ARTICLE IN PRESS
Journal of Computational and Applied Mathematics xxx (xxxx) xxx

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Efficient SSP low-storage Runge–Kutta methods ☆

I. Higueras, T. Roldán *

Departamento de Estadística, Informática y Matemáticas, Universidad Pública de Navarra, Campus de Arrosadia, 31006 Pamplona, Spain

ARTICLE INFO

Article history:
Received 31 January 2019
Received in revised form 31 July 2019

MSC:
65L06
65L05
65L04

Keywords:
Runge–Kutta
Strong stability preserving
Total variation diminishing
Low-storage
Local error term
Initial value problem

ABSTRACT

In this paper we study the efficiency of Strong Stability Preserving (SSP) Runge–Kutta methods that can be implemented with a low number of registers using their Shu–Osher representation. SSP methods have been studied in the literature and stepsize restrictions that ensure numerical monotonicity have been found. However, for some problems, the observed stepsize restrictions are larger than the theoretical ones. Aiming at obtaining additional properties of the schemes that may explain their efficiency, in this paper we study the influence of the local error term in the observed stepsize restrictions. For this purpose, we consider the family of 5-stage third order SSP explicit Runge–Kutta methods, namely SSP(5,3), and the Buckley–Leverett equation. We deal with optimal SSP(5,3) schemes whose implementation requires at least 3 memory registers, and non-optimal 2-register SSP(5,3) schemes. The numerical experiments done show that small error constants improve the efficiency of the method in the sense that larger observed SSP coefficients are obtained.

1. Introduction

Given an initial value problem of the form

$$\frac{d}{dt}y(t) = f(y(t)), \qquad t \geq t_0, \qquad (1)$$
$$y(t_0) = y_0,$$

a common class of schemes to solve it are explicit Runge–Kutta (RK) methods. An s-stage explicit RK method is defined by a strictly lower triangular $s \times s$ matrix A and a vector $b \in \mathbb{R}^s$. If y_n is the numerical approximation of the solution $y(t)$ at $t = t_n$, we obtain y_{n+1}, the numerical approximation of the solution at $t_{n+1} = t_n + h$, from

$$Y_i = y_n + h\sum_{j=1}^{i-1} a_{ij} f(Y_j), \qquad 1 \leq i \leq s, \qquad y_{n+1} = y_n + h\sum_{i=1}^{s} b_i f(Y_i), \qquad (2)$$

where the internal stage Y_i approximates $y(t_n + c_i h)$, and, as usual, $c_i = \sum_{j=1}^{s-1} a_{ij}$.

Strong Stability Preserving (SSP) methods were introduced in [1] to ensure numerical monotonicity for problems whose solutions satisfy a monotonicity property for the forward Euler method. In the SSP theory, numerical monotonicity is

☆ This work was partially supported by the Spanish Research Grant MTM2016-77735-C3-2-P.

* Corresponding author.
E-mail addresses: higueras@unavarra.es (I. Higueras), teo@unavarra.es (T. Roldán).

https://doi.org/10.1016/j.cam.2019.112500
0377-0427/

Please cite this article as: I. Higueras and T. Roldán, Efficient SSP low-storage Runge–Kutta methods, Journal of Computational and Applied Mathematics (2019) 112500, https://doi.org/10.1016/j.cam.2019.112500.

ensured under stepsize restrictions that involve the SSP coefficient of the RK method, and the larger the SSP coefficient is, the larger the stepsize restriction is. Consequently, optimal s-stage pth order SSP methods, denoted by SSP(s, p), give the largest theoretical stepsize restrictions. SSP(s, p) methods have been widely studied in the literature (see, e.g., [2] and the references therein). However, in practice, although for some problems the theoretical stepsize restrictions for numerical monotonicity are sharp, for some others the observed stepsize restrictions, that is, the ones that ensure numerical monotonicity for a given problem, are larger than the theoretical ones. One of the reasons may be the beneficial influence of additional non-SSP properties of the scheme like stability regions or the size of local error term. These issues have been outlined in the literature (see, e.g., [3] for implicit explicit SSP methods or [4] for SSP(5,3) methods).

SSP properties have also been studied for other kind of methods (e.g., two and multistep Runge–Kutta schemes [5–7], explicit SSP peer methods [8], etc.), and optimal SSP coefficients together with numerically optimal methods have been obtained. For these schemes it has also been observed that for some problems the observed stepsize restrictions for monotonicity are larger than the theoretical ones.

Implementation issues are also relevant for some problems. A naive implementation of a standard s-stage explicit RK method requires $s+1$ memory registers of length N, where N is the dimension of the differential problem (1). For systems with a large number of equations, the high dimension of the problem (1) compromises the computer memory capacity and thus it is important to incorporate low memory usage to some other properties of the scheme. These ideas have been developed, e.g., in [9–18], where different low-storage RK methods have been constructed. In particular, some low-storage methods have been studied in the context of strong stability preserving (SSP) schemes [10–16]. The idea in, e.g., [13,15] is to deal with the Shu–Osher form of explicit RK methods and exploit the sparsity of the Shu–Osher matrices to achieve an efficient implementation. In this way, in [13,15] it is proven that some optimal SSP schemes can be implemented with $2N$ memory registers. Besides, some $2N$ low-storage methods, denoted by $2N^*$, retain the computed approximation at the previous time step [13]. In [4], 5-stage third order SSP explicit RK methods were studied. It turns out that optimal SSP methods cannot be implemented in two memory registers. However, it is possible to construct $2N^*$ low-storage non-optimal 5-stage third order SSP RK methods.

In this paper we study properties that may benefit the efficiency of SSP low-storage RK methods. More precisely, we analyze the relationship between the size of the local error term and the observed SSP coefficient. Our goal is to obtain a criteria to be used in the construction of efficient SSP schemes. For this purpose, we consider SSP(5,3) RK explicit methods. These schemes were studied in [4], where the structure and low-storage properties of optimal SSP(5,3) were analyzed and some methods of this family were obtained; in particular, the ones named SSP53_R, SSP53_H, SSP53_1 and SSP53_2. As all the optimal SSP(5,3) methods have the same SSP coefficient and stability regions, variations of the observed SSP coefficient can be analyzed in terms of other properties of the schemes like variations of the local error term. Besides, in [4], some non-optimal $2N^*$ low-storage SSP(5,3) methods with maximum SSP coefficient were also constructed; however, these schemes have different SSP coefficients and stability regions and thus the observed SSP coefficient may depend on these two properties. In both cases, optimal SSP(5,3) methods and $2N^*$ low-storage SSP(5,3) schemes, the numerical results obtained in [4] lead us to conjecture that the smaller the $\|\cdot\|_2$-error constant is, the larger the observed SSP coefficient. In this paper, we go further in this analysis by means of the study of the local error and the low-storage arrangement of these schemes.

The rest of the paper is organized as follows. In Sections 2.1 and 2.2 we give a brief introduction to SSP RK methods and low-storage methods in the sense of [13]. In Section 2.3 we deal with low-storage SSP(5,3) methods; we review in Section 2.3.1 the family of optimal SSP(5,3) methods that can be implemented in 3N memory registers [4], while $2N^*$ low-storage SSP(5,3) methods [4] are reviewed in Section 2.3.2. New efficient SSP(5,3) schemes are obtained in Section 3; some optimal SSP(5,3) methods with minimum local error terms are obtained in Section 3.1, while non optimal $2N^*$ low-storage SSP(5,3) methods with minimum local error are constructed in Section 3.2. Section 4 is devoted to numerical experiments, where different SSP(5,3) methods are tested on the 1D Buckley–Leverett equation [19,20]. The paper ends with some conclusions in Section 5. The detailed coefficients of the different methods constructed in this paper are given in Appendix.

2. SSP and low-storage Runge–Kutta methods review

In this section we briefly review some known concepts on SSP RK methods and low-storage schemes, in the sense of [10–16], that will be used in the rest of the paper.

2.1. Strong stability preserving Runge–Kutta methods

SSP methods are relevant for dissipative problems (1), that is, problems such that the exact solution satisfies a monotonicity property of the form

$$\|y(t)\| \leq \|y(t_0)\|\,, \qquad \text{for all } t \geq t_0\,, \tag{3}$$

where $\|\cdot\| : \mathbb{R}^N \to \mathbb{R}$ denotes a convex functional, e.g., a norm or a semi-norm. A sufficient condition for (3) is monotonicity under forward Euler steps

$$\|\, y + h f(y) \,\| \leq \|\, y \,\|\,, \qquad \text{for } h \leq \Delta t_{FE}\,, \tag{4}$$

for all $y \in \mathbb{R}^N$ and a fixed $\Delta t_{FE} > 0$ (see, e.g., [21, p. 501] or [22, p. 1–2] for details).

