Available online at www.sciencedirect.com
ScienceDirect
Applied Numerical Mathematics 59 (2009) 373–392
APPLIED NUMERICAL MATHEMATICS
www.elsevier.com/locate/apnum

Optimal implicit strong stability preserving Runge–Kutta methods

David I. Ketcheson [a,*,1], Colin B. Macdonald [b,2], Sigal Gottlieb [c,3]

[a] *Department of Applied Mathematics, University of Washington, Seattle, WA 98195-2420, USA*
[b] *Department of Mathematics, Simon Fraser University, Burnaby, British Columbia, V5A 1S6, Canada*
[c] *Department of Mathematics, University of Massachusetts Dartmouth, North Dartmouth, MA 02747, USA*

Available online 1 April 2008

Abstract

Strong stability preserving (SSP) time discretizations were developed for use with spatial discretizations of partial differential equations that are strongly stable under forward Euler time integration. SSP methods preserve convex boundedness and contractivity properties satisfied by forward Euler, under a modified timestep restriction. We turn to implicit Runge–Kutta methods to alleviate this timestep restriction, and present implicit SSP Runge–Kutta methods which are optimal in the sense that they preserve convex boundedness properties under the largest timestep possible among all methods with a given number of stages and order of accuracy. We consider methods up to order six (the maximal order of SSP Runge–Kutta methods) and up to eleven stages. The numerically optimal methods found are all diagonally implicit, leading us to conjecture that optimal implicit SSP Runge–Kutta methods are diagonally implicit. These methods allow a larger SSP timestep, compared to explicit methods of the same order and number of stages. Numerical tests verify the order and the SSP property of the methods.

Keywords: Strong stability preserving; Runge–Kutta methods; Time discretization

1. Strong stability preserving Runge–Kutta methods

Strong stability preserving (SSP) Runge–Kutta methods are high-order time discretization methods that preserve the strong stability properties—in any norm or semi-norm—satisfied by a spatial discretization of a system of partial differential equations (PDEs) coupled with first-order forward Euler timestepping [30,28,9,10]. These methods were originally developed for the numerical solution of hyperbolic PDEs to preserve the total variation diminishing property satisfied by specially designed spatial discretizations coupled with forward Euler integration.

In this work we are interested in approximating the solution of the ODE

$$\boldsymbol{u}_t = F(\boldsymbol{u}), \tag{1}$$

* Corresponding author.
E-mail addresses: ketch@amath.washington.edu (D.I. Ketcheson), cbm@sfu.ca (C.B. Macdonald).

[1] The work of this author was funded by a U.S. Dept. of Energy Computational Science Graduate Fellowship under grant number DE-FG02-97ER25308.
[2] The work of this author was partially supported by a grant from NSERC Canada and a scholarship from the Pacific Institute for the Mathematical Sciences (PIMS).
[3] This work was supported by AFOSR grant number FA9550-06-1-0255.

0168-9274/$30.00
doi:10.1016/j.apnum.2008.03.034

arising from the discretization of the spatial derivatives in the PDE

$$u_t + f(u, u_x, u_{xx}, \ldots) = 0,$$

where the spatial discretization $F(\boldsymbol{u})$ is chosen so that the solution obtained using the forward Euler method

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

satisfies the monotonicity requirement

$$\|\boldsymbol{u}^{n+1}\| \leqslant \|\boldsymbol{u}^n\|,$$

in some norm, semi-norm or convex functional $\|\cdot\|$, for a suitably restricted timestep

$$\Delta t \leqslant \Delta t_{\mathrm{FE}}.$$

If we write an explicit Runge–Kutta method in the now-standard Shu–Osher form [30]

$$\begin{aligned}
&\boldsymbol{u}^{(0)} = \boldsymbol{u}^n,\\
&\boldsymbol{u}^{(i)} = \sum_{k=0}^{i-1} \big(\alpha_{i,k}\boldsymbol{u}^{(k)} + \Delta t \beta_{i,k} F(\boldsymbol{u}^{(k)})\big), \quad \alpha_{i,k} \geqslant 0,\ i = 1, \ldots, s,\\
&\boldsymbol{u}^{n+1} = \boldsymbol{u}^{(s)},
\end{aligned} \tag{3}$$

then consistency requires that $\sum_{k=0}^{i-1} \alpha_{i,k} = 1$. Thus, if $\alpha_{i,k} \geqslant 0$ and $\beta_{i,k} \geqslant 0$, all the intermediate stages $\boldsymbol{u}^{(i)}$ in (3) are simply convex combinations of forward Euler steps, each with Δt replaced by $\frac{\beta_{i,k}}{\alpha_{i,k}}\Delta t$. Therefore, any bound on a norm, semi-norm, or convex functional of the solution that is satisfied by the forward Euler method will be preserved by the Runge–Kutta method, under the timestep restriction $\frac{\beta_{i,k}}{\alpha_{i,k}}\Delta t \leqslant \Delta t_{\mathrm{FE}}$, or equivalently

$$\Delta t \leqslant \min \frac{\alpha_{i,k}}{\beta_{i,k}} \Delta t_{\mathrm{FE}}, \tag{4}$$

where the minimum is taken over all $k < i$ and $\beta_{i,k} \neq 0$.

These explicit SSP time discretizations can then be safely used with any spatial discretization that satisfies the required stability property when coupled with forward Euler.

Definition 1 *(Strong stability preserving (SSP)).* For $\Delta t_{\mathrm{FE}} > 0$, let $\mathcal{F}(\Delta t_{\mathrm{FE}})$ denote the set of all pairs $(F, \|\cdot\|)$ where the function $F : \mathbb{R}^m \to \mathbb{R}^m$ and convex functional $\|\cdot\|$ are such that the numerical solution obtained by the forward Euler method (2) satisfies $\|\boldsymbol{u}^{n+1}\| \leqslant \|\boldsymbol{u}^n\|$ whenever $\Delta t \leqslant \Delta t_{\mathrm{FE}}$. Given a Runge–Kutta method, the *SSP coefficient* of the method is the largest constant $c \geqslant 0$ such that, for all $(F, \|\cdot\|) \in \mathcal{F}(\Delta t_{\mathrm{FE}})$, the numerical solution given by (3) satisfies $\|\boldsymbol{u}^{(i)}\| \leqslant \|\boldsymbol{u}^n\|$ (for $1 \leqslant i \leqslant s$) whenever

$$\Delta t \leqslant c \Delta t_{\mathrm{FE}}. \tag{5}$$

If $c > 0$, the method is said to be *strong stability preserving* under the maximal timestep restriction (5).

A numerical method is said to be contractive if, for any two numerical solutions $\boldsymbol{u}, \boldsymbol{v}$ of (1) it holds that

$$\|\boldsymbol{u}^{n+1} - \boldsymbol{v}^{n+1}\| \leqslant \|\boldsymbol{u}^n - \boldsymbol{v}^n\|. \tag{6}$$

Strong stability preserving methods are also of interest from the point of view of preserving contractivity; in [4] it was shown that the SSP coefficient is equal to the radius of absolute monotonicity, which was shown in [22] to be the method-dependent factor in determining the largest timestep for contractivity preservation.

If a particular spatial discretization coupled with the explicit forward Euler method satisfies a strong stability property under some timestep restriction, then the implicit backward Euler method satisfies the same strong stability property, for any positive timestep [14]. However, all SSP Runge–Kutta methods of order greater than one suffer from some timestep restriction [10]. Much of the research in this field is devoted to finding methods that are optimal in terms of their timestep restriction. For this purpose, various implicit extensions and generalizations of the Shu–Osher form have been introduced [10,8,6,14]. The most general of these, and the form we use in this paper, was introduced independently in [6] and [14]. We will refer to it as the *modified Shu–Osher form*.

The necessity of a finite timestep restriction for strong stability preservation applies not only to Runge–Kutta methods, but also to linear multistep and all general linear methods [31]. Diagonally split Runge–Kutta methods lie outside this class and can be unconditionally contractive, but yield poor accuracy for semi-discretizations of PDEs when used with large timesteps [25].

Optimal implicit SSP Runge–Kutta methods with up to two stages were found in [16]. Recently, this topic was also studied by Ferracina & Spijker [7]; in that work, attention was restricted to the smaller class of singly diagonally implicit Runge–Kutta (SDIRK) methods. They present optimal SDIRK methods of order $p = 1$ with any number of stages, order $p = 2$ with two stages, order $p = 3$ with two stages, and order $p = 4$ with three stages. They find numerically optimal methods of orders two to four and up to eight stages. Based on these results, they conjecture the form of optimal SDIRK methods for second- and third-order and any number of stages.

In this work we consider the larger class of all Runge–Kutta methods, with up to eleven stages and sixth-order accuracy. Our search for new SSP methods is facilitated by known results on contractivity and absolute monotonicity of Runge–Kutta methods [31,22] and their connection to strong stability preservation [13,14,4,6]. For a more detailed description of the Shu–Osher form and the SSP property, we refer the interested reader to [30,9,10,29,8,20].

The structure of this paper is as follows. In Section 2 we use results from contractivity theory to determine order barriers and other limitations on implicit SSP Runge–Kutta methods. In Section 3, we present new numerically optimal implicit Runge–Kutta methods of up to sixth order and up to eleven stages, found by numerical optimization. A few of these numerically optimal methods are also proved to be truly optimal. We note that the numerically optimal implicit Runge–Kutta methods are all diagonally implicit, and those of order two and three are singly diagonally implicit. In Section 4 we present numerical experiments using the numerically optimal implicit Runge–Kutta methods, with a focus on verifying order of accuracy and the SSP timestep limit. Finally, in Section 5 we summarize our results and discuss future directions.

2. Barriers and limitations on SSP methods

The theory of strong stability preserving Runge–Kutta methods is very closely related to the concepts of *absolute monotonicity* and *contractivity* [13,14,4,6,16]. In this section we review this connection and collect some results on absolutely monotonic Runge–Kutta methods that allow us to draw conclusions about the class of implicit SSP Runge–Kutta methods. To facilitate the discussion, we first review two representations of Runge–Kutta methods.

