Relaxation Runge-Kutta Methods: Conservation and stability for Inner-Product Norms

David I. Ketcheson*

Abstract

We further develop a simple modification of Runge–Kutta methods that guarantees conservation or stability with respect to any inner-product norm. The modified methods can be explicit and retain the accuracy and stability properties of the unmodified Runge–Kutta method. We study the properties of the modified methods and show their effectiveness through numerical examples, including application to entropy-stability for first-order hyperbolic PDEs.

1 Motivation and background

Consider the initial value problem

$$u'(t) = f(t, u(t)) \tag{1a}$$

$$u(t_0) = u_0 \tag{1b}$$

where $u : \mathbb{R} \to \mathbb{R}^m$ and $f : \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}^m$. In this work we focus on problems that are dissipative with respect to some inner-product norm:

$$\frac{d}{dt}\|u(t)\|^2 = 2\langle u, f(t,u)\rangle \le 0.$$
(2)

Here and throughout, $\langle \cdot, \cdot \rangle$ denotes an inner product and $\|\cdot\|$ the corresponding norm; we will sometimes refer to $\|u\|^2$ as *energy*. For dissipative problems, it is desirable that the numerical solution mimic (2):

$$\|u^{n+1}\| \le \|u^n\|. \tag{3}$$

Herein, a method is called *monotonicity preserving* if it guarantees (3) for all problems satisfying (2).

We say the problem (1) is *conservative* if

$$\langle u, f(t, u) \rangle = 0. \tag{4}$$

For such problems it is desirable to discretely conserve energy:

$$\|u^{n+1}\| = \|u^n\|.$$
(5)

^{*}Computer, Electrical, and Mathematical Sciences & Engineering Division, King Abdullah University of Science and Technology, 4700 KAUST, Thuwal 23955, Saudi Arabia. (david.ketcheson@kaust.edu.sa)

 $^{^{1}}$ We consider real spaces for simplicity, but the methods developed here are also applicable in complex spaces.

A method is also called *conservative* if it guarantees (5) for all problems satisfying (4).

In many applications, numerical conservation or monotonicity preservation are of great importance. Not only do these properties guarantee that the solution remains bounded, but violation of these properties can lead to solutions that are unphysical and qualitatively wrong. Nevertheless, most numerical methods do not enforce these properties exactly, but only up to truncation errors. This is particularly true for explicit Runge–Kutta methods. Even if one considers only linear autonomous systems of ODEs, it has been shown that many explicit Runge–Kutta methods including the classical 4th-order method — are not even conditionally monotonicity preserving; see [26, 24, 21, 25]. For general nonlinear autonomous systems, no methods of order greater than one are known to be monotonicity preserving [19]. For conservative problems, no explicit Runge–Kutta method of any order can enforce discrete conservation, even for linear autonomous problems. Formulas for the production of a general convex entropy function (which includes the special case of an inner-product norm) have been derived in [18].

Monotonicity preservation for Runge–Kutta methods with respect to an inner-product norm was studied by Higueras [14], with results closely connected to an earlier study of contractivity preservation by Dahlquist & Jeltsch[6]. However, to obtain results for explicit RK methods in that framework, it is necessary to require strict inequality in (2), making the results inapplicable for conservative problems or for typical high-order semi-discretizations of hyperbolic PDEs.

Unconditionally conservative and monotonicity-preserving methods exist, for instance, among the classes of implicit Runge–Kutta and (for Hamiltonian systems) partitioned Runge–Kutta methods; see [12] and references therein.

