Optimal Implicit Strong Stability Preserving Runge–Kutta Methods

David I. Ketcheson*, Colin B. Macdonald†, Sigal Gottlieb‡

August 3, 2007

Abstract

Strong stability preserving (SSP) time discretizations were developed for use with the spatial discretization 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 time-step restriction. We turn to implicit Runge–Kutta methods to alleviate this time step restriction, and present implicit strong stability preserving (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 implicit SSP methods) and up to eleven stages. The numerically optimal Runge–Kutta methods found are all diagonally implicit, leading us to conjecture that optimal implicit SSP Runge–Kutta methods are diagonally implicit. These methods allow a significant increase in SSP time-step limit, compared to explicit methods of the same order and

*Department of Applied Mathematics, University of Washington, Seattle, WA 98195-2420 (ketch@amath.washington.edu). The work of this author was funded by a U.S. Dept. of Energy Computational Science Graduate Fellowship.

†Department of Mathematics, Simon Fraser University, Burnaby, British Columbia, V5A 1S6 Canada (cbm@sfu.ca). The work of this author was partially supported by a grant from NSERC Canada and a scholarship from the Pacific Institute of Mathematics (PIMS).

‡Department of Mathematics, University of Massachusetts Dartmouth, North Dartmouth MA 02747. This work was supported by AFOSR grant number FA9550-06-1-0255

number of stages. Numerical studies verify the order of the methods and the SSP property for several test cases.

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 time-stepping [26, 24, 9, 10]. These methods were originally developed for 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

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

arising from the discretization of the spatial derivatives in the PDE

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

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

$$||\boldsymbol{u}^{n+1}|| \leq ||\boldsymbol{u}^n|| \tag{3}$$

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

$$\Delta t \leq \Delta t_{\mathrm{FE}}. \tag{4}$$

If we write an explicit Runge–Kutta method in the now-standard Shu–Osher form [26]

$$\begin{aligned}
\boldsymbol{u}^{(0)} &= \boldsymbol{u}^n, \\
\boldsymbol{u}^{(i)} &= \sum_{k=0}^{i-1} \left(\alpha_{i,k}\boldsymbol{u}^{(k)} + \Delta t \beta_{i,k} F(\boldsymbol{u}^{(k)})\right), \quad \alpha_{i,k} \geq 0, \quad i = 1, \ldots, s, \\
\boldsymbol{u}^{n+1} &= \boldsymbol{u}^{(s)}.
\end{aligned} \tag{5}$$

consistency requires that $\sum_{k=0}^{i-1} \alpha_{i,k} = 1$. Thus, if $\alpha_{i,k} \geq 0$ and $\beta_{i,k} \geq 0$, all the intermediate stages in (5), $\boldsymbol{u}^{(i)}$, are simply convex combinations of forward Euler operators, each with Δt replaced by $\frac{\beta_{i,k}}{\alpha_{i,k}}\Delta t$. Therefore, any bound on a norm, semi-norm (or, in fact, any convex functional) of the solution that is satisfied by the forward Euler method will be preserved by the Runge–Kutta method, under the time step restriction $\frac{\beta_{i,k}}{\alpha_{i,k}}\Delta t \leq \Delta t_{\mathrm{FE}}$, or equivalently

$$\Delta t \leq \min \frac{\alpha_{i,k}}{\beta_{i,k}} \Delta t_{FE}, \tag{6}$$

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 which has the required stability properties when coupled with forward Euler.

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

$$\Delta t \leq c\Delta t_{FE}. \tag{7}$$

If $c > 0$, the method is said to be strong stability preserving.

If the forward Euler solution is contractive, i.e. any two numerical solutions $\boldsymbol{u}, \boldsymbol{v}$ of (1) satisfy

$$||\boldsymbol{u}^{n+1} - \boldsymbol{v}^{n+1}|| \leq ||\boldsymbol{u}^n - \boldsymbol{v}^n||, \tag{8}$$

then an SSP method will preserve this property as well, under the modified time step restriction (7) [20]. It is particularly useful to consider $\boldsymbol{v}$ to be a perturbation of $\boldsymbol{u}$ due to numerical errors or errors in the initial conditions. Then contractivity implies that these errors do not grow unduly as the solution is integrated.

If a particular spatial discretization coupled with the explicit forward Euler method satisfies a strong stability property for some time-step restriction, then the implicit backward Euler method satisfies the same strong stability property, for any positive time-step [16]. However, this unconditional SSP property does not extend to higher order implicit methods. All SSP Runge–Kutta methods of order greater than one suffer from some time-step restriction [20, 10]. Much of the research in this field is devoted to finding methods that are optimal in terms of time-step restriction. For this purpose, various implicit extensions and generalizations of the Shu–Osher form have been introduced [10, 8, 6, 16]. The most general of these, and the form we use in this paper, was introduced independently in [6] and [16]. We will refer to it as the *modified Shu–Osher form*.

As discussed in Section 2, the search for new SSP methods is facilitated by the connection between the SSP condition and the contractivity and absolute monotonicity conditions for Runge–Kutta methods [28, 20, 14, 16, 4, 6]. For a more detailed description of the Shu–Osher form, its generalization to implicit methods, and the connection with absolute monotonicity, we refer the interested reader to [26, 20, 9, 10, 25, 4, 14, 17, 6, 8, 15, 27]. For an investigation of the effect of the SSP property in practice, see [19].

The structure of this paper is as follows. In Section 2 we use some of the major results in contractivity theory [20, 3] 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 order up to six and up to eleven stages, found using numerical optimization routines. Some of these numerically optimal methods are also proved to be optimal. We will see that the best implicit Runge–Kutta methods, as found by numerical search, are all diagonally implicit, and those of order two and three are singly diagonally implicit. In Section 4 we present numerical experiments using the 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 *absolute monotonicity* and *contractivity*. These connections have been extensively explored [14, 16, 4, 6, 17]. In this section we collect the results on order barriers and other limitations found in contractivity theory [20], and use these to draw conclusions about the class of implicit SSP Runge–Kutta methods. To facilitate the discussion, we first present two important 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

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

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

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 (9) to be accurate of order p, the coefficients $\mathbf{K}$ must satisfy order conditions (see, e.g. [12]) denoted by

$$\Phi_p(\mathbf{K}) = 0.$$

A generalization that applies to implicit as well as explicit methods was introduced in [6, 16] 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

$$\boldsymbol{\lambda} = \begin{bmatrix} \boldsymbol{\lambda}_0 \\ \boldsymbol{\lambda}_1 \end{bmatrix}, \quad \boldsymbol{\lambda}_0 = \begin{bmatrix} \lambda_{11} & \cdots & \lambda_{1s} \\ \vdots & & \vdots \\ \lambda_{s1} & \cdots & \lambda_{ss} \end{bmatrix}, \quad \boldsymbol{\lambda}_1 = (\lambda_{s+1,1}, \ldots, \lambda_{s+1,s}), \tag{10a}$$

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_0 \\ \boldsymbol{\mu}_1 \end{bmatrix}, \quad \boldsymbol{\mu}_0 = \begin{bmatrix} \mu_{11} & \cdots & \mu_{1s} \\ \vdots & & \vdots \\ \mu_{s1} & \cdots & \mu_{ss} \end{bmatrix}, \quad \boldsymbol{\mu}_1 = (\mu_{s+1,1}, \ldots, \mu_{s+1,s}). \tag{10b}$$

These arrays define the method

$$\boldsymbol{y}^i = \left(1 - \sum_{j=1}^{s} \lambda_{ij}\right) \boldsymbol{u}^n + \sum_{j=1}^{s} \lambda_{ij} \boldsymbol{y}^j + \Delta t \mu_{ij} F(t_n + c_j \Delta t, \boldsymbol{y}^j), \quad (1 \le i \le s), \tag{11a}$$

$$\boldsymbol{u}^{n+1} = \left(1 - \sum_{j=1}^{s} \lambda_{s+1,j}\right) \boldsymbol{u}^n + \sum_{j=1}^{s} \lambda_{s+1,j} \boldsymbol{y}^j + \Delta t \mu_{s+1,j} F(t_n + c_j \Delta t, \boldsymbol{y}^j). \tag{11b}$$

Any $s \times s$ $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ in (5) can be written in the form (11) as

$$\boldsymbol{\lambda} = \begin{bmatrix} 0 \\ \boldsymbol{\alpha} \end{bmatrix}, \quad \boldsymbol{\mu} = \begin{bmatrix} 0 \\ \boldsymbol{\beta} \end{bmatrix},$$

thus the modified Shu–Osher form is more general. It also leads to simpler expressions in what follows.

Comparison of the Butcher representation (9) with the modified Shu–Osher representation (11) reveals that the two are related by

$$\boldsymbol{\mu} = \mathbf{K} - \boldsymbol{\lambda}\mathbf{A}. \tag{12}$$

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

$$\mathbf{A} = (\mathbf{I} - \boldsymbol{\lambda}_0)^{-1} \boldsymbol{\mu}_0, \tag{13a}$$

$$\boldsymbol{b}^{\mathrm{T}} = \boldsymbol{\mu}_1 + \boldsymbol{\lambda}_1 (\mathbf{I} - \boldsymbol{\lambda}_0)^{-1} \boldsymbol{\mu}_0, \tag{13b}$$

where $\boldsymbol{\lambda}_0$, $\boldsymbol{\mu}_0$, $\boldsymbol{\lambda}_1$, $\boldsymbol{\mu}_1$ are the submatrices of $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ as defined in (10).