2.1. Representations of Runge–Kutta methods

An s-stage Runge–Kutta method is usually represented by its Butcher tableau, consisting of an $s \times s$ matrix $\mathbf{A}$ and two $s \times 1$ vectors $\boldsymbol{b}$ and $\boldsymbol{c}$. The Runge–Kutta method defined by these arrays is

$$\boldsymbol{y}^i = \boldsymbol{u}^n + \Delta t \sum_{j=1}^{s} a_{ij} F(t_n + c_j \Delta t, \boldsymbol{y}^j), \quad 1 \leqslant i \leqslant s, \tag{7a}$$

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + \Delta t \sum_{j=1}^{s} b_j F(t_n + c_j \Delta t, \boldsymbol{y}^j). \tag{7b}$$

It is convenient to define the $(s+1) \times s$ matrix

$$\mathbf{K} = \begin{pmatrix} \mathbf{A} \\ \boldsymbol{b}^{\mathrm{T}} \end{pmatrix},$$

and we will also make the standard assumption $c_i = \sum_{j=1}^{s} a_{ij}$. For the method (7) to be accurate to order p, the coefficients $\mathbf{K}$ must satisfy order conditions (see, e.g., [11]) denoted here by $\Phi_p(\mathbf{K}) = 0$.

A generalization of the Shu–Osher form (3) that applies to implicit as well as explicit methods was introduced in [6,14] to more easily study the SSP property. We will refer to this formulation as the *modified Shu–Osher form*. Following the notation of [6], we introduce the coefficient matrices

$$\boldsymbol{\lambda} = \begin{bmatrix} \boldsymbol{\lambda}_0 \\ \boldsymbol{\lambda}_1 \end{bmatrix}, \quad \boldsymbol{\lambda}_0 = \begin{bmatrix} \lambda_{11} & \cdots & \lambda_{1s} \\ \vdots & & \vdots \\ \lambda_{s1} & \cdots & \lambda_{ss} \end{bmatrix}, \quad \boldsymbol{\lambda}_1 = (\lambda_{s+1,1}, \ldots, \lambda_{s+1,s}), \tag{8a}$$

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_0 \\ \boldsymbol{\mu}_1 \end{bmatrix}, \quad \boldsymbol{\mu}_0 = \begin{bmatrix} \mu_{11} & \cdots & \mu_{1s} \\ \vdots & & \vdots \\ \mu_{s1} & \cdots & \mu_{ss} \end{bmatrix}, \quad \boldsymbol{\mu}_1 = (\mu_{s+1,1}, \ldots, \mu_{s+1,s}). \tag{8b}$$

These arrays define the method

$$\boldsymbol{y}^i = \left(1 - \sum_{j=1}^{s} \lambda_{ij}\right)\boldsymbol{u}^n + \sum_{j=1}^{s} \lambda_{ij}\boldsymbol{y}^j + \Delta t \mu_{ij} F(t_n + c_j \Delta t, \boldsymbol{y}^j) \quad (1 \leqslant i \leqslant s), \tag{9a}$$

$$\boldsymbol{u}^{n+1} = \left(1 - \sum_{j=1}^{s} \lambda_{s+1,j}\right)\boldsymbol{u}^n + \sum_{j=1}^{s} \lambda_{s+1,j}\boldsymbol{y}^j + \Delta t \mu_{s+1,j} F(t_n + c_j \Delta t, \boldsymbol{y}^j). \tag{9b}$$

Comparison of the Butcher representation (7) with the modified Shu–Osher representation (9) reveals that the two are related by

$$\boldsymbol{\mu} = \mathbf{K} - \boldsymbol{\lambda}\mathbf{A}. \tag{10}$$

Hence the Butcher form can be obtained explicitly from the modified Shu–Osher form:

$$\mathbf{A} = (\mathbf{I} - \boldsymbol{\lambda}_0)^{-1}\boldsymbol{\mu}_0,$$
$$\boldsymbol{b}^{\mathrm{T}} = \boldsymbol{\mu}_1 + \boldsymbol{\lambda}_1(\mathbf{I} - \boldsymbol{\lambda}_0)^{-1}\boldsymbol{\mu}_0.$$

Note that the modified Shu–Osher representation is not unique for a given Runge–Kutta method. One particular choice, $\boldsymbol{\lambda} = 0$ yields $\mathbf{K} = \boldsymbol{\mu}$; i.e. the Butcher form is a special case of the modified Shu–Osher form.

2.2. Strong stability preservation

The SSP coefficient turns out to be related to the *radius of absolute monotonicity* $R(\mathbf{K})$, introduced originally by Kraaijevanger [22]. This relationship was proved in [4,13], where also a more convenient, equivalent definition of $R(\mathbf{K})$ was given:

Definition 2 *(Radius of absolute monotonicity (of a Runge–Kutta method)).* The radius of absolute monotonicity $R(\mathbf{K})$ of the Runge–Kutta method defined by Butcher array $\mathbf{K}$ is the largest value of $r \geqslant 0$ such that $(\mathbf{I} + r\mathbf{A})^{-1}$ exists and

$$\mathbf{K}(\mathbf{I} + r\mathbf{A})^{-1} \geqslant 0,$$
$$r\mathbf{K}(\mathbf{I} + r\mathbf{A})^{-1}\boldsymbol{e}_s \leqslant \boldsymbol{e}_{s+1}.$$

Here, the inequalities are understood component-wise and $\boldsymbol{e}_s$ denotes the $s \times 1$ vector of ones.

From [6, Theorem 3.4], we obtain:

Theorem 1. *Let an irreducible Runge–Kutta method be given by the Butcher array* $\mathbf{K}$*. Let* c *denote the SSP coefficient from Definition* 1*. Let* $R(\mathbf{K})$ *denote the radius of absolute monotonicity defined in Definition* 2*. Then*

$$c = R(\mathbf{K}).$$

Furthermore, there exists a modified Shu–Osher representation $(\boldsymbol{\lambda}, \boldsymbol{\mu})$ *such that* (10) *holds and*

$$c = \min_{i,j;\, i \neq j} \frac{\lambda_{i,j}}{\mu_{i,j}},$$

where the minimum is taken over all $\mu_{i,j} \neq 0$. In other words, the method preserves strong stability under the maximal timestep restriction

$$\Delta t \leqslant R(\mathbf{K})\Delta t_{\mathrm{FE}}.$$

For a definition of reducibility see, e.g., [6, Definition 3.1]. If we replace the assumption of irreducibility in the theorem with the assumption $c < \infty$, then the same results follow from [14, Propositions 2.1, 2.2 and 2.7]. Furthermore, the restriction $c < \infty$ is not unduly restrictive in the present work because if $c = \infty$ then $p = 1$ [10], and we will be concerned only with methods having order $p \geqslant 2$.

Although we are interested in strong stability preservation for general (nonlinear, nonautonomous) systems, it is useful for the purposes of this section to introduce some concepts related to strong stability preservation for linear autonomous systems.

When applied to a linear ODE

$$u_t = \tilde{\lambda} u,$$

any Runge–Kutta method reduces to the iteration

$$u^{n+1} = \phi(\Delta t \tilde{\lambda}) u^n,$$

where ϕ is a rational function called the *stability function* of the Runge–Kutta method. From [12, Section IV.3] we have the following equivalent expressions for the stability function for implicit Runge–Kutta methods

$$\phi(z) = 1 + z\boldsymbol{b}^{\mathrm{T}}(\mathbf{I} - z\mathbf{A})^{-1}\boldsymbol{e} \quad \text{and} \quad \phi(z) = \frac{\det(\mathbf{I} - z\mathbf{A} + z\boldsymbol{e}\boldsymbol{b}^{\mathrm{T}})}{\det(\mathbf{I} - z\mathbf{A})}. \tag{11}$$

Definition 3 *(Strong stability preservation for linear systems).* For $\Delta t_{\mathrm{FE}} > 0$, let $\mathcal{L}(\Delta t_{\mathrm{FE}})$ denote the set of all pairs $(\mathbf{L}, \|\cdot\|)$ where the matrix $\mathbf{L} \in \mathbb{R}^{m\times m}$ and convex functional $\|\cdot\|$ are such that the numerical solution obtained by forward Euler integration of the linear autonomous system of ODEs $\boldsymbol{u}_t = \mathbf{L}\boldsymbol{u}$ satisfies $\|\boldsymbol{u}^{n+1}\| \leqslant \|\boldsymbol{u}^n\|$ whenever $\Delta t \leqslant \Delta t_{\mathrm{FE}}$. Given a Runge–Kutta method, the *linear SSP coefficient* of the method is the largest constant $c_{\mathrm{lin}} \geqslant 0$ such that the numerical solution obtained with the Runge–Kutta method satisfies $\|\boldsymbol{u}^{n+1}\| \leqslant \|\boldsymbol{u}^n\|$ for all $(\mathbf{L}, \|\cdot\|) \in \mathcal{L}(\Delta t_{\mathrm{FE}})$ whenever

$$\Delta t \leqslant c_{\mathrm{lin}}\Delta t_{\mathrm{FE}}. \tag{12}$$

If $c_{\mathrm{lin}} > 0$, the method is said to be *strong stability preserving for linear systems* under the timestep restriction (12).

When solving a linear system of ODEs, the timestep restriction for strong stability preservation depends on the radius of absolute monotonicity of ϕ.

Definition 4 *(Radius of absolute monotonicity (of a function)).* The radius of absolute monotonicity of a function ψ, denoted by $R(\psi)$, is the largest value of $r \geqslant 0$ such that $\psi(x)$ and all of its derivatives exist and are nonnegative for $x \in (-r, 0]$.

The following result is due to Spijker [31]:

Theorem 2. *Let a Runge–Kutta method be given with stability function ϕ. Let c_{lin} denote the linear SSP coefficient of the method (see Definition 3). Then*

$$c_{\mathrm{lin}} = R(\phi).$$

In other words, the method preserves strong stability for linear systems under the maximal timestep restriction

$$\Delta t \leqslant R(\phi)\Delta t_{\mathrm{FE}}.$$

Because of this result, $R(\phi)$ is referred to as the *threshold factor* of the method [31,34]. Since $\mathcal{L}(h) \subset \mathcal{F}(h)$, clearly $c \leqslant c_{\mathrm{lin}}$, so it follows that

$$R(\mathbf{K}) \leqslant R(\phi). \tag{13}$$

Optimal values of $R(\phi)$, for various classes of Runge–Kutta methods, can be inferred from results found in [34], where the maximal value of $R(\psi)$ was studied for ψ belonging to certain classes of rational functions.

In the following section, we use this equivalence between the radius of absolute monotonicity and the SSP coefficient to apply results regarding $R(\mathbf{K})$ to the theory of SSP Runge–Kutta methods.

2.3. Order barriers for SSP Runge–Kutta methods

The SSP property is a very strong requirement, and imposes severe restrictions on other properties of a Runge–Kutta method. We now review these results and draw a few additional conclusions that will guide our search for optimal methods in the next section.

