial discretizations are appropriate and express an s-stage, explicit Runge-Kutta method using an $\alpha - \beta$ (or Shu-Osher [3]) representation

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

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

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

where all the $\alpha_{ik} \ge 0$ [21]. A given Runge-Kutta scheme will typically have many different representations of this type; see [24] for a discussion and an illustrative example. Throughout this paper, we construct representations that maximize the CFL coefficient, as defined in Theorem 1 or 2 below. We call an optimal representation of this type an *optimal $\alpha - \beta$ representation* and note that such representations have been constructed both numerically (e.g., [22, 8, 24]) and via a contractivity-based approach [3, 9].

By consistency we must have that $\sum_{k=0}^{i-1} \alpha_{ik} = 1$, $i = 1, 2, \ldots, s$. Assuming that both sets of coefficients α_{ik}, β_{ik} are nonnegative, it is clear that (2.1) is a convex combination of forward Euler steps with various step sizes $\frac{\beta ik}{\alpha_{ik}} \Delta t$. The strong stability property follows easily, and we conclude [22, 8]:

Theorem 1. *If the forward Euler method is strongly stable under the CFL restriction $\Delta t \le \Delta t_{FE}$, then the Runge-Kutta method* (2.1) *with $\beta_{ik} \ge 0$ is SSP provided*

$$\Delta t \le C \Delta t_{FE},$$

where C is the CFL coefficient

$$C \equiv \min \left\{ c_{ik} : 0 \le k < i \le s \right\}, \text{ where } c_{ik} = \begin{cases} \frac{\alpha_{ik}}{\beta_{ik}} & \text{if } \beta_{ik} \ne 0, \\ \infty & \text{otherwise.} \end{cases}$$

The CFL coefficient for strong stability has an interesting relationship with the time-stepping restriction associated with the more classical concept of *contractivity*, where the difference $||\tilde{U}^{n+1} - U^{n+1}||$ between any two sequences is required to be nonincreasing with increasing n (see, e.g., [23, 14, 15]). It turns out that many of the optimal SSP schemes found in [24] agree with optimal contractive schemes in [15]. In fact, for a given Runge-Kutta method involving only upwind-biased spatial discretrizations (2.1), Ferracina and Spijker [3, 4] have proven that the CFL coefficient C corresponding to an optimal $\alpha - \beta$ representation equals the related quantity $R(A, b)$ [15] arising in contractivity studies, provided $C \geq 0$. (If $C < 0$ for an optimal $\alpha - \beta$ representation, then $R(A, b) = 0$ [3].) See also the recent work of Higueras [9] for some interesting relationships between contractive and strong-stability-preserving Runge-Kutta methods when both upwind- and downwind-biased operators arise.

SSPRK schemes with negative coefficients β_{ik} are also accommodated by modifying the spatial discretization. Following the procedure first suggested in papers by Shu [21] and Shu and Osher [22], whenever $\beta_{ik} < 0$, the operator $\tilde{F}(\cdot)$ is used instead of $F(\cdot)$, where $\tilde{F}(\cdot)$ approximates the same derivatives as $F(\cdot)$ but is assumed to be strongly stable for Euler's method solved *backwards* in time under a suitable time-step restriction. In practice, this corresponds to differencing in the downwind direction.

Remark 1. Suppose that an appropriate spatial discretization has been derived for computing F for the hyperbolic conservation law (1.1). Then $-\tilde{F}$ is normally found by applying this same spatial discretization technique to the backwards-in-time variant

$$u_t + (-f(u)_x) = 0.$$

To explicitly illustrate, consider discretizing the linear, variable coefficient problem

$$u_t + a(x)u_x = 0 \tag{2.2}$$

using a uniform mesh (step size h) and first-order one-sided differencing. Then F can be found by applying first-order upwinding to (2.2) to give

$$F_j = -\begin{cases} a(x_j)(U_j - U_{j-1})/h & \text{if } a(x_j) > 0, \\ a(x_j)(U_{j+1} - U_j)/h & \text{otherwise.} \end{cases}$$

Similarly, first-order upwinding can also be applied to the backwards-in-time version of (2.2) to give

$$-\tilde{F}_j = \begin{cases} a(x_j)(U_{j+1} - U_j)/h & \text{if } a(x_j) > 0, \\ a(x_j)(U_j - U_{j-1})/h & \text{otherwise,} \end{cases}$$

from which $\tilde{F}$ is trivially obtained. Clearly $\tilde{F}$ approximates the same quantity as F, however, $\tilde{F}$ is based on first-order downwind differencing instead of first-order upwinding. See [21] for further discussion on constructing downwind-biased discretizations.

As shown by Shu [21] and Shu and Osher [22], an interesting generalization of Theorem 1 is obtained by using both upwind- and downwind-biased operators:

Theorem 2. *Let Euler's method applied forward in time combined with the spatial discretization $F(\cdot)$ be strongly stable under the CFL restriction $\Delta t \leq \Delta t_{FE}$. Let Euler's method applied backward in time combined with the spatial discretization*

$\tilde{F}(\cdot)$ also be strongly stable under the same CFL restriction $\Delta t \le \Delta t_{FE}$. Then the Runge-Kutta method (2.1) *is SSP provided*

$$\Delta t \le C \Delta t_{FE},$$

where C is the CFL coefficient

$$C \equiv \min\left\{c_{ik} : 0 \le k < i \le s\right\}, \text{ where } c_{ik} = \begin{cases} \frac{\alpha_{ik}}{|\beta_{ik}|} & \text{if } \beta_{ik} \neq 0, \\ \infty & \text{otherwise,} \end{cases} \tag{2.3}$$

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

We note that the assumptions on strong stability of Euler's method applied forward and backward in time restricts the theoretical advantages of this result to non-dissipative equations such as (1.1).

If every coefficient β_{ik} is nonnegative, then the number of stages is trivially equal to the number of function evaluations for an irreducible explicit Runge-Kutta method. However, if both $F(U^{(k)})$ and $\tilde{F}(U^{(k)})$ are required for some k, the Runge-Kutta method (2.1) has more function evaluations[2] than stages. As discussed by Ruuth and Spiteri [20], the first step in deriving optimal schemes is to create a fair comparison of the computational cost of a given Runge-Kutta method by considering general methods that allow precisely one (new) function evaluation per stage. This can be achieved by insisting that the nonzero coefficients β_{ik} for a given k are all of the same sign [20]. For the remainder of the paper, we will tacitly assume that the schemes under consideration are of this form, and we remark that schemes that are written combining nonnegative and negative coefficients β_{ik} within a given level k can be augmented with additional stages to be of this form [20]. Thus, without loss of generality, we have that the total number of evaluations of $F(\cdot)$ and $\tilde{F}(\cdot)$ is identically equal to the number of stages of the method.

We emphasize that with this formulation it is quite natural to search for the optimal scheme for a given order and a given number of stages (function evaluations) by maximizing the CFL coefficient. In earlier formulations the effective CFL coefficient could vary with the number of $\tilde{F}(\cdot)$ evaluations, complicating the development of optimal schemes.