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

As the modified Shu–Osher form (11) is not unique, any such particular representation may give a timestep restriction based on (6) that is not optimal. The optimal timestep restriction turns out to be related to the *radius of absolute monotonicity* $R(\mathbf{K})$, introduced originally by Kraaijevanger [20]. A more convenient, equivalent definition of $R(\mathbf{K})$ is given in [4, 14]:

Definition 2 Radius of absolute monotonicity *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 such that $(\mathbf{I}+r\mathbf{A})^{-1}$ exists and*

$$\begin{aligned} \mathbf{K}(\mathbf{I}+r\mathbf{A})^{-1} &\geq 0, \\ r\mathbf{K}(\mathbf{I}+r\mathbf{A})^{-1}\boldsymbol{e}_s &\leq \boldsymbol{e}_{s+1}. \end{aligned} \tag{14}$$

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

The following result follows from Propositions 2.2 and 2.7 of [16] or from Theorem 3.4 of [6]:

Theorem 1 *Let a Runge–Kutta method be given with Butcher array $\mathbf{K}$. Let c denote the SSP coefficient from definition 1. Let $R(\mathbf{K})$ denote the radius of absolute monotonicity defined by (14). Then*

$$c = R(\mathbf{K}). \tag{15}$$

Furthermore, there exists a modified Shu–Osher representation $(\boldsymbol{\lambda}, \boldsymbol{\mu})$ such that (12) holds and

$$c = \min_{i,j;i\neq j} \frac{\lambda_{i,j}}{\mu_{i,j}}, \quad \forall \mu_{i,j} \neq 0. \tag{16}$$

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

$$\Delta t \leq R(\mathbf{K})\Delta t_{FE}. \tag{17}$$

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 autonomous system of ODEs

$$\boldsymbol{u}_t = \mathbf{L}\boldsymbol{u}, \tag{18}$$

any Runge–Kutta method reduces to the iteration

$$\boldsymbol{u}^{n+1} = \phi(\Delta t \mathbf{L})\boldsymbol{u}^n, \tag{19}$$

where ϕ is a rational function called the *stability function* of the Runge–Kutta method [13].

Definition 3 Linear Strong Stability Preservation *For $\Delta t_{FE} > 0$, let $\mathcal{L}(\Delta t_{FE})$ denote the set of all pairs $(L, ||\cdot||)$ where the matrix $L \in R^{m\times m}$ and convex functional $||\cdot||$ are such that the numerical solution obtained by forward Euler integration of (18) satisfies $||\boldsymbol{u}^{n+1}|| \leq ||\boldsymbol{u}^n||$ whenever $\Delta t \leq \Delta t_{FE}$. Given a Runge–Kutta method, the linear SSP coefficient of the method is the largest constant $c_{lin} \geq 0$ such that the numerical solution obtained with the Runge–Kutta method satisfies $||\boldsymbol{u}^{n+1}|| \leq ||\boldsymbol{u}^n||$ for all $(L, ||\cdot||) \in \mathcal{L}(\Delta t_{FE})$ whenever*

$$\Delta t \leq c_{lin}\Delta t_{FE}. \tag{20}$$

If $c_{lin} > 0$, the method is said to be linearly strong stability preserving.

When solving (18), the timestep restriction for strong stability preservation depends on the radius of absolute monotonicity of ϕ.

Definition 4 Linear radius of absolute monotonicity *The linear radius of absolute monotonicity $R(\phi)$ of an Runge–Kutta method with stability function ϕ is the largest value of r such that $\phi(x)$ and all of its derivatives are nonnegative for $x \in (-r, 0]$.*

The following result is originally due to Spijker [28]:

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). Let $R(\phi)$ denote the linear radius of absolute monotonicity (definition 4). Then*

$$c_{lin} = R(\phi). \tag{21}$$

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

$$\Delta t \leq R(\phi)\Delta t_{FE}. \tag{22}$$

Since $\mathcal{L}(h) \subset \mathcal{F}(h)$, clearly $c_{\text{lin}} \leq c$, so $R(\mathbf{K}) \leq R(\phi)$. Optimal values of $R(\phi)$ for implicit Runge–Kutta methods with up to two stages and third order of accuracy were found in [29].

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. These restrictions have been carefully studied in [28, 3, 20]. We will review these results and draw a few additional conclusions. These results will guide our search for optimal methods in the next section.

Result 1 *([28] Thm. 1.3) Any Runge–Kutta method of order $p > 1$ has a finite radius of absolute monotonicity $R(\mathbf{K}) < \infty$.*

This is a disappointing result, which shows us that for methods of order greater than one we cannot avoid strong stability timestep restrictions altogether by using implicit methods (in contrast with linear stability where some high-order implicit methods (viz., the A-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.

The following result gives lower bounds on the coefficients that are useful in numerical searches. It is also useful in proving subsequent results.

Result 2 *([20] Thm. 4.2) Any Runge–Kutta method with positive radius of absolute monotonicity $R(\mathbf{K}) > 0$, must have all non-negative coefficients $\mathbf{A} \geq 0$ and positive weights $\boldsymbol{b} > 0$.*

The following three results deal with the stage order $\tilde{p}$ of Runge–Kutta methods. The stage order is a lower bound on the order of convergence when a method is applied to arbitrarily stiff problems [3]. Thus low stage order may lead to slow convergence (i.e., *order reduction*) when computing solutions of stiff ODEs. Although this is also the case for explicit methods, which have stage order one (see Result 4 below), this does not seem to be problematic for semi-discretizations of hyperbolic PDEs, where we observe that explicit methods do achieve their classical order of accuracy. It is possible that this may pose a larger problem for implicit methods when large timesteps are used, but we did not observe this in our numerical experiments in Section 4.

Result 3 *([20] Thm. 8.5) A Runge–Kutta method with non-negative coefficients* $\mathbf{A} \geq 0$*, must have stage order* $\tilde{p} \leq 2$*. If* $\tilde{p} = 2$*, then* $\mathbf{A}$ *must have a zero row.*

When constructing classical high-order implicit Runge–Kutta methods it is common to consider collocation methods; however since collocation methods have stage order equal to at least the number of stages [3], the result above implies that collocation methods with more than two stages cannot be SSP. This implies that $R(\mathbf{K}) = 0$ for most of the well-known implicit Runge–Kutta methods, such as the Gauss-Legendre, Radau IIA, and Lobatto IIIA methods with more than two stages, as well as the Lobatto IIIC methods with more than three stages. It is not difficult to show, as well, that the Radau IA methods with more than three stages and the Lobatto IIIB methods with more than four stages have stage order greater than two and hence $R(\mathbf{K}) = 0$. Most of the low stage number methods in these classes also have negative coefficients, so that many of the classical algebraically stable methods have $R(\mathbf{K}) = 0$. Notable exceptions include the 1-stage Gauss-Legendre, Radau IA, and Radau IIA methods and the 2-stage Lobatto IIIA and Lobatto IIIB methods.

When dealing with explicit methods, stage order is limited whether or not one requires non-negative coefficients:

Result 4 *([2] Thm. 4.4) Explicit Runge–Kutta methods all have stage order* $\tilde{p} = 1$*.*

For SSP methods, the stage order restriction leads to restrictions on the classical order as well:

Result 5 *([11]) For a general Runge–Kutta method, if all the weights* $\boldsymbol{b} > 0$ *then the stage order must satisfy* $\tilde{p} \geq \lfloor \frac{p-1}{2} \rfloor$*.*

Result 6 *([20] Lemma 8.6 and Corollary 8.7) Any SSP Runge–Kutta method with* $R(\mathbf{K}) > 0$ *has order* $p \leq 4$ *if it is explicit and* $p \leq 6$ *if it is implicit. Furthermore, if* $p \geq 5$*, then* $\mathbf{A}$ *has a zero row.*

The negative implications in Result 6 stem from the conditions $\mathbf{A} \geq 0$, $\boldsymbol{b} > 0$ in Result 2. Non-negativity of $\mathbf{A}$ leads to low stage order (Result 3), while positivity of $\boldsymbol{b}$ leads to a limit on the classical order (Result 5) relative to the stage order. The result is a severe restriction on the classical order of SSP methods.

2.4 Barriers for Singly Implicit and Singly 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 PDEs, m is typically very large and the system is generally 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. For details on efficient implementation of implicit Runge–Kutta methods see, e.g., [3].

Lemma 1 *(also appears in [7]) An SDIRK method with positive radius of absolute monotonicity $R(\mathbf{K}) > 0$ must have order $p \leq 4$.*

Proof: Assume $\mathbf{K}$ represents an SDIRK method with $R(\mathbf{K}) > 0$ and order $p > 4$. Result 6 means that $\mathbf{A}$ has a zero row. Since this is an SDIRK method all the diagonal entries are equal, so a zero row implies that all diagonal entries of $\mathbf{A}$ are zero, i.e., the method is explicit. However, Result 6 states that an explicit method cannot have order $p > 4$.

Result 7 *([29] Cor. 3.4) If a Runge–Kutta method has a positive radius of absolute monotonicity $R(\phi) > 0$ then its stability function ϕ has a real pole.*

Result 8 *([3] Thm. 3.5.11) If the stability function ϕ of a Runge–Kutta method has only real poles and numerator of degree s, then the order of the method $p \leq s+1$.*

Because the degree of the numerator of ϕ is at most the number of stages of the Runge–Kutta method [13], a consequence of these two results is

Corollary 1 *For SIRK and DIRK methods, if $R(\phi) > 0$, $p \leq s+1$.*

Proof: For SIRK methods, ϕ has a unique pole which by Result 7 must be real. For DIRK methods, the poles of ϕ are the diagonal entries of $\mathbf{A}$, which must be real. Hence Corollary 1 follows from Result 8.

Corollary 2 *If an s-stage SIRK method has positive radius of absolute monotonicity $R(\mathbf{K}) > 0$ and order $p \geq 5$, then its radius of absolute monotonicity is no larger than that of the optimal linearly SSP s-stage explicit method:*

$$R(\mathbf{K}) \leq \sup_{\phi \in \mathbb{P}_p} R(\phi) \tag{23}$$

where the supremum is taken over all polynomials ϕ of degree less than or equal to m that approximate the exponential function to order p near $x = 0$. Hence also the order is at most equal to the number of stages: $p \leq s$.

Proof: By Result 6, such methods have a zero row, so all eigenvalues of $\mathbf{A}$ must be zero. Then the stability function ϕ is a polynomial, and the result follows from the definition of $R(\phi)$.

Corollary 2 implies that $R(\mathbf{K})$ must be small for SIRK methods with $p > 4$.

3 Optimal Implicit Methods for Nonlinear Systems

In this section we present numerically optimal implicit 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 [23, 21, 22]. 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 optimal explicit and diagonally implicit SSP Runge–Kutta methods, and successfully found a solution at least as good as the 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 stages—may well 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 [7]. However, we have been able to find good 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

$$\max_{\mathbf{K}} r, \tag{24a}$$
$$\mathbf{K}(\mathbf{I}+r\mathbf{A})^{-1} \geq 0, \tag{24b}$$
$$r\mathbf{K}(\mathbf{I}+r\mathbf{A})^{-1}\boldsymbol{e}_s \leq \boldsymbol{e}_{s+1}, \tag{24c}$$
$$\Phi_p(\mathbf{K}) = 0, \tag{24d}$$

where the inequalities are understood component-wise and recall that $\Phi_p(\mathbf{K})$ represent 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, we are using this formulation to search for optimal explicit SSP methods [18].