Some results in this and the next section will deal with the optimal value of $R(\mathbf{K})$ when $\mathbf{K}$ ranges over some class of methods. This optimal value will be denoted by $R^{\text{IRK}}_{s,p}$ (resp., $R^{\text{ERK}}_{s,p}$) when $\mathbf{K}$ is permitted to be any implicit (resp., explicit) Runge–Kutta method with at most s stages and at least order p.

The result below follows from [31, Theorem 1.3] and Eq. (13) above.

Result 1. Any Runge–Kutta method of order $p > 1$ has a finite radius of absolute monotonicity; i.e. $R^{\text{IRK}}_{s,p} < \infty$ for $p > 1$.

This is a disappointing result, which shows us that for SSP Runge–Kutta methods of order greater than one we cannot avoid timestep restrictions altogether by using implicit methods (see also [10]). This is in contrast with linear stability and B-stability, where some high-order implicit methods (viz., the A-stable methods and the algebraically stable methods) have no timestep restriction. However, this does not indicate how restrictive the step-size condition is; it may still be worthwhile to consider implicit methods if the radius of absolute monotonicity is large enough to offset the additional work involved in an implicit solver.

From [22, Theorem 4.2] we can state the following result, which gives lower bounds on the coefficients that are useful in numerical searches. It is also useful in proving subsequent results.

Result 2. Any irreducible Runge–Kutta method with positive radius of absolute monotonicity $R(\mathbf{K}) > 0$, must have all nonnegative coefficients $\mathbf{A} \geqslant 0$ and positive weights $\boldsymbol{b} > 0$.

The following three results deal with the stage order $\tilde{p}$ of a Runge–Kutta method. The stage order is a lower bound on the order of convergence when a method is applied to arbitrarily stiff problems. Thus low stage order may lead to slow convergence (i.e., *order reduction*) when computing solutions of stiff ODEs. The stage order can be shown to be the largest integer $\tilde{p}$ such that the simplifying assumptions $B(\tilde{p})$, $C(\tilde{p})$ hold, where these assumptions are [3]

$$B(\xi): \quad \sum_{j=1}^{s} b_j c_j^{k-1} = \frac{1}{k} \quad (1 \leqslant k \leqslant \xi), \tag{14a}$$

$$C(\xi): \quad \sum_{j=1}^{s} a_{ij} c_j^{k-1} = \frac{1}{k} c_i^k \quad (1 \leqslant k \leqslant \xi). \tag{14b}$$

Result 3. (See [22, Theorem 8.5].) A Runge–Kutta method with nonnegative coefficients $\mathbf{A} \geqslant 0$ must have stage order $\tilde{p} \leqslant 2$. If $\tilde{p} = 2$, then $\mathbf{A}$ must have a zero row.

Result 4. (See [22, Lemma 8.6].) A Runge–Kutta method with weights $\boldsymbol{b} > 0$ must have stage order $\tilde{p} \geqslant \lfloor \frac{p-1}{2} \rfloor$.

When dealing with explicit methods, stage order is limited whether or not one requires nonnegative coefficients [22,2]:

Result 5. The stage order of an explicit Runge–Kutta method cannot exceed $\tilde{p} = 1$.

For SSP methods, the stage order restriction leads to restrictions on the classical order as well. Combining Results 2–4, and 5, we obtain:

Result 6. (See also [22, Corollary 8.7].) Any irreducible Runge–Kutta method with $R(\mathbf{K}) > 0$ has order $p \leqslant 4$ if it is explicit and $p \leqslant 6$ if it is implicit. Furthermore, if $p \geqslant 5$, then $\mathbf{A}$ has a zero row.

Result 6 shows that $R^{\mathrm{ERK}}_{s,p} = 0$ for $p > 4$ and $R^{\mathrm{IRK}}_{s,p} = 0$ for $p > 6$. The negative implications in Result 6 stem from the conditions $\mathbf{A} \geqslant 0$, $\boldsymbol{b} > 0$ in Result 2. Nonnegativity of $\mathbf{A}$ leads to low stage order (Result 3), while positivity of $\boldsymbol{b}$ leads to a limit on the classical order (Result 4) relative to the stage order. The result is a restriction on the classical order of SSP methods.

2.4. Barriers for singly implicit and diagonally implicit methods

An s-stage Runge–Kutta method applied to a system of m ODEs typically requires the solution of a system of sm equations. When the system results from the semi-discretization of a system of nonlinear PDEs, m is typically very large and the system of ODEs is nonlinear, making the solution of this system very expensive. Using a transformation involving the Jordan form of $\mathbf{A}$, the amount of work can be reduced [1]. This is especially efficient for *singly implicit* (SIRK) methods (those methods for which $\mathbf{A}$ has only one distinct eigenvalue), because the necessary matrix factorizations can be reused. On the other hand, *diagonally implicit* (DIRK) methods, for which $\mathbf{A}$ is lower triangular, can be implemented efficiently without transforming to the Jordan form of $\mathbf{A}$. The class of *singly diagonally implicit* (SDIRK) methods, which are both singly implicit and diagonally implicit (i.e., $\mathbf{A}$ is lower triangular with all diagonal entries identical), incorporate both of these advantages. Note that in the literature the term diagonally implicit has sometimes been used to mean singly diagonally implicit. We use $R^{\mathrm{DIRK}}_{s,p}$, $R^{\mathrm{SIRK}}_{s,p}$, and $R^{\mathrm{SDIRK}}_{s,p}$ to denote the optimal value of $R(\mathbf{K})$ over each of the respective classes of DIRK, SIRK, and SDIRK Runge–Kutta methods. Note that for a given s and p, these three quantities are each bounded by $R^{\mathrm{IRK}}_{s,p}$. For details on efficient implementation of implicit Runge–Kutta methods see, e.g., [3].

We now review some results regarding SSP methods in these classes.

Result 7. (See [7, Theorem 3.1].) An SDIRK method with positive radius of absolute monotonicity $R(\mathbf{K}) > 0$ must have order $p \leqslant 4$, i.e. $R^{\mathrm{SDIRK}}_{s,p} = 0$ for $p > 4$.

Proposition 1. *The order of an s-stage DIRK method when applied to a linear problem is at most $s + 1$.*

Proof. For a given s-stage DIRK method, let p denote its order and let ϕ denote its stability function. Then $\phi(x) = \exp(x) + \mathcal{O}(x^{p+1})$ as $x \to 0$. By equation (11), ϕ is a rational function whose numerator has degree at most s. For DIRK methods, $\mathbf{A}$ is lower triangular, so by Eq. (11) the poles of ϕ are the diagonal entries of $\mathbf{A}$, which are real. A rational function with real poles only and numerator of degree s approximates the exponential function to order at most $s + 1$ [3, Theorem 3.5.11]. Thus $p \leqslant s + 1$. □

Proposition 2. *The order of an s-stage SIRK method with positive radius of absolute monotonicity is at most $s + 1$. Hence $R^{\mathrm{SIRK}}_{s,p} = 0$ for $p > s + 1$.*

Proof. For a given s-stage SIRK method, let p denote its order and let ϕ denote its stability function. Assume $R(\mathbf{K}) > 0$; then by Eq. (13) $R(\phi) > 0$. By Eq. (11), ϕ is a rational function whose numerator has degree at most s. For SIRK methods, equation (11) also implies that ϕ has a unique pole. Since $R(\phi) > 0$, [34, Corollary 3.4] implies that this pole must be real. As in the proof above, [3, Theorem 3.5.11] then provides the desired result. □

Result 6 implies that all eigenvalues of $\mathbf{A}$ must be zero, hence the stability function ϕ must be a polynomial. We thus have

Table 1
Values of $R(\mathbf{K})$ for some well-known implicit methods

Family	stages	order	$R(\mathbf{K})$
Gauss–Legendre	1	2	2
Radau IA	1	1	∞
Radau IIA	1	1	∞
Lobatto IIIA	2	2	2
Lobatto IIIB	2	2	2

Corollary 1. *Consider the class of s-stage SIRK methods with order $5 \leqslant p \leqslant 6$ and $R(\mathbf{K}) > 0$. Let $\Pi_{n,p}$ denote the set of all polynomials ψ of degree less than or equal to n satisfying $\psi(x) = \exp(x) + \mathcal{O}(x^{p+1})$ as $x \to 0$. Then for $5 \leqslant p \leqslant 6$,*

$$R^{\mathrm{SIRK}}_{s,p} \leqslant \sup_{\psi \in \Pi_{s,p}} R(\psi)$$

where the supremum over an empty set is taken to be zero. Furthermore $R^{\mathrm{SIRK}}_{s,p} = 0$ for $4 \leqslant s < p \leqslant 6$.

The last statement of the corollary follows from the observation that, since a polynomial approximates $\exp(x)$ to order at most equal to its degree, $\Pi_{n,p}$ is empty for $p > s$.

Corollary 1 implies that for s-stage SIRK methods of order $p \geqslant 5$, $R(\mathbf{K})$ is bounded by the optimal linear SSP coefficient of s-stage explicit Runge–Kutta methods of the same order (see [21,17] for values of these optimal coefficients).

2.5. Absolute monotonicity of some classical methods

When constructing high-order implicit Runge–Kutta methods it is common to use the stage order conditions (14) as simplifying assumptions; however, Result 3 implies that any method satisfying the conditions $B(3)$ and $C(3)$ cannot be SSP. Examining the simplifying conditions satisfied by well-known methods (see [3, Table 3.3.1]), one sees that this implies $R(\mathbf{K}) = 0$ for the Gauss–Legendre, Radau IIA, and Lobatto IIIA methods with more than two stages, the Radau IA and Lobatto IIIC methods with more than three stages, and the Lobatto IIIB methods with more than four stages. Checking the signs of the Butcher arrays of the remaining (low stage number) methods in these families, by Result 2 we obtain that only the one-stage Gauss–Legendre, Radau IA, Radau IIA, and the two-stage Lobatto IIIA and Lobatto IIIB methods can have $R(\mathbf{K})$ different from zero. The values of $R(\mathbf{K})$ for these methods are listed in Table 1.

3. Optimal SSP implicit Runge–Kutta methods

In this section we present numerically optimal implicit SSP Runge–Kutta methods for nonlinear systems of ODEs. These methods were found via numerical search, and in general we have no analytic proof of their optimality. In a few cases, we have employed BARON, an optimization software package that provides a numerical certificate of global optimality [27]. BARON was used to find optimal explicit SSP Runge–Kutta methods in [24,26]. However, this process is computationally expensive and was not practical in most cases.

