Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

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

Embedded pairs for optimal explicit strong stability preserving Runge–Kutta methods

Imre Fekete [a,b,*], Sidafa Conde [c], John N. Shadid [c,d]

[a] Department of Applied Analysis and Computational Mathematics, Eötvös Loránd University, Pázmány P. s. 1/C, Budapest, H-1117, Hungary
[b] MTA-ELTE Numerical Analysis and Large Networks Research Group, Pázmány P. s. 1/C, Budapest, H-1117, Hungary
[c] Sandia National Laboratories, Albuquerque, NM 87123, United States
[d] Department of Mathematics and Statistics, University of New Mexico, MSC01 1115, Albuquerque, NM 87131, United States

ARTICLE INFO

Article history:
Received 23 April 2021
Received in revised form 30 January 2022

Keywords:
Runge–Kutta methods
Strong stability preserving methods
Step-size control
Embedded pairs
Variable step-size
Hyperbolic problems

ABSTRACT

We construct a family of embedded pairs for optimal explicit strong stability preserving Runge–Kutta methods of order $2 \leq p \leq 4$ to be used to obtain numerical solution of spatially discretized hyperbolic PDEs. In this construction, the goals include non-defective property, large stability region, and small error values as defined in Dekker and Verwer (1984) and Kennedy et al. (2000). The new family of embedded pairs offer the ability for strong stability preserving (SSP) methods to adapt by varying the step-size. Through several numerical experiments, we assess the overall effectiveness in terms of work versus precision while also taking into consideration accuracy and stability.

1. Introduction

We consider an initial value problem (IVP) of the form

$$u'(t) = f(t, u(t)), \qquad u(t_0) = u_0. \tag{1}$$

Using the method-of-lines approach, spatial discretization of the time-dependent partial differential equations (PDEs) gives rise to a large system of ordinary differential equations. The numerical solution of (1) at each time step with an s-stage explicit Runge–Kutta (ERK) method is given by

$$u_{n+1} = u_n + \Delta t \sum_{j=1}^{s} b_j f(t_n + c_j \Delta t, Y_j)$$

and the internal stages are computed as

$$Y_i = u_n + \Delta t \sum_{j=1}^{i-1} a_{ij} f(t_n + c_j \Delta t, Y_j), \qquad i = 1, \ldots, s,$$

where u_n is an approximation to the solution of (1) at time $t_n = t_0 + n\Delta t$. The coefficients of the method are $A = (a_{ij})$, $b^T = (b_j)$ and $c = (c_j)$. These are represented in the Butcher tableau (see Table 1). In some cases we will omit the vector c from the Butcher tableau since the methods have the property $c = Ae$.

* Corresponding author at: Department of Applied Analysis and Computational Mathematics, Eötvös Loránd University, Pázmány P. s. 1/C, Budapest, H-1117, Hungary.
E-mail addresses: imre.fekete@ttk.elte.hu (I. Fekete), sconde@sandia.gov (S. Conde), jnshadid@sandia.gov (J.N. Shadid).

https://doi.org/10.1016/j.cam.2022.114325

Table 1
The Butcher tableau of an s-stage explicit Runge–Kutta method.

0					
c_2	a_{21}				
c_3	a_{31}	a_{32}			
$\vdots$	$\vdots$	$\vdots$	$\ddots$		
c_s	a_{s1}	a_{s2}	$\dots$	$a_{s,s-1}$	
	b_1	b_2	$\dots$	b_{s-1}	b_s

In general, most of the time integration for the numerical solution of ODEs is computed with a single formula and a fixed step-size. This type of approach can be non-optimal if the solution varies rapidly over small subsets of the integration interval and slowly over larger ones [1]. In an attempt to minimize the computational cost and obtain the best possible result at ideal time-to-solution, it is necessary to use an adaptive method based on automatic step-size selection. A sophisticated adaptive theory has been worked out for Runge–Kutta methods using linear control theory and digital filters [2–5].

An embedded Runge–Kutta method provides the ability for time step adaptivity based on error control at minimal additional cost. As usual, integration is advanced using the higher order p approximation while the local error estimation is provided by embedded pairs. In this paper we denote the embedded pair of order $(p-1)$ by $\tilde{b}^T$. In the last decades several effective non-SSP embedded formulas have been proposed: the 3(2) pair of Bogacki and Shampine [6], and some other well-known embedded pairs such as Merson [7], Ceschino [8] and Zonneveld [9]. Coupled with robust step control strategy, it has been shown that embedded explicit Runge–Kutta technique is an efficient method for numerical solution of non-stiff initial value problems [10–12].

In this paper we are interested in designing embedded pairs for optimal explicit strong stability preserving (SSP) Runge–Kutta methods. Explicit SSP Runge–Kutta methods are extensively used in numerical computation of hyperbolic conservation laws with total variation diminishing spatial discretizations [13–16]. In many hyperbolic PDE applications, the step-size is controlled by monitoring the CFL number, often requiring an expensive evaluation of the right-hand side. Another motivation for providing error estimators for SSP methods is that several optimal SSP methods have good general properties (large stability regions, small error coefficients, etc.) and are frequently used even when SSP theory cannot be applied (when forward Euler condition does not hold), or even for non-hyperbolic PDEs. In such situations, how to control the time step-size in practice may be less obvious, while control based on error estimation will be even more useful.

Only a few attempts have been made in SSP sense to design embedded pairs for explicit Runge–Kutta methods. In [17], Macdonald constructed high-order Runge–Kutta methods with embedded SSP pairs. In [18], Ketcheson realized that the seven-stage second order method and the three-stage second order method can be used as an embedded method with nine-stage third order method and four-stage third order method for error control, respectively. These attempts treated only special cases. A general approach is missing in the literature to design embedded pairs for optimal explicit SSP Runge–Kutta methods. In this paper we fill this gap.

This paper is structured as follows: in the next subsection, we briefly review previous work on SSP methods, and present the analytical framework that enables us to construct the new family of embedded pairs. In Section 2 we construct the embedded pairs for optimal explicit SSP Runge–Kutta methods. Section 3 contains numerical experiments that compare the newly constructed embedded pairs with existing pairs on several test problems, using different step control strategies, to investigate their performance and robustness. Finally, in Section 4, we summarize our conclusions.

1.1. Explicit SSP Runge–Kutta methods

SSP time discretization methods are designed to ensure nonlinear stability properties in the numerical solution of spatially discretized hyperbolic PDEs. Typically after the spatial discretization we obtain a nonlinear system of ODEs

$$u_t = F(u), \tag{2}$$

where u is a vector of approximations to the exact solution of the PDE. We assume that the semi-discretization (2) and a convex functional $\|\cdot\|$ (or norm, semi-norm) are given, and that there exists a Δt_{FE} such that the forward Euler condition

$$\|u + \Delta t F(u)\| \le \|u\| \qquad \text{for} \;\; 0 \le \Delta t \le \Delta t_{\mathrm{FE}} \tag{3}$$

holds for all u. An ERK method is called SSP if the estimate

$$\|u_{n+1}\| \le \|u_n\|$$

holds for the numerical solution of (2), whenever (3) holds and $\Delta t \le \mathcal{C}\Delta t_{\mathrm{FE}}$. The constant $\mathcal{C}$ is called the SSP coefficient.

For a complete introduction to the topic we recommend monograph [14]. A brief summary about theoretical results can be found in review [15]. Next, we highlight two fundamental results of Kraaijevanger [19] which will be used in the paper. The notations in the following were introduced in [20].

Theorem 1.1. *Let us consider the matrix*

$$K = \begin{pmatrix} A & 0 \\ b^T & 0 \end{pmatrix} \in \mathbb{R}^{(s+1)\times(s+1)}$$

and the SSP conditions

$$K(I + rK)^{-1} \geq 0 \tag{4a}$$

$$rK(I + rK)^{-1}e \leq e. \tag{4b}$$

Then, the SSP coefficient of the ERK method is

$$\mathcal{C}(A, b^T) = \sup\left\{r : (I + rK)^{-1}\text{exists and conditions (4a)–(4b) hold}\right\}.$$

Theorem 1.2. *Consider an ERK method. If the method has positive SSP coefficient $\mathcal{C}(A, b^T)$, then $A \geq 0$ and $b^T \geq 0$.*

The inequalities in Theorems 1.1. and 1.2. are meant componentwise. Since there is no ERK method of order $p \geq 5$ with positive SSP coefficient, therefore we only give the order conditions up to order 4. These are

$$b^T e = 1, \qquad (p = 1) \tag{5a}$$

$$b^T c = \frac{1}{2}, \qquad (p = 2) \tag{5b}$$

$$b^T c^2 = \frac{1}{3}, \qquad (p = 3) \tag{5c}$$

$$b^T\left(\frac{c^2}{2!} - Ac\right) = 0, \qquad (p = 3) \tag{5d}$$

$$b^T c^3 = \frac{1}{4}, \qquad (p = 4) \tag{5e}$$

$$b^T A\left(\frac{c^2}{2!} - Ac\right) = 0, \qquad (p = 4) \tag{5f}$$

$$b^T\left(\frac{c^3}{3!} - \frac{Ac^2}{2!}\right) = 0, \qquad (p = 4) \tag{5g}$$

$$b^T \mathrm{diag}(c)\left(\frac{c^2}{2!} - Ac\right) = 0, \qquad (p = 4) \tag{5h}$$

where diag(c) is the square diagonal matrix with the elements of vector c on the main diagonal.

In the sequel we will consider only optimal explicit SSP Runge–Kutta methods, i.e. these methods have the largest possible SSP coefficient $\mathcal{C}$.

2. Embedded pairs for optimal explicit SSP Runge–Kutta methods

We introduce the notation SSPERK(s, p) for optimal explicit SSP Runge–Kutta methods, where s and p are the number of stages and order, respectively. Below we list the desired properties for embedded methods.