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

$$\max_{\mathbf{K},\boldsymbol{\mu}} r, \tag{25a}$$
$$\boldsymbol{\mu} \geq 0, \tag{25b}$$
$$r\boldsymbol{\mu}\boldsymbol{e}_s \leq \boldsymbol{e}_{s+1}, \tag{25c}$$
$$\mathbf{K} = \boldsymbol{\mu}(\mathbf{I}+r\mathbf{A}), \tag{25d}$$
$$\Phi_p(\mathbf{K}) = 0. \tag{25e}$$

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 [23]. Note that $\boldsymbol{\mu}$ in (25) corresponds to $\boldsymbol{\mu}$ in 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 timestep 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 per stage of 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 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- and Third-order Methods

Optimizing over the class of all $s \leq 11$-stage implicit Runge–Kutta methods of second- or third-order, we have found that the optimal methods are identical to the optimal SDIRK methods found in [5, 7]. 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 optimal second-order method with s stages has $R(\mathbf{K}) = 2s$ and coefficients

$$\boldsymbol{\lambda} = \begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ & 1 & \ddots & \\ & & \ddots & 0 \\ & & & 1 \end{bmatrix}, \quad \boldsymbol{\mu} = \begin{bmatrix} \frac{1}{s} & & & \\ \frac{1}{s} & \frac{1}{s} & & \\ & \frac{1}{s} & \ddots & \\ & & \ddots & \frac{1}{s} \\ & & & \frac{1}{s} \end{bmatrix}. \tag{26}$$

The one- and two-stage methods in this family are proven optimal because they achieve the maximum linear radius of absolute monotonicity in their respective classes [29, 17]. In addition to these duplicating these results, BARON was used to prove that the $s = 3$ scheme is globally optimal. The $s = 1$ and $s = 2$ cases required only several seconds but the $s = 3$ case took much longer, requiring approximately 14 hours of CPU time.

We conjecture that these methods are optimal for all values of s, which would imply that the effective SSP coefficient of any SSP Runge–Kutta method of order greater than one is at most equal to two. In fact, van de Griend and Kraaijevanger [29] showed that the optimal $R(\phi) \geq 2s$ for second-order methods with s stages, and conjectured that $R(\phi) = 2s$. They proved the conjecture only in the one- and two-stage cases (which implies the optimality of the one- and two-stage methods of the form (26)). In fact the conjecture is false for the three-stage case, as we demonstrate with the following counterexample:

$$\phi = \frac{1 + \frac{7969150767159903}{18014398509481984}x + \frac{4716995547632067}{72057594037927936}x^2 + \frac{1867769670100979}{576460752303423488}x^3}{1 - \frac{313913991947565}{562949953421312}x + \frac{8869189497956419}{72057594037927936}x^2 - \frac{1762527965732417}{144115188075855872}x^3}.$$

Because the numerator and denominator have degree three and approximates the exponential to second-order near $x = 0$, this corresponds to the stability function of a second-order implicit Runge–Kutta method with $s = 3$ stages. This function was found by numerical search; using the algorithm in [29] we have verified that $R(\phi) \geq 6.77 > 2s$. Thus our conjecture cannot be proved by analyzing $R(\phi)$.

Finally, we note that the one-stage method of this class is the implicit midpoint rule, while the s-stage method is essentially 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 both algebraic stability and A-stability [13]. Of course, since they all have the same effective SSP coefficient $R(\mathbf{K})/s = 2$, they are all essentially equivalent.

For $p = 3$ the optimal methods have $R(\mathbf{K}) = s - 1 + \sqrt{s^2 - 1}$ and coefficients

$$\boldsymbol{\mu} = \begin{bmatrix} \mu_{11} & & & \\ \mu_{21} & \ddots & & \\ & \ddots & \mu_{11} & \\ & & \mu_{21} & \mu_{11} \\ & & & \mu_{s+1,s} \end{bmatrix}, \quad \boldsymbol{\lambda} = \begin{bmatrix} 0 & & & \\ 1 & \ddots & & \\ & \ddots & 0 & \\ & & 1 & 0 \\ & & & \lambda_{s+1,s} \end{bmatrix}, \qquad (27)$$

where

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

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

In this form these methods can easily be implemented with very modest storage requirements.

The two-stage method in this family is proven optimal because it achieves the maximum linear radius of absolute monotonicity for two-stage, third-order methods [29, 17]. BARON was used to prove global optimality for the third-order $s = 3$ case, requiring about 11 hours of CPU time.

3.2 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 $p \geq 5$ since in that case $\mathbf{A}$ must have a zero row (cf. 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 coefficient arrays of the numerically optimal fourth-order methods have only a few non-zero entries below the first subdiagonal, leading to very modest storage requirements even for methods with many stages.

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 optimal methods in Table 1. For comparison, the table also lists the effective SSP coefficients of the optimal SDIRK methods found in [7]. Our optimal DIRK methods have larger SSP coefficients in every case. Furthermore, these methods have representations that allow for very efficient implementation in terms of storage. The non-zero coefficients in these representations are given in Tables 5–13.

s	$R(\mathbf{K})$ DIRK	Effective SSP coefficient DIRK	SDIRK
3	2.05	0.66	0.59
4	4.42	1.11	1.05
5	6.04	1.21	1.15
6	7.80	1.30	1.26
7	9.19	1.31	1.24
8	10.67	1.33	1.28
9	12.04	1.34	-
10	13.64	1.36	-
11	15.18	1.38	-

Table 1: SSP coefficients and effective SSP coefficients of optimal fourth-order DIRK methods. For comparison we also list the effective SSP coefficients of the optimal fourth-order SDIRK methods of [7].

3.3 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 was able to converge to a local optimum, the optimal method was diagonally implicit. The inability of the algorithm to converge to these DIRK methods (or perhaps to even more optimal methods) when searching over the class of fully implicit methods represents a failure of the optimization software. This is not surprising because the optimization problems involved are highly nonlinear with many variables and many constraints. The application of more sophisticated software to this problem is an area of future research. Nevertheless, the observation that all observed local optima correspond to DIRK methods leads us to believe that the globally optimal methods are likely to be DIRK methods.

Any 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 are due to searches for methods with more stages that resulted in convergence to a reducible or nearly-reducible method. Due to the high nonlinearity of the problem for $p \geq 5$, we found it helpful to explicitly limit the step sizes used by `fmincon` in the final steps of optimization.

3.3.1 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_{12} and a_{13} are simultaneously nonnegative, so there are no SSP methods in this class.

Four to Eleven stages We list the timestep coefficients and effective SSP coefficients of the optimal DIRK methods for $4 \leq s \leq 9$ in Table 2. 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 Corollary 2. Our optimal DIRK methods have larger effective SSP coefficients in every case. The non-zero coefficients of these methods are given in Tables 14–18.

3.3.2 Sixth-order Methods

Kraaijevanger [20] proved the bound $p \leq 6$ for contractive methods (cf. 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 $p \leq 6$ for implicit SSP methods is sharp.

The non-existence 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. Corollary 1 implies that sixth-order SSP DIRK methods must have at least five stages, and Corollary 2 shows that

s	$R(\mathbf{K})$ DIRK5	Effective SSP coefficient DIRK5	SIRK5 (bound)
4	1.07	0.27	0
5	1.07	0.21	0.20
6	4.97	0.83	0.40
7	6.21	0.89	0.38
8	7.56	0.94	0.42
9	8.90	0.99	0.46
10	10.13	1.01	0.48
11	11.33	1.03	0.50

Table 2: Comparison of effective SSP coefficients of optimal fifth-order DIRK methods with upper bounds on effective SSP coefficients of fifth-order SIRK methods from Result 6. No 5-stage method was found with SSP coefficient larger than the optimal 4-stage method.

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 optimal methods for $6 \leq s \leq 11$ are listed in Tables 3–4. All of these methods are diagonally implicit. The non-zero coefficients of these methods are given in Tables 21–24. We were unable to find a seven-stage method with larger SSP coefficient than that of the six-stage method, or an eleven-stage method with larger effective SSP coefficient than that of the ten-stage method (although we did find methods with larger values of $R(\mathbf{K})$ in the latter case).

s	$R(\mathbf{K})$	Effective SSP coeff.
6	0.18	0.03
7	0.18	0.026
8	2.25	0.28
9	5.80	0.63
10	8.10	0.81
11	8.85	0.80

Table 3: Radius of absolute monotonicity for optimal sixth-order methods. No 7-stage method was found with SSP coefficient larger than the optimal 6-stage method.

	Implicit Methods					Explicit Methods		
$s \setminus p$	2	3	4	5	6	2	3	4
2	2	1.37	-	-	-	0.5	-	-
3	2	1.61	0.66	-	-	0.67	0.33	-
4	2	1.72	1.11	0.27		0.75	0.5	-
5	2	1.78	1.21	0.21*		0.8	0.53	0.30
6	2	1.82	1.30	0.83	0.03	0.83	0.59	0.38
7	2	1.85	1.31	0.89	0.026*	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

Table 4: 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. An asterisk indicates that no s-stage method was found with effective SSP coefficient greater than that of the $(s-1)$-stage method.

Table 4 summarizes the effective SSP coefficients of the optimal diagonally implicit methods for $2 \leq p \leq 6$ and $2 \leq s \leq 11$. For comparison, Table 4 also includes the effective SSP coefficients of the best known explicit methods, including results from the forthcoming paper [18].

4 Numerical Experiments

We focus our numerical experiments on linear and nonlinear hyperbolic problems in Sections 4.1 and 4.2. The computations in Section 4.1 were performed with MATLAB version 7.1 on a Mac G5; those in Section 4.2 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 optimal 8-stage method of order 5 is SSP85.

4.1 Linear Advection

The prototypical hyperbolic PDE is the linear wave equation,

$$u_t + au_x = 0, \qquad 0 \le x \le 2\pi. \tag{28}$$

We consider (28) 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 time-stepping 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 (1), rather than to the exact solution of the underlying PDE. In lieu of the exact solution we use a very accurate numerical solution obtained using MATLAB's ode45 solver with minimal tolerances ($\mathsf{AbsTol} = 1 \times 10^{-14}$, $\mathsf{RelTol} = 1 \times 10^{-13}$).