Most of the methods were found using MATLAB's optimization toolbox. We applied the same computational approach to finding numerically optimal explicit and diagonally implicit SSP Runge–Kutta methods, and successfully found a solution at least as good as the previously best known solution in every case. Because our approach was able to find these previously known methods, we expect that some of new methods—particularly those of lower-order or lower number of stages—may be globally optimal.

The optimization problem for general Runge–Kutta methods involves approximately twice as many decision variables (dimensions) as the explicit or singly diagonally implicit cases, which have previously been investigated [9,10, 32,33,26,7]. Despite the larger number of decision variables, we have been able to find numerically optimal methods even for large numbers of stages. We attribute this success to the reformulation of the optimization problem in terms of the Butcher coefficients rather than the Shu–Osher coefficients, as suggested in [5]. Specifically, we solve the optimization problem

$$\max_{\mathbf{K}} r, \tag{15a}$$

$$\text{subject to} \quad \begin{cases} \mathbf{K}(\mathbf{I}+r\mathbf{A})^{-1} \geqslant 0, \\ r\mathbf{K}(\mathbf{I}+r\mathbf{A})^{-1}\boldsymbol{e}_s \leqslant \boldsymbol{e}_{s+1}, \\ \Phi_p(\mathbf{K}) = 0, \end{cases} \tag{15b}$$

where the inequalities are understood component-wise and recall that $\Phi_p(\mathbf{K})$ represents the order conditions up to order p. This formulation, implemented in MATLAB using a sequential quadratic programming approach (fmincon in the optimization toolbox), was used to find the methods given below. In a concurrent effort, this formulation has been used to search for optimal explicit SSP methods [17].

Because in most cases we cannot prove the optimality of the resulting methods, we use hats to denote the best value found by numerical search, e.g. $\hat{R}^{\text{IRK}}_{s,p}$, etc.

The above problem can be reformulated (using a standard approach for converting rational constraints to polynomial constraints) as

$$\max_{\mathbf{K},\boldsymbol{\mu}} r, \tag{16a}$$

$$\text{subject to} \quad \begin{cases} \boldsymbol{\mu} \geqslant 0, \\ r\boldsymbol{\mu}\boldsymbol{e}_s \leqslant \boldsymbol{e}_{s+1}, \\ \mathbf{K} = \boldsymbol{\mu}(\mathbf{I}+r\mathbf{A}), \\ \Phi_p(\mathbf{K}) = 0. \end{cases} \tag{16b}$$

This optimization problem has only polynomial constraints and thus is appropriate for the BARON optimization software which requires such constraints to be able to guarantee global optimality [27]. Note that $\boldsymbol{\mu}$ in (16) corresponds to one possible modified Shu–Osher form with $\boldsymbol{\lambda} = r\boldsymbol{\mu}$.

In comparing methods with different numbers of stages, one is usually interested in the relative time advancement per computational cost. For diagonally implicit methods, the computational cost per time-step is proportional to the number of stages. We therefore define the *effective SSP coefficient* of a method as $\frac{R(\mathbf{K})}{s}$; this normalization enables us to compare the cost of integration up to a given time using DIRK schemes of order $p > 1$. However, for non-DIRK methods of various s, it is much less obvious how to compare computation cost.

In the following, we give modified Shu–Osher arrays for the numerically optimal methods. To simplify implementation, we present modified Shu–Osher arrays in which the diagonal elements of $\boldsymbol{\lambda}$ are zero. This form is a simple rearrangement and involves no loss of generality.

3.1. Second-order methods

Optimizing over the class of all ($s \leqslant 11$)-stage second-order implicit Runge–Kutta methods we found that the numerically optimal methods are, remarkably, identical to the numerically optimal SDIRK methods found in [5,7]. This result stresses the importance of the second-order SDIRK methods found in [5,7]: they appear to be optimal not only among SDIRK methods, but also among the larger class of all implicit Runge–Kutta methods.

These methods are most advantageously implemented in a certain modified Shu–Osher form. This is because these arrays (if chosen carefully) are more sparse. In fact, for these methods there exist modified Shu–Osher arrays that are bidiagonal. We give the general formulae here.

The numerically optimal second-order method with s stages has $R(\mathbf{K}) = 2s$ and coefficients

$$\boldsymbol{\lambda} = \begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ & 1 & \ddots & \\ & & \ddots & 0 \\ & & & 1 \end{bmatrix}, \qquad \boldsymbol{\mu} = \begin{bmatrix} \frac{1}{2s} & & & \\ \frac{1}{2s} & \frac{1}{2s} & & \\ & \frac{1}{2s} & \ddots & \\ & & \ddots & \frac{1}{2s} \\ & & & \frac{1}{2s} \end{bmatrix}. \tag{17}$$

The one-stage method of this class is the implicit midpoint rule, while the s-stage method is equivalent to s successive applications of the implicit midpoint rule (as was observed in [5]). Thus these methods inherit the desirable properties of the implicit midpoint rule such as algebraic stability and A-stability [12]. Of course, since they all have the same effective SSP coefficient $R(\mathbf{K})/s = 2$, they are all essentially equivalent.

The one-stage method is the unique method with $s = 1$, $p = 2$ and hence is optimal. The two-stage method achieves the maximum radius of absolute monotonicity for rational functions that approximate the exponential to second order with numerator and denominator of degree at most two, hence it is optimal to within numerical precision [34,16,7]. In addition to duplicating these optimality results, BARON was used to numerically prove that the $s = 3$ scheme is globally optimal, verifying [7, Conjecture 3.1] for the case $s = 3$. The $s = 1$ and $s = 2$ cases required only several seconds but the $s = 3$ case took much longer, requiring approximately 11 hours of CPU time on an Athlon MP 2800+ processor.

While the remaining methods have not been proven optimal, it appears likely that they may be. From multiple random initial guesses, the optimization algorithm consistently converges to the same method, or to a reducible method corresponding to one of the numerically optimal methods with a smaller number of stages. Also, many of the inequality constraints are satisfied exactly for these methods. Furthermore, the methods all have a similar form, depending only on the stage number. These observations suggest:

Conjecture 1. *(An extension of [7, Conjecture 3.1]) The optimal second-order s-stage implicit SSP method is given by the SDIRK method* (17) *and hence* $R^{\text{IRK}}_{s,2} = 2s$.

This conjecture would imply that the effective SSP coefficient of any Runge–Kutta method of order greater than one is at most equal to two.

3.2. Third-order methods

The numerically optimal third-order implicit Runge–Kutta methods with $s \geqslant 2$ stages are also SDIRK and identical to the numerically optimal SDIRK methods found in [5,7], which have $R(\mathbf{K}) = s - 1 + \sqrt{s^2 - 1}$. Once again, these results indicate that the methods found in [5,7] are likely optimal over the entire class of implicit Runge–Kutta methods.

In this case, too, when implementing these methods it is possible to use bidiagonal Shu–Osher arrays. For $p = 3$ and $s \geqslant 2$ the numerically optimal methods have coefficients

$$\boldsymbol{\mu} = \begin{bmatrix} \mu_{11} & & & \\ \mu_{21} & \ddots & & \\ & \ddots & \mu_{11} & \\ & & \mu_{21} & \mu_{11} \\ & & & \mu_{s+1,s} \end{bmatrix}, \qquad \boldsymbol{\lambda} = \begin{bmatrix} 0 & & & \\ 1 & \ddots & & \\ & \ddots & 0 & \\ & & 1 & 0 \\ & & & \lambda_{s+1,s} \end{bmatrix}, \tag{18a}$$

where

$$\mu_{11} = \frac{1}{2}\left(1 - \sqrt{\frac{s-1}{s+1}}\right), \qquad \mu_{21} = \frac{1}{2}\left(\sqrt{\frac{s+1}{s-1}} - 1\right), \tag{18b}$$

$$\mu_{s+1,s} = \frac{s+1}{s(s+1+\sqrt{s^2-1})}, \qquad \lambda_{s+1,s} = \frac{(s+1)(s-1+\sqrt{s^2-1})}{s(s+1+\sqrt{s^2-1})}. \tag{18c}$$

The two-stage method in this family achieves the maximum value of $R(\phi)$ found in [34] for ϕ in the set of third-order rational approximations to the exponential with numerator and denominator of degree at most 2. Since the corresponding one-parameter optimization problem is easy to solve, then (in view of (13)) the method is clearly optimal to within numerical precision. BARON was used to numerically prove global optimality for the three-stage method (18), requiring about 12 hours of CPU time on an Athlon MP 2800+ processor. Note that this verifies [7, Conjecture 3.2] for the case $s = 3$.

Table 2
SSP coefficients and effective SSP coefficients of numerically optimal fourth-order implicit Runge–Kutta methods and SDIRK methods

s	$\hat{R}^{\text{IRK}}_{s,4}$	$\hat{R}^{\text{SDIRK}}_{s,4}$	$\hat{R}^{\text{IRK}}_{s,4}/s$	$\hat{R}^{\text{SDIRK}}_{s,4}/s$
3	2.05	1.76	0.68	0.59
4	4.42	4.21	1.11	1.05
5	6.04	5.75	1.21	1.15
6	7.80	7.55	1.30	1.26
7	9.19	8.67	1.31	1.24
8	10.67	10.27	1.33	1.28
9	12.04		1.34	
10	13.64		1.36	
11	15.18		1.38	

While the remaining methods (those with $s \geqslant 4$) have not been proven optimal, we are again led to suspect that they may be, because of the nature of the optimal methods and the convergent behavior of the optimization algorithm for these cases. These observations suggest:

Conjecture 2. *(An extension of [7, Conjecture 3.2]) For $s \geqslant 2$, the optimal third-order s-stage implicit Runge–Kutta SSP method is given by the SDIRK method (18) and hence $R^{\text{IRK}}_{s,3} = s - 1 + \sqrt{s^2 - 1}$.*

3.3. Fourth-order methods

Based on the above results, one might suspect that all optimal implicit SSP methods are singly diagonally implicit. In fact, this cannot hold for $p \geqslant 5$ since in that case $\mathbf{A}$ must have a zero row (see Result 6 above). The numerically optimal methods of fourth-order are not singly diagonally implicit either; however, all numerically optimal fourth-order methods we have found are diagonally implicit.