While classical explicit Runge–Kutta or linear multistep methods cannot preserve general quadratic invariants, some explicit energy-conservative methods have been developed by going outside these traditional classes (as is done also in the present work). Indeed, it is possible to modify any Runge–Kutta method to preserve energy or other first integrals by a technique known as projection; see e.g. [10, 4]. In this approach, at the end of each time step the solution is projected onto a desired set in order to ensure some property like conservation or monotonicity. The approach used in the current work can be viewed as a projection method where the projection is performed along a direction corresponding to the next time step update, but with an additional modification that the size of the time step is also modified. This approach was originally proposed by Dekker & Verwer [7, pp. 265-266], who noted that the classical four-stage RK method could be modified slightly to conserve energy while maintaining its accuracy. The idea was extended in [8] to a restricted class of fourth order methods. This was developed further in [5] by giving a general proof that applying the technique (without the step size adjustment) to a RK method of order p results in a method of order at least p-1. Subsequent development in [5, 4, 17] focuses on *embedded projection* methods; i.e. methods that project in a direction determined by an embedded Runge–Kutta pair. The approach described in the present work could be viewed as a variant in which the "embedded" method is simply the identity map, but with an additional twist that requires reinterpretation of the new step solution as an approximation at a slightly different time.

Like embedded projection methods, the methods proposed here are:

- explicit
- conditionally conservative and monotonicity-preserving for general nonlinear ODEs
- arbitrarily high order accurate
- linearly covariant

Furthermore, they do not require partitioning or temporal staggering, and they inherit other useful properties (such as strong stability preservation) of a selected Runge–Kutta method. Preservation of more general (non-inner-product) functionals is also possible with modification similar to that described herein; see [22].

The main contributions of the present work are: first, to further develop these methods that, while not entirely new, seem to have been overlooked; second, to put them on a rigorous footing in terms of accuracy and stability properties; and third, to explore their properties through analysis and numerical experiments.

1.1 Energy evolution by Runge–Kutta methods

A Runge–Kutta method applied to (1) takes the form

$$y_i = u^n + \Delta t \sum_{j=1}^s a_{ij} f(t_n + c_j \Delta t, y_j)$$
(6a)

$$u(t_n + \Delta t) \approx u^{n+1} = u^n + \Delta t \sum_{j=1}^s b_j f(t_n + c_j \Delta t, y_j).$$
(6b)

We make the usual assumption that $c_j = \sum_i a_{ij}$. For convenience we introduce the shorthand $f_{ij} = f(t_j + c_j \Delta t_j)$

$$f_i = f(t_n + c_i \Delta t, y_i)$$

for the *i*th stage derivative. The change in energy from one step to the next is

$$\|u^{n+1}\|^{2} - \|u^{n}\|^{2} = \left\|u^{n} + \Delta t \sum_{j=1}^{s} b_{j} f_{j}\right\|^{2} - \|u^{n}\|^{2}$$

= $2\Delta t \sum_{j=1}^{s} b_{j} \langle u^{n}, f_{j} \rangle + \Delta t^{2} \sum_{i,j=1}^{s} b_{i} b_{j} \langle f_{i}, f_{j} \rangle$
= $2\Delta t \sum_{j=1}^{s} b_{j} \langle y_{j}, f_{j} \rangle - 2\Delta t \sum_{j=1}^{s} b_{j} \langle u^{n} - y_{j}, f_{j} \rangle + \Delta t^{2} \sum_{i,j=1}^{s} b_{i} b_{j} \langle f_{i}, f_{j} \rangle,$

which can be rewritten using (6a) as

$$\|u^{n+1}\|^2 - \|u^n\|^2 = 2\Delta t \sum_{j=1}^s b_j \langle y_j, f_j \rangle - 2\Delta t^2 \sum_{i,j=1}^s b_i a_{ij} \langle f_j, f_i \rangle + \Delta t^2 \sum_{i,j=1}^s b_i b_j \langle f_i, f_j \rangle.$$
(7)

The first sum on the right side of (7) is zero for conservative systems, and it is negative for dissipative systems if $b_j \ge 0$ for all j. However, the remaining two terms may lead to violation of the conservation or monotonicity property. Those two terms can be written together as the bilinear form

$$-\Delta t^2 \sum_{i,j=1}^{s} m_{ij} \langle f_i, f_j \rangle, \tag{8}$$

where, letting B denote the diagonal matrix with entries b_j ,

$$M = BA + A^T B - bb^T$$

This is the traditional analysis used in studying symplecticity, algebraic stability, and other properties of RK methods (see e.g. [3, 7, 12]). If the matrix M is positive semidefinite and the weights are nonnegative, then the method is said to be algebraically stable. Clearly, such methods are unconditionally monotonicity-preserving. If M = 0, the method is said to be symplectic; clearly such methods are unconditionally conservative. Certain well-known implicit methods have these properties. However, explicit methods cannot be algebraically stable or symplectic.