Another advantage to this formulation is that schemes can be represented in Butcher array form via

$$\begin{aligned} a_{ik} &= \beta_{i-1,k-1} + \sum_{j=k+1}^{i-1} \alpha_{i-1,j-1} a_{jk}, k = 1, 2, \ldots, i-1, i = 2, 3, \ldots, s, \\ b_k &= \beta_{s,k-1} + \sum_{j=k+1}^{s} \alpha_{s,j-1} a_{jk}, \qquad k = 1, 2, \ldots, s, \end{aligned} \tag{2.4}$$

[2]The only difference between $\tilde{F}(\cdot)$ and $F(\cdot)$ is a change in the upwind direction, so $\tilde{F}(\cdot)$ can clearly be computed with the same cost as $F(\cdot)$ [8]. Recent studies make the assumption that if both $\tilde{F}(U^{(k)})$ and $F(U^{(k)})$ must be computed for some k, the cost as well as the storage requirements for that k doubles [7, 8, 24, 20].

since differences of the form $F(U^{(i)}) - \tilde{F}(U^{(i)})$ do not arise. Thus the method can be implemented as

$$K_i = \begin{cases} F\left(U^n + \Delta t \sum_{j=1}^{i-1} a_{ij} K_j\right) & \text{if } b_i \geq 0, \\ \tilde{F}\left(U^n + \Delta t \sum_{j=1}^{i-1} a_{ij} K_j\right) & \text{otherwise,} \end{cases} \qquad i = 1, 2, \ldots, s,$$

$$U^{n+1} = U^n + \Delta t \sum_{i=1}^{s} b_i K_i.$$

Butcher array form may be desirable for a variety of reasons such as to simplify optimality proofs, facilitate optimization (see Section 3), or (in some instances) reduce storage requirements. Note that by construction, if the $\beta_{ij}, j+1 \leq i \leq s$, are of one particular sign, then the corresponding $a_{i,j+1}, j+2 \leq i \leq s$, and b_{j+1} values must be of that same sign. We further remark that the differences $F(U^{(i)}) - \tilde{F}(U^{(i)})$ contribute to artificial dissipation and smearing [20]. For example, this difference is proportional to the discrete Laplacian when first-order upwinding is applied to the linear advection equation. A natural consequence of this formulation is that during optimization these dissipative differences do not arise, leading to schemes with smaller errors and less smearing than would otherwise occur [20].

In order to make a fair comparison of the relative efficiencies of these methods and to derive optimal schemes, we make the following definition.

Definition 2. The *effective CFL coefficient* C_{eff} of a SSPRK method is C/s, where C is the CFL coefficient of the method and s is the number of stages (function evaluations) required for one step of the method.

As conjectured in Shu and Osher [22] and subsequently proven in Gottlieb and Shu [7], the optimal two-stage, order-two SSPRK scheme with nonnegative coefficients is the modified Euler scheme. 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 a numerically optimal s-stage, order-p SSPRK scheme as SSP(s,p). This scheme is the first member of a class of s-stage, order-two SSPRK schemes [5, 24] with a CFL coefficient of $s-1$. For nonnegative coefficients, the optimality of these schemes was proven in [24]. Optimality for the case of s general function evaluations was shown in [20].

Shu and Osher [22] also conjectured that the optimal three-stage, order-three SSPRK scheme is

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

which has a CFL coefficient of 1 as well. This scheme is commonly called the *TVD Runge-Kutta scheme*, but we will refer to it as SSP(3,3). Optimality of this scheme was proved for nonnegative coefficients in [7] and for three general function evaluations in [20].

In [19], Ruuth and Spiteri derived a linear bound that is used to prove that the optimal four-stage, order three SSPRK scheme with nonnegative coefficients is

$$\begin{aligned} U^{(1)} &= U^n + \frac{1}{2}\Delta t F(U^n), \\ U^{(2)} &= U^{(1)} + \frac{1}{2}\Delta t F(U^{(1)}), \\ U^{(3)} &= \frac{2}{3}U^n + \frac{1}{3}U^{(2)} + \frac{1}{6}\Delta t F(U^{(2)}), \\ U^{n+1} &= U^{(3)} + \frac{1}{2}\Delta t F(U^{(3)}), \end{aligned}$$

which has a CFL coefficient of 2. Following [24] we will refer to this scheme as SSP(4,3). Optimality in the unrestricted case is shown in [20].

Moving on to nonnegative coefficient SSPRK schemes with five stages and order three gives a numerically optimized scheme, SSP(5,3), with a CFL coefficient of approximately 2.65 [24]. This scheme has a CFL coefficient that agrees with the contractivity bound $R(A, b)$ for linear constant-coefficient problems [14]. Because these restrictions are equivalent [3] and the time-stepping restriction for nonlinear problems cannot exceed that for linear problems, we conclude that SSP(5,3) is also an optimal five-stage, third-order nonnegative coefficient SSPRK scheme. Similarly, it was noted in [20] that third-order schemes with up to nine stages can be constructed that have CFL coefficients that agree with the contractivity bound $R(A, b)$ for linear constant-coefficient problems. Unfortunately, proving optimality for any of these schemes ($s \geq 5$) when s general function evaluations are involved is much more complicated than for the three- or four-stage cases. In Section 4.1 we use global optimization to directly guarantee the optimality of SSP(5,3) and SSP(6,3) over this broader class of schemes. We also give third-order schemes with seven and eight function evaluations and guarantee their optimality using global optimization in combination with the linear contractivity bound.

The analysis for orders greater than 3 is more complicated still, since even for the nonnegative coefficient case the linear contractivity bound fails to give a sharp bound for the nonlinear problem. However, by appropriately applying global optimization techniques, we are able to guarantee optimality of the five-stage, fourth-order SSPRK scheme given in [24, 15] in the general setting where downwind-biased spatial discretizations may arise. See Section 4.2. We remark that this scheme is particularly important since it is impossible for a fourth-order SSPRK scheme to use fewer than five (general) function evaluations [7, 20].

In [19] it is shown that explicit SSPRK schemes with only nonnegative coefficients do not exist with order greater than four. A similar restriction to orders four or less was shown for contractive schemes by Kraaijevanger [15]. This means that the search for schemes of order-five and higher must involve evaluations of the downwind-biased operator $\tilde{F}(\cdot)$. Fortunately, this still leads to schemes with competitive effective CFL coefficients. In Section 5 we derive a fifth-order scheme with an effective CFL coefficient of 0.3395, exceeding that of the popular SSP(3,3) scheme.

Optimal *low-storage* SSPRK schemes have also received some attention in the literature [7, 8, 24, 13]. In particular, Gottlieb and Shu [7] find an optimized three-stage, third-order scheme of Williamson [32] type and Kennedy, Carpenter and Lewis [13] find several optimized schemes of van der Houwen and Wray type. Using

global branch-and-bound optimization techniques, we are able to guarantee the optimality of these schemes and others. In the course of this work, improvements to some recently published low-storage schemes are also found. We remark that until now only nonnegative coefficients have been used to construct low-storage schemes, since it has been assumed that downwind-biased operators destroy the low-storage property. Section 5 disproves this assumption by exhibiting two examples of low-storage schemes that are more efficient than the corresponding nonnegative coefficient schemes.

The remainder of the paper commences with a discussion on our optimization techniques.

3. Global optimization of SSPRK schemes

Traditional nonlinear programs are guaranteed to converge to the optimal solution only under certain convexity assumptions [28]. On the other hand, deterministic global optimization algorithms of the branch-and-bound type are available for a variety of problems that are guaranteed to provide global optima under fairly general assumptions [28].

In this section we give a mathematical formulation appropriate for BARON [28, 29], and show how to bound the variables and their expressions in the nonlinear programming (`NLP`) problem. We also effectively break the problem into parallel subproblems to reduce the total number of CPU cycles required.

Subsequent sections use these ideas to determine and to guarantee the optimality of several third- and fourth-order schemes in both the low-storage and general settings.