The unique two-stage fourth-order Runge–Kutta method has a negative coefficient and so is not SSP. Thus we begin our search with three-stage methods. We list the SSP coefficients and effective SSP coefficients of the numerically optimal methods in Table 2. For comparison, the table also lists the effective SSP coefficients of the numerically optimal SDIRK methods found in [7]. Our numerically optimal DIRK methods have larger SSP coefficients in every case. Furthermore, they have representations that allow for very efficient implementation in terms of storage. However, SDIRK methods may be implemented in a potentially more efficient (in terms of computation) manner than DIRK methods. An exact evaluation of the relative efficiencies of these methods is beyond the scope of this work. The coefficients of the 4-stage method are included in Table 7. The coefficients of the remaining methods are available from [19,18].

BARON was run on the three-stage fourth-order case but was unable to prove the global optimality of the resulting method using 14 days of CPU time on an Athlon MP 2800+ processor. However, during that time BARON did establish an upper bound $R^{\text{IRK}}_{3,4} \leqslant 3.234$. BARON was not run on any other fourth-order cases, nor was it used for $p = 5$ or $p = 6$.

Although none of the fourth-order methods are proven optimal, it appears that they may be optimal. This is again because the optimization algorithm is able to converge to these methods from a range of random initial guesses, and because very many of the inequality constraints are satisfied exactly for these methods. Additionally, we were able to recover all of the optimal fourth-order SDIRK methods of [7] by restricting our search to the space of SDIRK methods.

3.4. Fifth- and sixth-order methods

We have found fifth- and sixth-order SSP methods with up to eleven stages. Two sets of numerical searches were conducted, corresponding to optimization over the full class of implicit Runge–Kutta methods and optimization over the subclass of diagonally implicit Runge–Kutta methods. More CPU time was devoted to the first set of searches; however, in most cases the best methods we were able to find resulted from the searches restricted to DIRK methods. Furthermore, when searching over fully implicit methods, in every case for which the optimization algorithm suc-

Table 3
Comparison of SSP coefficients of numerically optimal fifth-order IRK methods with theoretical upper bounds on SSP coefficients of fifth-order SIRK methods

s	$\hat{R}^{\text{IRK}}_{s,5}$	$R^{\text{SIRK}}_{s,5}$ (upper bound)	$\hat{R}^{\text{IRK}}_{s,5}/s$	$R^{\text{SIRK}}_{s,5}/s$ (upper bound)
4	1.14		0.29	
5	3.19	1.00	0.64	0.20
6	4.97	2.00	0.83	0.33
7	6.21	2.65	0.89	0.38
8	7.56	3.37	0.94	0.42
9	8.90	4.10	0.99	0.46
10	10.13	4.83	1.01	0.48
11	11.33	5.52	1.03	0.50

cessfully converged to a (local) optimum, the resulting method was diagonally implicit. Thus all of the numerically optimal methods found are diagonally implicit.

Because better results were obtained in many cases by searching over a strictly smaller class of methods, it seems likely that the methods found are not globally optimal. This is not surprising because the optimization problems involved are highly nonlinear with many variables, many constraints, and multiple local optima. The application of more sophisticated software to this problem is an area of future research. Nevertheless, the observation that all converged solutions correspond to DIRK methods leads us to believe that the globally optimal methods are likely to be DIRK methods.

Typically, an optimization algorithm may be expected to fail for sufficiently large problems (in our case, sufficiently large values of s). However, we found that the cases of relatively small s and large p (i.e., $p = 5$ and $s < 6$ or $p = 6$ and $s < 9$) also posed great difficulty. This may be because the feasible set in these cases is extremely small. The methods found in these cases were found indirectly by searching for methods with more stages and observing that the optimization algorithm converged to a reducible method. Due to the high nonlinearity of the problem for $p \geqslant 5$, we found it helpful to explicitly limit the step sizes used by fmincon in the final steps of optimization.

3.4.1. Fifth-order methods

Three stages Using the W transformation [3] we find the one parameter family of three-stage, fifth-order methods

$$\mathbf{A} = \begin{bmatrix} \frac{5}{36} + \frac{2}{9}\gamma & \frac{5}{36} + \frac{1}{24}\sqrt{15} - \frac{5}{18}\gamma & \frac{5}{36} + \frac{1}{30}\sqrt{15} + \frac{2}{9}\gamma \\ \frac{2}{9} - \frac{1}{15}\sqrt{15} - \frac{4}{9}\gamma & \frac{2}{9} + \frac{5}{9}\gamma & \frac{2}{9} + \frac{1}{15}\sqrt{15} - \frac{4}{9}\gamma \\ \frac{5}{36} - \frac{1}{30}\sqrt{15} + \frac{2}{9}\gamma & \frac{5}{36} - \frac{1}{24}\sqrt{15} - \frac{5}{18}\gamma & \frac{5}{36} + \frac{2}{9}\gamma \end{bmatrix}.$$

It is impossible to choose γ so that a_{21} and a_{31} are simultaneously nonnegative, so there are no SSP methods in this class.

Four to eleven stages We list the time-step coefficients and effective SSP coefficients of the numerically optimal fifth order implicit Runge–Kutta methods for $4 \leqslant s \leqslant 11$ in Table 3. It turns out that all of these methods are diagonally implicit.

For comparison, we also list the upper bounds on effective SSP coefficients of SIRK methods in these classes implied by combining Corollary 1 with [17, Table 2.1]. Our numerically optimal IRK methods have larger effective SSP coefficients in every case. The coefficients of the optimal five-stage method are listed in Table 8. Coefficients of the remaining methods are available from [19,18].

3.4.2. Sixth-order methods

Kraaijevanger [22] proved the bound $p \leqslant 6$ for contractive methods (see Result 6 above) and presented a single fifth-order contractive method, leaving the existence of sixth-order contractive methods as an open problem. The sixth-order methods we have found settle this problem, demonstrating that the order barrier $p \leqslant 6$ for implicit SSP/contractive methods is sharp.

Table 4
Radii of absolute monotonicity and effective SSP coefficients for numerically optimal sixth-order methods

s	$\hat{R}^{\text{IRK}}_{s,6}$	$\hat{R}^{\text{IRK}}_{s,6}/s$
6	0.18	0.030
7	0.26	0.038
8	2.25	0.28
9	5.80	0.63
10	8.10	0.81
11	8.85	0.80

Table 5
Effective SSP coefficients of best known methods. A dash indicates that SSP methods of this type cannot exist. A blank space indicates that no SSP methods of this type were found

s	p							
	Implicit methods					Explicit methods		
	2	3	4	5	6	2	3	4
1	2	–	–	–	–	–	–	–
2	2	1.37	–	–	–	0.5	–	–
3	2	1.61	0.68	–	–	0.67	0.33	–
4	2	1.72	1.11	0.29		0.75	0.5	–
5	2	1.78	1.21	0.64		0.8	0.53	0.30
6	2	1.82	1.30	0.83	0.030	0.83	0.59	0.38
7	2	1.85	1.31	0.89	0.038	0.86	0.61	0.47
8	2	1.87	1.33	0.94	0.28	0.88	0.64	0.52
9	2	1.89	1.34	0.99	0.63	0.89	0.67	0.54
10	2	1.90	1.36	1.01	0.81	0.9	0.68	0.60
11	2	1.91	1.38	1.03	0.80	0.91	0.69	0.59

The nonexistence of three-stage SSP Runge–Kutta methods of fifth-order, proved above, implies that sixth-order SSP Runge–Kutta methods must have at least four-stages. Proposition 1 implies that sixth-order SSP DIRK methods must have at least five stages, and Corollary 1 shows that sixth-order SSP SIRK methods require at least six stages. We were unable to find sixth-order SSP Runge–Kutta methods with fewer than six stages.

The SSP coefficients and effective SSP coefficients of the numerically optimal methods for $6 \leqslant s \leqslant 11$ are listed in Table 4. All of these methods are diagonally implicit. The coefficients of the optimal nine-stage method are listed in Table 9. Coefficients of the remaining methods are available from [19,18]. We were unable to find an eleven-stage method with larger *effective* SSP coefficient than that of the ten-stage method (although we did find a method with larger $R(\mathbf{K})$).

Table 5 summarizes the effective SSP coefficients of the numerically optimal diagonally implicit methods for $2 \leqslant p \leqslant 6$ and $2 \leqslant s \leqslant 11$. For comparison, Table 5 also includes the effective SSP coefficients of the best known explicit methods, including results from the forthcoming paper [17].

4. Numerical experiments

We begin our numerical examples with a convergence study on a linear advection problem with smooth initial conditions. We then proceed to show the effect of the linear SSP coefficient for this linear advection problem with discontinuous initial conditions. Finally, the effect of the SSP coefficient is demonstrated on the nonlinear Burgers' and Buckley–Leverett equations. The computations in Section 4.1 were performed with MATLAB version 7.1 on a Mac G5; those in Sections 4.2 and 4.3 were performed with MATLAB version 7.3 on x86-64 architecture. All calculations were performed in double precision. For the implicit solution of linear problems we used MATLAB's backslash operator, while for the nonlinear implicit solves we used the fsolve function with very small tolerances.

We refer to the numerically optimal methods as SSPsp where s, p are the number of stages and order, respectively. For instance, the numerically optimal eight-stage method of order five is SSP85.

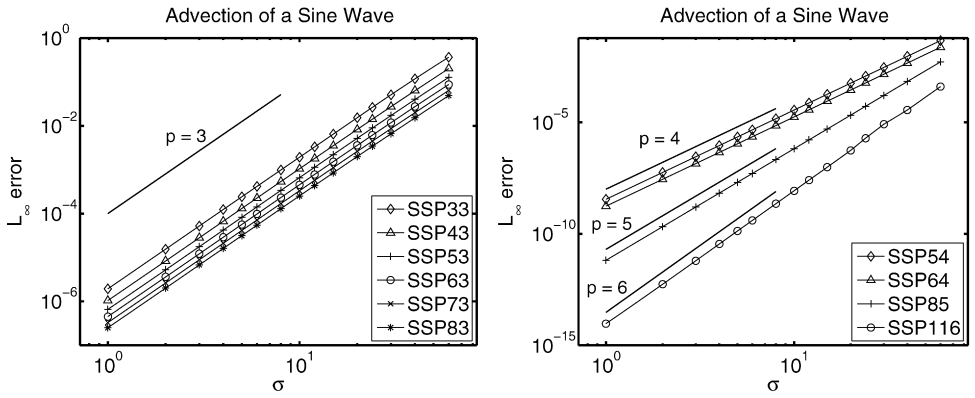


Fig. 1. Convergence of various numerically optimal SSP methods for linear advection of a sine wave.