2 Relaxation Runge–Kutta methods

The relaxation version of the method (6) is obtained by replacing (6b) with the update formula

$$u(t_n + \gamma_n \Delta t) \approx u_{\gamma}^{n+1} = u^n + \Delta t \gamma_n \sum_{j=1}^s b_j f(t_n + c_j \Delta t, y_j).$$
(9)

The only difference between (9) and (6b) is the factor γ_n that multiplies the step size. We can think of the original Runge-Kutta method (6) as determining only the direction in which the solution will be updated, while the choice of γ_n determines how far to step in that direction. From this point of view γ_n is similar to the relaxation parameter used in some iterative algebraic solvers, and for this reason we refer to these methods as relaxation Runge-Kutta (RRK) methods.

With this change, (7) becomes

$$|u_{\gamma}^{n+1}||^{2} - ||u^{n}||^{2} = 2\gamma_{n}\Delta t \sum_{j=1}^{s} b_{j}\langle y_{j}, f_{j}\rangle - 2\gamma_{n}\Delta t^{2} \sum_{i,j=1}^{s} b_{i}a_{ij}\langle f_{j}, f_{i}\rangle + \gamma_{n}^{2}\Delta t^{2} \sum_{i,j=1}^{s} b_{i}b_{j}\langle f_{i}, f_{j}\rangle.$$

$$(10)$$

We can eliminate the last two terms by setting

$$\gamma_n = \frac{2\sum_{i,j=1}^s b_i a_{ij} \langle f_i, f_j \rangle}{\sum_{i,j=1}^s b_i b_j \langle f_i, f_j \rangle},\tag{11}$$

so that

$$||u_{\gamma}^{n+1}||^2 - ||u^n||^2 = 2\gamma_n \Delta t \sum_{j=1}^s b_j \langle y_j, f_j \rangle$$

In case the denominator of (11) vanishes, we have $u^{n+1} = u^n$, so we can achieve conservation or monotonicity by taking simply $\gamma_n = 1$. We thus define in place of (11)

$$\gamma_n = \begin{cases} 1 & \|\sum_{j=1}^s b_j f_j\|^2 = 0\\ \frac{2\sum_{i,j=1}^s b_i a_{ij} \langle f_i, f_j \rangle}{\sum_{i,j=1}^s b_i b_j \langle f_i, f_j \rangle} & \|\sum_{j=1}^s b_j f_j\|^2 \neq 0. \end{cases}$$
(12)

Because we will interpret u_{γ}^{n+1} as an approximation to the solution at time $u(t_n + \gamma_n \Delta t)$, it is important that $\gamma_n > 0$. It is straightforward to show the following:

Lemma 1. Let $\sum b_i a_{ij} > 0$, let f be sufficiently smooth, and let γ_n be defined by (12). Then $\gamma_n > 0$ for sufficiently small $\Delta t > 0$.

Note that the condition in Lemma 1 holds for all methods of order two or higher.

Remark 1. A more detailed analysis indicates that *small enough* here means that Δt should be no more than about 1/L, where L is the Lipschitz constant of f. This is also the order of the absolutely stable step size when using any explicit method, so one may expect that using an absolutely stable step size will also yield $\gamma_n > 0$.

For conservative systems this gives exact energy conservation; for dissipative systems it preserves dissipativity as long as all weights are non-negative.

Theorem 2. Let (A, b) be the coefficients of a Runge-Kutta method of order at least two. The corresponding relaxation Runge-Kutta method defined by (6a) and (9) with γ_n defined by (12) is conservative. If the weights are nonnegative, then the relaxation method is also monotonicity preserving as long as Δt is chosen so that $\gamma_n \geq 0$.

Remark 2. The relaxation RK method still conserves linear invariants, which is important for instance in the semi-discretization of hyperbolic conservation laws.

Remark 3. A version of Theorem 2 applicable only to conservative systems appears in [5, Thm. 2.1], along with a formula for γ_n that is more computationally efficient (but correct only for conservative systems).