Taking into account (3), it makes sense to require numerical monotonicity, not only for the numerical solution y_n, but also for the internal stages $Y_i \approx y(t_n + c_i h)$, if $c_i \geq 0$. This is,

$$\|Y_i\| \leq \|y_n\|\,, \quad i = 1, \ldots, s\,, \qquad \|y_{n+1}\| \leq \|y_n\|\,, \tag{5}$$

for all $n \geq 0$, probably under a stepsize restriction $h \leq \Delta t_{\mathrm{MAX}}$. The seminal papers by Spijker [23–25] and Kraaijevanger [21,26] on numerical contractivity issues for RK schemes, settle a theoretical framework that is valid not only for contractivity but also for monotonicity.

With a different terminology and notation, the numerical preservation of monotonicity has also been investigated in the context of hyperbolic systems of conservation laws. In this setting, for different reasons, it is critical to deal with Total Variation Diminishing (TVD) schemes, and in the pioneering papers [1,27], monotonicity issues for the Total Variation semi-norm are analyzed. In these references, high order methods satisfying (5) when the forward Euler discretization of (1) satisfies (4) are studied. In this context, these methods are known as SSP methods.

The idea in [1,26,27] is to construct high order schemes by means of convex combinations of forward Euler steps. Thus, RK schemes (2), that in compact form are written as

$$Y = e \otimes y_n + (\mathbb{A} \otimes I_N) F(Y)\,, \tag{6}$$

with $Y = (Y_1, \ldots, Y_s, y_{n+1})^t \in \mathbb{R}^{(s+1)N}$, $F(Y) = (f(Y_1), \ldots, f(Y_s), 0)^t \in \mathbb{R}^{(s+1)N}$, and

$$\mathbb{A} = \begin{pmatrix} A & 0 \\ b^t & 0 \end{pmatrix}\,, \tag{7}$$

can be expressed as

$$Y = \alpha_r \otimes y_n + (\Lambda_r \otimes I_N)\left(Y + \frac{h}{r} F(Y) \right)\,, \tag{8}$$

where $r \in \mathbb{R}$ and

$$\alpha_r = (I + r\mathbb{A})^{-1} e\,, \qquad \Lambda_r = r(I + r\mathbb{A})^{-1}\mathbb{A}\,, \tag{9}$$

with $e = (1, \ldots, 1)^t$. If $\alpha_r \geq 0$ and $\Lambda_r \geq 0$, where the inequalities should be understood component-wise, then the right hand side of (8) is a convex combination of y_n and forward Euler steps. The radius of absolute monotonicity, also known as Kraaijevanger's coefficient or SSP coefficient, is defined by

$$R(\mathbb{A}) = \sup \left\{ r \mid r = 0 \text{ or } r > 0, (I + r\mathbb{A})^{-1} \text{ exits, and } \alpha_r \geq 0,\, \Lambda_r \geq 0 \right\}\,. \tag{10}$$

If the forward Euler method satisfies condition (4), then, from (8), numerical monotonicity (5) can be proven under the stepsize restriction

$$h \leq R(\mathbb{A})\, \Delta t_{FE}\,.$$

If $R(\mathbb{A}) > 0$, the method is said to be SSP. Irreducible coefficient RK schemes (A, b^t) are SSP if and only if

$$A \geq 0\,, \qquad b > 0\,, \qquad \mathrm{Inc}(A^2) \leq \mathrm{Inc}(A)\,, \tag{11}$$

where $\mathrm{Inc}(A)$ denotes the incidence matrix of the matrix A defined as $\mathrm{Inc}(A) = (g_{ij})$ with $g_{ij} = 1$ if $a_{ij} \neq 0$ and $g_{ij} = 0$ if $a_{ij} = 0$ [26, Theorem 4.2].

In the rest of the paper, we denote s-stage pth order SSP schemes by SSP(s, p). Optimal SSP(s, p) methods, in the sense that their SSP coefficient is the largest possible one for a given number of stages s and order p, are well known in the literature (see, e.g., [2] and the references therein). For some combinations of the pair (s, p) there is a unique optimal method, e.g., $(s, 1)$, $(s, 2)$, $(3, 3)$, $(4, 3)$ or $(5, 4)$ [26]. However, for some other values, e.g., $(s, p) = (5, 3)$ there is a family of optimal SSP schemes. Optimal SSP(5,3) methods were studied in [4] and two schemes of this family were given in [2,15]; furthermore, the package RK-Opt in [14,28] can be used to obtain other optimal SSP(5,3) methods.

Expression (8) is a particular case of Shu–Osher representations of a RK method (see, e.g., [29, Section 2]). Given a RK method with Butcher matrix $\mathbb{A}$, a representation is given in terms of two matrices (Λ, Γ) such that the matrix $I - \Lambda$ is invertible and $\mathbb{A} = (I - \Lambda)^{-1}\Gamma$; then, the numerical approximation of the RK scheme is written as

$$Y = \alpha \otimes y_n + (\Lambda \otimes I_N) Y + h(\Gamma \otimes I_N) F(Y)\,, \tag{12}$$

where $\alpha = (I - \Lambda)e$. For explicit RK methods, as $Y_1 = y_n$, we can consider $\alpha = (1, 0, \ldots, 0)^t$. Adding and subtracting the term $r(\Gamma \otimes I_N)Y$, Eq. (12) can also be written as

$$Y = \alpha \otimes y_n + ((\Lambda - r\,\Gamma) \otimes I_N)\, Y + r(\Gamma \otimes I_N)\left(Y + \frac{h}{r} F(Y) \right)\,. \tag{13}$$

If the following component-wise inequalities hold

$$\Lambda \geq 0\,, \quad \Gamma \geq 0\,, \quad \alpha \geq 0\,, \quad \Lambda - r\,\Gamma \geq 0\,, \tag{14}$$

then the right hand side of Eq. (13) is a convex combination of y_n, the internal stages and forward Euler steps. For $r = R(\mathbb{A})$, it can be proven [29, Proposition 2.7] that there exist Shu–Osher representations (Λ, Γ) such that inequalities (14) hold. Observe that the largest value r in (13) that satisfies $\Lambda - r\,\Gamma \geq 0$ is given by $r = \min_{ij}\{\lambda_{ij}/\gamma_{ij}\}$, that agrees with the SSP coefficient of a RK method defined in the context of TVD schemes (see, e.g., [27]; see too [2] and the references therein). In other words, these representations are optimal. For example, α_r and Λ_r in (9), together with $\Gamma_r := \Lambda_r/r$, give an optimal representation. Observe that, in this case, $\Lambda_r - r\Gamma_r = 0$ and (13) is reduced to (8).

For a detailed study on numerical monotonicity and SSP methods, see, e.g., [13,19,21,22,30–33]. Efficient SSP RK methods have also been analyzed in [1,10,11,15,16]; see too [2] and the references therein.

2.2. Low-storage 2N, 2N* and 3N methods

Low-storage RK methods are very desirable to solve problems where memory management considerations are at least as important as stability considerations. In the literature, different approaches to reduce the memory computer usage of forward RK methods have been proposed [9–18,34,35].

A naive implementation of an explicit s-stage RK method requires $s+1$ memory registers. However, more efficient implementations are possible if some algebraic relations on the coefficients are imposed. Most of these efficient implementations are based on the ideas of Williamson [18] and van der Houwen [17]. Although in a very different way, in both cases it is possible to implement these RK methods in two memory registers, and they are usually called $2N$ schemes, where N is the dimension of the differential problem (1).

More recently, in the context of SSP methods, low-storage implementations have been obtained from the sparse structure of the Shu–Osher form (12) of optimal SSP methods [11,13,15]; this is the case for optimal SSP(s,1), SSP(s,2), SSP(3,3), SSP(4,3) and SSP(n^2, 3) schemes. In this combined analysis, some optimal SSP RK methods turn out to be optimal also in terms of the storage required for their implementation.

In some cases, the sparse structure of the Shu–Osher matrices in (12) enables a $2N$ low-storage implementation. However, some of these low-storage schemes do not retain y_n, the previous time step approximation, and they require a third memory register to save this value. Recall that, if y_n is retained during all the step, it can be used to check some accuracy or stability condition (e.g., for a variable stepsize implementation) without additional memory usage. To differentiate both implementations, the $2N$ low-storage RK methods that retain the numerical solution of the previous step are marked with $2N^*$ [2, Section 6.1.3].