4.1. Linear advection

The prototypical hyperbolic PDE is the linear wave equation,

$$u_t + au_x = 0, \quad 0 \leqslant x \leqslant 2\pi. \tag{19}$$

We consider (19) with $a = -2\pi$, periodic boundary conditions and various initial conditions. We use a method-of-lines approach, discretizing the interval $(0, 2\pi]$ into m points $x_j = j\Delta x$, $j = 1, \ldots, m$, and then discretizing $-au_x$ with first-order upwind finite differences. We solve the resulting system (1) using our timestepping schemes. To isolate the effect of the time-discretization error, we exclude the effect of the error associated with the spatial discretization by comparing the numerical solution to the exact solution of the ODE system, rather than to the exact solution of the PDE (19). In lieu of the exact solution we use a very accurate numerical solution obtained using MATLAB's ode45 solver with minimal tolerances ($\mathsf{AbsTol} = 1 \times 10^{-14}$, $\mathsf{RelTol} = 1 \times 10^{-13}$).

Fig. 1 shows a convergence study for various numerically optimal schemes for the problem (19) with $m = 120$ points in space and smooth initial data

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

advected until final time $t_f = 1$. Here σ indicates the size of the timestep: $\Delta t = \sigma \Delta t_{\mathrm{FE}}$. The results show that all the methods achieve their design order.

Now consider the advection equation with discontinuous initial data

$$u(x, 0) = \begin{cases} 1 & \text{if } \frac{\pi}{2} \leqslant x \leqslant \frac{3\pi}{2}, \\ 0 & \text{otherwise.} \end{cases} \tag{20}$$

Fig. 2 shows a convergence study for the third-order methods with $s = 3$ to $s = 8$ stages, for $t_f = 1$ using $m = 64$ points and the first-order upwinding spatial discretization. Again, the results show that all the methods achieve their design order. Finally, we note that the higher-stage methods give a smaller error for the same timestep; that is as s increases, the error constant of the method decreases.

Fig. 3 shows the result of solving the discontinuous advection example using the two-stage third-order method over a single timestep with $m = 200$. For this linear autonomous system, the theoretical monotonicity-preserving timestep bound is $\sigma \leqslant c_{\mathrm{lin}} = 2.732$. We see that as the timestep is increased, the line steepens and forms a small step, which becomes an oscillation as the stability limit is exceeded, and worsens as the timestep is raised further.

4.2. Burgers' equation

In this section we consider the inviscid Burgers' equation, which consists of the conservation law

$$u_t + f(u)_x = 0 \tag{21}$$

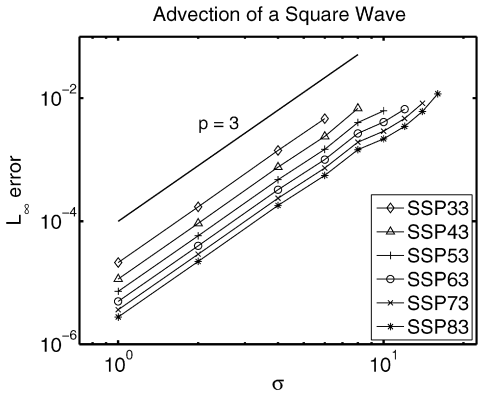


Fig. 2. Convergence of third-order methods for linear advection of a square wave.

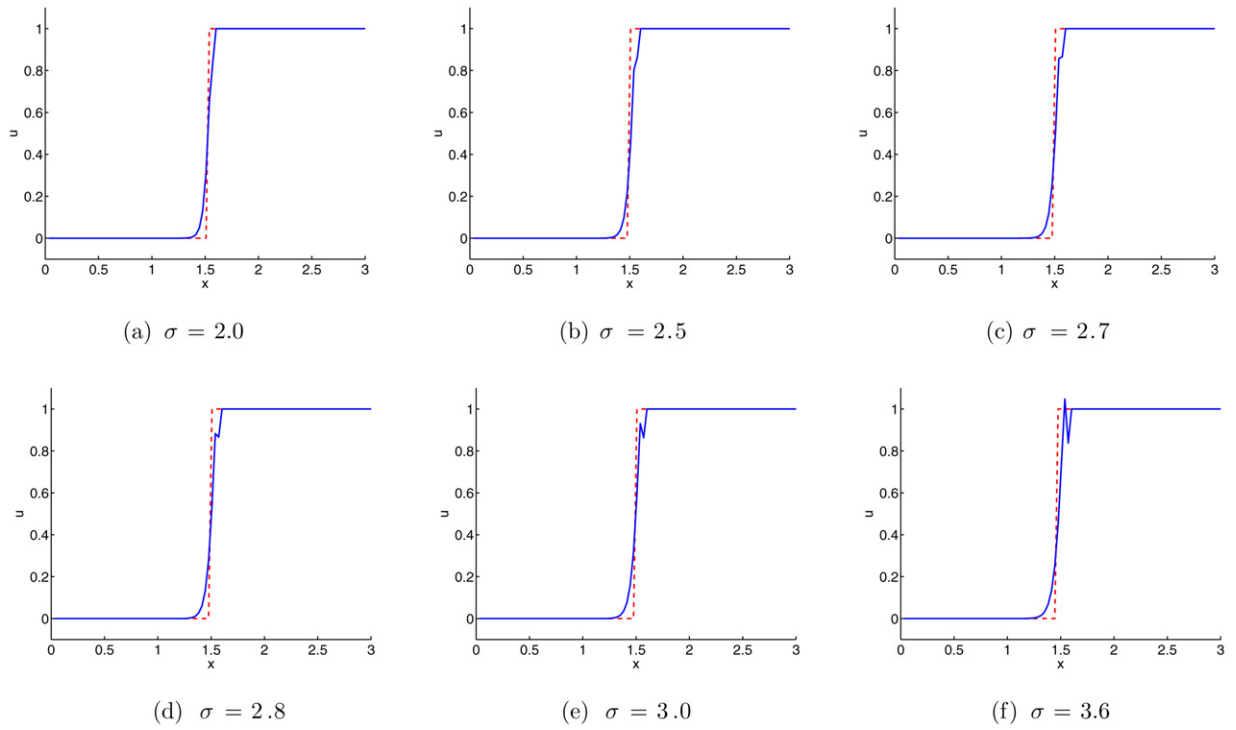


Fig. 3. Solution of the linear advection problem after one timestep with the two-stage third-order method ($c_{\text{lin}} = 2.732$).

with flux function $f(u) = \frac{1}{2}u^2$. We take initial conditions $u(0,x) = \frac{1}{2} - \frac{1}{4}\sin(\pi x)$ on the periodic domain $x \in [0,2)$. The solution is right-travelling and over time steepens into a shock. We discretize $-f(u)_x$ using the conservative upwind approximation

$$-f(u)_x \approx -\frac{1}{\Delta x}\big(f(u_i) - f(u_{i-1})\big) \tag{22}$$

with $m = 256$ points in space and integrate to time $t_f = 2$. The convergence study in Fig. 5 shows that the fourth-, fifth- and sixth-order s-stage methods achieve their respective orders of convergence when compared to a temporally very refined solution of the discretized system.

Fig. 4 shows that when the timestep is below the stability limit no oscillations appear, but when the stability limit is violated, oscillations are observed.

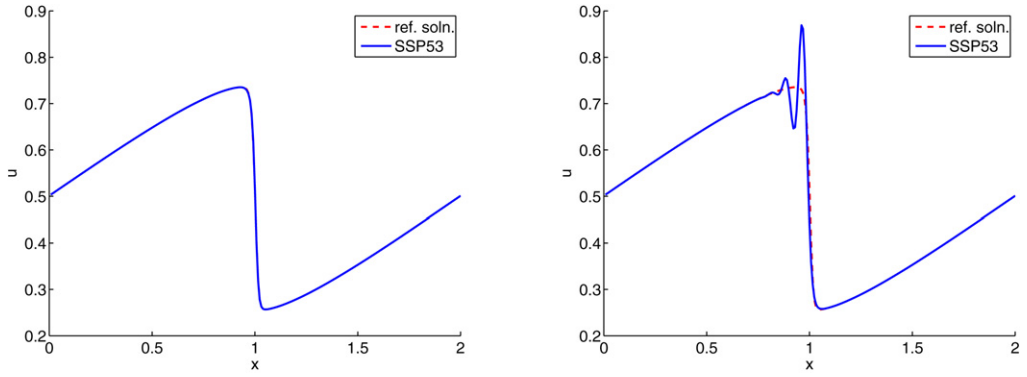


Fig. 4. Solution of Burgers' equation using the third-order, five-stage SSP timestepping method ($c = 8.90$).