(i), The embedded method is of order $p - 1$.
(ii), The embedded method is non-defective, i.e. it violates all of the conditions of order p.
(iii), Whenever possible, the embedded method has rational coefficients and a simple structure.
(iv), The embedded method has maximum SSP coefficient $\tilde{\mathcal{C}}$ where $\tilde{\mathcal{C}}$ is the SSP coefficient of the optimal SSPERK method. If this is not the case, then we are looking for embedded SSPERK methods with smaller SSP coefficients or simply embedded ERK methods.

Taking into account the desired properties (i)–(iv) and the coefficient matrix A of the optimal SSPERK method, we look for embedded pair $\tilde{b}^T$ such that the following constraints are satisfied:

$$\begin{pmatrix} A & 0 \\ \tilde{b}^T & 0 \end{pmatrix}\left(I + \tilde{\mathcal{C}}\begin{pmatrix} A & 0 \\ \tilde{b}^T & 0 \end{pmatrix}\right)^{-1} \geq 0, \tag{6}$$

$$\left\|\tilde{\mathcal{C}}\begin{pmatrix} A & 0 \\ \tilde{b}^T & 0 \end{pmatrix}\left(I + \tilde{\mathcal{C}}\begin{pmatrix} A & 0 \\ \tilde{b}^T & 0 \end{pmatrix}\right)^{-1}\right\|_\infty \leq 1, \tag{7}$$

the appropriate order conditions (5a)–(5h) and property (ii) are fulfilled. (8)

Inequalities (6)–(7) are equivalent with (4a)–(4b) and $\|\cdot\|_\infty$ denotes the induced matrix norm. Due to Theorem 1.2 and (5a) we have the componentwise inequality $0 \leq \tilde{b}^T \leq e$. The embedded methods should have nice stability related quantities. These will be discussed in detail below.

The stability region is given by

$$S = \{z \in \mathbb{C} : |\psi(z)| \leq 1\},$$

where $\psi(z)$ is the stability function of the given SSPERK method. The following definitions can be found in [19,21,22].

Definition 1. The absolute stability real axis inclusion is the radius of the largest interval on the real axis that is contained in the absolute stability region. Specifically,

$$\delta_R = \sup\{\gamma : \gamma \geq \ 0 \ \textit{and} \ l(-\gamma, \gamma) \subset S\},$$

where $l(z_1, z_2)$ is the line segment connecting $z_1, z_2 \in \mathbb{C}$.

Definition 2. The absolute stability imaginary axis inclusion is the radius of the largest interval on the imaginary axis that is contained in the absolute stability region. Specifically,

$$\delta_I = \sup\{\gamma : \gamma \geq \ 0 \ \textit{and} \ l(-i\gamma, i\gamma) \subset S\},$$

where $l(z_1, z_2)$ is the line segment connecting $z_1, z_2 \in \mathbb{C}$.

Definition 3. An ERK method is called circle contractive if for $r > 0$

$$|\psi(z)| \leq 1 \text{ for all } z \in D(r). \tag{9}$$

The radius of circle contractivity is the radius of the largest generalized disk $D(r)$ for which (9) holds, where

$$D(r) = \{\lambda \in \mathbb{C} : |\lambda + r| \leq r\}.$$

The radius of circle contractivity will be denoted by δ_C.

Definition 4. The radius of absolute monotonicity of the stability function $\psi(z)$ is the largest value of r such that $\psi(z)$ and all of its derivatives exist and are non-negative for $z \in (-r, 0]$. It will be denoted by $R(\psi)$.

Throughout the paper values of Definitions 1–4 are simply called stability radius values. The values δ_R and δ_I are relevant for absolute region plots. The values δ_C and $R(\psi)$ give important information for different stability properties. Next to the constraints (6)–(8) we impose the remaining desired properties (i), (ii) and (iv). We use MATLAB's built-in optimization package to get the set of suggested list of embedded pairs. Then we will choose the method with the highest sum of axis inclusion values. Thus, our stability related optimization problem is

$$\begin{array}{ll} \max & \delta_R + \delta_I \\ \text{subject to} & (6)\text{–}(8), \\ & \text{desired properties } (i),\ (ii) \text{ and } (iv). \end{array} \tag{10}$$

The other optimization problem will be related to error control metric values. Following [23], we quantify the L_2 and L_∞ principal errors of the pairs as

$$A_2^{(p+1)} = \|\tau^{(p+1)}\|_2 \quad \text{and} \quad \tilde{A}_2^{(p)} = \|\tilde{\tau}^{(p)}\|_2,$$
$$A_\infty^{(p+1)} = \|\tau^{(p+1)}\|_\infty \quad \text{and} \quad \tilde{A}_\infty^{(p)} = \|\tilde{\tau}^{(p)}\|_\infty,$$

where $\tau^{(p+1)}$ is the error coefficient vector of SSPERK method of order p. The vector $\tilde{\tau}^{(p)}$ corresponds to the error coefficient vector for the embedded pair of order $\tilde{p} = p - 1$. Furthermore, additional error controls from paper [23] are defined as

$$B_2^{(p+1)} = \frac{A_2^{(p+1)}}{A_2^{(p)}} \qquad \tilde{B}_2^{(p)} = \frac{\tilde{A}_2^{(p)}}{\tilde{A}_2^{(p-1)}},$$
$$B_\infty^{(p+1)} = \frac{A_\infty^{(p+1)}}{A_\infty^{(p)}} \qquad \tilde{B}_\infty^{(p)} = \frac{\tilde{A}_\infty^{(p)}}{\tilde{A}_\infty^{(p-1)}},$$
$$C_2^{(p+1)} = \frac{\|\tilde{\tau}^{(p+1)} - \tau^{(p+1)}\|_2}{\|\tilde{\tau}^{(p)}\|_2} \qquad C_\infty^{(p+1)} = \frac{\|\tilde{\tau}^{(p+1)} - \tau^{(p+1)}\|_\infty}{\|\tilde{\tau}^{(p)}\|_\infty},$$
$$D = \max\{|a_{ij}|, |c_i|, |b_i^T|, |\tilde{b}_i^T|\},$$

and impose that the values $\tilde{A}_2^{(p)}$ and $\tilde{A}_\infty^{(p+1)}$ should be as small as possible, while the values $B_2^{(p+1)}$, $B_\infty^{(p+1)}$, $C_2^{(p+1)}$ and $C_\infty^{(p+1)}$ should be close to one, and a small magnitude for D. This is equivalent to Dormand and Prince's constraints on $|c_i| \leq 1$ and avoiding large values of b_i^T and a_{ij} to circumvent considerable rounding errors in practical applications [24]. Throughout the paper these values will be simply called error metric values and they will support the choice of embedded pairs after the simplified optimization problem. We note that

- the quantities $A_2^{(p+1)}$ and $A_\infty^{(p+1)}$ are fixed and beyond our control since the original method is an optimal SSP method of order p,
- for a defective embedded pair, $\tilde{\tau}_i^p = 0$, since it satisfies all of the order $p = \tilde{p} + 1$ algebraic order conditions. Consequently $C_2^{(p+1)}, C_\infty^{(p+1)} \to \infty$. Since the higher-order method is the optimal SSP method, we know it is a non-defective method of order p. Otherwise, it would be a non-defective method of order $p + 1$.

Many of the optimal SSP methods, such as SSPERK(5,4) [19,25], optimal implicit SSP RK [14], and the newly developed optimized SSP IMEX methods in [26], were obtained numerically following the methodology in [27]. Similarly, in constructing an efficient and robust embedded pair for these numerically optimal methods, we construct an analogous optimization problem using certain error control metrics:

$$F(A, b^T, \tilde{w}^T) = \left[\tilde{A}_2^{(p)} \quad \tilde{A}_\infty^{(p)} \quad \left(B_2^{(p+1)} - 1\right) \quad \left(B_\infty^{(p+1)} - 1\right) \quad \left(C_2^{(p+1)} - 1\right) \quad \left(C_\infty^{(p+1)} - 1\right)\right] \tag{11}$$

$$\begin{aligned} &\underset{\tilde{w}^T}{\arg\min} && \|F(A, b^T, \tilde{w}^T)\|_\infty \\ &\text{subject to} && \tau_k(A, \tilde{w}^T) = 0, \ \ k = 1, \dots, \tilde{p} \\ & && \tilde{w}^T \geq 0, \end{aligned}$$

where $\tau_k(A, \tilde{w}^T)$ is the necessary algebraic order conditions for the embedded method of order $\tilde{p}$. Here, we have used $\tilde{w}^T$ to represent the embedded pair obtained numerically. Using MATLAB's *fmincon*, we search for $\tilde{w}$ that minimizes the cost function best with respect to stability. Due to Theorem 1.2 and (5a), the positivity constraint on the embedded weights is the componentwise inequality $0 \leq \tilde{w}^T \leq e$. Moreover, we find the above constraints always enforce $\|D\| \leq 1$. For the optimization problem here we do not impose the method to have rational coefficients since many of the numerically optimized SSP methods do not satisfy this restriction.

2.1. Embedded pairs for SSPERK(s,2) methods

The simplest method of this family, the SSPERK(2, 2) method was introduced in [13]. In general, optimal SSPERK(s, 2) methods have SSP coefficient $\mathcal{C} = s - 1$ [19,25]. Their Butcher tableau is given in Table 2.

Table 2
The Butcher tableau of SSPERK(s, 2) methods.

0					
$\frac{1}{s-1}$	$\frac{1}{s-1}$				
$\frac{2}{s-1}$	$\frac{1}{s-1}$	$\frac{1}{s-1}$			
$\vdots$	$\vdots$	$\vdots$	$\ddots$		
1	$\frac{1}{s-1}$	$\frac{1}{s-1}$	$\dots$	$\frac{1}{s-1}$	
	$\frac{1}{s}$	$\frac{1}{s}$	$\dots$	$\frac{1}{s}$	$\frac{1}{s}$

Optimization (10) led to embedded pair

$$\tilde{b}^T = \left[\frac{s+1}{s^2}, \frac{1}{s}, \dots, \frac{1}{s}, \frac{s-1}{s^2}\right] \in \mathbb{R}^s.$$

As we can see in Fig. 1 the embedded method has noticeably larger stability region than the original method until $s \leq 6$. However, for larger stage number s they have almost coincident stability regions.