3.1. Formulation of the optimization problem.

We seek to optimize an s-stage, order-p SSPRK scheme by maximizing the CFL coefficient defined in Theorem 2. To achieve this goal, we will transform the problem into a smooth version that has additional constraints, but is more amenable to treat using standard numerical optimization techniques. As a first step, we replace the nonsmooth objective function arising in the original formulation with a new variable, z, that is constrained to be a lower bound on all the ratios $\{\alpha_{ik}/|\beta_{ik}|\}$. This well-known technique applied to our problem gives the equivalent formulation [6, p. 96, 97]

$$\max_{(\alpha_{ik},\beta_{ik})} z, \tag{3.1a}$$

subject to

$$\alpha_{ik} \geq 0, \tag{3.1b}$$

$$\begin{aligned} \beta_{k+1,k}, \beta_{k+2,k}, \ldots, \beta_{sk} &\geq 0, \\ \text{or } \beta_{k+1,k}, \beta_{k+2,k}, \ldots, \beta_{sk} &\leq 0, \quad k = 0, \ldots, s-1, \end{aligned} \tag{3.1c}$$

$$\sum_{k=0}^{i-1} \alpha_{ik} = 1, \qquad i = 1, 2, \ldots, s, \tag{3.1d}$$

$$\sum_{j=1}^{s} b_j \Phi_j(t) = \frac{1}{\gamma(t)}, \quad t \in T_q, \quad q = 1, 2, \ldots, p, \tag{3.1e}$$

$$\alpha_{ik} - z|\beta_{ik}| \geq 0, \quad k = 0, 1, \ldots, i-1, \ i = 1, 2, \ldots, s, \tag{3.1f}$$

where the $\Phi_j(t)$ and b_j are nonlinear polynomials in $\alpha_{ik},\ \beta_{ik}$ and the optimal z-value equals the CFL coefficient. The notation b_j is used in the usual sense of the Butcher array representation of a Runge-Kutta method, and the symbol T_q stands for the set of all rooted trees of order equal to q. The number of constraints corresponding to the order conditions (3.1e) is 1,2,4,8 or 17 for orders p=1,2,3,4 or 5, respectively. We remark that this approach was first used for the optimization of SSPRK schemes with nonnegative coefficients in [24] and was first proposed for the optimization of SSPRK schemes with unrestricted coefficients in [20].

This formulation is effective for the nonnegative coefficient problem [24]. To study the unrestricted case, we wish to remove the nonsmooth absolute value function from our formulation, since general-purpose software for nonsmooth optimization problems is not expected to be as efficient or robust as general-purpose software designed for smooth problems. In particular, the GAMS User's Guide [1, p. 70] states

> Smooth functions can be used routinely in nonlinear models, but non-smooth ones may cause numerical problems and should be used only if unavoidable, and only in a special mode type called `DNLP`. However, the use of `DNLP` model type is strongly discouraged and the use of binary variables is recommended to model non-smooth functions.

Following this recommendation, we introduce a sign variable for each level,

$$\sigma(k) = \begin{cases} +1 & \text{if } \beta_{k+1,k}, \beta_{k+2,k}, \ldots, \beta_{sk} \geq 0, \\ -1 & \text{otherwise,} \end{cases}$$

indicating the sign of the coefficients at level $k, 1 \leq k \leq s$, and defining a variable, $\bar{\beta}_{ik}$, to represent the absolute value of β_{ik}. To formulate the problem in these new variables, we replace conditions (3.1c) and (3.1f) with $\bar{\beta}_{k+1,k}, \bar{\beta}_{k+2,k}, \ldots, \bar{\beta}_{sk} \geq 0$ and $\alpha_{ik} - z\bar{\beta}_{ik} \geq 0$, respectively. Our updated formulation (given in terms of $\sigma(k)$, $\bar{\beta}_{ik}$ and α_{ik}) is completed by replacing each β_{ik} by $\sigma(k)\bar{\beta}_{ik}$ in the order conditions (3.1e).

This mixed integer nonlinear programming (`MINLP`) formulation is comprised of polynomial objective and constraint functions, and thus is suitable for optimization in BARON.[3] However, the nonlinear order conditions take a simpler form when written in terms of the Butcher array entries a_{ik} and b_k rather than $\alpha_{ik},\ \beta_{ik}$, so we prefer to solve for the $\sigma(k)$, α_{ik} and the (unsigned) Butcher array entries directly. The $\bar{\beta}_{ik}$ are formed, where needed, as a linear combination of α_{ik} and Butcher array entries using (2.4). While this results in more complicated constraints on the bound, z, the overall speed of computation sometimes experiences a noteworthy improvement (e.g., a factor of two or more was observed in some tests), since the complicated nonlinear order conditions are simplified.

3.2. **Parallelization.** As prescribed above, the optimal SSP scheme is given as the global solution to a `MINLP` problem. While this is a suitable mathematical formulation for a variety of GAMS optimization software using the GAMS `MINLP` model type [2], the total computational effort can be reduced significantly by instead solving 2^s `NLP` problems, each corresponding to one of the possible sign combinations.

[3]Note that a variety of nonlinear functions are allowed in BARON [26, p. 10]: "In addition to multiplication and division, GAMS/BARON can handle nonlinear functions that involve $\exp(x)$, $\ln(x)$, $\exp(x)$, $\ln(x)$, x^α for real α, β^x for real β, x^y, and $|x|$."

This leads to a total of 2^s cases, which we treat in a parallel fashion using the GAMS `NLP` model type.

The total time savings was not found to have a predictable dependency on the number of stages. For example, for third order and $s = 3, 4, 5$, we found that treating separate `NLP` problems required about $40\%, 60\%$ and 40% of the total CPU time of a single `MINLP` problem, respectively. It is interesting that in the optimization of linear multistep SSP schemes, both approaches take about the same overall CPU time if *product disaggregation* [27] (or distributing products over their sums) is used in the `MINLP` problem [18]. Without product disaggregation the `MINLP` problem is much slower [18]. For many models, product disaggregation leads to overall improved efficiency by making use of tighter linear programming relaxations; see [27] for details. Product disaggregation was not found to improve the efficiency of the `MINLP` problem for the more complicated Runge-Kutta case considered here. Nonetheless, it is possible that the tightness of linear programming relaxations plays some role in the relative efficiencies of the two approaches.

We remark that the number of cases can be reduced somewhat by using the fact that the number of nonnegative levels must be equal or exceed the CFL coefficient [20]:

Lemma 1. *Suppose a consistent s-stage SSPRK method* (2.1) *has coefficients* $\beta_{ik} \geq 0$ *at* ℓ *distinct stages, i.e.,* $\beta_{ik} \geq 0$ *for all* i *and* $k = k_1, k_2, \ldots, k_\ell$ *with* $0 \leq k_1 < k_2 < \cdots < k_\ell \leq s - 1$. *Then the CFL coefficient* C *of the method satisfies* $C \leq \ell$.

We illustrate the usage of this lemma for eight stages and order three in Section 4.1.

3.3. **Parameter bounds.** For an efficient search and to guarantee optimality, the variables must be bounded. Fortunately we know that all of the Butcher array entries are bounded by the inverse of the CFL coefficient [20]:

Lemma 2. *If a method of the form* (2.1) *with* $\alpha_{ik} \geq 0$ *has a CFL coefficient* $c \geq m > 0$, *then*

$$-\frac{1}{m} \leq a_{ik} \leq \frac{1}{m} \text{ for all } k = 1, 2, \ldots, i-1,\ i = 2, 3, \ldots, s, \tag{3.2}$$

$$-\frac{1}{m} \leq b_k \leq \frac{1}{m} \text{ for all } k = 1, 2, \ldots, s. \tag{3.3}$$

According to the lemma, if we can find *any* feasible solution with a CFL coefficient of m, we know that each of the Butcher array entries of the optimal solution must be bounded in absolute value by $1/m$. Fortunately, finding good feasible solutions in BARON is a relatively inexpensive task which typically takes no more than a few seconds in the preprocessing step for orders four or less.

If a better bound is not known, we may choose zero as a lower bound on the variable z, since the CFL coefficient must be positive to be of any practical value. Note that Lemma 1 may be used to derive an upper bound if such a bound is not immediately clear from the problem or from numerical experience. We remark that, in practice, computational speed was much more critically affected by the bounds on the Butcher array entries than by bounds on z.

3.4. **Optimization software.** Deterministic global branch-and-bound software is particularly appropriate for this formulation since it can guarantee optimality of a

solution to within the given tolerance, provided bounds are supplied on each of the variables.

Briefly, branch-and-bound methods solve an optimization problem by constructing and solving a related (but much easier to solve) relaxed problem over successively refined partitions in the feasible space. In a minimization, the objective of this relaxation bounds the objective of the original problem from below, while local optimization and other bounding heuristics on the original problem give an upper bound on the desired objective. With enough refinement, the difference between the upper and lower bounds becomes sufficiently small, and the procedure terminates with the current upper bounding solution. See [28, 29] for a detailed exposition on deterministic branch-and-bound optimization methods.