Figure 1 shows a convergence study for various optimal schemes for the problem (28) with a fixed Δx and smooth initial data

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

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

Now consider the same advection equation but with discontinuous data

$$u(x, 0) = \begin{cases} 1 & \text{if } \frac{\pi}{2} \le x \le \frac{3\pi}{2}, \\ 0 & \text{otherwise.} \end{cases} \tag{29}$$

Figure 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 time-step; that is as s increases, the error constant of the method decreases.

Figure 3 shows typical results on the discontinuous advection example using the third-order methods after a single time-step. In this case, we have $s = 2$ stages for the third-order scheme. We see that as the time-step is increased, the line steepens, forms a small step, which becomes an oscillation

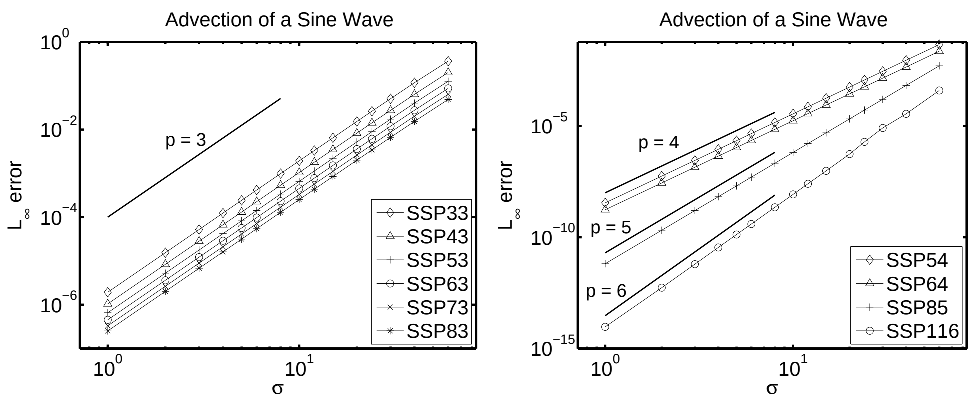


Figure 1: Convergence study for the third-order s-stage methods (left) and various optimal methods of orders four to six (right) for linear advection of a sine wave. The problem is solved to $t_f = 1$ using time-steps of length $\sigma \Delta t_{\text{FE}}$ for a range of values of σ. The spatial discretization uses $m = 120$ points and first-order upwinding.

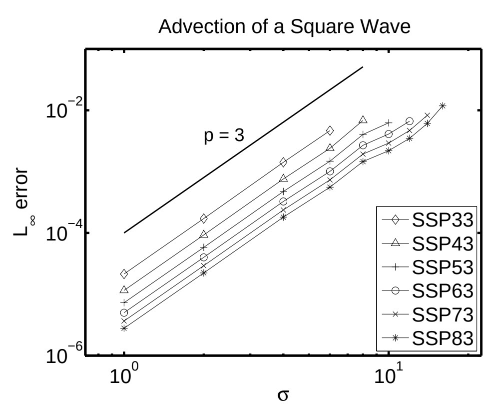
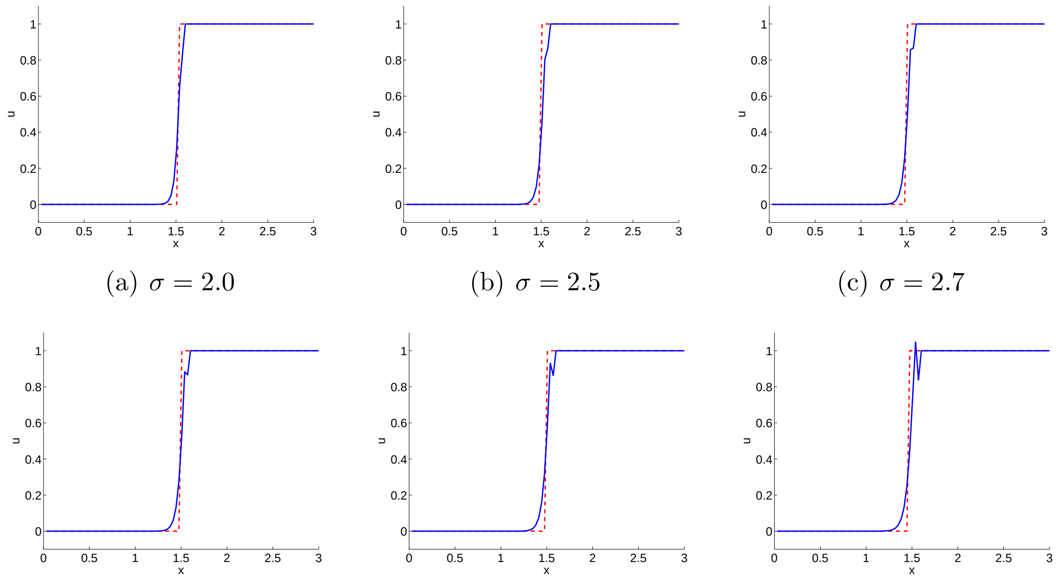


Figure 2: Convergence study for the third-order s-stage methods for linear advection of a square wave to $t_f = 1$ using N time-steps, $m = 64$ points and the first-order upwinding spatial discretization. Here σ measures the size of the step relative to Δt_{FE}.



(a) $\sigma = 2.0$ (b) $\sigma = 2.5$ (c) $\sigma = 2.7$

(d) $\sigma = 2.8$ (e) $\sigma = 3.0$ (f) $\sigma = 3.6$

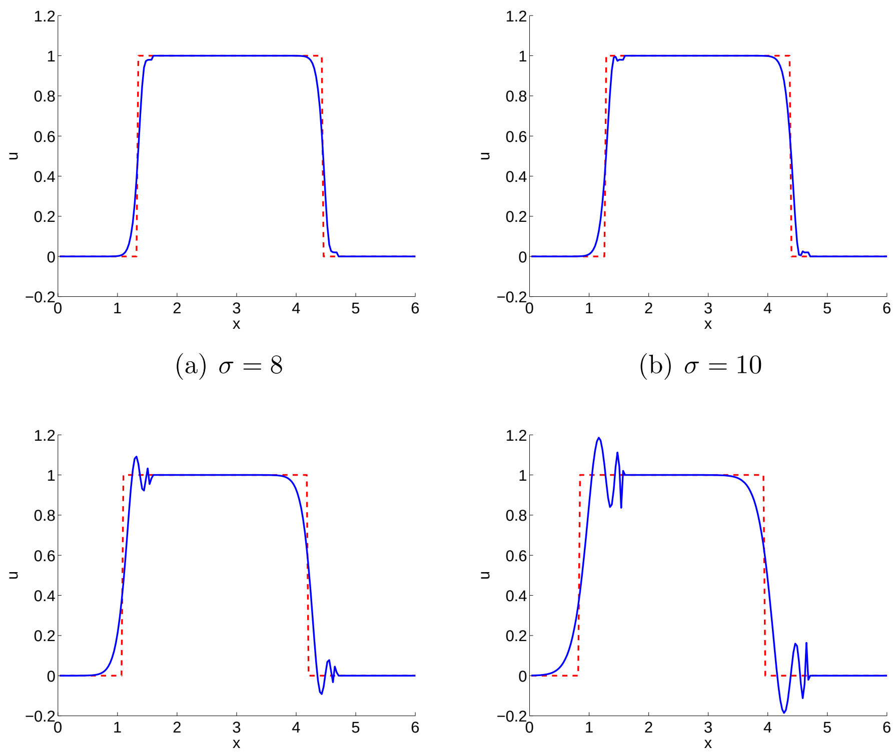
Figure 3: Solution of the linear advection problem after one time step with the third-order two-stage method in time. The spatial discretization is an upwind spatial differencing with 200 points in space. No oscillations form when the stability limit $R(\mathbf{K}) = 2.7320$ is respected (e.g., $\sigma = 2.7$) but oscillations immediately form when it is exceeded (e.g., $\sigma = 2.8$) and become worse as σ is increased further.

as the stability limit is exceeded, and worsens as the time-step is raised further. Figure 4 shows how the oscillations worsen as the size of the time-step increases (corresponding to $\sigma = 8, 10, 16, 24$) in the numerical results for the linear advection problem when using $s = 5$ stages with the third-order scheme.

4.2 Burgers' Equation

In this section we consider the inviscid Burgers' equation

$$u_t = -f(u)_x = -\left(\frac{1}{2}u^2\right)_x,$$



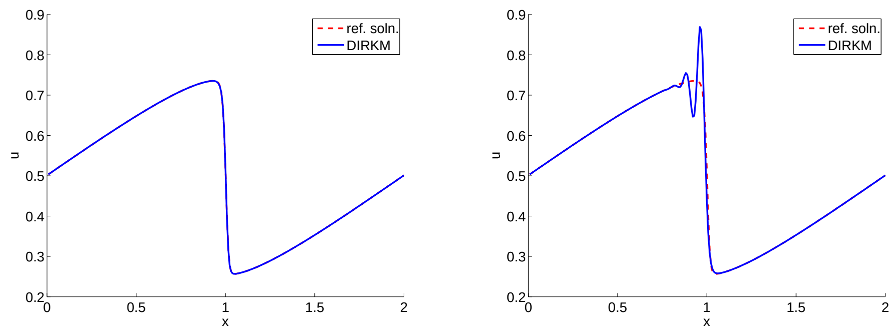
(a) $\sigma = 8$

(b) $\sigma = 10$

(c) $\sigma = 16$

(d) $\sigma = 24$

Figure 4: Solution of the linear advection problem after one time step with the third-order five-stage method in time ($R(\mathbf{K}) \approx 8.899$). The spatial discretization is first-order upwind spatial differencing with 200 points in space. No oscillations form when the stability limit is respected $\sigma = 8$ (top left) but minor oscillations begin to form when the stability limit is exceeded $\sigma = 10$ (top right), and become worse with increasing time-step corresponding to $\sigma = 16$ (bottom left), $\sigma = 24$ (bottom right).