In this paper we consider 5-stage Runge–Kutta methods with a canonical Shu–Osher representation of the form

$$\Lambda = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ \lambda_{31} & \lambda_{32} & 0 & 0 & 0 & 0 \\ \lambda_{41} & \lambda_{42} & \lambda_{43} & 0 & 0 & 0 \\ \lambda_{51} & \lambda_{52} & \lambda_{53} & \lambda_{54} & 0 & 0 \\ \lambda_{61} & \lambda_{62} & \lambda_{63} & \lambda_{64} & \lambda_{65} & 0 \end{pmatrix}, \qquad \Gamma = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ \gamma_{21} & 0 & 0 & 0 & 0 & 0 \\ 0 & \gamma_{32} & 0 & 0 & 0 & 0 \\ 0 & 0 & \gamma_{43} & 0 & 0 & 0 \\ 0 & 0 & 0 & \gamma_{5,4} & 0 & 0 \\ 0 & 0 & 0 & 0 & \gamma_{6,4} & 0 \end{pmatrix}, \tag{15}$$

where $\Lambda e = (0, 1, \ldots, 1)^t$. If $\lambda_{42} = \lambda_{52} = \lambda_{53} = \lambda_{62} = \lambda_{63} = \lambda_{64} = 0$, then a $2N^*$ implementation is possible (see [4, Algorithm 1]). In particular, optimal SSP(s,2), SSP(3,3) and SSP(4,3) schemes are $2N^*$ low-storage methods [2,13]. On the other hand, if either

$$\lambda_{42} = \lambda_{52} = \lambda_{62} = \lambda_{64} = 0\,, \qquad \text{or} \qquad \lambda_{53} = \lambda_{63} = \lambda_{64} = 0\,, \tag{16}$$

then a $3N$ implementation is possible (see [4, Algorithms 2 and 3]).

2.3. Low-storage SSP(5,3) methods

In this section we review the most relevant properties of SSP(5,3) methods that will be used in the rest of the paper. First, we analyze the structure of the family of optimal SSP(5,3) methods and next, the form of $2N^*$ low-storage SSP(5,3) methods.

2.3.1. Optimal SSP(5,3) methods

The first optimal SSP(5,3) methods were found by numerical search in [15,16]; nowadays, different optimal SSP(5,3) methods can be numerically constructed with the code RK-Opt [28]. The recent study done in [4] showed that the family of optimal SSP(5,3) methods has a Butcher tableau of the form

$$\begin{array}{c|ccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ c_2 & \frac{1}{r} & 0 & 0 & 0 & 0 \\ c_3 & \frac{1}{r} & \frac{1}{r} & 0 & 0 & 0 \\ c_4 & a_{41} & a_{41} & a_{41} & 0 & 0 \\ c_5 & a_{51} & a_{52} & a_{52} & a_{54} & 0 \\ \hline & b_1 & b_2 & b_3 & b_4 & b_5 \end{array} \tag{17}$$

where

$$a_{41}=\frac{r}{60\,b_4}\,,\quad a_{52}=\frac{r}{60\,b_5}\,,\quad a_{54}=\frac{b_4}{b_5\,r}\,,\quad b_3=\frac{r^2}{60}\,, \tag{18}$$

and $r=R(\mathbb{A})$ is the real root of the polynomial $x^3-5x^2+10x-10=0$. The stability function of these schemes is

$$R(z)=1+z+\frac{z^2}{2}+\frac{z^3}{6}+\frac{z^4}{12r}+\frac{z^5}{60r^2}\,, \tag{19}$$

that agrees with the optimal polynomial $\Phi_{5,3}$ that gives the optimal threshold factor $r=R_{5,3}$ for linear problems [21, Theorem 5.2]. In this way, the third order condition $b^tAc=1/6$ is fulfilled and only the following order conditions

$$b^te=1\,,\qquad b^tc=\frac{1}{2}\,,\qquad b^tc^2=\frac{1}{3}\,, \tag{20}$$

must be imposed to (17)–(18). After imposing conditions (20), the five free parameters in (17)–(18) namely, a_{51}, b_1, b_2, b_4 and b_5, can be reduced to two free parameters. If we assume that b_4 and b_5 are the free parameters and a_{51}, b_1 and b_2 are functions of b_4 and b_5, then the unique solution of (20) with positive solutions is the following one:

$$a_{51}=\frac{-2b_4\left(30b_4-15b_5+r^2\right)+3\sqrt{d}}{60b_4b_5r}\,,\quad b_1=1-b_4-\frac{b_5}{2}-\frac{r}{2}-\frac{r^2}{12}+\frac{\sqrt{d}}{20b_4}\,,\quad b_2=-\frac{b_5}{2}+\frac{r}{2}+\frac{r^2}{15}-\frac{\sqrt{d}}{20b_4}\,, \tag{21}$$

where $d=b_4b_5\left(20b_4(5b_5+r(7r-10))-r^4\right)$. In this way, not only the construction but also the study of relevant properties (e.g., size of local error constant) of optimal SSP(5,3) methods are simplified considerably.

As it has been pointed out in the introduction, the Butcher coefficients (A,b^t) for SSP Runge–Kutta methods must satisfy conditions (11). For schemes of the form (17)–(18) the sign conditions are satisfied if and only if $a_{51}\geq 0$ and b_1, b_2, b_4, $b_5>0$. Furthermore, $a_{51}>0$; otherwise, if $a_{51}=0$, the condition on the incidence matrices (11) implies that $a_{52}a_{21}+a_{52}a_{31}+a_{54}a_{41}=0$; but this is impossible for methods (17)–(18) as the left hand side of this expression is different from zero.

Canonical Shu–Osher representations for the 2-parametric family of optimal methods (17)–(20) were also studied in [4]. There the minimum number of memory registers required to implement optimal SSP(5,3) methods was determined by using the sparse optimal Shu–Osher representation (Λ,Γ) with the subdiagonal matrix Γ given below (see [4] for details).

$$\Lambda=\begin{pmatrix}0&0&0&0&0&0\\1&0&0&0&0&0\\0&1&0&0&0&0\\\lambda_{41}&0&\frac{r^2}{60b_4}&0&0&0\\\lambda_{51}&\lambda_{52}&0&\frac{b_4}{b_5}&0&0\\0&\lambda_{62}&\lambda_{63}&0&b_5r&0\end{pmatrix}\,,\qquad \Gamma=\begin{pmatrix}0&0&0&0&0&0\\\frac{1}{r}&0&0&0&0&0\\0&\frac{1}{r}&0&0&0&0\\0&0&\frac{r}{60b_4}&0&0&0\\0&0&0&\frac{b_4}{b_5r}&0&0\\0&0&0&0&b_5&0\end{pmatrix}\,, \tag{22}$$

$$\lambda_{41}=1-\frac{r^2}{60b_4}\,, \tag{23a}$$

$$\lambda_{51}=1-\frac{b_4}{b_5}-r\left(a_{51}-\frac{r}{60b_5}\right)\,,\quad \lambda_{52}=r\left(a_{51}-\frac{r}{60b_5}\right)\,, \tag{23b}$$

$$\lambda_{62}=r\left(b_1-b_2-rb_5\left(a_{51}-\frac{r}{60b_5}\right)\right)\,,\quad \lambda_{63}=r\left(b_2-\frac{r^2}{60}\right)\,. \tag{23c}$$

Again the coefficients a_{51}, b_1 and b_2 must be understood as the functions defined in (21) depending on the parameters b_4 and b_5. In order to obtain a representation (22) with non-negative coefficients we only require the inequalities

$$\lambda_{52}\geq 0,\ \lambda_{62}\geq 0,\ \lambda_{63}\geq 0\,, \tag{24}$$

as we have numerically obtained that the non-negativity of the other coefficients in (23), namely λ_{41} and λ_{51}, is redundant. The curves defined by equations $\lambda_{52}=0$, $\lambda_{62}=0$ and $\lambda_{63}=0$, in (24), are represented in Fig. 1. The region limited by these curves encloses all optimal SSP(5,3) methods. Although all of them are optimal methods, their local error norm $\|C_{err}\|_2$, where C_{err} is the vector containing the coefficients of the leading truncation error [36, p. 158], namely

$$C_{err}=\frac{1}{4!}\left(1-12\,b^tAc^2,\ 1-24\,b^tA^2c,\ 3(1-8\,b^t(Ac\cdot c)),\ 1-4\,b^tc^3\right)\,, \tag{25}$$

or their low-storage properties are not the same; to stress the first fact we have added some contour lines for the error norm in Fig. 1.