Furthermore, compared to the optimal methods in Appendix A. Table 6, embedded pair $\tilde{b}^T$ has the same value of radius of circle contractivity values δ_C and radius of absolute monotonicity of stability function values $R(\psi)$ up to four digits for stage number $s \geq 3$. In addition, for stage number $s = 2$ the embedded method has larger radius of circle contractivity value δ_C than the optimal explicit method. The error metric values in Appendix A. Table 7 also strengthen the quality of embedded pair $\tilde{b}^T$.

For stage numbers $s = 2$ and $s = 3$ optimization (11) led to embedded pairs

$$\tilde{w}^T = [0.694021459207626, \ 0.305978540792374]$$

and

$$\tilde{w}^T = [0.635564950337195, \ 0.033488381714827, \ 0.330946667947978].$$

Their stability regions can be seen in Fig. 2.

The two-stage embedded method has larger stability region than the original method. However, its real axis inclusion value δ_R is significantly smaller than the value for the embedded method with pair $\tilde{b}^T$. The embedded method has nice

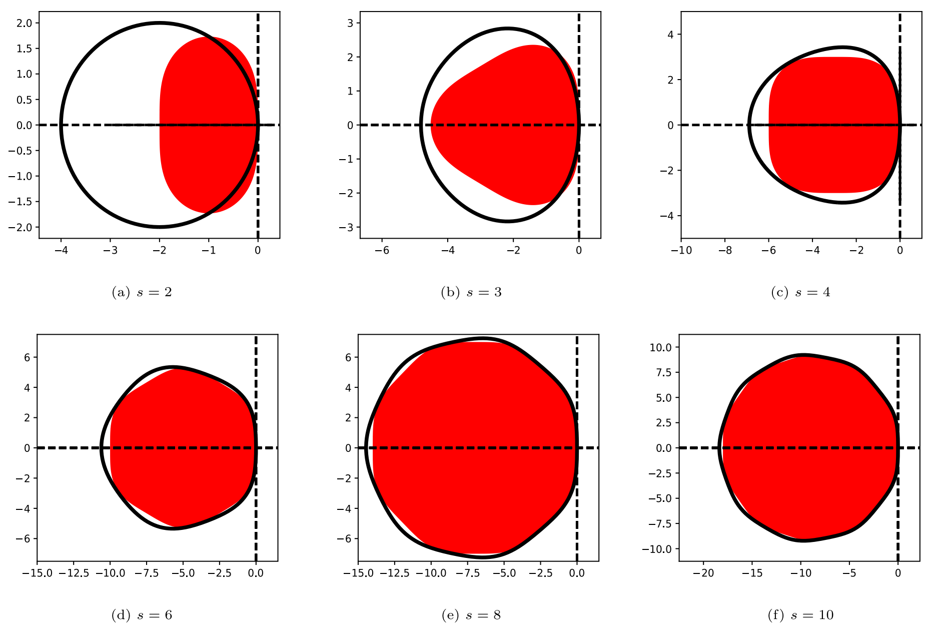


Fig. 1. The stability regions of SSPERK(s, 2) method (red) and the embedded method (black contours) for different stage numbers s.

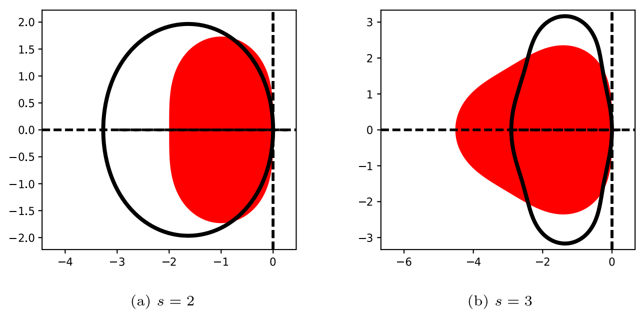


Fig. 2. The stability regions of SSPERK(s, 2) method (red) and the corresponding embedded method (black contour) for stage numbers $s = 2$ and $s = 3$.

radius of circle contractivity, radius of absolute monotonicity of stability function and error metric values. These can be found in Appendix A. Tables 6 and 7.

For stage number $s = 3$, the embedded method has moderate stability region. Furthermore, based on Appendix A. Table 6, it has small stability radius values, too. However, as we can see in Appendix A. Table 7 it has nice error metric values. Compared with the pair obtained by optimization (10) this pair has significantly smaller stability radius values and slightly better error metric values.

2.2. Embedded pairs for SSPERK(s,3) methods

Kraaijevanger treated the case SSPERK(4, 3) in [19]. Later the SSPERK(s, 3) methods were characterized, where $s = n^2$ and $n \geq 2$ is an integer [27]. These methods have SSP coefficient $n^2 - n = s - \sqrt{s}$. Their Butcher form is given by (12)–(13)

where the submatrix in the rectangle is a $\left(\frac{n(n-1)}{2}\right) \times (2n-1)$ dimensional matrix.

$$A = \left(\begin{array}{cccc|ccc|cccc}
0 & & & & & & & & & & \\
\frac{1}{n(n-1)} & \ddots & & & & & & & & & \\
\frac{1}{n(n-1)} & \ddots & & & & & & & & & \\
\frac{1}{n(n-1)} & & \ddots & & & & & & & & \\
\vdots & & & \ddots & & & & & & & \\
\vdots & & & & \ddots & & & & & & \\
\frac{1}{n(n-1)} & \cdots & \frac{1}{n(n-1)} & \cdots & \cdots & & \frac{1}{n(n-1)} & & & & \\
\cline{5-7}
\frac{1}{n(n-1)} & & \frac{1}{n(n-1)} & & \frac{1}{n(2n-1)} & \cdots & \frac{1}{n(2n-1)} & \frac{1}{n(n-1)} & & & \\
\vdots & & \vdots & & \vdots & & \vdots & \vdots & \ddots & & \\
\frac{1}{n(n-1)} & \cdots & \frac{1}{n(n-1)} & & \frac{1}{n(2n-1)} & \cdots & \frac{1}{n(2n-1)} & \frac{1}{n(n-1)} & \cdots & \frac{1}{n(n-1)} & 0 \\
\cline{5-7}
\multicolumn{3}{c}{\underbrace{\hphantom{xxxxxxxxxxxx}}_{\frac{(n-2)(n-1)}{2}}} & \multicolumn{4}{c}{} & \multicolumn{3}{c}{\underbrace{\hphantom{xxxxxxxxxxxx}}_{\frac{n(n-1)}{2}-1}} &
\end{array}\right) \in \mathbb{R}^{n^2 \times n^2}, \tag{12}$$

$$b^T = \left[\underbrace{\frac{1}{n(n-1)}, \dots, \frac{1}{n(n-1)}}_{\frac{(n-1)(n-2)}{2}}, \underbrace{\frac{1}{n(2n-1)}, \dots, \frac{1}{n(2n-1)}}_{2n-1}, \underbrace{\frac{1}{n(n-1)}, \dots, \frac{1}{n(n-1)}}_{\frac{n(n-1)}{2}}\right] \in \mathbb{R}^{n^2}. \tag{13}$$

The case SSPERK(4, 3)

A possible embedded pair for the SSPERK(4, 3) method was suggested in [18]. The pair from the literature will be denoted by $\tilde{b}^T_{1\text{it.}}$. It is

$$\tilde{b}^T_{1\text{it.}} = \left[\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 0\right].$$

The construction of this pair is related to the SSPERK(3, 2) method. Taking into account the Butcher forms of SSPERK(s, 2) and SSPERK(s, 3) methods, one can realize that this kind of embedded pair can be only achieved in this exceptional case.

In general, optimization (10) suggests the pair

$$\tilde{b}^T = \left[\frac{1}{n^2}, \dots, \frac{1}{n^2}\right] \in \mathbb{R}^{n^2}.$$

For the special case $n^2 = 4$ we have

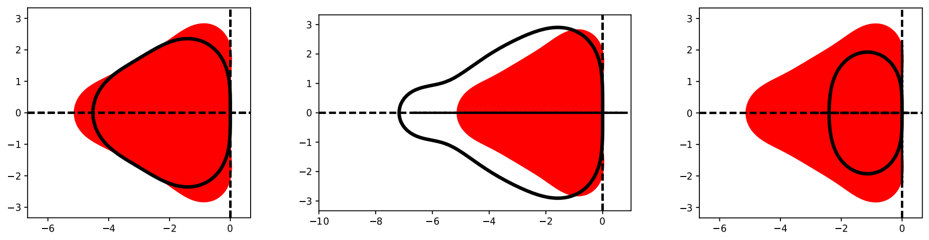
$$\tilde{b}^T = \left[\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right].$$

Optimization (11) suggests the pair

$$\tilde{w}^T = [0.138870252716866,\ 0.722259494566267,\ 0.138870252716866, 0].$$

The stability regions of the three embedded methods are in Fig. 3. As we can see the stability region of the embedded method obtained by optimization (10) is significantly larger than the stability regions of the other embedded methods and the original method.

As we can see in Appendix A. Table 8, the embedded method obtained by optimization (11) has significantly smaller radius of circle contractivity values δ_C and radius of absolute monotonicity of stability function values $R(\psi)$ than the original method. However, the literature embedded method and the embedded method obtained by optimization (10) have the same δ_C and $R(\psi)$ values as the original method. Furthermore, based on Table 9 in Appendix A., the embedded method obtained by optimization (10) has the best error metric values. Thus, the pair $\tilde{b}^T$ is recommended for the SSPERK(4, 3) method.