(a) $\sigma = 8$ (b) $\sigma = 32$

Figure 5: Solution of Burgers' equation using a conservative first-order upwind spatial discretization with $m = 256$ points in space, and the third-order, five-stage SSP time stepping method, with final time $t_f = 2$.

with initial condition $u(0,x) = \frac{1}{2} - \frac{1}{4}\sin(\pi x)$ on the periodic domain $x \in [0,2)$. The solution is a right-travelling, steepening shock. We discretize $-f(u)_x$ using a conservative simple upwind approximation

$$-f(u)_x \approx -\frac{1}{\Delta x}\left(f(u_i) - f(u_{i-1})\right).$$

The convergence study in Figure 6 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. Figure 5 shows that when the time step is below the stability limit no oscillations appear, but when the stability limit is violated, oscillations are observed.

5 Conclusions and Future Work

Using numerical optimization, we have found numerically optimal 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 using BARON global optimization software. Remarkably, the numerically optimal methods are all diagonally implicit. Furthermore, all of the *local* optima found in our searches correspond to diagonally implicit methods. Based

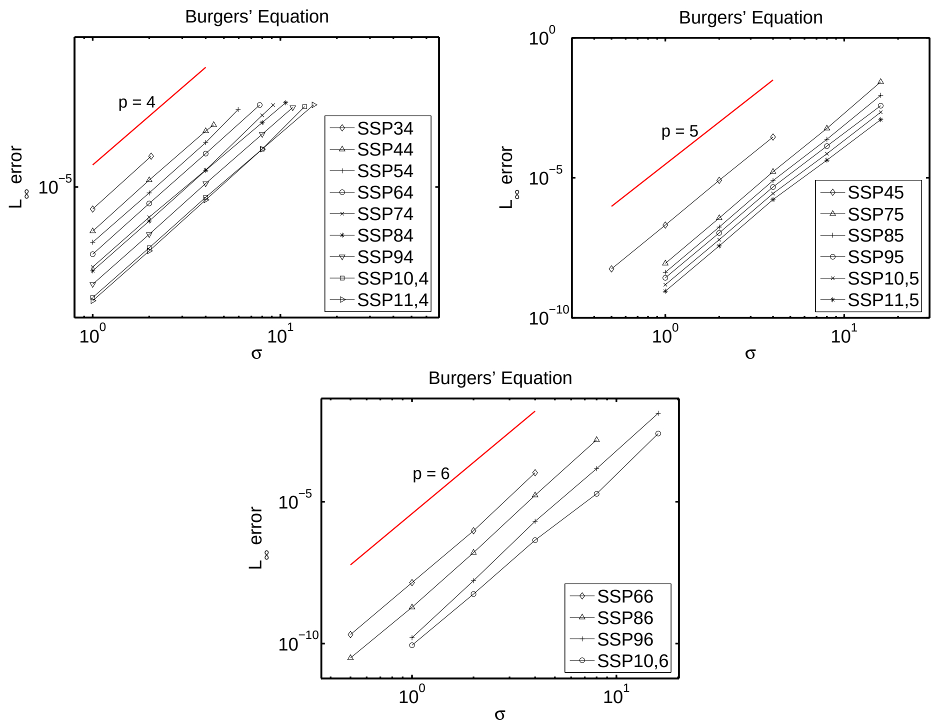


Figure 6: Convergence study for the optimal fourth-, fifth- and sixth-order schemes on Burgers' equation using a conservative first-order upwind spatial discretization with $m = 256$ points in space with final time $t_f = 2$.

on these results, we conjecture that the optimal implicit SSP Runge–Kutta methods of any number of stages are diagonally implicit. Also of note, the second- and third-order numerically optimal methods are singly diagonally implicit. Future work will 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.

Numerical experiments demonstrate that these methods converge at the expected order, do not seem to suffer from order reduction despite having lower stage order, and violate the SSP property precisely when the theoretical time-step restriction is exceeded. 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. However, these advantages in accuracy and time-step 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.

References

[1] J. C. Butcher, *On the implementation of implicit Runge–Kutta methods*, BIT, 17 (1976), pp. 375–378.

[2] G. Dahlquist and 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 and J. G. Verwer, *Stability of Runge–Kutta methods for stiff nonlinear differential equations*, Elsevier Science Publishers B. V., 1984.

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

[5] ——, *Computing optimal monotonicity-preserving Runge–Kutta methods*, (2005).

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

[7] L. Ferracina and M. Spjker, *Strong stability of singly-diagonally-implicit Runge–Kutta methods*. 2007.

[8] S. Gottlieb, *On high order strong stability preserving Runge–Kutta and multi step time discretizations*, Journal of Scientific Computing, 25 (2005), pp. 105–127.

[9] S. Gottlieb and C.-W. Shu, *Total variation diminishing Runge–Kutta schemes*, Mathematics of Computation, 67 (1998), pp. 73–85.

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

[11] E. Hairer, *Highest possible order of algebraically stable diagonally implicit Runge–Kutta methods*, BIT, 20 (1980), pp. 254–256.

[12] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving ordinary differential equations I: Nonstiff problems*, vol. 8 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, second ed., 1993.

[13] E. Hairer and G. Wanner, *Solving ordinary differential equations II: Stiff and differential-algebraic problems*, vol. 14 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 1991.

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

[15] ——, *Monotonicity for Runge–Kutta methods: inner product norms*, Journal of Scientific Computing, 24 (2005), pp. 97–117.

[16] ——, *Representations of Runge–Kutta methods and strong stability preserving methods*, SIAM J. Numer. Anal., 43 (2005), pp. 924–948.

[17] D. I. Ketcheson, *An algebraic characterization of strong stability preserving Runge–Kutta schemes*, 2004.

[18] ——, *Highly efficient low-storage strong stability preserving Runge-Kutta methods and implementations*, (2007). in preparation.

[19] D. I. Ketcheson and 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), pp. 271–303.

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

[21] C. B. Macdonald, *Constructing high-order Runge–Kutta methods with embedded strong-stability-preserving pairs*, Master's thesis, Simon Fraser University, August 2003.

[22] S. J. Ruuth, *Global optimization of explicit strong-stability-preserving Runge–Kutta methods*, Math. Comp., 75 (2006), pp. 183–207 (electronic).

[23] N. V. Sahinidis and M. Tawarmalani, *BARON 7.2: Global Optimization of Mixed-Integer Nonlinear Programs,* User's Manual, 2004. Available at `http://www.gams.com/dd/docs/solvers/baron.pdf`.

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

[25] ——, *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.

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

[27] M. N. Spijker, *Stepsize conditions for general monotonicity in numerical initial value problems.* manuscript.

[28] ——, *Contractivity in the numerical solution of initial value problems*, Numerische Mathematik, 42 (1983), pp. 271–290.

[29] J. A. van de Griend and J. F. B. M. Kraaijevanger, *Absolute monotonicity of rational functions occurring in the numerical solution of initial value problems*, Numerische Mathematik, 49 (1986), pp. 413–424.

A Coefficients of Optimal Methods

A.1 Fourth-order Methods

$\mu_{11} = 0.157330905682085$	$\mu_{33} = 0.157021682372699$	$\lambda_{21} = 0.703541497995214$	$\lambda_{43} = 0.549902549377947$
$\mu_{21} = 0.342491639470766$	$\mu_{41} = 0.081822264233578$	$\lambda_{32} = 0.694594303739345$	
$\mu_{22} = 0.047573123554705$	$\mu_{42} = 0.079106848361263$	$\lambda_{41} = 0.168078141811591$	
$\mu_{32} = 0.338136048168635$	$\mu_{43} = 0.267698531248384$	$\lambda_{42} = 0.162500172803529$	

Table 5: Non-zero coefficients of the optimal 3-stage method of order 4.

$\mu_{11} = 0.119309657880174$	$\mu_{33} = 0.070606483961727$	$\mu_{52} = 0.034154109552284$	$\lambda_{43} = 0.939878564212065$
$\mu_{21} = 0.226141632153728$	$\mu_{43} = 0.212545672537219$	$\mu_{54} = 0.181099440898861$	$\lambda_{51} = 0.048147179264990$
$\mu_{22} = 0.070605579799433$	$\mu_{44} = 0.119309875536981$	$\lambda_{21} = 1$	$\lambda_{52} = 0.151029729585865$
$\mu_{32} = 0.180764254304414$	$\mu_{51} = 0.010888081702583$	$\lambda_{32} = 0.799340893504885$	$\lambda_{54} = 0.800823091149145$

Table 6: Non-zero coefficients of the optimal 4-stage method of order 4.

$\mu_{11} = 0.072154507748981$	$\mu_{43} = 0.154799860761964$	$\mu_{52} = 0.007472809894781$	$\lambda_{54} = 0.954864191619538$
$\mu_{21} = 0.165562779595956$	$\mu_{44} = 0.077017601068238$	$\mu_{62} = 0.017471397966712$	$\lambda_{65} = 0.894472670673021$
$\mu_{22} = 0.071232036614272$	$\mu_{54} = 0.158089969701175$	$\lambda_{21} = 1$	$\lambda_{52} = 0.045135808380468$
$\mu_{32} = 0.130035287184462$	$\mu_{55} = 0.106426690493882$	$\lambda_{32} = 0.785413771753555$	$\lambda_{62} = 0.105527329326976$
$\mu_{33} = 0.063186062090477$	$\mu_{65} = 0.148091381629243$	$\lambda_{43} = 0.934991917505507$	

Table 7: Non-zero coefficients of the optimal 5-stage method of order 4.

$\mu_{11} = 0.077219435861458$	$\mu_{43} = 0.103230521234296$	$\mu_{66} = 0.077016336936138$	$\lambda_{43} = 0.805203213502341$
$\mu_{21} = 0.128204308556198$	$\mu_{44} = 0.058105933032597$	$\mu_{73} = 0.013804194371285$	$\lambda_{54} = 1$
$\mu_{22} = 0.063842903854499$	$\mu_{54} = 0.128204308556197$	$\mu_{76} = 0.114400114184912$	$\lambda_{63} = 0.062741759593964$
$\mu_{32} = 0.128204308556197$	$\mu_{55} = 0.064105484788524$	$\lambda_{21} = 1$	$\lambda_{65} = 0.937258240406037$
$\mu_{33} = 0.058359965096908$	$\mu_{63} = 0.008043763906343$	$\lambda_{32} = 1$	$\lambda_{73} = 0.107673404480272$
$\mu_{41} = 0.008458154338733$	$\mu_{65} = 0.120160544649854$	$\lambda_{41} = 0.065974025631326$	$\lambda_{76} = 0.892326595519728$

Table 8: Non-zero coefficients of the optimal 6-stage method of order 4.