With regard to low-storage implementation, in [4] it was proven that optimal SSP(5,3) methods cannot be implemented in $2N$ memory registers. However, if $\lambda_{52}=\lambda_{62}=0$ or $\lambda_{63}=0$ in (22), then $3N$ implementations are possible. In the first

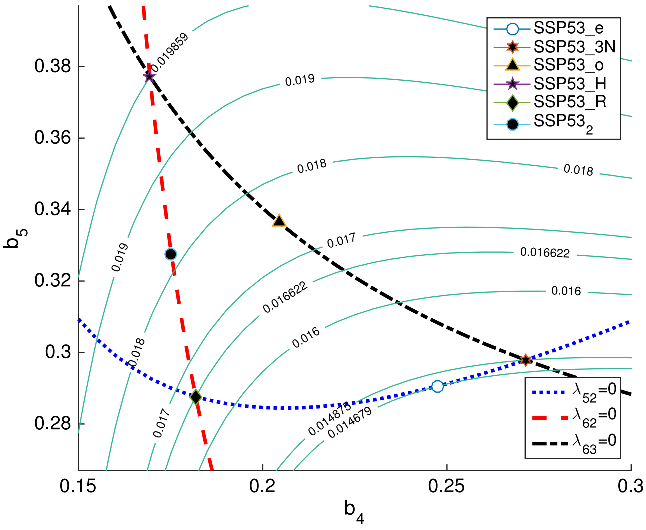

Fig. 1. (b_4, b_5)-region for optimal SSP(5,3) methods (17)–(18) and level sets of the local error norm $\|C_{err}\|_2$. The six optimal SSP(5,3) methods considered in this paper are shown with different symbols.

case, $\lambda_{52} = \lambda_{62} = 0$, there is only a method satisfying both restrictions, the one with $(b_4, b_5) = (0.181803, 0.287632)$ (see the intersection of curves $\lambda_{52} = 0$ and $\lambda_{62} = 0$ in Fig. 1). This method was first numerically obtained in [15] and in this paper we will refer to it as SSP53_R. For the second case, $\lambda_{63} = 0$, there exist a family of methods that can be implemented in three memory registers: all the schemes obtained with the values of the parameters (b_4, b_5) along the curve $\lambda_{63} = 0$ in Fig. 1.

In terms of the Butcher coefficients, for optimal SSP(5,3) methods (17)–(18), trivially $\lambda_{63} = 0$ in (23c) is equivalent to $b_2 = r^2/60$, and thus, from (18), condition $b_2 = b_3$ holds. In a similar way, $\lambda_{52} = \lambda_{62} = 0$ in (23) is equivalent to $a_{51} = r/(60\, b_5)$ and $b_1 = b_2$. In this case, from (18) we obtain that $a_{51} = a_{52}$. Consequently, if one of the following statements holds,

1. $b_2 = b_3$,
2. $b_1 = b_2$ and $a_{51} = a_{52}$,

then optimal SSP(5,3) methods (17)–(18) can be implemented in $3N$ memory registers.

2.3.2. $2N^*$ Low-storage SSP(5,3) methods

In this section we consider 5-stage $2N^*$ methods of the form (15). These schemes, studied in [4], have a simple Butcher tableau of the form

$$\begin{array}{c|ccccc}
0 & 0 & 0 & 0 & 0 & 0 \\
ub_1 & ub_1 & 0 & 0 & 0 & 0 \\
v(b_1+b_2) & vb_1 & vb_2 & 0 & 0 & 0 \\
w(b_1+b_2+b_3) & wb_1 & wb_2 & wb_3 & 0 & 0 \\
x(b_1+b_2+b_3+b_4) & xb_1 & xb_2 & xb_3 & xb_4 & 0 \\
\hline
 & b_1 & b_2 & b_3 & b_4 & b_5
\end{array} \tag{26}$$

where b_i , $i = 1, \ldots, 5$, u, v, w and x are free parameters. This representation allows an easier way of dealing with third order conditions (see [4] for details).

In [4] we obtained two methods, namely SSP53_2N$_1^*$ and SSP53_2N$_2^*$, that will be used later in the numerical experiments. In the table below we summarize some properties of these schemes.

Method	$R(\mathbb{A})$	Coefficients in (26)	Λ matrix in (15)	$\|C_{err}\|_2$
SSP53_2N$_1^*$	2.1807	$u = v = w,\ x = 1$	Λ_1 in (27)	0.027841
SSP53_2N$_2^*$	2.1487	$u = v,\ w = x$	Λ_2 in (27)	0.022736

Method SSP53_2N$_1^*$ has the largest possible SSP coefficient and, although the SSP coefficient for the other scheme SSP53_2N$_2^*$ is not the largest one, this method has some other relevant properties (see [4] for details). Each of these

Please cite this article as: I. Higueras and T. Roldán, Efficient SSP low-storage Runge–Kutta methods, Journal of Computational and Applied Mathematics (2019) 112500, https://doi.org/10.1016/j.cam.2019.112500.

methods have a particular sparse structure for the Shu–Osher matrix Λ in (15):

$$\Lambda_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \lambda_{51} & 0 & 0 & \lambda_{54} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \qquad \Lambda_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ \lambda_{41} & 0 & \lambda_{43} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ \lambda_{61} & 0 & 0 & 0 & \lambda_{65} & 0 \end{pmatrix}. \tag{27}$$

3. Efficient SSP(5,3) methods

In this section we construct some new SSP(5,3) methods that will be tested in the numerical experiments. First, we consider the family of optimal SSP(5,3) methods. They have the same SSP coefficient and, therefore, comparisons of observed SSP coefficients are more reliable. In the second part we consider $2N^*$ low-storage SSP(5,3) methods.

3.1. Efficient 5-stage third order optimal SSP methods

For the family of optimal SSP(5,3) methods (17)–(18), we focus our attention on the local error constants $\|C_{err}\|_2$ of the schemes (see (25)) and their low-storage properties. First, we use standard numerical optimization techniques to get the 5 unknowns in (17), namely a_{51}, b_1, b_2, b_4 and b_5, that minimize $\|C_{err}\|_2$. More precisely, we solve the following optimization problem:

$$\begin{aligned} &\text{Minimize } \|C_{err}\|_2 \text{ subject to:} \\ &\quad \text{Method of the form (17)–(18)}, \\ &\quad a_{51}, b_1, b_2, b_4, b_5 \geq 0, \\ &\quad \lambda_{52} \geq 0,\ \lambda_{62} \geq 0,\ \lambda_{63} \geq 0, \\ &\quad \text{Third order conditions (20).} \end{aligned} \tag{28}$$

We have used the function *fmincon* in Matlab to get the five positive unknowns in (28). There we have considered $\|C_{err}\|_2$ as the objective function, the third order conditions (20) as the nonlinear equality constrains, and λ_{52}, λ_{62}, $\lambda_{63} \geq 0$, as the nonlinear inequality constraints. The minimum value for the local error is $\|C_{err}\|_2 = 0.014679$, obtained when $(b_4, b_5) = (0.247411, 0.290558)$. We have represented this method, named SSP53_e, with an empty circle in Fig. 1. Observe that though this method is on the curve $\lambda_{52} = 0$, the corresponding scheme cannot be implemented in $3N$ memory registers as $\lambda_{53} \neq 0$ and $\lambda_{63} \neq 0$ (see (16)). The Butcher and the Shu–Osher coefficients of this method are given in the Appendix, Eq. (A.1).

Next, we get the $3N$ low-storage optimal SSP(5,3) method with minimum error constant $\|C_{err}\|_2$. As all points along curve $\lambda_{63} = 0$ correspond to methods that can be implemented in $3N$ memory registers, we simply have to modify the optimization problem (28) by setting $\lambda_{63} = 0$. For these $3N$ methods the minimum value for the error norm is $\|C_{err}\|_2 = 0.014875$. This scheme is achieved at $(b_4, b_5) = (0.271439, 0.297885)$, the point where curves $\lambda_{62} = 0$ and $\lambda_{52} = 0$ meet. We have denoted this method by SSP53_3N. Its Butcher coefficients (A.2) can be seen in the Appendix.

Finally, modifying accordingly problem (28), we construct the optimal SSP(5,3) method with the highest error, $\|C_{err}\|_2 = 0.019859$. This method, referred as SSP53_H in [4], can also be obtained from the intersection of curves $\lambda_{63} = 0$ and $\lambda_{62} = 0$, that happens at point $(0.169383, 0.377269)$. Observe that this method can be implemented in $3N$ memory registers.

3.2. Efficient $2N^*$ low-storage SSP(5,3) methods

In this section we consider the family of methods of the form (26) and look for a numerically optimal third order method with respect the 2-norm of the coefficients in the leading term of the local error. For this purpose, we use standard numerical optimization techniques to get the 9 unknowns in (26), namely b_i, $i = 1, \ldots, 5$; u, v, w, x. More precisely, we have solved the following optimization problem.

$$\begin{aligned} &\text{Minimize } \|C_{err}\|_2 \text{ subject to:} \\ &\quad \text{Method of the form (26)}, \\ &\quad b_i \geq 0,\ i = 1, \ldots, 5, \\ &\quad u \geq v \geq w \geq x \geq 1, \\ &\quad \text{Third order conditions (20).} \end{aligned} \tag{29}$$

We have used again the function *fmincon* in Matlab to get the nine positive unknowns in (29). We have proceeded in a similar way as in the optimization problem (28), but now for methods with the structure defined in (26). Unfortunately, the minimum value of the error, namely $\|C_{err}\|_2 = 0.01212$, is obtained for a method $\mathbb{A}$ with a poor SSP coefficient,

$R(\mathbb{A}) = 0.7451$. Due to this bad result, we restrict the study to methods of the form (26) satisfying $u = v = w$ and $x = 1$ whose matrix Λ in the Shu–Osher representation (15) is of the form Λ_1 in (27). We proceed in this way because scheme SSP53_2N$_1^*$ in [4], with the largest SSP coefficient, belongs to this family (see Section 2.3.2). For this subfamily of schemes we have used the function *fmincon* in Matlab to get the 6 unknowns, namely b_i, $i = 1, \ldots, 5$ and u. The optimization problem is the following one:

$$\begin{aligned}
&\text{Minimize } \|C_{err}\|_2 \text{ subject to:}\\
&\quad \text{Method of the form (26)},\\
&\quad b_i \geq 0\,,\ \ i = 1, \ldots, 5\,,\\
&\quad u = v = w \geq x = 1\,,\\
&\quad \text{Third order conditions (20)}.
\end{aligned} \tag{30}$$

In this case, the minimum value for the error, $\|C_{err}\|_2 = 0.025407$, is obtained for a Runge–Kutta method $\mathbb{A}$ with SSP coefficient $R(\mathbb{A}) = 1.8229$. The coefficients of this method (A.4), named SSP53_2N$_3^*$, are given in the Appendix Section.