(a) $\tilde{b}^T_{\text{lit.}}$ (b) $\tilde{b}^T$ (c) $\tilde{w}^T$

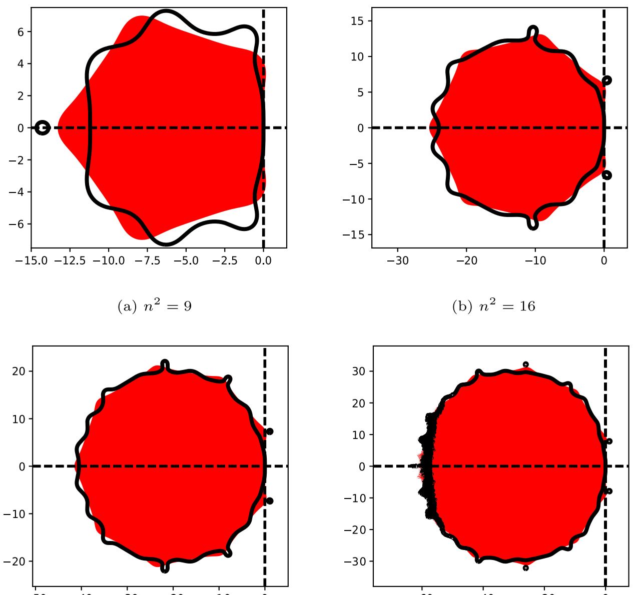
Fig. 3. The stability regions of SSPERK(4, 3) method (red) and the embedded methods (black contour).

The case SSPERK(n^2*, 3)*

Now, we turn our attention to analyze the stability radius and error metric values of SSPERK(n^2, 3) method and its embedded method, where $n \geq 3$ is an integer. As we have mentioned optimization (10) suggests the pair

$$\tilde{b}^T = \left[\frac{1}{n^2}, \ldots, \frac{1}{n^2}\right] \in \mathbb{R}^{n^2}.$$

Based on Fig. 4 the original and the embedded method have similar stability regions.



(a) $n^2 = 9$ (b) $n^2 = 16$ (c) $n^2 = 25$ (d) $n^2 = 36$

Fig. 4. The stability regions of SSPERK(n^2, 3) methods (red) and the embedded method (black contour).

As we have seen, the suggested embedded pair $\tilde{b}^T$ has big stability radius and nice error metric values for $n^2 = 4$. For $n^2 = 9$ it has acceptable stability radius values. However, for higher stage numbers, expect the nice absolute stability regions, the embedded methods have fairly low stability radius values and acceptable error metric values. The exact values can be found in Appendix A. Tables 10 and 11.

*The case SSPERK(*3, 3*)*

One of the most popular SSP methods in the literature is the SSPERK(3, 3) or the so-called Shu–Osher method [28]. Its Butcher tableau is given in Table 3.

Table 3
The Butcher tableau of the SSPERK(3, 3) method.

$$\begin{array}{c|ccc} 0 & & & \\ 1 & 1 & & \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4} & \\ \hline & \frac{1}{6} & \frac{1}{6} & \frac{2}{3} \end{array}$$

Using optimization (11) we obtained the embedded pair

$$\tilde{w}^T = [0.291485418878409,\ 0.291485418878409,\ 0.417029162243181].$$

As we can see in Fig. 5 the embedded method has larger stability region than the original method. Based on Tables 12 and 13 in Appendix A., the original and embedded methods have the same δ_C, $R(\psi)$ and error metric values.

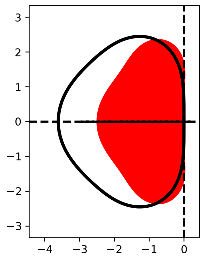


Fig. 5. The stability regions of SSPERK(3, 3) method (red) and the embedded method (black contour).

2.3. Embedded pairs for fourth order methods

The optimal SSPERK(10, 4) method has simple rational coefficients. It is quite popular in the literature and applications since it allows for a low-storage implementation [27]. It has $\mathcal{C} = 6$. Its Butcher matrix is

$$A = \begin{pmatrix} 0 & & & & & & & & & \\ \frac{1}{6} & \ddots & & & & & & & & \\ \frac{1}{6} & \frac{1}{6} & \ddots & & & & & & & \\ \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \ddots & & & & & & \\ \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \ddots & & & & & \\ \frac{1}{15} & \frac{1}{15} & \frac{1}{15} & \frac{1}{15} & \frac{1}{15} & \ddots & & & & \\ \frac{1}{15} & \frac{1}{15} & \frac{1}{15} & \frac{1}{15} & \frac{1}{15} & \frac{1}{6} & \ddots & & & \\ \frac{1}{15} & \frac{1}{15} & \frac{1}{15} & \frac{1}{15} & \frac{1}{15} & \frac{1}{6} & \frac{1}{6} & \ddots & & \\ \frac{1}{15} & \frac{1}{15} & \frac{1}{15} & \frac{1}{15} & \frac{1}{15} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \ddots & \\ \frac{1}{15} & \frac{1}{15} & \frac{1}{15} & \frac{1}{15} & \frac{1}{15} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & 0 \end{pmatrix} \in \mathbb{R}^{10\times 10}$$

and its Butcher array is

$$b^T = \left[\frac{1}{10}, \frac{1}{10}, \frac{1}{10}, \frac{1}{10}, \frac{1}{10}, \frac{1}{10}, \frac{1}{10}, \frac{1}{10}, \frac{1}{10}, \frac{1}{10}\right] \in \mathbb{R}^{10}.$$

We cannot expect non-defective embedded pairs for this method since condition (5g) is always satisfied. Thus, we are looking for embedded pairs which violate the other three fourth-order conditions. During our experimental investigation we have found eight potential embedded pairs which have nice structures in terms of coefficients. These are listed below.

$$\tilde{b}_1^T = \left[0, \frac{3}{8}, 0, \frac{1}{8}, 0, 0, 0, \frac{3}{8}, 0, \frac{1}{8}\right] \qquad \tilde{b}_2^T = \left[\frac{3}{14}, 0, 0, \frac{2}{7}, 0, 0, 0, \frac{3}{7}, 0, \frac{1}{14}\right]$$

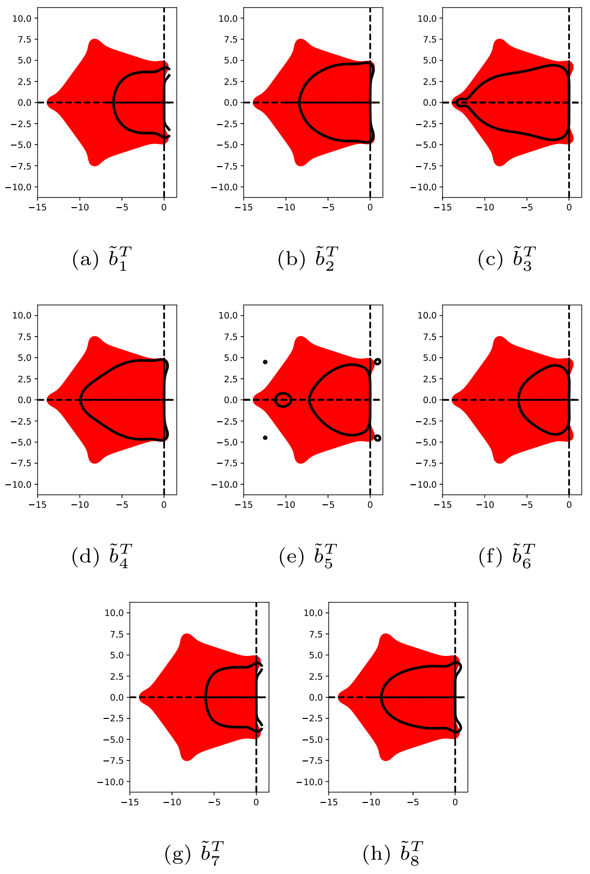
$$\tilde{b}_3^T = \left[0, \frac{2}{9}, 0, 0, \frac{5}{18}, \frac{1}{3}, 0, 0, 0, \frac{1}{6}\right] \qquad \tilde{b}_4^T = \left[\frac{1}{5}, 0, 0, \frac{3}{10}, 0, 0, \frac{1}{5}, 0, \frac{3}{10}, 0\right]$$

$$\tilde{b}_5^T = \left[\frac{1}{10}, 0, 0, \frac{2}{5}, 0, \frac{3}{10}, 0, 0, 0, \frac{1}{5}\right] \qquad \tilde{b}_6^T = \left[\frac{1}{6}, 0, 0, 0, \frac{1}{3}, \frac{5}{18}, 0, 0, \frac{2}{9}, 0\right]$$

$$\tilde{b}_7^T = \left[0, \frac{2}{5}, 0, \frac{1}{10}, 0, 0, 0, \frac{1}{5}, \frac{3}{10}, 0\right] \qquad \tilde{b}_8^T = \left[\frac{1}{7}, 0, \frac{5}{14}, 0, 0, 0, 0, \frac{3}{14}, \frac{2}{7}, 0\right]$$

As we can see in Fig. 6, the embedded methods have moderate stability regions. The embedded methods have zero δ_C and $R(\psi)$ values (see in Appendix A. Table 14). We suggest pair $\tilde{b}_4^T$ since this embedded method has the highest sum of δ_R and δ_I. The exact absolute stability real and imaginary inclusion values can be found in Appendix A. Table 15.

In an effort to showcase the effectiveness of the SSPERK(10, 4) pairs above, we explored a moderate stage number optimized fourth order pair. Using optimization (11) we obtained the embedded pair for SSPERK(6, 4) method [29]. The Butcher tableau of SSPERK(6, 4) method and its embedded pair are given in Table 4.



(a) $\tilde{b}_1^T$ (b) $\tilde{b}_2^T$ (c) $\tilde{b}_3^T$

(d) $\tilde{b}_4^T$ (e) $\tilde{b}_5^T$ (f) $\tilde{b}_6^T$

(g) $\tilde{b}_7^T$ (h) $\tilde{b}_8^T$

Fig. 6. The stability regions of the SSPERK(10, 4) method (red) and the embedded methods (black contour).

Table 4
The Butcher tableau of the embedded SSPERK(6, 4) pair.

0						
0.3552975516919	0.3552975516919					
0.6022749207532	0.2704882223931	0.3317866983600				
0.4697506438335	0.1223997401356	0.1501381660925	0.1972127376054			
0.5648149343849	0.0763425067155	0.0936433683640	0.1230044665810	0.2718245927242		
1.0006305886426	0.0763425067155	0.0936433683640	0.1230044665810	0.2718245927242	0.4358156542577	
	0.1522491819555	0.1867521364225	0.1555370561501	0.1348455085546	0.2161974490441	0.1544186678729
	0.1210663237182	0.2308844004550	0.0853424972752	0.3450614904457	0.0305351538213	0.1871101342844

As we can see in Fig. 7, the embedded method has quite large stability region. However, based on Table 16 in Appendix A., the embedded method has moderate stability radius values compared to the original method's stability radius values.