As explained in [28, p. 4], "convergence of this algorithm (branch-and-bound) is well established as long as the partitioning and bounding schemes obey certain properties (cf. [12])." In this project we use BARON 5.0 from the GAMS suite of software to solve our reformulated problem. Given sufficient CPU time, BARON will provide the optimal solution within the prescribed tolerances as long as the user-supplied variable bounds (or the ones BARON infers from the problem constraints) are such that all variables and expressions are bounded [28]. If these bounds are missing, BARON reports upon termination:

```
*** User did not provide appropriate variable bounds ***
     *** Globality is therefore not guaranteed ***
```

Thus the results in this article will be either *guaranteed optimal* or *numerically optimal*. The former means that BARON is able to guarantee that the problem has been solved (globally) to within the given tolerances. *Numerically optimal* results are not guaranteed to be globally optimal and are instead based on extensive numerical searching. Globality will not be guaranteed if some variable bound is missing (see above) or if BARON is terminated before a guarantee of optimality can be found (e.g., due to practical limitations on CPU time).

Because our SSP-based formulation involves polynomial functions which are (typically) defined on bounded sets, it is very naturally treated by global branch-and-bound optimization software such as BARON. On the other hand, BARON cannot be directly applied to the contractivity-based formulation proposed in [3] since BARON cannot treat the nonlinear objective function appearing there (see footnote 3).

We recommend the use of BARON over Matlab's Optimization Toolbox (cf. [24]) for several reasons. Not only does BARON have the potential to guarantee optimality of the solutions found, but it often finds solutions with larger effective CFL coefficients. Furthermore, it is straightforward to satisfy constraints to full double precision accuracy within GAMS by using the result from BARON as a starting point for a local optimization with MINOS [17].

Throughout this manuscript, all schemes are provided to the full precision of the optimization software (15 decimal digits). We conjecture that schemes satisfying the order conditions to higher precision could be designed by taking our time-stepping schemes as initial guesses for local optimizations in higher precision arithmetic. Alternatively, near-optimal schemes that satisfy the order conditions exactly may be sought among schemes with fractional coefficients. One approach to finding such methods is to rewrite the corresponding class of schemes in terms of its parameters (using, for example, a symbolic computation package such as Maple). Assuming

that this step can be achieved, the parameters may be chosen as fractions which give a scheme that closely approximates the optimal scheme, yet satisfies the order conditions exactly. See Gottlieb, Shu and Tadmor [8] for several examples using this technique.

To automate the construction of GAMS models, modifications of Macdonald's Maple scripts [16] for nonnegative coefficient SSPRK schemes were used. All optimizations were carried out on a (shared) cluster of 96 dual 1.2 GHz Athlon processors.

3.5. **BARON parameters.** In all computations, MINOS was chosen as the `NLP` solver and CPLEX [2] was chosen as the linear programming (LP) solver.

We terminate a run when the upper and lower bounds on the global maximum differ by 10^{-10} or less. This is accomplished by setting `epsa` (ϵ_a) $= 10^{-10}$ and `epsr` (ϵ_r) $= 0$. In difficult problems, where a guarantee of optimality is not practical, a CPU time limit is also assumed.

The number of probing problems (`pdo`) [28] had a strong effect on the speed of computation and on the memory requirements. Larger values led to smaller memory requirements. The effect on computational speed was more difficult to predict. We have found that probing on the unsigned Butcher array variables was usually satisfactory when a guarantee of global optimality was sought. The related parameter `pxdo` [28] was found to have a weaker influence. Typically we took `pxdo=pdo` which corresponds to a probing strategy whereby optimization problems were solved over all the probing variables [28]. In difficult problems where CPU time limits the computation, the default value (no probing) worked well.

4. Optimal schemes

We now give new existence and optimality results in the context of effective CFL coefficient. We allow for the possibility of downwind-biased operations. This contrasts with most previous work which focuses on optimizing CFL coefficients for methods with nonnegative coefficients. Following [22, 21], we report our schemes in optimal $\alpha - \beta$ form in this section. Butcher arrays are easily recovered via equation (2.4).

4.1. **Third-order schemes.** As discussed in the introduction, SSP(3,3) and SSP(4,3) are optimal three- and four-stage third-order schemes.

Moving on to methods with five stages and order-three gives a numerically optimized scheme, SSP(5,3), with a CFL coefficient of approximately 2.65. See Table 1. BARON guarantees optimality of this scheme in about 90 minutes.

Considering six stages and order-three gives a numerically optimized scheme, SSP(6,3), with a CFL coefficient of approximately 3.52. See Table 1 for the optimal $\alpha - \beta$ form of this scheme to double precision. BARON guarantees the optimality of this scheme for the general case of unrestricted coefficients in about eight days of CPU time. We remark that the bulk of this computational work is used to verify that there are no nonnegative coefficient schemes that exceed the theoretical efficiency of SSP(6,3).

Moving up to seven stages, it is no longer practical to verify the optimality of the nonnegative coefficient case directly. However, seven-stage schemes which have a time-stepping restriction that agrees with the contractivity bound $R(A, b)$ for linear constant-coefficient problems [14] can be found in less than two seconds during

preprocessing (see, e.g., Table 2). Because the SSP and contractivity restrictions are equivalent [3] and the time-stepping restriction for nonlinear problems cannot exceed that for linear problems, we conclude that SSP(7,3) is an optimal seven-stage, third-order nonnegative coefficient SSPRK scheme. Using BARON, we can verify (in 14 hours) that this scheme exceeds the theoretical efficiency of any scheme involving downwind-biased operators. Thus, SSP(7,3) is an optimal seven-stage, third-order SSPRK scheme.

This approach can also be applied to the eight-stage case to guarantee that the SSP(8,3) scheme shown in Table 2 is an optimal third-order SSPRK scheme. The primarily computational effort comes from verifying that schemes with downwind-biased operators are suboptimal. It takes about five days to check the most computationally intensive case. In this example, Lemma 1 is particularly useful for reducing the number of cases corresponding to the signs of the coefficients. BARON's preprocessing step immediately (i.e., in less than one second) finds a feasible nonnegative coefficient scheme with a CFL coefficient exceeding 5.1, which implies that the CFL coefficient of the optimal scheme must also exceed 5.1. Thus, Lemma 1 indicates that the number of nonnegative levels, ℓ, satisfies $\ell > 5.1$. But the number of nonnegative levels is an integer, so ℓ must satisfy $6 \le \ell \le 8$. Over this range we have $\binom{8}{8} + \binom{8}{7} + \binom{8}{6} = 37$ cases to check, which is a significant reduction over the full $2^8 = 256$ cases that correspond to all possibilities.

4.2. **An optimal fourth-order scheme.** Unfortunately optimality is more difficult to study for fourth-order schemes since the linear contractivity restriction does not provide a sharp bound on the time-stepping restriction for the nonlinear case. This implies that we must resort to applying global optimization software to all cases, including the nonnegative coefficient case.

It was practical to treat the class of five-stage, fourth-order schemes (approximately one day was required to verify the nonnegative coefficient case). Here, BARON guarantees that the SSP(5,4) scheme given in [15, 24] is optimal. See Table 3 for this scheme in optimal $\alpha - \beta$ form (in double precision) or [15] for this scheme in Butcher array form (in quadruple precision).

4.3. **Fifth-order schemes.** We were unable to guarantee optimality of any fifth-order schemes using our approach. Instead we carry out extensive numerical searches with a limited amount of CPU time for each job.

For a large numbers of stages, it is impractical to check all 2^s possibilities. However, extensive numerical searches for seven-, eight-, nine- and ten-stage schemes using one, two and three negative levels were carried out. In all cases we found that for a fixed number of stages the maximal CFL coefficient was a strictly increasing function of the number of nonnegative levels. This fact encouraged us to limit our searches to schemes with precisely one, two or three negative levels. Results based on this approach follow.