Finally, we have used again numerical optimization for the restricted family of $2N^*$ methods with $u = v$ and $w = x$ in (26) and Shu–Osher matrix of the form Λ_2 in (27). Remember that the scheme SSP53_2N$_2^*$ in [4] belongs to this family (see Section 2.3.2). We have used again the function *fmincon* in Matlab to solve the optimization problem

$$\begin{aligned}
&\text{Minimize } \|C_{err}\|_2 \text{ subject to:}\\
&\quad \text{Method of the form (26)},\\
&\quad b_i \geq 0\,,\ \ i = 1, \ldots, 5\,,\\
&\quad u = v \geq w = x \geq 1\,,\\
&\quad \text{Third order conditions (20)}.
\end{aligned} \tag{31}$$

We have denoted SSP53_2N$_4^*$ the method with the minimum local error $\|C_{err}\|_2 = 0.015458$. This method has SSP coefficient $R(\mathbb{A}) = 1.4252$, and its coefficients (A.5) are given in the Appendix Section.

4. Numerical experiments

In this section we analyze the performance of some optimal and non optimal SSP(5,3) methods when the hyperbolic 1-dimensional Buckley–Leverett problem is solved. The exact solution for this problem is Total Variation Diminishing (TVD) and our goal is to study if there is any relationship between the observed SSP coefficient and the norm of the leading term of the local error.

The hyperbolic 1-dimensional Buckley–Leverett equation is defined by (see, e.g., [19,20])

$$\frac{\partial}{\partial t}u(x,t) + \frac{\partial}{\partial x}\Phi(u(x,t)) = 0\,, \qquad \text{with} \qquad \Phi(u) = \frac{3u^2}{(1-v)^2}\,. \tag{32}$$

We consider $0 \leq x \leq 1$, $0 \leq t \leq 1/8$, periodic boundary condition $u(0,t) = u(1,t)$ and initial condition

$$u(x,0) = \begin{cases} 0 & \text{for } 0 < x \leq 1/2,\\ \frac{1}{2} & \text{for } \frac{1}{2} < x \leq 1\,. \end{cases}$$

We semi-discretize this problem using a uniform grid with mesh-points $x_j = j\Delta x$ where $j = 1, 2, \ldots, N$ and $\Delta x = 1/N$, $N = 100$. We denote $U_j(t) \approx u(x_j, t)$ and we approximate (32) by the system of ordinary differential equations

$$U_j'(t) = \frac{1}{\Delta x}(\Phi(U_{j-1/2}(t)) - \Phi(U_{j+1/2}(t))), \qquad j = 1, 2, \ldots, N, \tag{33}$$

where

$$U_{j+1/2}(t) = U_j + \frac{1}{2}\phi(\theta_j)\left(U_{j+1} - U_j\right)\,,$$

and $\phi(\theta)$ is the Koren's limiter defined by

$$\phi(\theta) = \max\left(0, \min\left(2, \frac{2}{3} + \frac{1}{3}\theta, 2\theta\right)\right)\,, \quad \text{where} \quad \theta_j = \frac{U_j - U_{j-1}}{U_{j+1} - U_j}\,.$$

In order to compute the observed SSP coefficient for a given explicit RK, we have integrated (33) with different stepsizes, from $\Delta t = 2 \cdot 10^{-3}$ to $\Delta t = 10^{-2}$. For each stepsize Δt, the maximal ratio of the TV-seminorm of two consecutive numerical approximations, in the time interval $[0, 1/8]$, is computed

$$\mu(\Delta t) = \max\left\{\frac{\|u_n\|_{TV}}{\|u_{n-1}\|_{TV}} \mid n \geq 1\,, \text{with } n\Delta t \leq 1/8\right\}.$$

If $\mu(\Delta t) = 1$, then the explicit RK method is Total Variation Diminishing (TVD) on the interval $[0, 1/8]$, that is $\|u_n\|_{TV} \leq \|u_{n-1}\|_{TV}$ (see [19] for details).

Table 1
Theoretical and observed SSP coefficients, error constant and number of memory registers.

		SSP coefficient $R(\mathbb{A})$	Observed SSP coefficient $R_{obs}(\mathbb{A})$	Gain %	Error constant	Number of registers
SSP53_o	(A.3)	2.6506	3.088	16.50%	1.75000e−02	$3N$
SSP53_e	(A.1)	2.6506	3.008	13.48%	1.46786e−02	$\geq 3N$
SSP53_3N	(A.2)	2.6506	2.968	11.97%	1.48753e−02	$3N$
SSP53_R	[4,15]	2.6506	2.916	10.01%	1.66219e−02	$3N$
SSP53$_2$	[4]	2.6506	2.788	5.18%	1.81787e−02	$\geq 3N$
SSP53_H	[4]	2.6506	2.740	3.37%	1.98589e−02	$3N$
SSP53_2N_1^*	[4]	2.1807	2.300	5.47%	2.78407e−02	$2N^*$
SSP53_2N_2^*	[4]	2.1487	2.444	13.74%	2.27362e−02	$2N^*$
SSP53_2N_3^*	(A.4)	1.8229	2.292	25.73%	2.54073e−02	$2N^*$
SSP53_2N_4^*	(A.5)	1.4252	2.184	53.20%	1.54584e−02	$2N^*$

Table 2
Optimal SSP(5,3) methods with the maximum observed SSP coefficient for a prescribed error constant.

		Error constant	Max. observed SSP coefficient	Gain %	$3N$ method	# registers $\geq 3N$
SSP53_H	[4]	1.98589e-02	2.740	3.37%	✓	
SSP53_1.95		1.95000e-02	2.772	4.58%	✓	
SSP53_1.90		1.90000e-02	2.880	8.65%	✓	
SSP53_1.85		1.85000e-02	2.996	13.03%	✓	
SSP53_1.80		1.80000e-02	3.072	15.90%	✓	
SSP53_o	(A.3)	1.75000e-02	3.088	16.50%	✓	
SSP53_1.70		1.70000e-02	3.084	16.35%	✓	
SSP53_1.65		1.65000e-02	3.076	16.05%	✓	
SSP53_1.60		1.60000e-02	3.060	15.45%	✓	
SSP53_1.55		1.55000e-02	3.032	14.39%		✓
SSP53_3N	(A.2)	1.48753e-02	3.008	13.48%		✓
SSP53_e	(A.1)	1.46786e-02	3.008	13.48%		✓

For this problem we have obtained that forward Euler method is TVD for $0 \leq \Delta t \leq \Delta t_{FE}^{obs} \simeq 0.0025$. For different schemes $\mathbb{A}$ we have repeated this computation to obtain the value $\Delta t_{\mathbb{A}}^{obs}$ such that $\mu(\Delta t_{\mathbb{A}}^{obs}) = 1$; then the quotient $\Delta t_{\mathbb{A}}^{obs}/\Delta t_{FE}^{obs}$ gives the observed SSP coefficient of scheme $\mathbb{A}$, that we will denote by $R_{obs}(\mathbb{A})$.

In Table 1 we summarize the numerical results obtained as well as some information on the schemes considered. More precisely, for each method, we give the theoretical SSP coefficient and the observed SSP coefficient for this hyperbolic problem. We also add the gain of the observed SSP coefficient with regard to the theoretical one, the error constant norm $\|C_{err}\|_2$ defined in (25) and, finally, the number of memory registers needed for the implementation. Next, we discuss the results obtained.