Remark 4. The denominator of the expression for γ_n in (11) is simply the norm of the step update, and can be computed with a single inner product. It has been pointed out by Hendrik Ranocha that by solving (7) for the term that is the numerator of the expression for γ_n , it can be computed using only *s* inner products [20]. Thus γ_n can be computed with just s + 1 inner products.

The update formula (9) is equivalent to replacing the coefficients b_j in the Runge–Kutta method with $\gamma_n b_j$. It can also be viewed roughly as taking a Runge–Kutta step of size $\gamma_n \Delta t$ in place of Δt , but notice that the original step size Δt is still used in the computation of the stages y_i . Both viewpoints (rescaling b and rescaling Δt) will be useful in our analysis.

We will see below in Lemma 4 that, for reasonable values of Δt , γ_n is close to unity. Although the proof of this fact is technical, the result itself is not surprising, at least for conservative systems. For such systems, the value of γ_n given by (12) is a solution of

$$\Delta E(\gamma) := \|u_{\gamma}^{n+1}\|^2 - \|u^{n+1}\|^2 = 0,$$

which is a quadratic function of γ with one root at zero. Given that $\Delta E(1) = \mathcal{O}(\Delta t^{p+1})$ and that according to (10) $\Delta E \propto \Delta t^2$ for conservative problems, it is natural to expect that $\Delta E(\gamma)$ has a zero within $\mathcal{O}(\Delta t^{p-1})$ of unity. The method described here can be viewed as using a line search to determine the step size that solves $\Delta E(\gamma) = 0$ and thus exactly conserves energy.

For dissipative systems also, we will see that the last two terms in (7) are not important (in the sense of accuracy) to the numerical approximation of the energy evolution; i.e. the term $2\Delta t \sum_{j} b_j \langle y_j, f_j \rangle$ approximates the energy evolution over one step to order p:

$$\Delta E(1) = \|u(t_n + \Delta t)\|^2 - \|u(t_n)\|^2 = 2\Delta t \sum_j b_j \langle y_j, f_j \rangle + \mathcal{O}(\Delta t^{p+1}).$$

2.1 Accuracy

At first glance, the RRK method (given by (6a) with (9)) seems to be not even consistent, since $\sum_j \gamma_n b_j = \gamma_n \neq 1$ in general. For a classical RK method, the condition $\sum_j b_j = 1$, along with

higher order conditions, is necessary for local consistency of a given order. But for an RRK method, because the coefficients depend on Δt , we can still obtain high order accuracy if the order conditions are *nearly* satisfied, which is true if γ_n is sufficiently close to unity.

Theorem 3. Let (a_{ij}, b_j) be the coefficients of a Runge-Kutta method of order p. Consider the RRK method defined by (6a), (9) and suppose that

$$\gamma_n = 1 + \mathcal{O}(\Delta t^{p-1}). \tag{13}$$

Then:

- 1. (IDT method) If the solution u_{γ}^{n+1} is interpreted as an approximation to $u(t_n + \Delta t)$, the method has order p 1.
- 2. (RRK method) If the solution u_{γ}^{n+1} is interpreted as an approximation to $u(t_n + \gamma_n \Delta t)$, the method has order p.

Proof. First we note that by taking appropriate linear combinations of the usual order conditions, the conditions for a RK method to have order p can be written as (see [1])

$$b^T c^{k-1} - 1/k = 0$$
 $1 \le k \le p$ (14a)

$$b^T v = 0 \qquad \qquad \forall v \in V_p, \tag{14b}$$

where V_p is a set of vectors depending only on a_{ij} whose specific elements are not important here. For the RRK method, we replace b with $\gamma_n b$, obtaining the conditions

$$\gamma_n b^T c^{k-1} - 1/k = 0 \qquad \qquad 1 \le k \le p \tag{15a}$$

$$\gamma_n b^T v = 0 \qquad \qquad \forall v \in V_p. \tag{15b}$$

Given a method that satisfies (14), clearly (15b) is satisfied as well, while the left hand side of (15a) is $\mathcal{O}(\Delta t^{p-1})$. In the error expansion, this value gets multiplied by $\mathcal{O}(\Delta t^k)$, so the leading error term is $\mathcal{O}(\Delta t^p)$ and the method has order p-1.