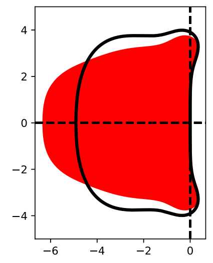


Fig. 7. The stability regions of SSPERK(6, 4) method (red) and the embedded method (black contour).

3. Numerical results

First we briefly summarize the technical details about step-size control strategies. Then we apply our methods to two classical PDEs and compare them with well-known embedded pairs from the literature.

3.1. Step-size control strategies

The starting step-size is computed following the algorithm in [10, p. 169.]. We take the minimum of the starting step-size returned from the algorithm and the step-size restriction based on the CFL constraints to avoid any stability issues from influencing the numerical study.

Following the classical approach from [10] the optimal step-size is computed in the multiplicative form

$$\Delta t_{opt} = \Delta t \cdot \beta_{n+1},$$

where β_{n+1} is determined by the choice of error control algorithm. As we pointed out in the introduction there exists a variety of error control algorithms [3–5,11]. In our numerical experiment we used the following four error control strategies with the usual estimated local truncation error err:

(1) The standard time adaptivity I controller provides a prospective time step estimate entirely based on the current local error estimate

$$\beta_{n+1} = err_{n+1}^{-k_1/p}.$$

By default, we take $k_1 = 1$.

(2) The PI controller uses the two most recent local truncation errors in the form

$$\beta_{n+1} = err_{n+1}^{-k_1/p} err_{n}^{k_2/p}.$$

Here, the default values are $k_1 = 0.8$ and $k_2 = 0.31$.

(3) The PID controller uses the information from the three most recent time steps to provide an optimal step-size:

$$\beta_{n+1} = err_{n+1}^{-k_1/p} err_{n}^{k_2/p} err_{n-1}^{-k_3/p}.$$

The default values are $k_1 = 0.58$, $k_2 = 0.21$ and $k_3 = 0.1$.

(4) The explicit Gustafsson controller [2] has the form

$$\beta_{n+1} = \begin{cases} err_1^{-1/p} & \text{on the first step,} \\ err_{n+1}^{-k_1/p}\,(err_{n+1}/err_n)^{k_2/p} & \text{on the subsequent steps.} \end{cases}$$

The parameters are $k_1 = 0.367$ and $k_2 = 0.268$. The default values chosen are similarly used by SUNDIALS/CVODE [30]. In this estimate, a floor of $err_n > 1e-10$ is enforced to avoid division by zero errors.

However, often the optimal step-size Δt_{opt} is scaled conservatively by safety factors fac, $facmin$ and $facmax$ in the form

$$\Delta t_{opt} = \Delta t \cdot \min(facmax, \max(facmin, fac \cdot \beta_{n+1})).$$

We chose a maximal step-size increase factor of $facmax = 5$ to limit large increases in the step-size. In computing the approximate solution immediately following a step-rejection, we set $facmax = fac = 0.9$ to prevent an infinite loop that we observed on rare occasions — similar strategies are employed by [6,10,12].

If $err_{n+1} \le 1$, then the computed solution u_{n+1} is accepted and advanced. The next time step is computed with Δt_{opt} as step-size. If $err_{n+1} > 1$, then step is rejected and the computations are repeated with the new step-size Δt_{opt}.

3.2. Numerical results for the PDE test problems

3.2.1. The problems

We consider two examples of the one-dimensional hyperbolic conservation law

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

spatially discretized by WENO5 [28,31].

The first problem is the scalar linear advection equation, i.e. $f(u) = u$ with periodic boundary conditions and a square wave initial condition on $x \in [-1, 1]$. The integration interval is $[0, 0.2]$.

The second problem is the one-dimensional Euler equations

$$\frac{\partial \mathbf{q}}{\partial t} + \nabla \cdot \mathbf{F} = 0,$$

where the conserved variables are given as

$$\mathbf{q} = \begin{bmatrix} \rho \\ \rho u \\ E \end{bmatrix} \text{ and } \mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ (E+p)u \end{bmatrix}.$$

The equations are closed by the ideal gas law as

$$p = (\gamma - 1)\left(E - \frac{1}{2}\rho u^2\right) \text{ and } c_s = \sqrt{\frac{\gamma p}{\rho}},$$

where c_s represents the speed of sound and γ is a fluid dependent constant. We take $\gamma = 7/5$ for air in typical atmospheric conditions. We consider the Sod shock tube [32–34] with initial conditions

$$\rho(x,0) = \begin{cases} 1.0 & x < 0.5, \\ 0.125 & x \geq 0.5, \end{cases} \qquad \rho u(x,0) = 0, \qquad E(x,0) = \frac{1}{\gamma - 1}\begin{cases} 1.0 & x < 0.5, \\ 0.1 & x \geq 0.5. \end{cases}$$

These also serve as boundary conditions since any disturbance is assumed not to reach the boundaries of the computational domain taken as $x \in [0, 1]$. The integration interval is $[0, 0.2]$.

3.2.2. The numerical results

In this section we present some experiments to test the numerical efficiency of the new embedded pairs constructed in this paper and compare them with well-known embedded pairs from the literature. The embedded pairs from the literature are listed in Table 5. The corresponding stability radius values and stability regions can be found in Appendix B. Table 17 and Fig. 9, respectively. As we can see in Appendix B. Table 17 almost all of the methods have zero radius of circle contractivity values δ_C and radius of absolute monotonicity values $R(\psi)$, therefore in general these methods cannot be efficient enough.

We note that the name of Merson45 method could be misleading in Table 5. The embedded method is fifth order for linear problems with constant coefficients and it is only third order for nonlinear problems.

Table 5
Well-known methods from the literature.

Method	Order	Embedded order
RKF23 [35]	2	3
RKF23b [35]	2	3
Ceschino24 [8]	2	4
BogackiShampine32 [6]	3	2
Merson45 [7]	4	3
Zonneveld43 [9]	4	3
Fehlberg45 [35]	4	5

For a better comparison, we compare the methods using relative work versus precision diagrams. In this paper, we use precision to refer to the temporal error. The problems are computed with varying number of spatial points $N = 128–512$ until we obtain a constant error indicating a spatial error plateau less than $1e-8$. Thus, the reported precision indicated the temporal accuracy. For the sake of completeness, the usual total work versus precision diagrams are also given in Appendix C. Fig. 10. In the sequel, each method is compared against a reference method for the sake of easy comparison. The reference methods are the following:

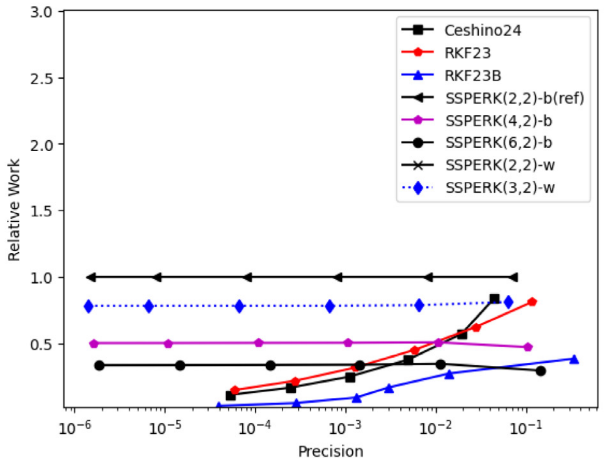
- For second order methods it is SSPERK$(2,2) - \tilde{b}^T$,
- for third order methods it is BogackiShampine32,
- and finally for fourth order methods it is Fehlberg45.

All of the following simulations can be reproduced using the paper's GitHub repository [36]. Furthermore, there we report the error coefficients for all the second, third and fourth order explicit embedded pairs. The global errors are calculated by using a very accurate solution calculated by MATLAB's *ODE45* with absolute and relative tolerances 10^{-14}.

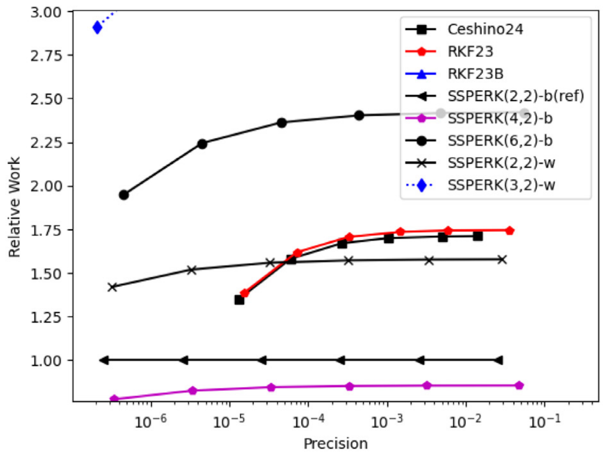
In all the numerical tests, PID controller performed best and so we only present this case. In Fig. 8 we can see the relative work versus precision diagrams for different orders in case of the PDE test problems.

SSPERK$(s,2) - \tilde{b}^T$ pairs are able to obtain a global error that is very close to the prescribed tolerance for the linear advection problem (see Fig. 8(a)). However, the second order methods from the literature (RKF23, Ceshino24) fall short. Although it appears that second order methods with third and fourth order (RKF23, Ceshino24) are less costly than the newly developed SSPERK pairs, these methods fail to obtain a global error better than $1e-5$ even with a restrictive tolerance of $1e-7$. For the Euler equations the SSPERK$(s,2) - \tilde{b}^T$ easily outperform the literature methods (see Fig. 8(b)). Summarizing the second order case we can say that, as our constructions predicted, SSPERK$(s,2) - \tilde{b}^T$ work robustly.