$\mu_{11} = 0.081324471088377$	$\mu_{54} = 0.108801609187400$	$\mu_{83} = 0.001561606596621$	$\lambda_{73} = 0.189624069894518$
$\mu_{21} = 0.108801609187400$	$\mu_{55} = 0.040474271914787$	$\mu_{87} = 0.107240002590779$	$\lambda_{76} = 0.810375930105481$
$\mu_{22} = 0.051065224656204$	$\mu_{65} = 0.108801609187400$	$\lambda_{21} = 1$	$\lambda_{83} = 0.014352789524754$
$\mu_{32} = 0.108801609187400$	$\mu_{66} = 0.061352000212100$	$\lambda_{32} = 1$	$\lambda_{87} = 0.985647210475246$
$\mu_{33} = 0.036491713577701$	$\mu_{73} = 0.020631403945188$	$\lambda_{43} = 0.865661994183934$	
$\mu_{43} = 0.094185417979586$	$\mu_{76} = 0.088170205242212$	$\lambda_{54} = 1$	
$\mu_{44} = 0.037028821732794$	$\mu_{77} = 0.080145231879588$	$\lambda_{65} = 1$	

Table 9: Non-zero coefficients of the optimal 7-stage method of order 4.

$\mu_{11} = 0.080355939553359$	$\mu_{51} = 0.004426522032754$	$\mu_{84} = 0.021977226754808$	$\lambda_{51} = 0.047220157287989$
$\mu_{21} = 0.093742212796061$	$\mu_{54} = 0.083174746150582$	$\mu_{87} = 0.071764986041253$	$\lambda_{54} = 0.887270992114641$
$\mu_{22} = 0.054617345411549$	$\mu_{55} = 0.030116385482588$	$\mu_{88} = 0.055606577879005$	$\lambda_{65} = 1$
$\mu_{32} = 0.093742212796061$	$\mu_{65} = 0.093742212796061$	$\mu_{98} = 0.093742212796061$	$\lambda_{76} = 1$
$\mu_{33} = 0.039438131644116$	$\mu_{66} = 0.038334326442344$	$\lambda_{21} = 1$	$\lambda_{84} = 0.234443225728203$
$\mu_{43} = 0.093742212796061$	$\mu_{76} = 0.093742212796061$	$\lambda_{32} = 1$	$\lambda_{87} = 0.765556774271797$
$\mu_{44} = 0.032427875074076$	$\mu_{77} = 0.058861620081910$	$\lambda_{43} = 1$	$\lambda_{98} = 1$

Table 10: Non-zero coefficients of the optimal 8-stage method of order 4.

$\mu_{11} = 0.068605696784244$	$\mu_{61} = 0.008747971137402$	$\mu_{98} = 0.065270626421385$	$\lambda_{65} = 0.878687985294225$
$\mu_{21} = 0.082269487560004$	$\mu_{62} = 0.001326570052113$	$\mu_{99} = 0.057552171403649$	$\lambda_{76} = 1$
$\mu_{22} = 0.048685583036902$	$\mu_{65} = 0.072971983212453$	$\mu_{10,9} = 0.083046524401968$	$\lambda_{87} = 1$
$\mu_{32} = 0.077774790319743$	$\mu_{66} = 0.029699905991308$	$\lambda_{21} = 0.990643355064403$	$\lambda_{95} = 0.214047464461523$
$\mu_{33} = 0.039925150083662$	$\mu_{76} = 0.083046524401968$	$\lambda_{32} = 0.936520713898770$	$\lambda_{98} = 0.785952535538477$
$\mu_{43} = 0.083046524401968$	$\mu_{77} = 0.035642110881905$	$\lambda_{43} = 1$	$\lambda_{10,9} = 1$
$\mu_{44} = 0.031928917146492$	$\mu_{87} = 0.083046524401969$	$\lambda_{54} = 1$	
$\mu_{54} = 0.083046524401968$	$\mu_{88} = 0.050978240433952$	$\lambda_{61} = 0.105338196876962$	
$\mu_{55} = 0.029618614941264$	$\mu_{95} = 0.017775897980583$	$\lambda_{62} = 0.015973817828813$	

Table 11: Non-zero coefficients of the optimal 9-stage method of order 4.

$\mu_{11} = 0.053637857412307$	$\mu_{65} = 0.073302847899924$	$\mu_{10,6} = 0.012892211367605$	$\lambda_{72} = 0.121369109867354$
$\mu_{21} = 0.073302847899924$	$\mu_{66} = 0.029485772863308$	$\mu_{10,9} = 0.060410636532319$	$\lambda_{76} = 0.878630890132646$
$\mu_{22} = 0.042472343576273$	$\mu_{72} = 0.008896701400356$	$\mu_{10,10} = 0.053275700719583$	$\lambda_{87} = 1$
$\mu_{32} = 0.063734820131903$	$\mu_{76} = 0.064406146499568$	$\mu_{11,10} = 0.073302847899924$	$\lambda_{98} = 1$
$\mu_{33} = 0.039816143518898$	$\mu_{77} = 0.033369849008191$	$\lambda_{21} = 1$	$\lambda_{10,6} = 0.175875995775857$
$\mu_{43} = 0.072590353622503$	$\mu_{87} = 0.073302847899924$	$\lambda_{32} = 0.869472632481021$	$\lambda_{10,9} = 0.824124004224143$
$\mu_{44} = 0.034233821696022$	$\mu_{88} = 0.037227578299133$	$\lambda_{43} = 0.990280128291965$	$\lambda_{11,10} = 1$
$\mu_{54} = 0.073302847899924$	$\mu_{98} = 0.073302847899924$	$\lambda_{54} = 1$	
$\mu_{55} = 0.030626774272464$	$\mu_{99} = 0.046126339053885$	$\lambda_{65} = 1$	

Table 12: Non-zero coefficients of the optimal 10-stage method of order 4.

$\mu_{11} = 0.056977945207836$	$\mu_{65} = 0.065880156369595$	$\mu_{10,10} = 0.042024506703707$	$\lambda_{65} = 1$
$\mu_{21} = 0.065880156369595$	$\mu_{66} = 0.025442782369057$	$\mu_{11,7} = 0.012021727578515$	$\lambda_{76} = 1$
$\mu_{22} = 0.043484869703481$	$\mu_{76} = 0.065880156369595$	$\mu_{11,10} = 0.053858428791080$	$\lambda_{83} = 0.150816289869158$
$\mu_{32} = 0.065880156369595$	$\mu_{77} = 0.029602951078198$	$\mu_{11,11} = 0.045164424313434$	$\lambda_{87} = 0.849183710130842$
$\mu_{33} = 0.035790792116714$	$\mu_{83} = 0.009935800759662$	$\mu_{12,11} = 0.065880156369595$	$\lambda_{98} = 1$
$\mu_{41} = 0.000026595081404$	$\mu_{87} = 0.055944355609932$	$\lambda_{21} = 1$	$\lambda_{10,9} = 1$
$\mu_{43} = 0.061212831485396$	$\mu_{88} = 0.027887296332663$	$\lambda_{32} = 1$	$\lambda_{11,7} = 0.182478734735714$
$\mu_{44} = 0.029306212740362$	$\mu_{98} = 0.065880156369595$	$\lambda_{41} = 0.000403688802047$	$\lambda_{11,10} = 0.817521265264286$
$\mu_{54} = 0.065880156369595$	$\mu_{99} = 0.033340440672342$	$\lambda_{43} = 0.929154313811668$	$\lambda_{12,11} = 1$
$\mu_{55} = 0.028274789742965$	$\mu_{10,9} = 0.065880156369595$	$\lambda_{54} = 1$	

Table 13: Non-zero coefficients of the optimal 11-stage method of order 4.

A.2 Fifth-order Methods

$\mu_{21} = 0.123278811294077$	$\mu_{43} = 0.401279315201748$	$\lambda_{21} = 0.132472176321996$	$\lambda_{52} = 0.152838011871931$
$\mu_{22} = 0.123278811294077$	$\mu_{44} = 0.130922344408366$	$\lambda_{32} = 0.381520524096177$	$\lambda_{53} = 0.183936851467070$
$\mu_{32} = 0.355043587270334$	$\mu_{51} = 0.025096216568591$	$\lambda_{41} = 0.104471653720944$	$\lambda_{54} = 0.294739759754655$
$\mu_{33} = 0.058564865652410$	$\mu_{52} = 0.142231289220490$	$\lambda_{42} = 0.017621069446845$	
$\mu_{41} = 0.097221481840379$	$\mu_{53} = 0.171171917240332$	$\lambda_{43} = 0.431204224308822$	
$\mu_{42} = 0.016398194363903$	$\mu_{54} = 0.274285274330639$	$\lambda_{51} = 0.026967735910097$	

Table 14: Non-zero coefficients of the optimal 4-stage method of order 5.

$\mu_{21} = 0.086178504812481$	$\mu_{54} = 0.174378360369685$	$\mu_{74} = 0.041213090387125$	$\lambda_{62} = 0.080127417074354$
$\mu_{22} = 0.086178504230788$	$\mu_{55} = 0.061419240496304$	$\mu_{76} = 0.146947097961295$	$\lambda_{63} = 0.129789388273218$
$\mu_{32} = 0.150240097189194$	$\mu_{62} = 0.016112966258044$	$\lambda_{21} = 0.428553060148464$	$\lambda_{65} = 0.789958727494273$
$\mu_{33} = 0.062153125773256$	$\mu_{63} = 0.026099581270141$	$\lambda_{32} = 0.747121959791843$	$\lambda_{73} = 0.023257255649335$
$\mu_{43} = 0.175903357484878$	$\mu_{65} = 0.158854219767892$	$\lambda_{43} = 0.874741587876977$	$\lambda_{74} = 0.204946651627447$
$\mu_{44} = 0.056168662884210$	$\mu_{66} = 0.080304315695241$	$\lambda_{51} = 0.116834490082770$	$\lambda_{76} = 0.730746357786987$
$\mu_{51} = 0.023494457517988$	$\mu_{73} = 0.004676843322988$	$\lambda_{54} = 0.867158000973773$	

Table 15: Non-zero coefficients of the optimal 6-stage method of order 5.