For six stages, the best scheme that was found had a CFL coefficient of 0.19 which is too small to be of practical use. Using seven or more stages, however, reasonable CFL coefficients are observed. The best CFL coefficients using 14 hours per job are reported in Table 4. We also report the fraction of total CPU time required to find a feasible solution with a CFL coefficient that is within 1% of the best found. Our formulation under BARON finds good solutions quickly, and we suspect that these schemes may be optimal or nearly optimal.

TABLE 1. The coefficients of the optimal SSPRK (5,3) and SSPRK (6,3) schemes.

Stages $s = 5$					
1.000000000000000					α_{ik}
0.000000000000000	1.000000000000000				
0.355909775063327	0.000000000000000	0.644090224936674			
0.367933791638137	0.000000000000000	0.000000000000000	0.632066208361863		
0.000000000000000	0.000000000000000	0.237593836598569	0.000000000000000	0.762406163401431	
0.377268915331368					β_{ik}
0.000000000000000	0.377268915331368				
0.000000000000000	0.000000000000000	0.242995220537396			
0.000000000000000	0.000000000000000	0.000000000000000	0.238458932846290		
0.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000	0.287632146308408	
CFL number $c = 2.65062919143939$					

Stages $s = 6$						
1.000000000000000						α_{ik}
0.000000000000000	1.000000000000000					
0.000000000000000	0.000000000000000	1.000000000000000				
0.476769811285196	0.098511733286064	0.000000000000000	0.424718455428740			
0.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000	1.000000000000000		
0.000000000000000	0.000000000000000	0.155221702560091	0.000000000000000	0.000000000000000	0.844778297439909	
0.284220721334261						β_{ik}
0.000000000000000	0.284220721334261					
0.000000000000000	0.000000000000000	0.284220721334261				
0.000000000000000	0.000000000000000	0.000000000000000	0.120713785765930			
0.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000	0.284220721334261		
0.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000	0.240103497065900	
CFL number $c = 3.51839230899685$						

TABLE 2. The coefficients of the optimal SSPRK (7,3) and SSPRK (8,3) schemes.

Stages $s = 7$							
1.000000000000000							α_{ik}
0.000000000000000	1.000000000000000						
0.000000000000000	0.000000000000000	1.000000000000000					
0.184962588071072	0.000000000000000	0.000000000000000	0.815037411928928				
0.180718656570380	0.314831034403793	0.000000000000000	0.000000000000000	0.504450309025826			
0.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000	1.000000000000000		
0.000000000000000	0.000000000000000	0.000000000000000	0.120199000000000	0.000000000000000	0.000000000000000	0.879801000000000	
0.233213863663009							β_{ik}
0.000000000000000	0.233213863663009						
0.000000000000000	0.000000000000000	0.233213863663009					
0.000000000000000	0.000000000000000	0.000000000000000	0.190078023865845				
0.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000	0.117644805593912			
0.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000	0.233213863663009		
0.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000	0.205181790464579	
CFL number $c = 4.28790975070412$							

Stages $s = 8$								
1.000000000000000								α_{ik}
0.000000000000000	1.000000000000000							
0.000000000000000	0.000000000000000	1.000000000000000						
0.000000000000000	0.000000000000000	0.000000000000000	1.000000000000000					
0.421366967085359	0.005949401107575	0.000000000000000	0.000000000000000	0.572683631807067				
0.000000000000000	0.004254010666365	0.000000000000000	0.000000000000000	0.000000000000000	0.995745989333635			
0.000000000000000	0.000000000000000	0.104380143093325	0.243265240906726	0.000000000000000	0.000000000000000	0.652354615999950		
0.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000	1.000000000000000	
0.195804015330143								β_{ik}
0.000000000000000	0.195804015330143							
0.000000000000000	0.000000000000000	0.195804015330143						
0.000000000000000	0.000000000000000	0.000000000000000	0.195804015330143					
0.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000	0.112133754621673				
0.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000	0.194971062960412			
0.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000	0.127733653231944		
0.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000	0.195804015330143	
CFL number $c = 5.10714756443533$								

TABLE 3. The coefficients of the optimal SSPRK (5,4) scheme.

Stages			5		
	1				
	0.444370493651235	0.555629506348765			
α_{ik}	0.620101851488403	0	0.379898148511597		
	0.178079954393132	0	0	0.821920045606868	
	0	0	0.517231671970585	0.096059710526147	0.386708617503269
	0.391752226571890				
	0	0.368410593050371			
β_{ik}	0	0	0.251891774271694		
	0	0	0	0.544974750228521	
	0	0	0	0.063692468666290	0.226007483236906
CFL coefficient			1.50818004918983		

TABLE 4. The CFL coefficients of some numerically optimal fifth-order SSPRK schemes. We also report the fraction of total CPU time required to find a feasible solution with a CFL coefficient that is within 1% of the best found.

stages	CFL coefficient	Fraction of CPU time
7	1.1785083484719	0.076
8	1.8756849616413	0.054
9	2.6957882892949	0.00028
10	3.3953368327742	0.00052

TABLE 5. Coefficients of SSP(10,5). CFL coefficient is 3.39533683277420.

(i,k)	α_{ik}	β_{ik}
(1,0)	1	0.173586107937995
(2,0)	0.258168167463650	0
(2,1)	0.741831832536350	0.218485490268790
(3,0)	0	0
(3,1)	0.037493531856076	0.011042654588541
(3,2)	0.962506468143924	0.283478934653295
(4,0)	0.595955269449077	0
(4,1)	0	0
(4,2)	0.404044730550923	0.118999896166647
(4,3)	0	0
(5,0)	0.331848124368345	0.025030881091201
(5,1)	0	0
(5,2)	0	0
(5,3)	0.008466192609453	-0.002493476502164
(5,4)	0.659685683022202	0.194291675763785
(6,0)	0.086976414344414	0
(6,1)	0	0
(6,2)	0	0
(6,3)	0	0
(6,4)	0	0
(6,5)	0.913023585655586	0.268905157462563
(7,0)	0.075863700003186	0
(7,1)	0	0
(7,2)	0.267513039663395	0.066115378914543
(7,3)	0	0
(7,4)	0	0
(7,5)	0	0
(7,6)	0.656623260333419	0.193389726166555
(8,0)	0.005212058095597	0
(8,1)	0	0
(8,2)	0	0
(8,3)	0.407430107306541	-0.119996962708895
(8,4)	0	0
(8,5)	0	0
(8,6)	0	0
(8,7)	0.587357834597862	0.172989562899406
(9,0)	0.122832051947995	0.000000000000035
(9,1)	0	0
(9,2)	0	0
(9,3)	0	0
(9,4)	0	0
(9,5)	0	0
(9,6)	0	0
(9,7)	0	0
(9,8)	0.877167948052005	0.258344898092277
(10,0)	0.075346276482673	0.016982542367506
(10,1)	0.000425904246091	0
(10,2)	0	0
(10,3)	0	0
(10,4)	0	0
(10,5)	0.064038648145995	0.018860764424857
(10,6)	0.354077936287492	0.098896719553054
(10,7)	0	0
(10,8)	0	0
(10,9)	0.506111234837749	0.149060685217562

The seven-, eight- and nine-stage schemes obtained using this approach are reported in Ruuth and Spiteri [20]. The ten-stage scheme is reported in Table 5. Note that the effective CFL coefficient of this scheme is 0.3395, exceeding that of the popular SSPRK scheme SSP(3,3).

We consider this to be a possible reason for further study of downwind-biased spatial discretizations for hyperbolic conservation laws. As we shall see in the next section, another possible reason to use these schemes arises when storage considerations are paramount.

5. Low-storage 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 storage are very desirable (see, e.g., [32]).

It is commonly assumed that two-register low-storage schemes cannot utilize downwind-biased operators without destroying the low-storage property. See, e.g., [8]. However, if we assume that the β's at a level are the same sign, then the low-storage property is preserved since either $F(U^{(j)})$ or $\tilde{F}(U^{(j)})$ appears, but not both. As we shall see, downwind-biased operators provide a means to obtain improved theoretical efficiency in certain low-storage SSPRK schemes.