Optimal SSP(5,3) methods

First, we have considered six schemes of the family of optimal SSP(5,3) Runge–Kutta methods: the method SSP53_o with the highest observed SSP coefficient (we explain in the next paragraph how we have obtained this method), and the new methods SSP53_e and SSP53_3N obtained in Section 3.1; besides, we have considered other methods from the literature, namely SSP53_R, SSP53$_2$ and SSP53_H [4,15]. We observe that, although all of them have the same optimal SSP coefficient, $R(\mathbb{A}) = 2.6506$, the observed coefficients vary; indeed, from the smallest one, $R_{obs}(\mathbb{A}) = 2.740$, to the largest one, $R_{obs}(\mathbb{A}) = 3.088$, there is a gain of 16.50%. If we study the local error constants, it seems that the smaller these constants are, the larger the observed SSP coefficients are. However, this is not true for the method SSP53_o.

In order to further analyze the relationship between the error constant and the observed SSP coefficient, we have also computed the observed SSP coefficient for many other optimal SSP(5,3) methods: for the 2-parameter family of optimal SSP(5,3) methods (17), we have fixed the value of the local error, say $\|C_{err}\|_2 = c$, and we have numerically obtained both, the method with the maximum and the one with the minimum observed SSP coefficient; we have repeated this process for different values of c in the range of possible values $[0.014679, 0.019859]$. In Table 2 we can see, for these values of c, the method with the maximum value of the observed SSP coefficient. We have also added information about the gain with respect to the theoretical SSP coefficient, and the number of registers needed in the implementation. The maximum observed SSP coefficient, $R_{obs}(\mathbb{A}) = 3.088$, is obtained when $c = 0.0175$. We have denoted this method SSP53_o; its coefficients (A.3) have been written down in the Appendix. In Fig. 2 we show the observed SSP coefficient vs. the 2-norm of the local error for the different optimal SSP(5,3) considered. There we have denoted with white circles (red squares), joined with a blue line (joined with a green line), the methods with the maximum (minimum) observed SSP coefficient. In the range $[0.0175, 0.019859]$ it is true that the smaller the local error is, the larger the maximum observed SSP coefficient is. However, once the maximum value $R_{obs}(\mathbb{A}) = 3.088$ is achieved at $\|C_{err}\|_2 = 0.0175$, there is a light decrease of the maximum observed SSP coefficient.

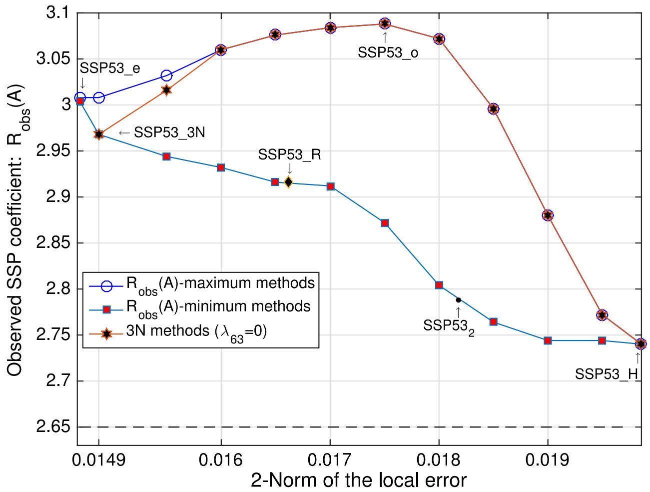

Fig. 2. Observed SSP coefficient $R_{obs}(\mathbb{A})$ vs. 2-norm for the local error for optimal SSP(5,3) methods (17)–(18).

We have also considered the subfamily of optimal SSP(5,3) methods with $\lambda_{63} = 0$ (see Fig. 1). This is a 1-parameter family of optimal SSP(5,3) methods that can be implemented in 3N memory registers. In this family each value of the local error, $\|C_{err}\|_2 = c$, ranged in the interval [0.014875, 0.019859], determines a unique optimal SSP(5,3) method. In Fig. 2 we have denoted with six-pointed stars, joined with a red line, these $3N$ methods. Surprisingly, circles and six-pointed stars coincide in the range of values [0.016, 0.019859]. This means that in this interval the maximum observed SSP coefficient $R_{obs}(\mathbb{A})$ is obtained for optimal SSP(5,3) methods that can be implemented in $3N$ memory registers. On the other hand, when the error c is in the interval [0.01467859, 0.016), the maximum observed SSP coefficient corresponds to optimal SSP(5,3) methods that cannot be implemented in $3N$ memory registers.

We have also found that, for each value of c, the minimum observed SSP coefficient (red squares in Fig. 2) is obtained when $\lambda_{52} = 0$ or $\lambda_{62} = 0$ (see also Fig. 1). Optimal SSP(5,3) method SSP53_R in [15], with error constant $\|C_{err}\|_2 = 0.0166219$ and obtained when $\lambda_{52} = \lambda_{62} = 0$, has been represented in Fig. 2 with a diamond symbol (see also Fig. 1). Observe that the line joining methods SSP53_R and SSP53_H in Fig. 2 corresponds with the curve $\lambda_{62} = 0$ in Fig. 1, while the line joining the methods SSP53_e and SSP53_R corresponds with the curve $\lambda_{52} = 0$.

Non-optimal $2N^*$ SSP(5,3) methods

Finally, we have considered some other SSP(5,3) Runge–Kutta methods that, although they are not optimal, they can be implemented in $2N^*$ memory registers: the new methods SSP53_2N_3^* and SSP53_2N_4^* obtained in Section 3.2, and other similar methods, namely SSP53_2N_1^* and SSP53_2N_2^*, aforementioned in Section 2.3.2. The numerical results have been added at the bottom of Table 1. In this case, the schemes have different SSP coefficients. We observe that the smaller the error constant is, the largest the gain with respect to the theoretical SSP coefficient is. The largest observed SSP coefficient, 2.444, is not obtained with the scheme with the largest theoretical SSP coefficient but with SSP53_2N_2^*, whose observed SSP coefficient is lightly lower.

5. Conclusions

In this paper we have studied the efficiency of SSP Runge–Kutta methods that can be implemented with a low number of memory registers using Shu–Osher representations. Our goal was to study the influence of the local error term in the observed stepsize restrictions for numerical monotonicity. For this purpose, we have considered the family of SSP(5,3) methods and the Buckley–Leverett equation. We have dealt with optimal SSP(5,3) methods, whose implementation requires at least 3 memory registers, and non-optimal $2N^*$ low-storage SSP(5,3) schemes. Several methods have been constructed. The numerical experiments done show that, for optimal SSP(5,3) methods:

- In general, schemes with small error constants provide larger observed SSP coefficients.
- If the local error constant is fixed, then in most of the cases the maximum observed SSP coefficient is obtained for a $3N$ low-storage method ($\lambda_{63} = 0$).
- From the point of view of the observed SSP coefficient, the unique $3N$ scheme with $\lambda_{52} = \lambda_{62} = 0$, that is, method SSP53_R, is not competitive with a conveniently constructed $3N$ schemes with $\lambda_{63} = 0$.

For $2N^*$ low-storage non-optimal SSP(5,3) methods, we have observed that:

- The smaller the error constant is, the larger the gain of the observed SSP coefficient with respect to the theoretical SSP one.
- As the error constant decreases, the theoretical SSP coefficient of the method decreases.

Consequently, in the search of efficient low-storage SSP methods, besides the size of the SSP coefficients, the magnitude of the error constants should also be taken into account. There must be a compromise between large theoretical SSP coefficients and small error constants as these variables seem to be inversely correlated. Furthermore, our experience with $3N$ low-storage optimal SSP(5,3) schemes highlights the interest of low-storage methods as a class of efficient schemes not only because of their low-storage properties but also because they may provide methods with large observed SSP coefficients.

In this paper we have considered explicit Runge–Kutta methods, but the study done can also be extended to other kind of SSP methods (e.g., two and multistep Runge–Kutta schemes [5–7], explicit SSP peer methods [8], etc.). This may be a topic for future work.

Acknowledgments

We would like to thank the anonymous reviewers for their suggestions and comments.

Appendix. Coefficients of the methods

In this section we show the coefficients of the methods obtained in this paper: the three new optimal SSP(5,3) methods named SSP53_e, SSP53_3N, and SSP53_o, and the two non optimal $2N^*$ low-storage SSP(5,3) schemes named SSP53_$2N^*_3$ and SSP53_$2N^*_4$. The coefficients of other methods considered in the numerical experiments can be found in the corresponding references. For each Runge–Kutta method we show both, its Butcher coefficients and its Shu–Osher form (Λ, Γ) with $\Lambda e = (0, 1, \ldots, 1)^t$ and subdiagonal matrix Γ. This subdiagonal structure can be obtained directly from (22)–(23c).