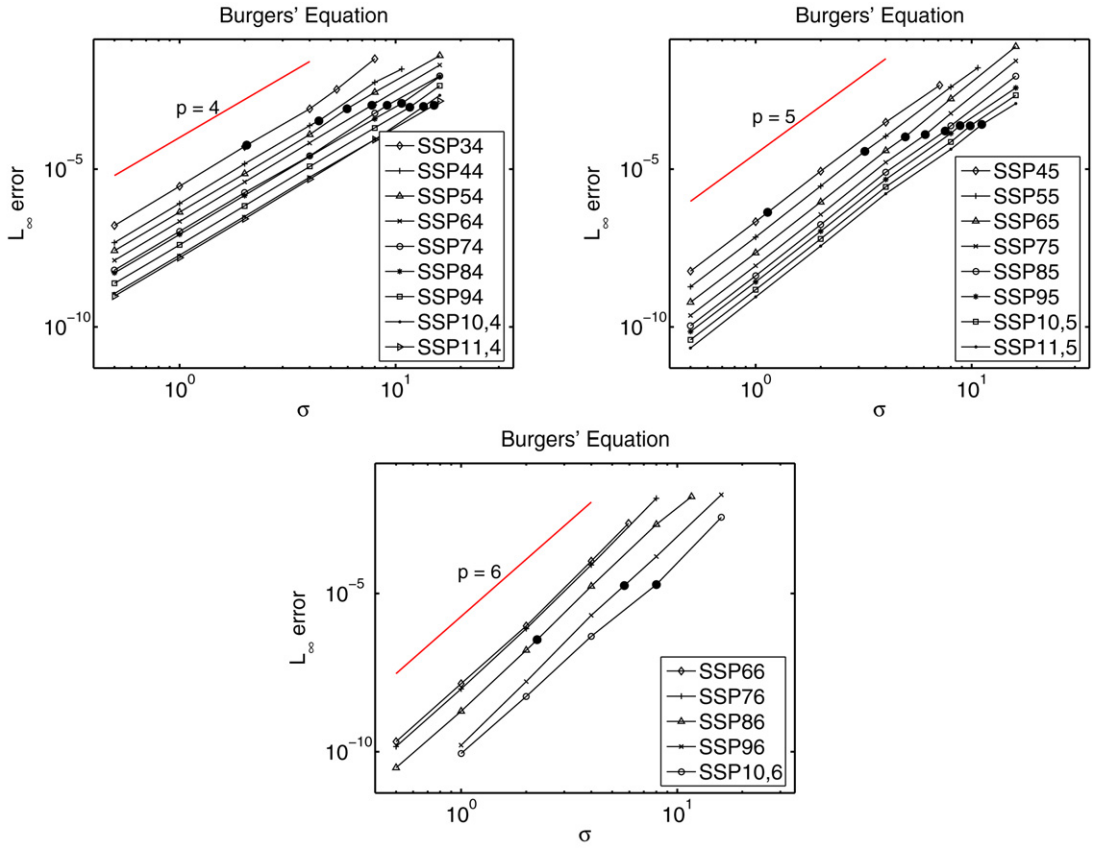
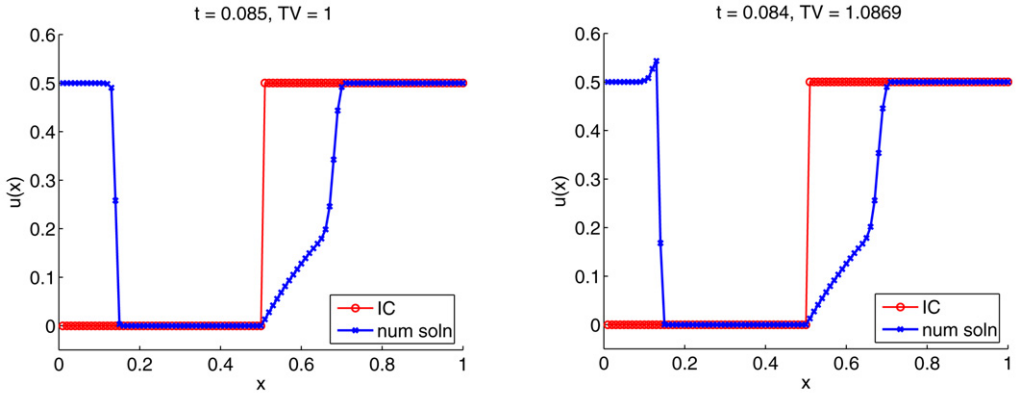


Fig. 5. Convergence of the numerically optimal fourth-, fifth- and sixth-order schemes on Burgers' equation. The solid circles indicate $\sigma = c$ for each scheme.

4.3. Buckley–Leverett equation

The Buckley–Leverett equation is a model for two-phase flow through porous media [23] and consists of the conservation law (21) with flux function

$$f(u) = \frac{u^2}{u^2 + a(1-u)^2}.$$



(a) $\sigma = 2$ (b) $\sigma = 2.4$

Fig. 6. Solution of the Buckley–Leverett equation using the second-order, one-stage SSP timestepping method ($c = 2$).

Table 6
SSP coefficients versus largest timesteps exhibiting the TVD property on the Buckley–Leverett example

s	p									
	$R(\mathbf{K})$					σ_{BL}				
	2	3	4	5	6	2	3	4	5	6
1	2	–	–	–	–	2.03	–	–	–	–
2	4	2.73	–	–	–	4.08	3.68	–	–	–
3	6	4.83	2.05	–	–	6.11	5.39	4.01	–	–
4	8	6.87	4.42	1.14		8.17	7.13	5.59	4.04	
5	10	8.90	6.04	3.21		10.25	9.06	6.46	4.91	
6	12	10.92	7.80	4.97	0.18	12.33	11.18	7.98	6.92	4.83
7	14	12.93	9.19	6.21	0.26	14.43	13.33	9.31	9.15	5.14
8	16	14.94	10.67	7.56	2.25	16.53	15.36	11.42	8.81	5.66
9	18	16.94	12.04	8.90	5.80	18.60	17.52	15.01	11.04	7.91
10	20	18.95	13.64	10.13	8.10	20.66	19.65	13.84	12.65	10.80
11	22	20.95	15.18	11.33	8.85	22.77	21.44	15.95	14.08	11.82

We take $a = \frac{1}{3}$ and initial conditions

$$u(x,0) = \begin{cases} 1 & \text{if } x \leqslant \frac{1}{2}, \\ 0 & \text{otherwise,} \end{cases}$$

on $x \in [0,1)$ with periodic boundary conditions. Our spatial discretization uses $m = 100$ points and we use the conservative scheme with Koren limiter used in [7] and [15, Section III.1]. The nonlinear system of equations for each stage of the Runge–Kutta method is solved with MATLAB's fsolve, with the Jacobian approximated [15] by that of the first-order upwind discretization (22). We compute the solution for $n = \lceil \frac{1}{8} \frac{1}{\Delta t} \rceil$ timesteps.

For this problem, as in [7], we find that the forward Euler solution is total variation diminishing (TVD) for $\Delta t \leqslant \Delta t_{\mathrm{FE}} = 0.0025$. Fig. 6 shows typical solutions for the SSP(1,2) scheme with timestep $\Delta t = \sigma \Delta t_{\mathrm{FE}}$. Table 6 compares the SSP coefficient $R(\mathbf{K})$ with $\sigma_{\mathrm{BL}} = \Delta t_{\mathrm{RK}} / \Delta t_{\mathrm{FE}}$, where Δt_{RK} is the largest *observed* timestep for which the numerical solution obtained with the Runge–Kutta method is TVD. We note that, for each method, the value of σ_{BL} is greater than the SSP coefficient. In fact, at least for either lower order p or high number of stages s, the values are in good correspondence. For $p = 2$ and $p = 3$, our results agree with those of [7].

5. Conclusions and future work

By numerical optimization we have found implicit strong stability preserving Runge–Kutta methods of order up to the maximum possible of $p = 6$ and stages up to $s = 11$. Methods with up to three stages and third order of

accuracy have been proven optimal by analysis or by the global optimization software package BARON. Remarkably, the numerically optimal methods of up to third-order are singly diagonally implicit, and the numerically optimal methods of all orders are diagonally implicit. Furthermore, all of the *local* optima found in our searches correspond to diagonally implicit methods. Based on these results, we conjecture that the optimal implicit SSP Runge–Kutta methods of any number of stages are diagonally implicit. Future work may involve numerical experiments with more powerful numerical optimization software, which will allow us to search more thoroughly and among methods with more stages to support this conjecture.

The likelihood that our numerically optimal methods are nearly or truly optimal can be inferred to some extent from the behavior of MATLAB's optimization toolbox. For the methods of up to fourth-order, the software is able to repeatedly converge to the optimal solution from a wide range of initial guesses. Hence we expect that these methods are optimal, or very nearly so. For methods of fifth- and sixth-order, the behavior of MATLAB's toolbox is more erratic and it is difficult to determine how close to optimal the methods are. By comparing them with methods of the same number of stages and lower order, however, we see that in most cases the SSP coefficients of the globally optimal methods cannot be dramatically larger than those we have found.

Numerical experiments confirm the theoretical properties of these methods. The implicit SSP Runge–Kutta methods we found have SSP coefficients significantly larger than those of optimal explicit methods for a given number of stages and order of accuracy. Furthermore, we have provided implicit methods of orders five and six, whereas explicit methods can have order at most four. However, these advantages in accuracy and timestep restriction must be weighed against the cost of solving the implicit set of equations. In the future we plan to compare in practice the relative efficiency of these methods with explicit methods.

Acknowledgements

The authors thank Luca Ferracina and Marc Spijker for sharing their results [7] before publication, and the anonymous referees for very thorough comments that have led to many improvements in the paper.

Appendix A. Coefficients of some optimal methods

Some of the methods discussed in the paper are given below. The various other methods mentioned throughout the paper can be found in [18,19].

Table 7
Non-zero coefficients of the optimal 4-stage method of order 4

$\mu_{11} = 0.119309657880174$	$\mu_{33} = 0.070606483961727$	$\mu_{52} = 0.034154109552284$	$\lambda_{43} = 0.939878564212065$
$\mu_{21} = 0.226141632153728$	$\mu_{43} = 0.212545672537219$	$\mu_{54} = 0.181099440898861$	$\lambda_{51} = 0.048147179264990$
$\mu_{22} = 0.070605579799433$	$\mu_{44} = 0.119309875536981$	$\lambda_{21} = 1$	$\lambda_{52} = 0.151029729585865$
$\mu_{32} = 0.180764254304414$	$\mu_{51} = 0.010888081702583$	$\lambda_{32} = 0.79934089350488$	$\lambda_{54} = 0.8008230911491455$

Table 8
Non-zero coefficients of the optimal 5-stage method of order 5

$\mu_{21} = 0.107733237609082$	$\mu_{44} = 0.079032059834967$	$\mu_{65} = 0.194911604040485$	$\lambda_{52} = 0.036331447472278$
$\mu_{22} = 0.107733237609079$	$\mu_{51} = 0.040294985548405$	$\lambda_{21} = 0.344663606249694$	$\lambda_{53} = 0.077524819660326$
$\mu_{31} = 0.000009733684024$	$\mu_{52} = 0.011356303341111$	$\lambda_{31} = 0.000031140312055$	$\lambda_{54} = 0.706968664080396$
$\mu_{32} = 0.205965878618791$	$\mu_{53} = 0.024232322953809$	$\lambda_{32} = 0.658932601159987$	$\lambda_{63} = 0.255260385110718$
$\mu_{33} = 0.041505157180052$	$\mu_{54} = 0.220980752503271$	$\lambda_{41} = 0.035170229692428$	$\lambda_{64} = 0.075751744720289$
$\mu_{41} = 0.010993335656900$	$\mu_{55} = 0.098999612937858$	$\lambda_{42} = 0.000000100208717$	$\lambda_{65} = 0.623567413728619$
$\mu_{42} = 0.000000031322743$	$\mu_{63} = 0.079788022937926$	$\lambda_{43} = 0.786247596634378$	
$\mu_{43} = 0.245761367350216$	$\mu_{64} = 0.023678103998428$	$\lambda_{51} = 0.128913001605754$	