$\mu_{21} = 0.077756487471956$	$\mu_{62} = 0.011454172434127$	$\mu_{86} = 0.000177781270869$	$\lambda_{65} = 0.760345962143127$
$\mu_{22} = 0.077756487471823$	$\mu_{63} = 0.027138257330487$	$\mu_{87} = 0.124996366168017$	$\lambda_{73} = 0.125302322168346$
$\mu_{32} = 0.126469010941083$	$\mu_{65} = 0.122441492758580$	$\lambda_{21} = 0.482857811904546$	$\lambda_{76} = 0.874697677831654$
$\mu_{33} = 0.058945597921853$	$\mu_{66} = 0.037306165750735$	$\lambda_{32} = 0.785356333370487$	$\lambda_{84} = 0.059945182887979$
$\mu_{43} = 0.143639250502198$	$\mu_{73} = 0.020177924440034$	$\lambda_{43} = 0.891981318293413$	$\lambda_{85} = 0.157921009644458$
$\mu_{44} = 0.044443238891736$	$\mu_{76} = 0.140855998083160$	$\lambda_{51} = 0.074512829695468$	$\lambda_{86} = 0.001103998884730$
$\mu_{51} = 0.011999093244164$	$\mu_{77} = 0.077972159279168$	$\lambda_{54} = 0.900717090387559$	$\lambda_{87} = 0.776211398253764$
$\mu_{54} = 0.145046006148787$	$\mu_{84} = 0.009653207936821$	$\lambda_{62} = 0.071128941372444$	
$\mu_{55} = 0.047108760907057$	$\mu_{85} = 0.025430639631870$	$\lambda_{63} = 0.168525096484428$	

Table 16: Non-zero coefficients of the optimal 7-stage method of order 5.

$\mu_{21} = 0.068228425119547$	$\mu_{63} = 0.019703233696280$	$\mu_{95} = 0.002607774587593$	$\lambda_{65} = 0.794939486396848$
$\mu_{22} = 0.068228425081188$	$\mu_{65} = 0.105180973170163$	$\mu_{96} = 0.024666705635997$	$\lambda_{73} = 0.115904148048060$
$\mu_{32} = 0.105785458668142$	$\mu_{66} = 0.045239659320409$	$\mu_{98} = 0.104666894951906$	$\lambda_{76} = 0.884095226988328$
$\mu_{33} = 0.049168429086829$	$\mu_{73} = 0.015335646668415$	$\lambda_{21} = 0.515658560550227$	$\lambda_{84} = 0.085067722561958$
$\mu_{43} = 0.119135238085849$	$\mu_{76} = 0.116977452926909$	$\lambda_{32} = 0.799508082567950$	$\lambda_{85} = 0.049438833770315$
$\mu_{44} = 0.040919294063196$	$\mu_{77} = 0.050447703819928$	$\lambda_{43} = 0.900403391614526$	$\lambda_{87} = 0.865488353423280$
$\mu_{51} = 0.009164078944895$	$\mu_{84} = 0.011255581082016$	$\lambda_{51} = 0.069260513476804$	$\lambda_{95} = 0.019709106398420$
$\mu_{54} = 0.120257079939301$	$\mu_{85} = 0.006541409424671$	$\lambda_{54} = 0.908882077064212$	$\lambda_{96} = 0.186426667470161$
$\mu_{55} = 0.039406904101415$	$\mu_{87} = 0.114515518273119$	$\lambda_{62} = 0.056144626483417$	$\lambda_{98} = 0.791054172708715$
$\mu_{62} = 0.007428674198294$	$\mu_{88} = 0.060382824328534$	$\lambda_{63} = 0.148913610539984$	

Table 17: Non-zero coefficients of the optimal 8-stage method of order 5.

$\mu_{21} = 0.057541273792734$	$\mu_{66} = 0.040395338505384$	$\mu_{10,6} = 0.000157554758807$	$\lambda_{74} = 0.161972606843345$
$\mu_{22} = 0.057541282875429$	$\mu_{73} = 0.003606182878823$	$\mu_{10,7} = 0.023607648002010$	$\lambda_{76} = 0.805943410735452$
$\mu_{32} = 0.089687860942851$	$\mu_{74} = 0.018205434656765$	$\mu_{10,9} = 0.088454624345414$	$\lambda_{84} = 0.098497788983963$
$\mu_{33} = 0.041684970395150$	$\mu_{76} = 0.090586614534056$	$\lambda_{21} = 0.511941093031398$	$\lambda_{87} = 0.901502211016037$
$\mu_{43} = 0.101622955619526$	$\mu_{77} = 0.042925976445877$	$\lambda_{32} = 0.797947256574797$	$\lambda_{95} = 0.091469767162319$
$\mu_{44} = 0.040743690263377$	$\mu_{84} = 0.011070977346914$	$\lambda_{43} = 0.904133043080300$	$\lambda_{98} = 0.908530232837680$
$\mu_{51} = 0.009276188714858$	$\mu_{87} = 0.101327254746568$	$\lambda_{51} = 0.082529667434119$	$\lambda_{10,6} = 0.001401754777391$
$\mu_{54} = 0.101958242208571$	$\mu_{88} = 0.046669302312152$	$\lambda_{54} = 0.907116066770269$	$\lambda_{10,7} = 0.210035759124536$
$\mu_{55} = 0.040815264589441$	$\mu_{95} = 0.010281040119047$	$\lambda_{62} = 0.100295062538531$	$\lambda_{10,9} = 0.786975228149903$
$\mu_{62} = 0.011272987717036$	$\mu_{98} = 0.102117191974435$	$\lambda_{65} = 0.899704937426848$	
$\mu_{65} = 0.101125244372555$	$\mu_{99} = 0.050500143250113$	$\lambda_{73} = 0.032083982209117$	

Table 18: Non-zero coefficients of the optimal 9-stage method of order 5.

$\mu_{21} = 0.052445615058994$	$\mu_{74} = 0.015255382390056$	$\mu_{10,10} = 0.043749770437420$	$\lambda_{76} = 0.812371189661489$
$\mu_{22} = 0.052445635165954$	$\mu_{76} = 0.080215515252923$	$\mu_{11,7} = 0.001872759401284$	$\lambda_{84} = 0.097617319434729$
$\mu_{32} = 0.079936220395519$	$\mu_{77} = 0.035768398609662$	$\mu_{11,8} = 0.017616881402665$	$\lambda_{87} = 0.902382678155958$
$\mu_{33} = 0.038724845476313$	$\mu_{84} = 0.009638972523544$	$\mu_{11,10} = 0.079160150775900$	$\lambda_{95} = 0.093186499255038$
$\mu_{43} = 0.089893189589075$	$\mu_{87} = 0.089103469454345$	$\lambda_{21} = 0.531135486241871$	$\lambda_{98} = 0.906813500744962$
$\mu_{44} = 0.037676214671832$	$\mu_{88} = 0.040785658461768$	$\lambda_{32} = 0.809542670828687$	$\lambda_{10,6} = 0.057065598977612$
$\mu_{51} = 0.007606429497294$	$\mu_{95} = 0.009201462517982$	$\lambda_{43} = 0.910380456183399$	$\lambda_{10,7} = 0.066440169285130$
$\mu_{54} = 0.090180506502554$	$\mu_{98} = 0.089540979697808$	$\lambda_{51} = 0.077033029836054$	$\lambda_{10,9} = 0.876494226842443$
$\mu_{55} = 0.035536573874530$	$\mu_{99} = 0.042414168555682$	$\lambda_{54} = 0.913290217244921$	$\lambda_{11,7} = 0.018966103726616$
$\mu_{62} = 0.009295158915663$	$\mu_{10,6} = 0.005634796609556$	$\lambda_{62} = 0.094135396158718$	$\lambda_{11,8} = 0.178412453726484$
$\mu_{65} = 0.089447242753894$	$\mu_{10,7} = 0.006560464576444$	$\lambda_{65} = 0.905864193215084$	$\lambda_{11,10} = 0.801683136446066$
$\mu_{66} = 0.036490114423762$	$\mu_{10,9} = 0.086547180546464$	$\lambda_{73} = 0.033130514796271$	
$\mu_{73} = 0.003271387942850$		$\lambda_{74} = 0.154496709294644$	

Table 19: Non-zero coefficients of the optimal 10-stage method of order 5.

$\mu_{21} = 0.048856948431570$	$\mu_{76} = 0.073016254277378$	$\mu_{11,10} = 0.075746112223043$	$\lambda_{76} = 0.827494171134198$
$\mu_{22} = 0.048856861697775$	$\mu_{77} = 0.033377699686911$	$\mu_{11,11} = 0.042478561828713$	$\lambda_{84} = 0.093508818968334$
$\mu_{32} = 0.072383163641108$	$\mu_{84} = 0.008251011235053$	$\mu_{12,8} = 0.004170617993886$	$\lambda_{87} = 0.906491181031666$
$\mu_{33} = 0.035920513887793$	$\mu_{87} = 0.079986775597087$	$\mu_{12,9} = 0.011637432775226$	$\lambda_{95} = 0.091745217287743$
$\mu_{43} = 0.080721632683704$	$\mu_{88} = 0.035640440183022$	$\mu_{12,11} = 0.072377330912325$	$\lambda_{98} = 0.908254782302260$
$\mu_{44} = 0.034009594943671$	$\mu_{95} = 0.008095394925904$	$\lambda_{21} = 0.553696439876870$	$\lambda_{10,6} = 0.066947714363965$
$\mu_{51} = 0.006438090160799$	$\mu_{98} = 0.080142391870059$	$\lambda_{32} = 0.820319346617409$	$\lambda_{10,7} = 0.061140603801867$
$\mu_{54} = 0.081035022899306$	$\mu_{99} = 0.036372965664654$	$\lambda_{43} = 0.914819326070196$	$\lambda_{10,9} = 0.871911681834169$
$\mu_{55} = 0.032672027896742$	$\mu_{10,6} = 0.005907318148947$	$\lambda_{51} = 0.072962960562995$	$\lambda_{11,7} = 0.040471104837131$
$\mu_{62} = 0.007591099341932$	$\mu_{10,7} = 0.005394911565057$	$\lambda_{54} = 0.918370981510030$	$\lambda_{11,8} = 0.101097207986272$
$\mu_{63} = 0.000719846382100$	$\mu_{10,9} = 0.076935557118137$	$\lambda_{62} = 0.086030028794504$	$\lambda_{11,10} = 0.858431687176596$
$\mu_{65} = 0.079926841108108$	$\mu_{10,10} = 0.032282094274356$	$\lambda_{63} = 0.008158028526592$	$\lambda_{12,8} = 0.047265668639449$
$\mu_{66} = 0.033437798720082$	$\mu_{11,7} = 0.003571080721480$	$\lambda_{65} = 0.905811942678904$	$\lambda_{12,9} = 0.131887178872293$
$\mu_{73} = 0.003028997848550$	$\mu_{11,8} = 0.008920593887617$	$\lambda_{73} = 0.034327672500586$	$\lambda_{12,11} = 0.820253244225314$
$\mu_{74} = 0.012192534706212$		$\lambda_{74} = 0.138178156365216$	

Table 20: Non-zero coefficients of the optimal 11-stage method of order 5.