Scheme SSP53_e

This is the optimal SSP(5,3) method with the lowest value of the leading local error, $\|C_{err}\|_2 = 0.01467859$. It is obtained by solving the optimization problem (28). It cannot be implemented in $3N$ memory registers. The Butcher coefficients for this scheme are

$$
\begin{array}{c|ccccc}
0 & 0 & 0 & 0 & 0 & 0 \\
0.377268915331368 & 0.377268915331368 & 0 & 0 & 0 & 0 \\
0.754537830662736 & 0.377268915331368 & 0.377268915331368 & 0 & 0 & 0 \\
0.535673936262144 & 0.178557978754048 & 0.178557978754048 & 0.178557978754048 & 0 & 0 \\
0.777371470949368 & 0.152042242678717 & 0.152042242678717 & 0.152042242678717 & 0.321244742913218 & 0 \\
\hline
 & 0.203807751220298 & 0.141125888396921 & 0.117097251841844 & 0.247410692588023 & 0.290558415952914
\end{array}
\tag{A.1}
$$

and the Shu–Osher form is

$$
\Lambda = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ \lambda_{41} & 0 & \lambda_{43} & 0 & 0 & 0 \\ \lambda_{51} & 0 & 0 & \lambda_{54} & 0 & 0 \\ 0 & \lambda_{62} & \lambda_{63} & 0 & \lambda_{65} & 0 \end{pmatrix}, \qquad
\Gamma = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ \gamma_{21} & 0 & 0 & 0 & 0 & 0 \\ 0 & \gamma_{32} & 0 & 0 & 0 & 0 \\ 0 & 0 & \gamma_{43} & 0 & 0 & 0 \\ 0 & 0 & 0 & \gamma_{54} & 0 & 0 \\ 0 & 0 & 0 & 0 & \gamma_{65} & 0 \end{pmatrix}.
$$

$\lambda_{41} = 0.526709009150106$, $\lambda_{43} = 0.473290990849893$,
$\lambda_{51} = 0.148499306837781$, $\lambda_{54} = 0.851500693162219$,
$\lambda_{62} = 0.166146375373442$, $\lambda_{63} = 0.063691005483375$, $\lambda_{65} = 0.770162619143183$;
$\gamma_{21} = 0.377268915331368$, $\gamma_{32} = 0.377268915331368$, $\gamma_{43} = 0.178557978754048$, $\gamma_{54} = 0.321244742913218$,
$\gamma_{65} = 0.290558415952914$.

Scheme SSP53_3N

This is the optimal SSP(5,3) method with the lowest value of the leading local error, namely $\|C_{err}\|_2 = 0.01487531$, that can be implemented in $3N$ memory registers. It is obtained by solving the optimization problem (28) with the additional condition $\lambda_{63} = 0$. The Butcher coefficients for this scheme are

$$\begin{array}{c|cccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0.377268915331368 & 0.377268915331368 & 0 & 0 & 0 & 0 \\ 0.754537830662736 & 0.377268915331368 & 0.377268915331368 & 0 & 0 & 0 \\ 0.488254447100037 & 0.162751482366679 & 0.162751482366679 & 0.162751482366679 & 0 & 0 \\ 0.788683186188260 & 0.148302591520154 & 0.148302591520154 & 0.148302591520154 & 0.343775411627798 & 0 \\ \hline & 0.196480926343466 & 0.117097251841844 & 0.117097251841844 & 0.271439329143100 & 0.297885240829746 \end{array} \tag{A.2}$$

and the Shu–Osher form is

$$\Lambda = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ \lambda_{41} & 0 & \lambda_{43} & 0 & 0 & 0 \\ \lambda_{51} & 0 & 0 & \lambda_{54} & 0 & 0 \\ 0 & \lambda_{62} & 0 & 0 & \lambda_{65} & 0 \end{pmatrix}, \qquad \Gamma = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ \gamma_{21} & 0 & 0 & 0 & 0 & 0 \\ 0 & \gamma_{32} & 0 & 0 & 0 & 0 \\ 0 & 0 & \gamma_{43} & 0 & 0 & 0 \\ 0 & 0 & 0 & \gamma_{54} & 0 & 0 \\ 0 & 0 & 0 & 0 & \gamma_{65} & 0 \end{pmatrix}.$$

$\lambda_{41} = 0.568606169888847$, $\lambda_{43} = 0.4313938301111528$,
$\lambda_{51} = 0.088778858640267$, $\lambda_{54} = 0.911221141359733$,
$\lambda_{62} = 0.210416684957724$, $\lambda_{65} = 0.789583315042277$;
$\gamma_{21} = 0.377268915331368$, $\gamma_{32} = 0.377268915331368$, $\gamma_{43} = 0.162751482366679$, $\gamma_{54} = 0.343775411627798$,
$\gamma_{65} = 0.297885240829746$.

Scheme SSP53_o

This is the optimal SSP(5,3) method showing the highest value of the observed SSP coefficient $R_{obs}(\mathbb{A}) = 3.088$. It is numerically obtained by fixing the value of the local error, say $\|C_{err}\|_2 = c$, and analyzing the values of the observed SSP coefficient for these c-methods. The maximum value is obtained when $c = 0.0175$. It can be implemented in $3N$ memory registers and the norm of the leading truncation error is $\|C_{err}\|_2 = 0.0175$. The Butcher coefficients for this scheme are

$$\begin{array}{c|cccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0.377268915331368 & 0.377268915331368 & 0 & 0 & 0 & 0 \\ 0.754537830662736 & 0.377268915331368 & 0.377268915331368 & 0 & 0 & 0 \\ 0.648537741845154 & 0.216179247281718 & 0.216179247281718 & 0.216179247281718 & 0 & 0 \\ 0.698265024354585 & 0.206522632400617 & 0.131300520276274 & 0.131300520276274 & 0.229141351401419 & 0 \\ \hline & 0.224992896536234 & 0.117097251841844 & 0.117097251841844 & 0.204354274270769 & 0.336458325509300 \end{array} \tag{A.3}$$

and the Shu–Osher form is

$$\Lambda = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ \lambda_{41} & 0 & \lambda_{43} & 0 & 0 & 0 \\ \lambda_{51} & \lambda_{52} & 0 & \lambda_{54} & 0 & 0 \\ 0 & \lambda_{62} & 0 & 0 & \lambda_{65} & 0 \end{pmatrix}, \qquad \Gamma = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ \gamma_{21} & 0 & 0 & 0 & 0 & 0 \\ 0 & \gamma_{32} & 0 & 0 & 0 & 0 \\ 0 & 0 & \gamma_{43} & 0 & 0 & 0 \\ 0 & 0 & 0 & \gamma_{54} & 0 & 0 \\ 0 & 0 & 0 & 0 & \gamma_{65} & 0 \end{pmatrix}.$$

$\lambda_{41} = 0.426988976571684$, $\lambda_{43} = 0.5730110234283154$,
$\lambda_{51} = 0.193245318771018$, $\lambda_{52} = 0.199385926238509$, $\lambda_{54} = 0.607368754990473$,
$\lambda_{62} = 0.108173740702208$, $\lambda_{65} = 0.891826259297792$;
$\gamma_{21} = 0.377268915331368$, $\gamma_{32} = 0.377268915331368$, $\gamma_{43} = 0.216179247281718$, $\gamma_{54} = 0.229141351401419$,
$\gamma_{65} = 0.336458325509300$.

Scheme SSP53_$2N_3^*$

This is the 5-stage third order $2N^*$ method of the form (26) that solves the optimization problem (30). For this method the leading local error is $\|C_{err}\|_2 = 0.02540727$. The SSP coefficient is 1.822952 and the observed SSP coefficient is $R_{obs}(\mathbb{A}) = 2.292$.

$$\begin{array}{c|cccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0.266541020678955 & 0.266541020678955 & 0 & 0 & 0 & 0 \\ 0.815101729727488 & 0.266541020678955 & 0.548560709048532 & 0 & 0 & 0 \\ 1.104618743881888 & 0.266541020678955 & 0.548560709048532 & 0.289517014154401 & 0 & 0 \\ 0.537056421518187 & 0.108739964320909 & 0.223794715642056 & 0.118113413497299 & 0.086408328057923 & 0 \\ \hline & 0.108739964320909 & 0.223794715642056 & 0.118113413497299 & 0.086408328057923 & 0.462943578481813 \end{array} \tag{A.4}$$

Please cite this article as: I. Higueras and T. Roldán, Efficient SSP low-storage Runge–Kutta methods, Journal of Computational and Applied Mathematics (2019) 112500, https://doi.org/10.1016/j.cam.2019.112500.

and the Shu–Osher form is

$$\Lambda = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \lambda_{51} & 0 & 0 & 1-\lambda_{51} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \quad \Gamma = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ \gamma_{21} & 0 & 0 & 0 & 0 & 0 \\ 0 & \gamma_{32} & 0 & 0 & 0 & 0 \\ 0 & 0 & \gamma_{43} & 0 & 0 & 0 \\ 0 & 0 & 0 & \gamma_{54} & 0 & 0 \\ 0 & 0 & 0 & 0 & \gamma_{65} & 0 \end{pmatrix}.$$

$\lambda_{51} = 0.592032910942121\,;$
$\gamma_{21} = 0.266541020678955\,,\ \gamma_{32} = 0.548560709048532\,,\ \gamma_{43} = 0.289517014154401\,,\ \gamma_{54} = 0.086408328057923\,,$
$\gamma_{65} = 0.462943578481813\,.$