Table 9
Non-zero coefficients of the optimal 9-stage method of order 6

$\mu_{21} = 0.060383920365295$	$\mu_{77} = 0.019840674620006$	$\mu_{10,7} = 0.017872872156132$	$\lambda_{82} = 0.000000092581509$
$\mu_{22} = 0.060383920365140$	$\mu_{81} = 0.000000149127775$	$\mu_{10,8} = 0.027432316305282$	$\lambda_{83} = 0.198483904509141$
$\mu_{31} = 0.000000016362287$	$\mu_{82} = 0.000000015972341$	$\mu_{10,9} = 0.107685980331284$	$\lambda_{84} = 0.099500236576982$
$\mu_{32} = 0.119393671070984$	$\mu_{83} = 0.034242827620807$	$\lambda_{21} = 0.350007201986739$	$\lambda_{85} = 0.000000002211499$
$\mu_{33} = 0.047601859039825$	$\mu_{84} = 0.017165973521939$	$\lambda_{31} = 0.000000094841777$	$\lambda_{86} = 0.007174780797111$
$\mu_{42} = 0.000000124502898$	$\mu_{85} = 0.000000000381532$	$\lambda_{32} = 0.692049215977999$	$\lambda_{87} = 0.694839938634174$
$\mu_{43} = 0.144150297305350$	$\mu_{86} = 0.001237807078917$	$\lambda_{42} = 0.000000721664155$	$\lambda_{91} = 0.000000420876394$
$\mu_{44} = 0.016490678866732$	$\mu_{87} = 0.119875131948576$	$\lambda_{43} = 0.835547641163090$	$\lambda_{92} = 0.000002244169749$
$\mu_{51} = 0.014942049029658$	$\mu_{88} = 0.056749019092783$	$\lambda_{51} = 0.086609559981880$	$\lambda_{93} = 0.002320726117116$
$\mu_{52} = 0.033143125204828$	$\mu_{91} = 0.000000072610411$	$\lambda_{52} = 0.192109628653810$	$\lambda_{94} = 0.000634542179300$
$\mu_{53} = 0.020040368468312$	$\mu_{92} = 0.000000387168511$	$\lambda_{53} = 0.116161276908552$	$\lambda_{95} = 0.074293052394615$
$\mu_{54} = 0.095855615754989$	$\mu_{93} = 0.000400376164405$	$\lambda_{54} = 0.555614071795216$	$\lambda_{96} = 0.066843552689032$
$\mu_{55} = 0.053193337903908$	$\mu_{94} = 0.000109472445726$	$\lambda_{61} = 0.000037885959162$	$\lambda_{97} = 0.000167278634186$
$\mu_{61} = 0.000006536159050$	$\mu_{95} = 0.012817181286633$	$\lambda_{62} = 0.004669151960107$	$\lambda_{98} = 0.834466572009306$
$\mu_{62} = 0.000805531139166$	$\mu_{96} = 0.011531979169562$	$\lambda_{63} = 0.088053362494510$	$\lambda_{10,1} = 0.009141400274516$
$\mu_{63} = 0.015191136635430$	$\mu_{97} = 0.000028859233948$	$\lambda_{64} = 0.317839263219390$	$\lambda_{10,2} = 0.000051643216195$
$\mu_{64} = 0.054834245267704$	$\mu_{98} = 0.143963789161172$	$\lambda_{65} = 0.519973146034093$	$\lambda_{10,3} = 0.000018699502726$
$\mu_{65} = 0.089706774214904$	$\mu_{99} = 0.060174596046625$	$\lambda_{71} = 0.000035341304071$	$\lambda_{10,4} = 0.000000360342058$
$\mu_{71} = 0.000006097150226$	$\mu_{10,1} = 0.001577092080021$	$\lambda_{72} = 0.108248004479122$	$\lambda_{10,5} = 0.052820347381733$
$\mu_{72} = 0.018675155382709$	$\mu_{10,2} = 0.000008909587678$	$\lambda_{73} = 0.150643488255346$	$\lambda_{10,6} = 0.050394050390558$
$\mu_{73} = 0.025989306353490$	$\mu_{10,3} = 0.000003226074427$	$\lambda_{74} = 0.001299063147749$	$\lambda_{10,7} = 0.103597678603687$
$\mu_{74} = 0.000224116890218$	$\mu_{10,4} = 0.000000062166910$	$\lambda_{75} = 0.000727575773504$	$\lambda_{10,8} = 0.159007699664781$
$\mu_{75} = 0.000125522781582$	$\mu_{10,5} = 0.009112668630420$	$\lambda_{76} = 0.727853067743022$	$\lambda_{10,9} = 0.624187175011814$
$\mu_{76} = 0.125570620920810$	$\mu_{10,6} = 0.008694079174358$	$\lambda_{81} = 0.000000864398917$	

References

[1] J.C. Butcher, On the implementation of implicit Runge–Kutta methods, BIT 17 (1976) 375–378.
[2] G. Dahlquist, R. Jeltsch, Generalized disks of contractivity for explicit and implicit Runge–Kutta methods, Tech. rep., Dept. of Numer. Anal. and Comp. Sci., Royal Inst. of Techn., Stockholm, 1979.
[3] K. Dekker, J.G. Verwer, Stability of Runge–Kutta Methods for Stiff Nonlinear Differential Equations, CWI Monographs, vol. 2, North-Holland Publishing Co., Amsterdam, 1984.
[4] L. Ferracina, M.N. Spijker, Stepsize restrictions for the total-variation-diminishing property in general Runge–Kutta methods, SIAM Journal of Numerical Analysis 42 (2004) 1073–1093.
[5] L. Ferracina, M.N. Spijker, Computing optimal monotonicity-preserving Runge–Kutta methods, Tech. Rep. MI2005-07, Mathematical Institute, Leiden University, 2005.
[6] L. Ferracina, M.N. Spijker, An extension and analysis of the Shu–Osher representation of Runge–Kutta methods, Mathematics of Computation 249 (2005) 201–219.
[7] L. Ferracina, M.N. Spijker, Strong stability of singly-diagonally-implicit Runge–Kutta methods, Applied Numerical Mathematics 58 (2007) 1675–1686.
[8] S. Gottlieb, On high order strong stability preserving Runge–Kutta and multi step time discretizations, Journal of Scientific Computing 25 (2005) 105–127.
[9] S. Gottlieb, C.-W. Shu, Total variation diminishing Runge–Kutta schemes, Mathematics of Computation 67 (1998) 73–85.
[10] S. Gottlieb, C.-W. Shu, E. Tadmor, Strong stability preserving high-order time discretization methods, SIAM Review 43 (2001) 89–112.
[11] E. Hairer, S.P. Nørsett, G. Wanner, Solving Ordinary Differential Equations I: Nonstiff Problems, second ed., Springer Series in Computational Mathematics, vol. 8, Springer-Verlag, Berlin, 1993.
[12] E. Hairer, G. Wanner, Solving Ordinary Differential Equations. II: Stiff and Differential-Algebraic Problems, second ed., Springer Series in Computational Mathematics, vol. 14, Springer-Verlag, Berlin, 1996.
[13] I. Higueras, On strong stability preserving time discretization methods, Journal of Scientific Computing 21 (2004) 193–223.
[14] I. Higueras, Representations of Runge–Kutta methods and strong stability preserving methods, SIAM Journal of Numerical Analysis 43 (2005) 924–948.
[15] W. Hundsdorfer, J. Verwer, Numerical Solution of Time-Dependent Advection–Diffusion–Reaction Equations, Springer Series in Computational Mathematics, vol. 14, Springer, 2003.
[16] D.I. Ketcheson, An algebraic characterization of strong stability preserving Runge–Kutta schemes, B.Sc. thesis, Brigham Young University, Provo, Utah, USA, 2004.
[17] D.I. Ketcheson, Highly efficient strong stability preserving Runge–Kutta methods with low-storage implementations, SIAM Journal on Scientific Computing (2008), doi: 10.1137/07070485X.
[18] D.I. Ketcheson, High order numerical methods for wave propagation, unpublished doctoral thesis, University of Washington.

[19] D.I. Ketcheson, C.B. Macdonald, S. Gottlieb, Numerically optimal SSP Runge–Kutta methods, website, http://www.cfm.brown.edu/people/sg/ssp.html, 2007.
[20] D.I. Ketcheson, A.C. Robinson, On the practical importance of the SSP property for Runge–Kutta time integrators for some common Godunov-type schemes, International Journal for Numerical Methods in Fluids 48 (2005) 271–303.
[21] J.F.B.M. Kraaijevanger, Absolute monotonicity of polynomials occurring in the numerical solution of initial value problems, Numerische Mathematik 48 (1986) 303–322.
[22] J.F.B.M. Kraaijevanger, Contractivity of Runge–Kutta methods, BIT 31 (1991) 482–528.
[23] R.J. LeVeque, Finite Volume Methods for Hyperbolic Problems, Cambridge University Press, 2002.
[24] C.B. Macdonald, Constructing high-order Runge–Kutta methods with embedded strong-stability-preserving pairs, Master's thesis, Simon Fraser University, August 2003.
[25] C.B. Macdonald, S. Gottlieb, S.J. Ruuth, A numerical study of diagonally split Runge–Kutta methods for PDEs with discontinuities, Journal of Scientific Computing (2008), doi:10.1007/s10915-007-9180-6.
[26] S.J. Ruuth, Global optimization of explicit strong-stability-preserving Runge–Kutta methods, Mathematics of Computation 75 (253) (2006) 183–207 (electronic).
[27] N.V. Sahinidis, M. Tawarmalani, BARON 7.2: Global Optimization of Mixed-Integer Nonlinear Programs, User's Manual, available at http://www.gams.com/dd/docs/solvers/baron.pdf, 2004.
[28] C.-W. Shu, Total-variation diminishing time discretizations, SIAM Journal on Scientific and Statistical Computing 9 (1988) 1073–1084.
[29] C.-W. Shu, A survey of strong stability-preserving high-order time discretization methods, in: Collected Lectures on the Preservation of Stability under Discretization, SIAM, Philadelphia, PA, 2002.
[30] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, Journal of Computational Physics 77 (1988) 439–471.
[31] M.N. Spijker, Contractivity in the numerical solution of initial value problems, Numerische Mathematik 42 (1983) 271–290.
[32] R.J. Spiteri, S.J. Ruuth, A new class of optimal high-order strong-stability-preserving time discretization methods, SIAM Journal of Numerical Analysis 40 (2002) 469–491.
[33] R.J. Spiteri, S.J. Ruuth, Nonlinear evolution using optimal fourth-order strong-stability-preserving Runge–Kutta methods, Mathematics and Computers in Simulation 62 (2003) 125–135.
[34] J.A. van de Griend, J.F.B.M. Kraaijevanger, Absolute monotonicity of rational functions occurring in the numerical solution of initial value problems, Numerische Mathematik 49 (1986) 413–424.