To prove the second part of Theorem 3, we view the solution given by (9) as an interpolant for the RK solution, evaluated at $\gamma_n \Delta t$. The order conditions for this interpolated solution are

$$\gamma_n b^T c^{k-1} - \gamma_n^k / k = 0 \qquad \qquad 1 \le k \le p \tag{16a}$$

$$\gamma_n b^T v = 0 \qquad \qquad \forall v \in V_p. \tag{16b}$$

We see that the conditions (16b) and the first bushy-tree condition ((16a) with k = 1) are still exactly fulfilled. The remaining bushy-tree conditions (16a) are not exactly satisfied; for $k \ge 2$ we have, using (14a) and (13),

$$\gamma_n(b^T c^{k-1} - \gamma_n^{k-1}/k) = \frac{\gamma_n}{k} (1 - (1 + \Delta t^{(k-1)(p-1)})) = \mathcal{O}(\Delta t^{(k-1)(p-1)}).$$

In the error expansion, each of these residuals is multiplied by $\mathcal{O}(\Delta t^k)$, so that the overall error incurred is $\mathcal{O}(\Delta t^{(k-1)p+1})$. The exponent in this expression is at least p+1 for $k \geq 2$.

It turns out that γ_n satisfies the condition required in Theorem 3.

Lemma 4. Let a_{ij}, b_j denote the coefficients of a Runge-Kutta method (6) of order p, let f be a sufficiently smooth function, and let γ_n be defined by (12). Then

$$\gamma_n = 1 + \mathcal{O}(\Delta t^{p-1}).$$



Figure 1: Convergence of γ_n to unity for various methods applied to problem (26). The observed rate is at least the expected $\mathcal{O}(\Delta t^{p-1})$ for all methods. The SSPRK(2,2) and RK(4,4) methods show convergence at one order higher than expected.

We illustrate Lemma 4 in Figure 1, which shows the convergence of γ_n to 1 as the step size is reduced for the problem (26). Due to the symmetry of the problem, even faster convergence is observed for some methods.

Remark 5. Lemma 4 was proved in [8, Proposition 4] for the special case of a certain family of four-stage, fourth-order methods.

Before proving Lemma 4 we state the main consistency result, which follows immediately from Theorem 3 and Lemma 4.

Corollary 5. Let (A, b) be the coefficients of a Runge-Kutta method of order p, and consider the RRK method defined by (6a) and (9) with γ_n defined by (12).

- (IDT method) If the solution u_{γ}^{n+1} is interpreted as an approximation to $u(t_n + \Delta t)$, the method has order p 1.
- (RRK method) If the solution u_{γ}^{n+1} is interpreted as an approximation to $u(t_n + \gamma_n \Delta t)$, the method has order p; i.e. the solution after one step satisfies

$$\|u_{\gamma}^{1} - u(\gamma_{1}\Delta t)\| = \mathcal{O}(\Delta t^{p+1}).$$

Remark 6. The first part of Corollary 5 was proved by different means in [5, Thm. 2.1 (ii)].

In order to prove Lemma 4 we use the theory of B-series and follow the notation of [13]. In the remainder of this section, t denotes a tree rather than time and $\gamma(t)$ denotes the density of a tree rather than the relaxation step length (see [13, Dfn. 2.10]). Let $\rho(t)$ denote the order of tree t, and let t' denote the tree obtained by attaching a new root node to the root of t. Let $[t_1, t_2]$ denote the tree obtained by attaching a new root node to the root of (t_1, t_2) as in [13, Dfn. 2.12]. Then

$$\rho(t') = \rho(t) + 1$$

and

$$\gamma([t_1, t_2]) = \gamma(t_1)\gamma(t_2)(\rho(t_1) + \rho(t_2) + 1).$$

Finally, let $\Phi_j(t)$ be defined as in [13, Dfn. 2.9]. We recall [13, Thm. 2.13]:

Theorem 6. A Runge–Kutta method is of order p iff

$$\sum_{j=1}^{s} b_j \Phi_j(t) = \frac{1}{\gamma(t)}$$
(17)

for all trees of order $\leq p$.