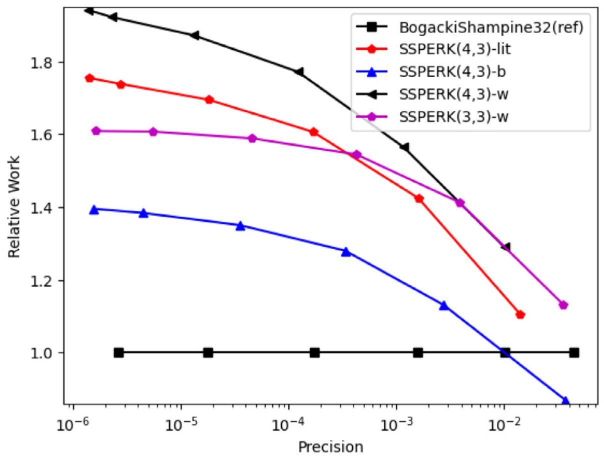
Fig. 8(c) and (d) show the relative work versus precision results for the third order methods. The results show the new SSPERK pairs are more expensive when compared to the reference method BogackiShampine32. This four stage method



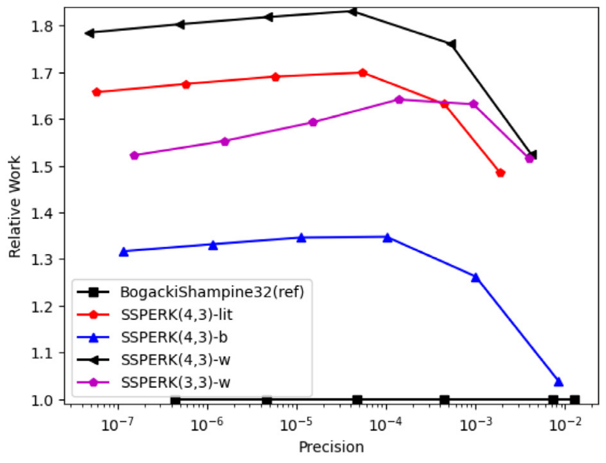
(a) Second order methods - Advection



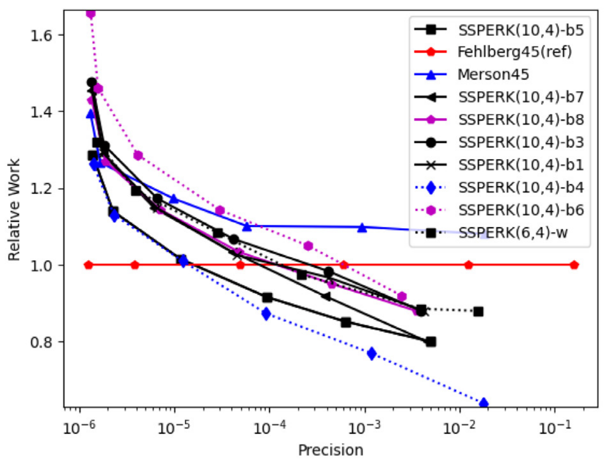
(b) Second order methods - Euler



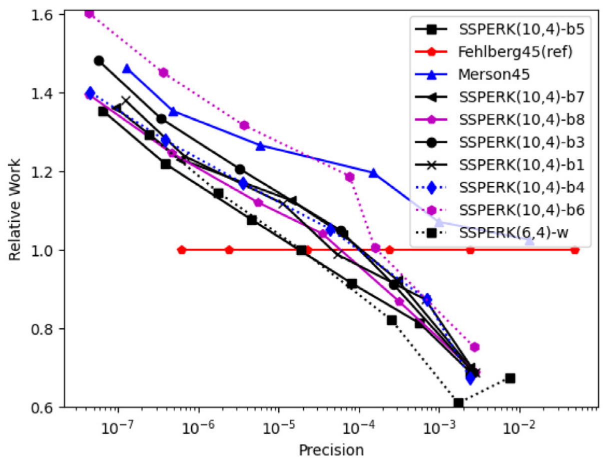
(c) Third order methods - Advection



(d) Third order methods - Euler



(e) Fourth order methods - Advection



(f) Fourth order methods - Euler

Fig. 8. Second, third and fourth order relative work versus precision diagrams using PID controller. The ordinate is the number of function evaluations (relative work) compared against the reference methods. The abscissa is the global error at the endpoint of integration (the precision).

has positive coefficients but is not an SSP method. For the hyperbolic problems, as it is expected, the newly constructed method SSPERK$(4, 3) - \tilde{b}^T$ performs better than the previous literature one SSPERK$(4, 3) - \tilde{b}^T_{\text{lit.}}$.

Similarly, Fig. 8(e) and (f) show the relative work versus precision results for the fourth order methods. As it is predicted, the best performing pair among the SSPERK$(10, 4)$ pairs is the pair $\tilde{b}^T_4$. The new pair is more efficient than the other methods for tolerances larger than $1e-4$. For tolerances larger than $1e-4$ the step-size restriction is more laxed and the run is vulnerable to stability violation at larger step-size. At this regime, larger step-size is recommended and may violate the stability requirement and ultimately resulting in poor accuracy of the solution. However, the newly constructed pairs perform best in this region since the higher order method used to advance the solution is SSP and these schemes were designed to be strongly stable with the largest possible step-size. The relative work versus precision figure shows this trade off more clearly in Fig. 8 than the total work versus precision in Appendix C. Fig. 10. At a very restrictive tolerance, i.e. when the tolerance is less than $1e-5$, the optimal step-sizes are small enough that stability is not an issue and SSPERK pairs are no longer favorable. With the high stage count of the schemes, these methods roughly perform 20%–50% more number of function evaluations than the reference method Fehlberg45.

4. Conclusions

Due to stability issues SSP methods are heavily used for the time integration of spatially discretized hyperbolic problems. Modern robust IVP solvers include many important features such as error estimation and automatic step-size control. Most of these important features have not yet been developed for existing higher-order optimal explicit SSP Runge–Kutta methods. The current work provides these important features. The numerical results provide evidence of the effectiveness of the newly created pairs.

Based on the theoretical investigation supported by the numerical experiments we recommend the following pairs for optimal explicit SSP Runge–Kutta methods:

- Second order methods: we recommend the lower stage pairs SSPERK$(s, 2) - \tilde{b}^T$ pairs, especially when $s = 4$.
- Third order methods: we recommend the SSPERK$(4, 3)$–$\tilde{b}^T$ pair. Based on the recent paper [37], our method performs really well on challenging industrial compressible computational fluid dynamics problems.
  Due to its popularity we have also created an embedded pair for SSPERK$(3, 3)$ method.
- Fourth order methods: we recommend the SSPERK$(10, 4) - \tilde{b}^T_4$ pair. For the sake of users' requirement we have created a lower stage number embedded pair for SSPERK$(6, 4)$ method, too.

We can conclude, depending on the problem being integrated and length of integration, the new SSPERK embedded pairs are effective, practical and robust. At moderate tolerance level, the embedded pairs are capable of integrating at near stability limit without the need for an expensive evaluation of the right-hand side often required for CFL computation. One can reproduce all of the results using the paper's GitHub repository [36].

Acknowledgments

The work of Imre Fekete was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences. EFOP-3.6.3-VEKOP-16-2017-00001: Talent Management in Autonomous Vehicle Control Technologies — The Project is supported by the Hungarian Government and co-financed by the European Social Fund. "Application Domain Specific Highly Reliable IT Solutions" project has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme TKP2020-NKA-06 (National Challenges Subprogramme) funding scheme. The project has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002).

The work of Sidafa Conde and John Shadid at Sandia was partially supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics Program and by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research and Office of Fusion Energy Sciences, Scientific Discovery through Advanced Computing (SciDAC) program. This manuscript has been authored by National Technology & Engineering Solutions of Sandia, LLC., USA, under contract DE-NA0003525 with the U.S. Department of Energy/National Nuclear Security Administration. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

In addition, this work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation, USA grant number ACI-1548562 [38]. This work used the Extreme Science and Engineering Discovery Environment (XSEDE) Stampede at the Texas Advanced Computing Center (TACC) through allocation DMS170002.

We thank Sigal Gottlieb and David Ketcheson for the initial stimulating discussions on the topic and their helpful remarks.

We are indebted to the referees for their suggestions that helped us improving the quality and the presentation of the material.

Appendix A. Stability radius and error metric values for optimal SSPERK methods and their embedded pairs

All of the computations in Appendices A and B. regarding stability region plots, stability radius values and error metric values were done by the NodePy Python package [39] and all of them can be reproduced using the Jupyter notebooks at the paper's GitHub repository [36].

SSPERK(s, 2)

See Tables 6 and 7.

Table 6
The δ_C and $R(\psi)$ values for the SSPERK(s, 2) and the embedded methods.

	$s=2$		$s=3$		$s=4$	
	δ_C	$R(\psi)$	δ_C	$R(\psi)$	δ_C	$R(\psi)$
SSPERK(s, 2)	1.0000	0.9999	2.0000	1.9999	3.0000	2.9999
$\tilde{b}^T$	1.2679	0.9999	2.1577	1.9999	3.0000	2.9999
$\tilde{w}^T$	1.2805	0.9999	0.6028	0.2024	–	–
	$s=6$		$s=8$		$s=10$	
	δ_C	$R(\psi)$	δ_C	$R(\psi)$	δ_C	$R(\psi)$
SSPERK(s, 2)	5.0000	4.9999	7.0000	6.9999	9.0000	8.9999
$\tilde{b}^T$	5.0000	4.9999	7.0000	6.9999	9.0000	8.9999

Table 7
The error metric values for the SSPERK(s, 2) and the embedded methods.