5.1. **Williamson schemes.** We begin our discussion with SSPRK schemes of Williamson type [32] that require two units of storage per step,[4] although more general methods requiring more storage per step are possible. These schemes take the form

$$dU^{(i)} = A_i dU^{(i-1)} + \Delta t F(U^{(i-1)}), \tag{5.1a}$$

$$U^{(i)} = U^{(i-1)} + B_i dU^{(i)}, \qquad i = 1, 2, \ldots, s, \tag{5.1b}$$

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 numerically optimal s-stage, order-p low-storage SSPRK scheme by Williamson(s,p), and remark that $B_i = a_{i+1,i}, 1 \le i \le s-1$, and $B_s = b_s$ in the usual Butcher notation, so that the B_i are bounded according to Lemma 2. Unfortunately, an analytical bound on the A_i is not known for general schemes of this form. For this reason, we do not expect BARON to be able to guarantee optimality except perhaps in specialized cases where it can determine bounds on the A_i based on the constraints of the problem (see [28, 29] for further details on the construction of bounds based on problem constraints). Nonetheless, improvements to the schemes reported in [24] are possible via our approach.

TABLE 6. The coefficients of the first few numerically optimal Williamson low-storage schemes of order three: Williamson(3,3), Williamson(4,3) and Williamson(5,3). BARON guarantees optimality in the three-stage case.

Stages	A_i	B_i	CFL coefficient
3	0	0.924574112262461	0.322349301195940
	-2.915493957701923	0.287712943868770	
	0	0.626538293270800	
4	0	1.086620745813428	0.634274456962008
	-0.449336503268844	0.854115548251602	
	0	-1.576604558206099	
	-4.661555711601366	-0.278475500113052	
5	0	0.713497331193829	1.40154693827206
	-4.344339134485095	0.133505249805329	
	0	0.713497331193829	
	-3.770024161386381	0.149579395628565	
	-0.046347284573284	0.384471116121269	

[4]Note that if some form of error control is required, then additional storage for the current solution vector is also needed [13].

First, we remark that in the first-order case SSP(s,1) is the optimal scheme since this is the provably optimal first-order, s-stage scheme and it may be written in low-storage form with two registers. Moving on to second order, it is clear that a traditional implementation of any two-stage scheme must be low-storage in the sense we are considering, so the optimal low-storage method with $s = p = 2$ corresponds to SSP(2,2) [24]. The SSP(2,2) scheme has an effective CFL coefficient of 1/2. Extensive searching with more stages ($s = 3, 4, 5, 6$) did not find any schemes with improved effective CFL coefficients, and we conjecture that an optimal scheme with an even number of stages is just SSP(2,2) repeated ($s/2$) times:

$$\begin{aligned} dU^{(i-1)} &= \Delta t F(U^{(i-2)}), \\ U^{(i-1)} &= U^{(i-2)} + (2/s)\, dU^{(i-1)}, \\ dU^{(i)} &= -dU^{(i-1)} + \Delta t F(U^{(i-1)}), \\ U^{(i)} &= U^{(i-1)} + (1/s)\, dU^{(i)}, \qquad i = 2, 4, 6, \ldots, s, \end{aligned}$$

where $U^{(0)} = U^n$, $U^{n+1} = U^{(s)}$. We further remark that all our numerical optimizations for low-storage schemes with an odd number of stages lead to schemes with effective CFL coefficients that were strictly less than 1/2. Thus, we conjecture that low-storage schemes of this type and with an odd number of stages have smaller effective CFL coefficients than the simple SSP(2,2) scheme.

The results for the coefficients A_i, B_i are given in Table 6 for up to five stages and order three. We note that the optimal three-stage, order-three, low-storage method reported in Table 6 agrees with that reported in [7]. In this particular case, BARON is able to use the constraints of the problem to automatically construct bounds on the A_i, and subsequently guarantee the optimality of the scheme. In the more complicated four- and five-stage cases, the software was unable to determine such bounds, and a guarantee of optimality was not obtained.

Using four stages, the third-order numerically optimal scheme involved two downwind-biased operators per step and had a CFL coefficient of about 0.634. This represents a 20% improvement over the nonnegative coefficient scheme reported in [24]. For completeness, Table 7 supplies the coefficients of that nonnegative coefficient scheme to 15 digits.

Similar to [24], we find that the optimal five-stage scheme does not require downwind-biased operators. However, here we find a numerically optimized scheme with a 40% larger CFL coefficient. See Table 6.

TABLE 7. The coefficients of the four-stage Williamson low-storage scheme of order three having nonnegative coefficients. This result comes from extensive searching.

Stages	A_i	B_i	CFL coefficient
4	0	1.032161930751755	0.528418106518184
	-4.946517279341980	0.187941555751458	
	0	0.152152605134959	
	-0.151274934922161	0.656749852605931	

5.2. **Van der Houwen and Wray schemes.** Van der Houwen and Wray devised another class of low-storage schemes which alternate information between two available storage registers at each successive stage [30, 31, 33]. See Kennedy, Carpenter and Lewis [13] for details. Starting with $X^{(j)}$ and $F(U^{(j)})$ stored in Registers 1 and 2 respectively, two intermediate stages would proceed according to

$$\begin{aligned}
&\text{(5.3a)} \quad \text{(Register 1) } U^{(i+1)} &&= X^{(i)} + a_{i+1,i}\Delta t F(U^{(i)}),\\
&\text{(5.3b)} \quad \text{(Register 2) } X^{(i+1)} &&= U^{(i+1)} + (b_i - a_{i+1,i})\Delta t F(U^{(i)}),\\
&\text{(5.3c)} \quad \text{(Register 2) } U^{(i+2)} &&= X^{(i+1)} + a_{i+2,i+1}\Delta t F(U^{(i+1)}),\\
&\text{(5.3d)} \quad \text{(Register 1) } X^{(i+2)} &&= U^{(i+2)} + (b_{i+1} - a_{i+2,i+1})\Delta t F(U^{(i+1)}),
\end{aligned}$$

where a_{ij} and b_j are the usual Butcher array entries [13]. By overwriting, the three vectors U, F, and X never fully coexist [13]. In particular, during the function evaluation, the previous solution vector is overwritten. While this will not be acceptable in all situations, compressible Navier-Stokes equations provide a situation where this may be profitably utilized [13]. Full details on implementing these schemes are given in the comprehensive article of Kennedy, Carpenter and Lewis [13].

To distinguish these methods from Williamson schemes, we shall refer to them as vdH schemes (cf. [13]). Note that these schemes are easily generalized to accommodate more than two registers of storage. We shall consider the cases where two or three registers of storage are available, and for simplicity refer to the numerically optimal r-register, s-stage, pth-order scheme as vdHr(s,p), and the corresponding numerically optimal nonnegative coefficient scheme as vdHr+(s,p).

TABLE 8. The coefficients of the first few numerically optimal vdH low-storage schemes of order three with two registers of storage: vdH2(3,3), vdH2(4,3) and vdH2(5,3). BARON guarantees optimality in all cases.

Stages	$a_{i+1,i}$	b_i	CFL coefficient
3	0.755726313669390	0.245170292105110	0.838384821388215
	0.386954492646558	0.184896041116058	
		0.569933666778832	
4	0.410502506371045	0.222722477423144	1.067414323404809
	0.508294264771036	0.167687843505189	
	0.309067503393721	0.151218171982708	
		0.458371507088958	
5	0.674381436593749	0.174481959220521	1.482840341885634
	0.116638367147961	0.116638367147961	
	0.674381436593749	0.162995387938952	
	0.162995387938952	0.106256369067643	
		0.439627916624922	

5.2.1. *Two registers of storage.* A two-register, s-stage vdH Runge-Kutta scheme is specified by the Butcher array entries $a_{i+1,i}, 1 \le i \le s-1$, and $b_i, 1 \le i \le s$. The remaining a_{ij} entries are defined according to $a_{ij} = b_j$, so the Butcher array takes the particularly simple form [13]

$$\begin{array}{c|ccccccc}
0 & & & & & & & \\
c_2 & a_{21} & & & & & & \\
c_3 & b_1 & a_{32} & & & & & \\
\vdots & \vdots & b_2 & a_{43} & & & & \\
 & & \vdots & b_3 & a_{54} & & & \\
 & & & \vdots & \ddots & \ddots & & \\
c_s & & & & & b_{s-2} & a_{a,s-1} & \\
\hline
 & b_1 & b_2 & b_3 & \cdots & b_{s-2} & b_{s-1} & b_s
\end{array}$$

Because each of the entries is bounded according to Lemma 2, we may directly apply the algorithms of Section 3 to derive optimal schemes.