A.3 Sixth-order Methods

$\mu_{21} = 0.306709397198437$	$\mu_{54} = 0.331296656179688$	$\mu_{75} = 0.303060566272042$	$\lambda_{61} = 0.000006020335333$
$\mu_{22} = 0.306709397198281$	$\mu_{55} = 0.107322255666019$	$\mu_{76} = 0.135975816243004$	$\lambda_{62} = 0.000000003205153$
$\mu_{31} = 0.100402778173265$	$\mu_{61} = 0.000033015066992$	$\lambda_{21} = 0.055928810359256$	$\lambda_{63} = 0.072039142196788$
$\mu_{32} = 0.000000014622272$	$\mu_{62} = 0.000000017576816$	$\lambda_{31} = 0.018308561756789$	$\lambda_{64} = 0.002650837430364$
$\mu_{33} = 0.100402700098726$	$\mu_{63} = 0.395057247524893$	$\lambda_{32} = 0.000000002666388$	$\lambda_{65} = 0.076936194272824$
$\mu_{41} = 0.000015431349319$	$\mu_{64} = 0.014536993458566$	$\lambda_{41} = 0.000002813924247$	$\lambda_{71} = 0.009870541274021$
$\mu_{42} = 0.000708584139276$	$\mu_{65} = 0.421912313467517$	$\lambda_{42} = 0.000129211130507$	$\lambda_{72} = 0.000379944400556$
$\mu_{43} = 0.383195003696784$	$\mu_{66} = 0.049194928995335$	$\lambda_{43} = 0.069876048429340$	$\lambda_{73} = 0.042665841426363$
$\mu_{44} = 0.028228318307509$	$\mu_{71} = 0.054129307323559$	$\lambda_{51} = 0.018587746937629$	$\lambda_{74} = 0.033716209818106$
$\mu_{51} = 0.101933808745384$	$\mu_{72} = 0.002083586568620$	$\lambda_{52} = 0.000004866574675$	$\lambda_{75} = 0.055263441854804$
$\mu_{52} = 0.000026687930165$	$\mu_{73} = 0.233976271277479$	$\lambda_{53} = 0.024929494718837$	$\lambda_{76} = 0.024795346049276$
$\mu_{53} = 0.136711477475771$	$\mu_{74} = 0.184897163424393$	$\lambda_{54} = 0.060412325234826$	

Table 21: Non-zero coefficients of the optimal 6-stage method of order 6.

$\mu_{21} = 0.078064586430339$	$\mu_{74} = 0.001997541632150$	$\mu_{97} = 0.029504401738350$	$\lambda_{81} = 0.275406645480353$
$\mu_{22} = 0.078064586430334$	$\mu_{75} = 0.177750742549303$	$\mu_{98} = 0.000167109498102$	$\lambda_{82} = 0.001937467969363$
$\mu_{31} = 0.000000000128683$	$\mu_{76} = 0.099344022703332$	$\lambda_{21} = 0.175964293749273$	$\lambda_{83} = 0.018605123379003$
$\mu_{32} = 0.207887720440412$	$\mu_{77} = 0.025183595544641$	$\lambda_{31} = 0.000000000290062$	$\lambda_{84} = 0.518868675379274$
$\mu_{33} = 0.051491724905522$	$\mu_{81} = 0.122181071065616$	$\lambda_{32} = 0.468596806556916$	$\lambda_{85} = 0.104656154246370$
$\mu_{41} = 0.039407945831803$	$\mu_{82} = 0.000859535946343$	$\lambda_{41} = 0.088828900190110$	$\lambda_{86} = 0.039349722004217$
$\mu_{43} = 0.256652317630585$	$\mu_{83} = 0.008253954430873$	$\lambda_{43} = 0.578516403866171$	$\lambda_{87} = 0.040422284523661$
$\mu_{44} = 0.062490509654886$	$\mu_{84} = 0.230190271515289$	$\lambda_{51} = 0.021817144198582$	$\lambda_{91} = 0.024950675444873$
$\mu_{51} = 0.009678931461971$	$\mu_{85} = 0.046429529676480$	$\lambda_{52} = 0.256377915663045$	$\lambda_{92} = 0.024730907022402$
$\mu_{52} = 0.113739188386853$	$\mu_{86} = 0.017457063072040$	$\lambda_{54} = 0.513470441684846$	$\lambda_{93} = 0.155143002154553$
$\mu_{54} = 0.227795405648863$	$\mu_{87} = 0.017932893410781$	$\lambda_{62} = 0.023037388973687$	$\lambda_{94} = 0.110144297841125$
$\mu_{55} = 0.076375614721986$	$\mu_{88} = 0.322331010725841$	$\lambda_{63} = 0.304490034708070$	$\lambda_{95} = 0.309707532056893$
$\mu_{62} = 0.010220279377975$	$\mu_{91} = 0.011069087473717$	$\lambda_{65} = 0.530062554633790$	$\lambda_{96} = 0.203650883489192$
$\mu_{63} = 0.135083590682973$	$\mu_{92} = 0.010971589676607$	$\lambda_{72} = 0.000000021253185$	$\lambda_{97} = 0.066505459796630$
$\mu_{65} = 0.235156310567507$	$\mu_{93} = 0.068827453812950$	$\lambda_{73} = 0.254322947692795$	$\lambda_{98} = 0.000376679185235$
$\mu_{66} = 0.033370798931382$	$\mu_{94} = 0.048864283062331$	$\lambda_{74} = 0.004502630688369$	
$\mu_{72} = 0.000000009428737$	$\mu_{95} = 0.137398274895655$	$\lambda_{75} = 0.400665465691124$	
$\mu_{73} = 0.112827524882246$	$\mu_{96} = 0.090347431612516$	$\lambda_{76} = 0.223929973789109$	

Table 22: Non-zero coefficients of the optimal 8-stage method of order 6.

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

Table 23: Non-zero coefficients of the optimal 9-stage method of order 6.

$\mu_{21} = 0.054638144097621$	$\mu_{84} = 0.020931607249912$	$\mu_{11,7} = 0.015039465910057$	$\lambda_{86} = 0.000000038106595$
$\mu_{22} = 0.054638144097609$	$\mu_{85} = 0.007491225374492$	$\mu_{11,8} = 0.004816543620956$	$\lambda_{87} = 0.768392593572726$
$\mu_{32} = 0.094708145223810$	$\mu_{86} = 0.000000004705702$	$\mu_{11,9} = 0.031302441038151$	$\lambda_{94} = 0.000000003546047$
$\mu_{33} = 0.044846931722606$	$\mu_{87} = 0.094887152674486$	$\mu_{11,10} = 0.071672462436845$	$\lambda_{95} = 0.109198714839684$
$\mu_{43} = 0.108958403164940$	$\mu_{88} = 0.041052752299292$	$\lambda_{21} = 0.442457635916190$	$\lambda_{96} = 0.099613661566658$
$\mu_{44} = 0.031071352647397$	$\mu_{94} = 0.000000000437894$	$\lambda_{32} = 0.766942997969774$	$\lambda_{98} = 0.786948084216732$
$\mu_{51} = 0.004498251069701$	$\mu_{95} = 0.013484714992727$	$\lambda_{43} = 0.882341050812911$	$\lambda_{10,1} = 0.007993221037648$
$\mu_{52} = 0.005530448043688$	$\mu_{96} = 0.012301077330264$	$\lambda_{51} = 0.036426667979449$	$\lambda_{10,2} = 0.000002813781560$
$\mu_{54} = 0.107851443619437$	$\mu_{98} = 0.097178530400423$	$\lambda_{52} = 0.044785360253007$	$\lambda_{10,6} = 0.035121034164983$
$\mu_{55} = 0.018486380725450$	$\mu_{99} = 0.039273658398104$	$\lambda_{54} = 0.873376934047102$	$\lambda_{10,7} = 0.092804768098049$
$\mu_{62} = 0.015328210231111$	$\mu_{10,1} = 0.000987065715240$	$\lambda_{62} = 0.124127269944714$	$\lambda_{10,8} = 0.017181361859997$
$\mu_{63} = 0.014873940010974$	$\mu_{10,2} = 0.000000347467847$	$\lambda_{63} = 0.120448606787528$	$\lambda_{10,9} = 0.844926230212794$
$\mu_{64} = 0.000000013999299$	$\mu_{10,6} = 0.004337021151393$	$\lambda_{64} = 0.000000113365798$	$\lambda_{11,3} = 0.005319886250823$
$\mu_{65} = 0.093285690103096$	$\mu_{10,7} = 0.011460261685365$	$\lambda_{65} = 0.755424009901960$	$\lambda_{11,7} = 0.121789029292733$
$\mu_{66} = 0.031019852663844$	$\mu_{10,8} = 0.002121689510807$	$\lambda_{73} = 0.189047812082446$	$\lambda_{11,8} = 0.039004189088262$
$\mu_{73} = 0.023345108682580$	$\mu_{10,9} = 0.104338127248348$	$\lambda_{74} = 0.000003741673193$	$\lambda_{11,9} = 0.253485990215933$
$\mu_{74} = 0.000000462051194$	$\mu_{10,10} = 0.042268075457472$	$\lambda_{76} = 0.810948446244362$	$\lambda_{11,10} = 0.580400905152248$
$\mu_{76} = 0.100142283610706$	$\mu_{11,3} = 0.000656941338471$	$\lambda_{84} = 0.169503368254511$	
$\mu_{77} = 0.037191650574052$		$\lambda_{85} = 0.060663661331375$	

Table 24: Non-zero coefficients of the optimal 10-stage method of order 6.