	Method	$A_2^{(p+1)}$	$A_\infty^{(p+1)}$	$\tilde{A}_2^{(p)}$	$\tilde{A}_\infty^{(p)}$	D
$s=2$	SSPERK(s, 2)	0.186339	0.166667	0.144338	0.125	1
	$\tilde{b}^T$	0.25	0.25	0.171796	0.166667	1
	$\tilde{w}^T$	0.194021	0.194021	0.167227	0.166667	1
$s=3$	SSPERK(s, 2)	0.093170	0.083333	0.065881	0.041667	1
	$\tilde{b}^T$	0.111111	0.111111	0.111976	0.111111	1
	$\tilde{w}^T$	0.152309	0.152309	0.083983	0.083930	1
$s=4$	SSPERK(s, 2)	0.062113	0.055556	0.044406	0.032407	1
	$\tilde{b}^T$	0.0625	0.0625	0.076468	0.07639	1
$s=6$	SSPERK(s, 2)	0.037268	0.033333	0.027285	0.021667	1
	$\tilde{b}^T$	0.027778	0.027778	0.044531	0.044444	1
$s=8$	SSPERK(s, 2)	0.026620	0.023810	0.019760	0.016157	1
	$\tilde{b}^T$	0.015625	0.015625	0.030780	0.030506	1
$s=10$	SSPERK(s, 2)	0.020704	0.018519	0.015501	0.012860	1
	$\tilde{b}^T$	0.010000	0.010000	0.023355	0.022963	1

SSPERK(s, 3)

See Tables 8–13.

Table 8
The δ_C and $R(\psi)$ values for the SSPERK(4, 3) and the embedded methods.

	δ_C	$R(\psi)$
SSPERK(4, 3)	2.0000	1.9999
$\tilde{b}^T_{\text{lit.}}$	2.0000	1.9999
$\tilde{b}^T$	2.0000	1.9999
$\tilde{w}^T$	0.7282	0.3314

Table 9
The error metric values for the SSPERK(4, 3) and the embedded methods.

Method	$A_2^{(p+1)}$	$A_\infty^{(p+1)}$	$\tilde{A}_2^{(p)}$	$\tilde{A}_\infty^{(p)}$	D
SSPERK(4, 3)	0.036084	0.020833	0.030230	0.022917	1
$\tilde{b}^T_{\text{lit.}}$	0.093170	0.083333	0.065881	0.041667	1
$\tilde{b}^T$	0.046585	0.041667	0.045405	0.031250	1
$\tilde{w}^T$	0.132132	0.131950	0.104820	0.090282	1

Table 10
The δ_C and $R(\psi)$ values for the SSPERK(n^2, 3) method and the embedded method.

	$n^2 = 9$		$n^2 = 16$		$n^2 = 25$		$n^2 = 36$	
	δ_C	$R(\psi)$	δ_C	$R(\psi)$	δ_C	$R(\psi)$	δ_C	$R(\psi)$
SSPERK(n^2, 3)	6.0000	5.9999	12.0000	11.9594	20.0000	18.2029	30.0000	26.3612
$\tilde{b}^T$	3.6886	1.1441	4.7470	1.4618	5.4710	1.7148	6.0286	1.9260

Table 11
The error metric values for the SSPERK(n^2, 3) and the embedded methods.

	Method	$A_2^{(p+1)}$	$A_\infty^{(p+1)}$	$\tilde{A}_2^{(p)}$	$\tilde{A}_\infty^{(p)}$	D
$n^2 = 9$	SSPERK(n^2, 3)	0.008965	0.006944	0.009064	0.006674	0.833333
	$\tilde{b}^T$	0.019088	0.018519	0.023315	0.020062	0.833333
$n^2 = 16$	SSPERK(n^2, 3)	0.004311	0.003472	0.004763	0.003800	0.916667
	$\tilde{b}^T$	0.013997	0.012153	0.018097	0.015914	0.916667
$n^2 = 25$	SSPERK(n^2, 3)	0.002564	0.002083	0.002949	0.002404	0.95
	$\tilde{b}^T$	0.011335	0.009167	0.014752	0.013000	0.95
$n^2 = 36$	SSPERK(n^2, 3)	0.001705	0.001389	0.002001	0.001646	0.966667
	$\tilde{b}^T$	0.009544	0.007407	0.012389	0.010910	0.966667

Table 12
The δ_C and $R(\psi)$ values for the SSPERK(3, 3) and the embedded method.

	δ_C	$R(\psi)$
SSPERK(3, 3)	1.0000	0.9999
$\tilde{w}^T$	1.0000	0.9999

Table 13
The error metric values for the SSPERK(3, 3) and the embedded method.

Method	$A_2^{(p+1)}$	$A_\infty^{(p+1)}$	$\tilde{A}_2^{(p)}$	$\tilde{A}_\infty^{(p)}$	D
SSPERK(3, 3)	0.072169	0.041667	0.056486	0.033333	1
$\tilde{w}^T$	0.072169	0.041667	0.056486	0.033333	1

SSPERK(10, 4)

See Tables 14 and 15.

Table 14
The δ_C and $R(\psi)$ values for the SSPERK(10, 4) and the embedded methods.

	δ_C	$R(\psi)$
SSPERK(10, 4)	6.0000	5.9999
$\tilde{b}_1^T, \tilde{b}_2^T, \tilde{b}_3^T, \tilde{b}_4^T, \tilde{b}_5^T, \tilde{b}_6^T, \tilde{b}_7^T, \tilde{b}_8^T$	0.0000	0.0000

Table 15
The δ_R and δ_I values for the SSPERK(10, 4) and the embedded methods.

	SSPERK(10, 4)	$\tilde{b}_1^T$	$\tilde{b}_2^T$	$\tilde{b}_3^T$	$\tilde{b}_4^T$	$\tilde{b}_5^T$	$\tilde{b}_6^T$	$\tilde{b}_7^T$	$\tilde{b}_8^T$
δ_R	13.92	6.00	8.40	13.34	9.90	7.23	6.00	6.00	8.75
δ_I	0.0008	0	4.61	0	4.70	0	2.51	0	0

SSPERK(6, 4)

See Table 16.

Table 16
The δ_C and $R(\psi)$ values for SSPERK(6, 4) and embedded method.

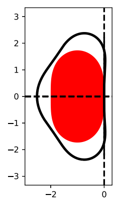
	δ_C	$R(\psi)$
SSPERK(6, 4)	2.5055	2.2944
$\tilde{w}$	0.8915	0.3745

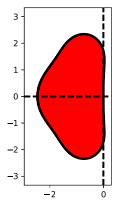
Appendix B. Stability radius values and stability regions of the well-known embedded methods

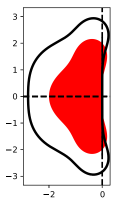
See Table 17 and Fig. 9.

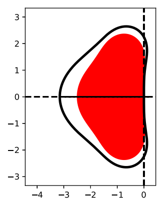
Table 17
The error metric values for the well-known embedded methods.

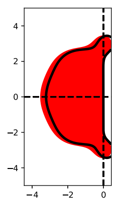
Method		δ_C	$R(\psi)$
RKF23	b^T	0.2083	0
	$\tilde{b}^T$	0	0
RKF23b	b^T	1	0.9999
	$\tilde{b}^T$	1	0.9999
Ceschino24	b^T	0	0
	$\tilde{b}^T$	0	0
BogackiShampine32	b^T	0.8990	0
	$\tilde{b}^T$	0.9583	0
Merson45	b^T	0	0
	$\tilde{b}^T$	0	0
Zonneveld43	b^T	1	0
	$\tilde{b}^T$	0	0
Fehlberg45	b^T	0	0
	$\tilde{b}^T$	0	0

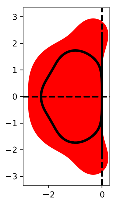

(a) RKF23


(b) RKF23b


(c) Ceschino24


(d) BogackiShampine32


(e) Merson45


(f) Zonneveld43

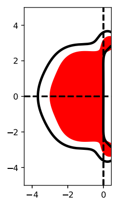
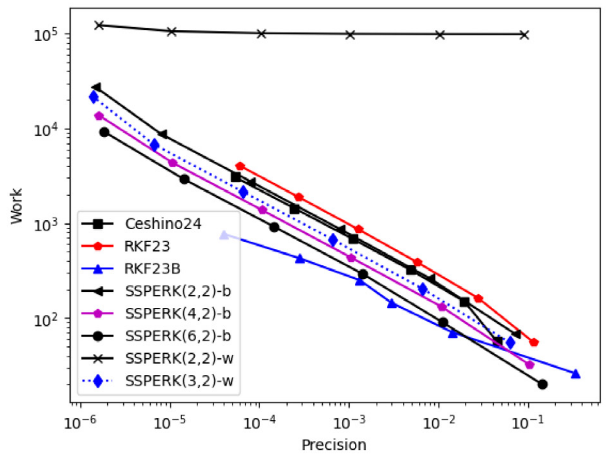

(g) Fehlberg45

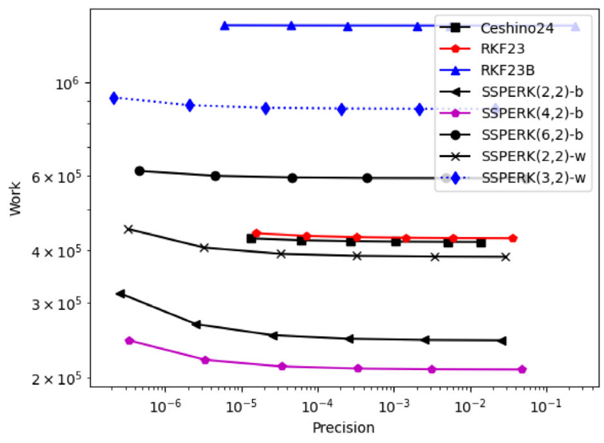
Fig. 9. The stability regions of the well-known methods (red) and the corresponding embedded methods (black contour).

Appendix C. Extended numerical results

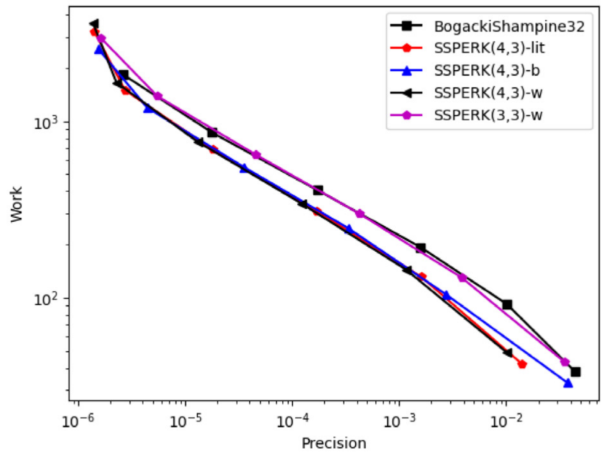
All of the numerical results below can be reproduced using the MATLAB codes at the paper's GitHub repository [36]. Furthermore, we report the error coefficients for all of the embedded pairs. In addition, we also provide the MATLAB codes corresponding to the I, PI, and explicit Gustafsson controllers.