Proof of Lemma 4. We can write $\gamma = 1 - \delta/\eta$ where $\eta = \mathcal{O}(1)$ and

$$\delta = \sum_{i} b_i \sum_{j} (b_j - 2a_{ij}) \langle f_i, f_j \rangle.$$
(18)

Thus it suffices to show that

$$\sum_{i} b_i \sum_{j} (b_j - 2a_{ij}) \langle f_i, f_j \rangle = \mathcal{O}(\Delta t^{p-1}).$$

From [13, Thm. II.2.11] we have the Taylor series for the *i*th stage derivative:

$$f_i = \sum_{q=1}^{\infty} \frac{\Delta t^q}{q!} \sum_{t \in LT_q} \gamma(t) \sum_k a_{ik} \Phi_k(t) F(t)(u_0),$$

where F(t) is the elementary differential corresponding to t. Thus $\langle f_i, f_j \rangle$ can be expressed as a linear combination of inner products of elementary differentials:

$$\langle f_i, f_j \rangle = \sum_{t_1} \sum_{t_2} \frac{\Delta t^{\rho(t_1) + \rho(t_2)}}{\rho(t_1)! \rho(t_2)!} \gamma(t_1) \gamma(t_2) \beta_{ij}(t_1, t_2) \langle F(t_1)(u_0), F(t_2)(u_0) \rangle,$$

where $\rho(t)$ is the order (number of nodes) of tree t, and t_1, t_2 range over the set of all labelled rooted trees. Here

$$\beta_{ij}(t_1, t_2) = \sum_k a_{ik} \Phi_k(t_1) \sum_l a_{jl} \Phi_j(t_2) = \Phi_i(t_1') \Phi_j(t_2').$$

Because of the symmetry of the inner product, it is sufficient to show that

$$\sum_{i} b_i \sum_{j} (b_j - 2a_{ij}) (\beta_{ij}(t_1, t_2) + \beta_{ij}(t_2, t_1)) = 0$$

for all pairs of trees (t_1, t_2) satisfying

$$\rho(t_1) + \rho(t_2) \le p - 2. \tag{19}$$

We have

$$\begin{split} \sum_{i} b_{i} \sum_{j} (b_{j} - 2a_{ij}) (\beta_{ij}(t_{1}, t_{2}) + \beta_{ij}(t_{2}, t_{1})) \\ &= \sum_{i} b_{i} \sum_{j} (b_{j} - 2a_{ij}) \left(\Phi_{i}(t_{1}') \Phi_{j}(t_{2}') + \Phi_{i}(t_{2}') \Phi_{j}(t_{1}') \right) \\ &= \frac{2}{\gamma(t_{1}')\gamma(t_{2}')} - 2 \sum_{i} b_{i} \left(\Phi_{i}(t_{1}') \Phi_{i}(t_{2}'') + \Phi_{i}(t_{2}') \Phi_{i}(t_{1}'') \right) \\ &= \frac{2}{\gamma(t_{1}')\gamma(t_{2}')} - 2 \sum_{i} b_{i} \left(\Phi_{i}([t_{1}, t_{2}']) + \Phi_{i}([t_{2}, t_{1}']) \right) \\ &= \frac{2}{\gamma(t_{1}')\gamma(t_{2}')} - \frac{2}{\rho(t_{1}) + \rho(t_{2}) + 2} \left(\frac{1}{\gamma(t_{1})\gamma(t_{2}')} \frac{1}{\gamma(t_{2})\gamma(t_{1}')} \right) \\ &= \frac{2}{\gamma(t_{1}')\gamma(t_{2}')} \left(1 - \frac{1}{\rho(t_{1}) + \rho(t_{2}) + 2} \left(\frac{\gamma(t_{1}')}{\gamma(t_{1})} + \frac{\gamma(t_{2}')}{\gamma(t_{2})} \right) \right) = 0. \end{split}$$

Here we have applied (17) in various places, using the fact that our method is of order p and that (due to (19)) the trees in question all have order less than or equal to p.