Scheme SSP53_2N_4^*

This is the 5-stage third order $2N^*$ method of the form (26) that solves the optimization problem (31). For this method the leading local error is $\|C_{err}\|_2 = 0.01545843$. The SSP coefficient is 1.425159 and the observed SSP coefficient is $R_{obs}(\mathbb{A}) = 2.184$.

$$\begin{array}{c|cccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0.292845746913355 & 0.292845746913355 & 0 & 0 & 0 & 0 \\ 0.632378540889763 & 0.292845746913355 & 0.339532793976408 & 0 & 0 & 0 \\ 0.385276307296290 & 0.085552377928378 & 0.099191599043240 & 0.200532330324672 & 0 & 0 \\ 1.086952476303169 & 0.085552377928378 & 0.099191599043240 & 0.200532330324672 & 0.701676169006879 & 0 \\ \hline & 0.066486721228291 & 0.077086392610822 & 0.155842975571268 & 0.545305098127742 & 0.155278812461877 \end{array} \tag{A.5}$$

and the Shu–Osher form is

$$\Lambda = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ \lambda_{41} & 0 & 1-\lambda_{41} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \lambda_{61} & 0 & 0 & 0 & 1-\lambda_{61} & 0 \end{pmatrix}, \quad \Gamma = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ \gamma_{21} & 0 & 0 & 0 & 0 & 0 \\ 0 & \gamma_{32} & 0 & 0 & 0 & 0 \\ 0 & 0 & \gamma_{43} & 0 & 0 & 0 \\ 0 & 0 & 0 & \gamma_{54} & 0 & 0 \\ 0 & 0 & 0 & 0 & \gamma_{65} & 0 \end{pmatrix}.$$

$\lambda_{41} = 0.707858560931430\,,\ \lambda_{61} = 0.222853615080669\,,$
$\gamma_{21} = 0.292845746913355\,,\ \gamma_{32} = 0.339532793976408\,,\ \gamma_{43} = 0.200532330324672\,,\ \gamma_{54} = 0.701676169006879\,,$
$\gamma_{65} = 0.155278812461877\,.$

References

[1] C.W. Shu, S. Osher, Efficient implementation of essentially nonoscillatory shock-capturing schemes, J. Comput. Phys. 77 (2) (1988) 439–471.
[2] S. Gottlieb, D.I. Ketcheson, C.W. Shu, Strong Stability Preserving Runge–Kutta and Multistep Time Discretizations, World Scientific, 2011.
[3] I. Higueras, N. Happenhofer, O. Koch, F. Kupka, Optimized strong stability preserving IMEX Runge–Kutta methods, J. Comput. Appl. Math. 272 (2014) 116–140.
[4] I. Higueras, T. Roldan, New third order low-storage SSP explicit Runge–Kutta methods, J. Sci. Comput. 79 (2019) 1882–1906.
[5] C. Bresten, S. Gottlieb, Z. Grant, D. Higgs, D.I. Ketcheson, A. Németh, Explicit strong stability preserving multistep Runge–Kutta methods, Math. Comp. 86 (304) (2017) 747–769.
[6] L. Isherwood, Z.J. Grant, S. Gottlieb, Strong stability preserving integrating factor two-step Runge–Kutta methods. arXiv preprint arXiv:1904.07194, 2019.
[7] D.I. Ketcheson, S. Gottlieb, C.B. Macdonald, Strong stability preserving two-step Runge–Kutta methods, SIAM J. Numer. Anal. 49 (6) (2011) 2618–2639.
[8] Z. Horváth, H. Podhaisky, R. Weiner, Strong stability preserving explicit peer methods, J. Comput. Appl. Math. 296 (2016) 776–788.
[9] M. Calvo, J.M. Franco, J.I. Montijano, L. Rández, On some new low-storage implementations of time advancing Runge–Kutta methods, J. Comput. Appl. Math. 236 (15) (2012) 3665–3675.
[10] S. Gottlieb, C.W. Shu, Total variation diminishing Runge–Kutta schemes, Math. Comp. 67 (221) (1998) 73–85.
[11] S. Gottlieb, C.W. Shu, E. Tadmor, Strong stability preserving high-order time discretization methods, SIAM Rev. 43 (1) (2001) 89–112.
[12] C.A. Kennedy, M.H. Carpenter, R.M. Lewis, Low-storage, explicit Runge–Kutta schemes for the compressible Navier–Stokes equations, Appl. Numer. Math. 35 (3) (2000) 177–219.
[13] D.I. Ketcheson, Highly efficient strong stability preserving Runge–Kutta methods with low-storage implementations, SIAM J. Sci. Comput. 30 (4) (2008) 2113–2136.
[14] D.I. Ketcheson, Runge–Kutta methods with minimum storage implementations, J. Comput. Phys. 229 (5) (2010) 1763–1773.
[15] S.J. Ruuth, Global optimization of explicit strong-stability-preserving Runge–Kutta methods, Math. Comp. 75 (253) (2006) 183–208.
[16] R.J. Spiteri, S.J. Ruuth, A new class of optimal high-order strong stability preserving time discretization methods, SIAM J. Numer. Anal. 40 (2) (2002) 469–491.
[17] P.J. Van Der Houwen, Construction of Integration Formulas for Initial Value Problems, North Holland, 1977.
[18] J.H. Williamson, Low-storage Runge–Kutta schemes, J. Comput. Phys. 35 (1) (1980) 48–56.
[19] L. Ferracina, M.N. Spijker, Strong stability of singly-diagonally-implicit Runge–Kutta methods, Appl. Numer. Math. 58 (11) (2008) 1675–1686.

[20] R.J. LeVeque, Finite Volume Methods for Hyperbolic Problems, Cambridge University Press, 2002.
[21] J.F.B.M. Kraaijevanger, Absolute monotonicity of polynomials occurring in the numerical solution of initial value problems, Numer. Math. 48 (3) (1986) 303–322.
[22] I. Higueras, Strong stability for additive Runge-Kutta methods, SIAM J. Numer. Anal. 44 (4) (2006) 1735–1758.
[23] M.N. Spijker, Contractivity in the numerical solution of initial value problems, Numer. Math. 42 (3) (1983) 271–290.
[24] M.N. Spijker, Stepsize restrictions for stability of one-step methods in the numerical solution of initial value problems, Math. Comp. 45 (172) (1985) 377–392.
[25] M.N. Spijker, Stepsize conditions for general monotonicity in numerical initial value problems, SIAM J. Numer. Anal. 45 (3) (2008) 1226–1245.
[26] J.F.B.M. Kraaijevanger, Contractivity of Runge–Kutta methods, BIT 31 (3) (1991) 482–528.
[27] C.W. Shu, Total-Variation-Diminishing time discretizations, SIAM J. Sci. Comput. 9 (6) (1988) 1073–1084.
[28] D.I. Ketcheson, M. Parsani, A.J. Ahmadia, Rk-opt: Software for the design of Runge–Kutta methods, version 0.2. 2013.
[29] I. Higueras, Representations of Runge–Kutta methods and strong stability preserving methods, SIAM J. Numer. Anal. 43 (3) (2005) 924–948.
[30] L. Ferracina, M.N. Spijker, Stepsize restrictions for the Total-Variation-Diminishing property in general Runge-Kutta methods, SIAM J. Numer. Anal. 42 (3) (2004) 1073–1093.
[31] S. Gottlieb, D.I. Ketcheson, C.W. Shu, High order strong stability preserving time discretizations, J. Sci. Comput. 38 (3) (2009) 251–289.
[32] D.I. Ketcheson, C.B. Macdonald, S. Gottlieb, Optimal implicit strong stability preserving Runge–Kutta methods, Appl. Numer. Math. 59 (2) (2009) 373–392.
[33] R.J. Spiteri, S.J. Ruuth, Non-linear evolution using optimal fourth-order strong-stability-preserving Runge–Kutta methods, Math. Comput. Simulation 62 (1–2) (2003) 125–135.
[34] M. Calvo, J.M. Franco, L. Rández, Minimum storage Runge–Kutta schemes for computational acoustics, Comput. Math. Appl. 45 (1) (2003) 535–545.
[35] D. Cavaglieri, T. Bewley, Low-storage implicit/explicit Runge–Kutta schemes for the simulation of stiff high-dimensional ode systems, J. Comput. Phys. 286 (2015) 172–193.
[36] E. Hairer, S.P. Nørsett, G. Wanner, Solving Ordinary Differential Equations I, Nonstiff Problems, Second Revised ed., Springer, Berlin, 1993.

Please cite this article as: I. Higueras and T. Roldán, Efficient SSP low-storage Runge–Kutta methods, Journal of Computational and Applied Mathematics (2019) 112500, https://doi.org/10.1016/j.cam.2019.112500.