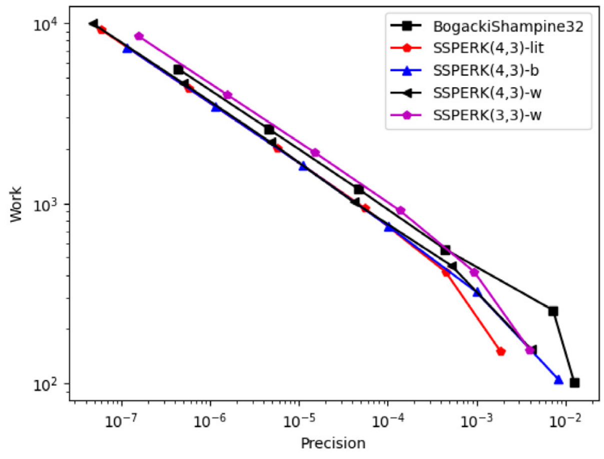
(a) Second order methods - Advection



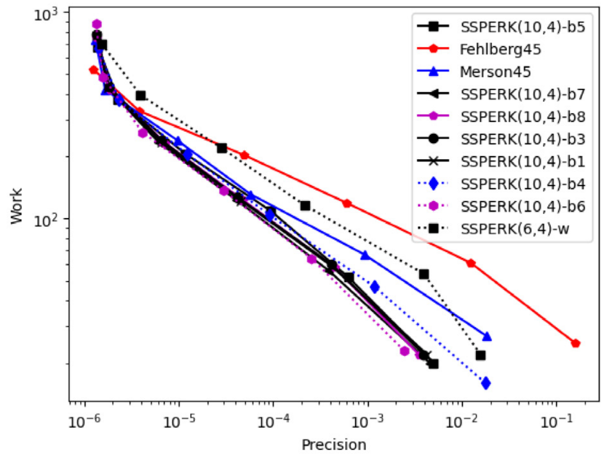
(b) Second order methods - Euler



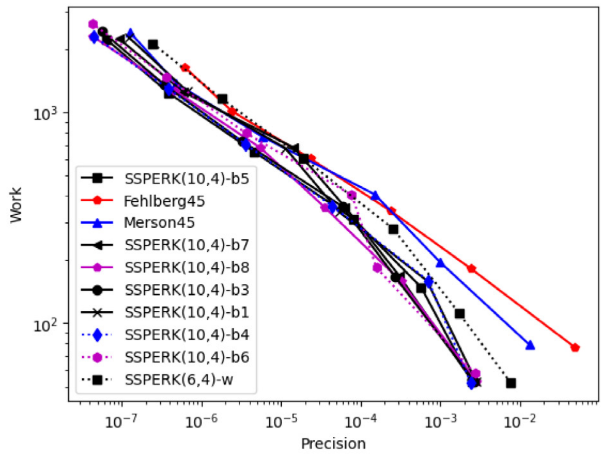
(c) Third order methods - Advection



(d) Third order methods - Euler



(e) Fourth order methods - Advection



(f) Fourth order methods - Euler

Fig. 10. Second, third and fourth order work versus precision diagrams using PID controller. The ordinate is the total number of function evaluations (work) compared against the reference methods. The abscissa is the global error at the endpoint of integration (the precision).

References

[1] U.M. Ascher, L.R. Petzold, Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations, SIAM, Philadelphia, 1998.
[2] K. Gustafsson, Control theoretic techniques for stepsize selection in explicit Runge–Kutta methods, ACM Trans. Math. Software 17 (1991) 533–554.
[3] G. Söderlind, Automatic control and adaptive time-stepping, Numer. Algorithms 31 (2002) 281–310.
[4] G. Söderlind, Digital filters in adaptive time-stepping, ACM Trans. Math. Software 29 (2003) 1–26.
[5] G. Söderlind, Time-step selection algorithms: Adaptivity, control, and signal processing, Appl. Num. Math. 56 (2006) 488–502.
[6] P. Bogacki, L.F. Shampine, A 3(2) pair of Runge–Kutta formulas, Appl. Math. Lett. 2 (1989) 321–325.
[7] R.H. Merson, An operational method for the study of integration processes, in: Proc. Symp. Data Processing, Vol. 1, 1957, pp. 110–125.
[8] F. Ceschino, Evaluation de l'erreur par pas dans les problemes différentiels, Chriffres 5 (1962) 223–229.
[9] J.A. Zonneveld, Automatic Integration of Ordinary Differential Equations, Mathematisch Centrum, Amsterdam, 1964.
[10] E. Hairer, S.P. Nørsett, G. Wanner, Solving Ordinary Differential Equations. I, second ed., Springer-Verlag, Berlin, 1993.
[11] C.A. Kennedy, M.H. Carpenter, Additive Runge–Kutta schemes for convection–diffusion-reaction equations, Appl. Numer. Math. 44 (2003) 139–181.
[12] L.F. Shampine, M.W. Reichelt, The MATLAB ODE suite, SIAM J. Sci. Comput. 18 (1997) 1–22.
[13] S. Gottlieb, C.-W. Shu, Total variation diminishing Runge-Kutta schemes, Math. Comp. 67 (1998) 73–85.
[14] S. Gottlieb, D.I. Ketcheson, C.-W. Shu, Strong Stability Preserving Runge-Kutta and Multistep Time Discretizations, World Scientific, Publishing Co. NJ, 2011.
[15] S. Gottlieb, Strong stability preserving time discretizations: A review, ICOSAHOM 2014 (2015) 17–30.
[16] C.-W. Shu, Total-variation-diminishing time discretization, SIAM J. Sci. Stat. Comput. 9 (1988) 1073–1084.
[17] C.B. Macdonald, Constructing High-Order Runge–Kutta Methods with Embedded Strong-Stability-Preserving Pairs (Master Thesis), Simon Fraser University, Canada, 2003.
[18] D.I. Ketcheson, High Order Strong Stability Preserving Time Integrators and Numerical Wave Propagation Methods for Hyperbolic PDEs (PhD Dissertation), University of Washington, United States, 2009.
[19] J.F.B.M. Kraaijevanger, Contractivity of Runge-Kutta methods, BIT 31 (1991) 482–528.
[20] I. Higueras, On strong stability preserving time discretization methods, J. Sci. Comput. 21 (2004) 193–223.
[21] K. Dekker, J.G. Verwer, Stability of Runge–Kutta Methods for Stiff Nonlinear Differential Equations, North-Holland Publishing, Amsterdam, 1984.
[22] R.P. van der Marel, Stability radius of polynomials occurring in the numerical solution of initial value problems, BIT 30 (1990) 516–528.
[23] 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 (2000) 177–219.
[24] J.R. Dormand, P.J. Prince, A family of embedded Runge–Kutta formulae, J. Comput. Appl. Math. 6 (1980) 19–26.
[25] R.J. Spiteri, S.J. Ruuth, A new class of optimal high-order strong-stability-preserving time discretization methods, SIAM J. Numer. Anal. 40 (2002) 469–491.
[26] S. Conde, S. Gottlieb, Z.J. Grant, J.N. Shadid, Implicit and implicit–explicit strong stability preserving Runge–Kutta methods with high linear order, J. Sci. Comput. 73 (2017) 667–690.
[27] D.I. Ketcheson, Highly efficient strong stability-preserving Runge–Kutta methods with low-storage implementations, SIAM J. Sci. Comput. 30 (2008) 2113–2136.
[28] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, J. Comput. Phys. 77 (1988) 439–471.
[29] S. Gottlieb, Z. Grant, D. Higgs, Optimal explicit strong stability preserving Runge—Kutta methods with high linear order and optimal nonlinear order, Math. Comp. 84 (2015) 2743–2761.
[30] A.C. Hindmarsh, P.N. Brown, K.E. Grant, S.L. Lee, R. Serban, D.E. Shumaker, C.S. Woodward, SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers, ACM Trans. Math. Software 31 (2005) 363–396.
[31] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, J. Comput. Phys. 126 (1996) 202–228.
[32] J. Hesthaven, Numerical Methods for Conservation Laws, SIAM, Philadelphia, 2017.
[33] R.J. LeVeque, Numerical Methods for Conservation Laws, Birkhäuser Basel, 1992.
[34] R.J. LeVeque, Finite Volume Methods for Hyperbolic Problems, Cambridge University Press, 2002.
[35] E. Fehlberg, Low-order Classical Runge-Kutta Formulas with Step Size Control and Their Application to Some Heat Transfer Problems, NASA Technical Report 315, 1969.
[36] I. Fekete, S. Conde, J.N. Shadid, GitHub repository, 2022, https://github.com/feipaat/Embedded-pairs-for-optimal-SSPERK.
[37] H. Ranocha, L. Dalcin, M. Parsani, D.I. Ketcheson, Optimized Runge–Kutta methods with automatic step size control for compressible computational fluid dynamics, 2021, arXiv:2104.06836v2.
[38] J. Towns, T. Cockerill, M. Dahan, I. Foster, K. Gaither, A. Grimshaw, V. Hazlewood, S. Lathrop, D. Lifka, G.D. Peterson, R. Roskies, J.R. Scott, N. Wilkins-Diehr, XSEDE: Accelerating scientific discovery, Comput. Sci. Eng. 16 (2014) 62–74.
[39] D.I. Ketcheson, H. Ranocha, M. Parsani, U. bin Waheed, Y. Hadjimichael, NodePy: A package for the analysis of numerical ODE solvers, J. Open Source Softw. 5 (2020).