Remark 7. It is possible to prove a generalization of Lemma 4 without the use of B-series; see [22]. The proof above is included here to show a direct approach that is very different from that used in [22].

2.2 Comparison with projection methods

A projection Runge–Kutta method [12, Section IV.4] consists of the traditional Runge–Kutta formula (6) followed by a projection step:

$$u_{\lambda}^{n+1} = u^{n+1} - \lambda \Phi,$$

where Φ is the projection direction and λ is chosen so that u_{λ}^{n+1} lies on a desired manifold. A relaxation Runge–Kutta method can be viewed as a projection method along the direction $\Phi = u^{n+1} - u^n$ with step length $\lambda = 1 - \gamma_n$. However, the existing formalism for such methods does not include the possibility of interpreting the new solution as an approximation at a time different from $t_n + \Delta t$. In [5], this projection perspective was applied and it was shown that the resulting method is indeed of order p - 1 when the original RK method is order p. We follow the terminology of [5] and refer to this interpretation as the *incremental direction technique*, or IDT.

To properly write a relaxation method as a projection method we can write (1) as the equivalent autonomous system of m + 1 ODEs v' = g(v) in the standard way with

$$v = \begin{bmatrix} u \\ t \end{bmatrix}, \qquad \qquad g(v) = \begin{bmatrix} f(v_{m+1}, v_{1:m}) \\ 1 \end{bmatrix}$$

where $g(v) = f(v_{m+1}, v_{1:m})$. The projection of the solution of this problem in the direction $\Phi = v^{n+1} - v^n$ requires also projecting the updated value of t to obtain the value $t_n + \gamma_n \Delta t$.

3 Stability properties of explicit relaxation RK methods

All of the results in the previous section apply to general Runge–Kutta methods. In the rest of this work, we focus on explicit methods.

An important question not explicitly answered in the foregoing analysis is how large the step size Δt can be taken in practice. Theorem 2 guarantees unconditional stability for RRK methods applied to any conservative problem, and guarantees stability for dissipative problems as long as the step size is small enough. Due to the overall explicit nature of the methods, we should not expect in either case to obtain accurate results using step sizes much larger than what the original explicit RK method allows; in this respect RRK methods behave similarly to rational Runge–Kutta methods, to which their form is also very similar [27, 11]. In this section we investigate linear stability of RRK methods and also study how the strong stability preserving (SSP) property is affected by the use of RRK methods.

The behavior of an RRK method may be quite challenging to analyze, since γ depends in a nonlinear way on the method coefficients and the numerical solution itself. On the other hand, over a single step we can think of γ as a fixed value close to unity, and study the properties of the Runge–Kutta method

$$\frac{c \mid A}{\mid \gamma b^T}.$$
(20)

3.1 The stability function

The stability function of a Runge-Kutta method is

$$R(z) = 1 + zb^T (I - zA)^{-1}e,$$

where e is vector with all entries equal to unity. Thus the stability function corresponding to one step of the relaxation method (20) is

$$R_{\gamma}(z) = 1 + z\gamma b^{T} (I - zA)^{-1} e.$$
(21)

Letting α_k denote the coefficients of R(z) for an explicit method:

$$R(z) = 1 + \sum_{k=1}^{s} \alpha_k z^k,$$

we have

$$R_{\gamma}(z) = 1 + \gamma \sum_{k=1}^{s} \alpha_k z^k.$$

Let $S(A, b) \subset \mathbb{C}$ denote the region of absolute stability of RK method (A, b). It turns out that $S(A, \gamma b)$ grows as γ decreases.

Theorem 7. Let γ_1, γ_2 be given such that $0 \leq \gamma_1 \leq \gamma_2$. Then $S(A, \gamma_2 b) \subseteq S(A, \gamma_1 b)$.