Similar to the case of Williamson low-storage schemes, we know that SSP(s,1) is the optimal first-order, s-stage vdH SSPRK scheme and that SSP(2,2) is the optimal second-order, two-stage vdH SSPRK scheme. For order two, BARON guarantees that three- and five-stage schemes both have effective CFL coefficients that are less than 0.5, and we conjecture that *any* scheme with an odd number of stages will have a smaller effective CFL coefficient than SSP(2,2). For four- and six-stage schemes and order-two BARON guarantees that an optimal scheme is just SSP(2,2) repeated $(s/2)$ times. We conjecture that this also holds for any even number of stages.

The results for up to five stages and order three are given in Table 8. The three-stage scheme is guaranteed optimal by BARON and agrees to single precision with the RK3(2)3[2R+]N contractive scheme reported in [13]. We also report new four- and five-stage stage schemes which are also guaranteed to be optimal. It is interesting to note that in each case these schemes have improved CFL coefficients over the corresponding Williamson schemes.

5.2.2. *Three registers of storage.* A three-register, s-stage vdH SSPRK scheme is specified by the Butcher array entries $a_{i+1,i}, 1 \le i \le s-1$, $a_{i+2,i}, 1 \le i \le s-2$, and $b_i, 1 \le i \le s$. Similar to the two-register case, the remaining a_{ij} entries are simply

TABLE 9. The coefficients of the optimal five-stage vdH low-storage scheme of order three with three registers of storage: vdH3(5,3). BARON guarantees optimality.

Stages	$a_{i+1,i}$	$a_{i+2,i}$	b_i	CFL coefficient
5	0.390109487215461	0.390109487215461	0.226019654979362	2.56338292907932
	0.390109487215461	0.226019654979362	0.120298131155176	
	0.226019654979362	0.120298131155176	0.109515534018255	
	0.207634341633736		0.189023599836438	
			0.355143080010770	

defined according to $a_{ij} = b_j$, and the Butcher array takes the form [13]

$$\begin{array}{c|ccccccc}
0 & & & & & & & \\
c_2 & a_{21} & & & & & & \\
c_3 & a_{31} & a_{32} & & & & & \\
\vdots & b_1 & a_{42} & a_{43} & & & & \\
 & \vdots & b_2 & a_{53} & a_{54} & & & \\
 & & \vdots & \ddots & \ddots & \ddots & & \\
c_s & & & & b_{s-3} & a_{s,s-2} & a_{s,s-1} & \\
\hline
 & b_1 & b_2 & \cdots & b_{s-3} & b_{s-2} & b_{s-1} & b_s
\end{array}$$

which allows us to apply the algorithms of Section 3 to derive optimal schemes.

It is clear that the optimal SSP(3,3) and SSP(4,3) schemes fit within this class of schemes, so they must be optimal three-register vdH SSPRK schemes [13]. Moving on to five stages, we find a new five-stage scheme which BARON guarantees to be optimal (see Table 9). It is noteworthy that the CFL coefficient for this scheme is only about 3% smaller than the optimal SSP(5,3) scheme.

Moving up to fourth order and five stages proved more computationally intensive. Nonetheless, BARON was able to guarantee that the optimal scheme (see Table 10) involved one downwind-biased operator and had a CFL coefficient of 0.935. It took 12 days of CPU time for BARON to verify that schemes involving only nonnegative coefficients must have smaller CFL coefficients. The best nonnegative coefficient scheme (see Table 11) had a CFL coefficient of 0.531, an 11% improvement over the scheme found in [13]. The optimality of this scheme was guaranteed by BARON in 21 days of CPU time.

TABLE 10. The coefficients of the optimal five-stage vdH low-storage scheme of order four with three registers of storage: vdH3(5,4). BARON guarantees optimality.

Stages	$a_{i+1,i}$	$a_{i+2,i}$	b_i	CFL coefficient
5	0.537734210467782	0.000000000000000	0.299925395513371	0.935322006941531
	0.000000000000000	0.596326927126672	0.449146588599927	
	-0.472823582086688	-0.545978886296898	-0.137488645434953	
	0.796906902183600		0.225363552896202	
			0.163053108425452	

TABLE 11. The coefficients of the optimal five-stage vdH low-storage scheme of order four with three registers of storage and nonnegative coefficients: vdH3+(5,4). BARON guarantees optimality.

Stages	$a_{i+1,i}$	$a_{i+2,i}$	b_i	CFL coefficient
5	0.216747619157064	0.059060301553258	0.049733200550301	0.530770344137093
	0.513374951629630	0.113274666138869	0.370241294854764	
	0.415710952208246	0.080866763251240	0.051983506105255	
	0.366498283222966		0.235595750064777	
			0.292446248424903	

6. Conclusions

We have studied explicit high-order strong-stability-preserving Runge-Kutta methods with downwind-biased spatial discretizations. We have developed a practical approach to guarantee the optimality of a variety of schemes of up to order four and have applied the technique to guarantee the optimality of SSP(5,3), SSP(5,4) as well as several new third-order schemes: SSP(6,3), SSP(7,3) and SSP(8,3). We also find an efficient fifth-order scheme that has an effective CFL coefficient that exceeds the popular SSP(3,3) scheme.

Several new results for low-storage schemes were found. Our approach guarantees the optimality of the known Williamson(3,3) and vdH2(3,3) schemes. It also guarantees the optimality of the third-order low-storage schemes vdH2(4,3) and vdH2(5,3). Interestingly, we found that in two instances, Williamson(4,3) and vdH3(5,4), significantly improved schemes arise when downwind-biased operators are utilized, and we find and guarantee the optimality of the corresponding schemes. Our approach also derives two nonnegative coefficient schemes, vdH3+(5,4) and Williamson(5,3), which are more efficient than the corresponding schemes that were previously proposed in the literature. These results demonstrate that global branch-and-bound software may be applied to our mathematical formulation to achieve a practical, constructive way of guaranteeing the optimality of SSPRK schemes.

This manuscript has considered the optimization of explicit SSPRK schemes. However, we expect that general implicit schemes, e.g.,

$$U^{(0)} = U^n, \tag{6.1a}$$

$$U^{(i)} = \sum_{k=0}^{s} \Big(\alpha_{ik} U^{(k)} + \Delta t \max(\beta_{ik}, 0) F(U^{(k)}) \tag{6.1b}$$

$$+ \Delta t \min(\beta_{ik}, 0) \tilde{F}(U^{(k)}) \Big), \qquad i = 1, 2, \ldots, s, \tag{6.1c}$$

$$U^{n+1} = U^{(s)}, \tag{6.1d}$$

could also be treated using the techniques outlined in this paper. To proceed, assume Euler's method applied forward in time combined with the spatial discretization $F(\cdot)$ is strongly stable under the CFL restriction $\Delta t \le \Delta t_{FE}$. Also assume Euler's method applied backward in time combined with the spatial discretization $\tilde{F}(\cdot)$ is strongly stable under the same CFL restriction $\Delta t \le \Delta t_{FE}$. Then the implicit (or backward) Euler method applied to these problems is unconditionally stable [11], and it easily seen that the method (6.1) will be SSP provided $\Delta t \le C \Delta t_{FE}$, where C is the CFL coefficient

$$C \equiv \min \{ c_{ik} : 1 \le i \le s,\ 0 \le k \le s \}, \text{ where } c_{ik} = \begin{cases} \frac{\alpha_{ik}}{|\beta_{ik}|} & \text{if } \beta_{ik} \ne 0 \text{ and } i \ne k, \\ \infty & \text{otherwise.} \end{cases}$$

The development of appropriate optimization techniques then follows in a similar manner to the techniques described in this paper. See also [3] for relevant discussions on the underlying theory for general implicit SSPRK schemes with nonnegative coefficients.

Nonlinearly stable explicit multistep methods with guaranteed optimal CFL coefficients have been derived using the ideas described in this article [18], and we believe the derivation of optimal general linear methods (cf. [21, 8]) is another

natural application area for these techniques. Ultimately, one might consider developing techniques to guarantee optimality over a combination of properties such as error constant, nonlinear stability and linear stability (cf. [13, 16]). On the other hand, our approach is not suitable to show the nonexistence of a scheme, since as the CFL coefficient tends to zero, we no longer enjoy the property that our intervals are bounded. Similarly, certain schemes such as the Williamson class of low-storage schemes include parameters which are not obviously bounded. This typically prohibits the use of our approach to guarantee optimality.

Nonetheless, as software and hardware advances take place and improved bounds on variables are derived, we expect that the utility of the techniques outlined in this paper for optimal explicit SSPRK schemes will experience a correspondingly rapid growth in applicability and importance.

Acknowledgments

The author thanks Colin Macdonald, Nick Sahinidis, and Luis Vicente for helpful discussions related to this project.

References

1. A. Brooke, D. Kendrick, A. Meeraus, and R. Raman, *Gams-a users guide*, GAMS Development Corporation, Washington, 1998.
2. *Gams-the solver manuals*, GAMS Development Corporation, Washington, 2001.
3. L. Ferracina and M. N. Spijker, *An extension and analysis of the Shu-Osher representation of Runge-Kutta methods*, Math. Comp. **74** (2005), 201–219. MR2085408
4. ———, *Stepsize restrictions for the total-variation-diminishing property in general Runge-Kutta methods*, SIAM J. Numer. Anal. **42** (2004), no. 3, 1073–1093. MR2113676
5. S. Gottlieb and L. J. Gottlieb, *Strong stability preserving properties of Runge-Kutta time discretization methods for linear constant coefficient operators*, J. Scientific Computation **18** (2003), no. 1, 83–109. MR1958936 (2003m:65161)
6. P. E. Gill, W. Murray, and M. H. Wright, *Practical optimization*, Academic Press, San Diego, 1981. MR0634376 (83d:65195)
7. S. Gottlieb and C.-W. Shu, *Total variation diminishing Runge-Kutta schemes*, Math. Comput. **67** (1998), no. 221, 73–85. MR1443118 (98c:65122)
8. S. Gottlieb, C.-W. Shu, and E. Tadmor, *Strong-stability-preserving high-order time discretization methods*, SIAM Review **43** (2001), no. 1, 89–112. MR1854647 (2002f:65132)
9. I. Higueras, *Representations of Runge-Kutta methods and strong stability preserving methods*, Technical Report No. 2, Departamento de Matematica e Informatica, Universidad Publica de Navarra, 2003.
10. ———, *On strong stability preserving time discretization methods*, J. Scientific Computation **21** (2004), no. 2, 193–223. MR2069949 (2005d:65112)
11. W. Hundsdorfer, S. J. Ruuth, and R. J. Spiteri, *Monotonicity-preserving linear multistep methods*, SIAM J. Numer. Anal. **41** (2003), no. 2, 605–623. MR2004190 (2004g:65093)
12. R. Horst and H. Tuy, *Global optimization: Deterministic approaches*, third ed., Springer Verlag, Berlin, 1996. MR1274246 (94m:90004)
13. C. A. Kennedy, M. H. Carpenter, and R. M. Lewis, *Low-storage, explicit Runge-Kutta schemes for the compressible Navier-Stokes equations*, Appl. Numer. Math. **35** (2000), no. 3, 177–219. MR1793508 (2001k:65111)
14. J. F. B. M. Kraaijevanger, *Absolute monotonicity of polynomials occurring in the numerical solution of initial value problems*, Numer. Math. **48** (1986), 303–322. MR0826471 (87c:65084)
15. J.F.B.M. Kraaijevanger, *Contractivity of Runge-Kutta methods*, BIT **31** (1991), 482–528. MR1127488 (92i:65120)
16. C. B. Macdonald, *High-order embedded Runge-Kutta pairs for the time evolution of hyperbolic conservation laws*, Master's thesis, Simon Fraser University, Burnaby, BC, Canada, 2003.
17. B. A. Murtagh and M. A. Saunders, *MINOS 5.1 User's Guide*, Report SOL 83-20R, Department of Operations Research, Stanford University, 1987.

18. S. J. Ruuth and W. Hundsdorfer, *High-order linear multistep methods with general monotonicity and boundedness properties*, J. Comput. Phys. **209** (2005), no. 1, 226–248. MR2145787
19. S. J. Ruuth and R. J. Spiteri, *Two barriers on strong-stability-preserving time discretization methods*, J. Scientific Computation **17** (2002), no. 4, 211–220. MR1910562
20. Steven J. Ruuth and Raymond J. Spiteri, *High-order strong-stability-preserving Runge–Kutta methods with downwind-biased spatial discretizations*, SIAM J. Numer. Anal. **42** (2004), no. 3, 974–996. MR2112790
21. Chi-Wang Shu, *Total-variation-diminishing time discretizations*, SIAM J. Sci. Statist. Comput. **9** (1988), no. 6, 1073–1084. MR0963855 (90a:65196)
22. Chi-Wang Shu and Stanley Osher, *Efficient implementation of essentially nonoscillatory shock-capturing schemes*, J. Comput. Phys. **77** (1988), no. 2, 439–471. MR0954915 (89g:65113)
23. M. N. Spijker, *Contractivity in the numerical solution of initial value problems*, Numer. Math. **42** (1983), 271–290. MR0723625 (85b:65067)
24. Raymond J. Spiteri and Steven J. Ruuth, *A new class of optimal high-order strong-stability-preserving time-stepping schemes*, SIAM J. Numer. Anal. **40** (2002), no. 2, 469–491. MR1921666 (2003g:65083)
25. ———, *Nonlinear evolution using optimal fourth-order strong-stability-preserving Runge-Kutta methods*, Mathematics and Computers in Simulation **62** (2003), nos. 1–2, 125–135, Special issue on "Nonlinear Waves: Computation and Theory II". MR1983581
26. N. V. Sahinidis and M. Tawarmalani, GAMS The Solver Manuals, GAMS Development Corporation, Washington, 2004, pp. 9–20.
27. M. Tawarmalani, S. Ahmed, and N. V. Sahinidis, *Product disaggregation in global optimization and relaxations of rational programs*, Optimization and Engineering **3** (2002), 281–303. MR1955959 (2003k:90051)
28. M. Tawarmalani and N. V. Sahinidis, *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications*, Nonconvex Optimization and Its Applications, vol. 65, Kluwer Academic Publishers, Dordrecht, 2002. MR1961018 (2004a:90001)
29. M. Tawarmalani and N. V. Sahinidis, *Global optimization of mixed-integer nonlinear programs: A theoretical and computational study*, Mathematical Programming **99** (2004), 563–591. MR2051712 (2004m:90096)
30. P.J. van der Houwen, *Explicit Runge-Kutta formulas with increased stability boundaries*, Numer. Math. **20** (1972), no. 2, 149–164. MR0317547 (47:6094)
31. ———, *Construction of integration formulas for initial value problems*, North-Holland, Amsterdam, 1977. MR0519726 (58:24960)
32. J. H. Williamson, *Low-storage Runge-Kutta schemes*, J. Comput. Phys. **35** (1980), no. 1, 48–56. MR81a:65070 MR0566473 (81a:65070)
33. A.A. Wray, *Minimal storage time advancement schemes for spectral methods*, Tech. report, NASA Ames Research Center, Moffett Field, CA, 1986.

Department of Mathematics and Statistics, Simon Fraser University, Burnaby, British Columbia, Canada V5A 1S6

E-mail address: sruuth@sfu.ca