Proof. It is sufficient to show that if $|R_{\gamma_2}(z_*)| \leq 1$ for some $z_* \in \mathbb{C}$ then $|R_{\gamma_1}(z_*)| \leq 1$. Let $w = \sum_{k=1}^s \alpha_k z_*^k$; then clearly $|R_{\gamma_2}(z_*)| = |1 + \gamma_2 w| \leq 1$ implies $|R_{\gamma_1}(z_*)| = |1 + \gamma_1 w| \leq 1$. \Box

Theorem 7 is illustrated in Figure 2, which shows the stability region for relaxation RK methods with 2–4 stages and order equal to the stage number. Stability region boundaries are shown for different values of γ in the range [0.7, 1.3]. This is an exaggerated range compared to values of γ that are used in practice; for practical values of γ , the change in the stability region is visually too small to notice. It is particularly interesting to note that small reductions in γ lead to significantly enhanced stability along the imaginary axis. Note also that as $\gamma \to 0$, the RRK method tends to the identity map and $S(A, \gamma b) \to \mathbb{C}$.



Figure 2: Stability regions for RRK methods. Regions are shown for γ ranging from 0.7 to 1.3. Larger regions correspond to smaller values of γ . The stability region of the original RK method

Consideration of absolute stability along the imaginary axis leads to the so-called Epolynomial, which for explicit RK methods is

$$E(y) = 1 - |R(iy)|^2 = 1 - R(iy)R(-iy).$$

Clearly the method is stable for z = iy such that $E(y) \ge 0$. Direct calculation shows that for any RRK method of order at least p (where $p \ge 2$), we have

$$E_{\gamma}(y) = \sum_{j=1}^{p/2} \frac{2}{(2j)!} \gamma(1-\gamma) y^{2j} + \mathcal{O}(y^{p+2}).$$

The leading terms up to y^p (which vanish for a standard RK method) are positive for $0 \le \gamma < 1$, so we see that for every method of order two or higher $S(A, \gamma b)$ includes a segment of the imaginary axis containing z = 0 when $\gamma \in [0, 1)$. This can be observed for instance in Figure 2a, where the RK method is unstable over the whole imaginary axis, but for $\gamma < 1$ the stability regions include part of the imaginary axis.

3.2 Strong stability preservation

(corresponding to $\gamma = 1$) is outlined in black.

Because the RRK method is only a small perturbation of the original RK method, desirable properties of the original method may remain in effect when using the RRK version. We illustrate this idea by studying strong stability preserving RRK methods.

In the following, C(A, b) denotes the SSP coefficient (or radius of absolute monotonicity) of the Runge–Kutta method with coefficients (A, b). Recall that C(A, b) is equal to the largest value $r \geq 0$ such that the method (A, b) is absolutely monotonic at -r [9].

Method	γ_*
SSPRK(2,2)	2
SSPRK(s,2)	s/(s-1)
SSPRK(3,3)	3/2
SSPRK(4,3)	1
SSPRK(5,3)	1
SSPRK(9,3)	1
SSPRK(5,4)	1.312
SSPRK(10,4)	25/24

Table 1: Values of γ_* from Theorem 9 for some well-known SSP methods. For $0 \leq \gamma \leq \gamma_*$, the relaxation method has the same SSP coefficient as the original method.

For $0 \leq \gamma \leq 1$, u_{γ}^{n+1} is a convex combination of u^n and u^{n+1} , which implies that $\mathcal{C}(A, \gamma b) \geq \mathcal{C}(A, b)$ for $0 \leq \gamma \leq 1$. For many methods, $\mathcal{C}(A, \gamma b)$ also does not decrease when γ is taken a little larger than 1.

Lemma 8. Let the RK method with coefficients (A, b) be absolutely monotonic at z = -r. Then the method $(A, \gamma b)$ with $\gamma \geq 0$ is also absolutely monotonic at z = -r iff $R_{\gamma}(-r) \geq 0$.

Proof. The conditions for absolute monotonicity of method (A, b) at z can be written (see, e.g. [9, p. 211]

$$A(I - zA)^{-1} \ge 0 \qquad (I - zA)^{-1}e \ge 0 \qquad (22a)$$

$$b^{T}(I - zA)^{-1} \ge 0 \qquad R(z) \ge 0. \qquad (22b)$$

$$(-zA)^{-1} \ge 0$$
 $R(z) \ge 0.$ (22b)

Conditions (22a) do not depend on b, while the first condition of (22b) will hold for any positive multiple of b if it holds for b.

Theorem 9. Given a RK method with coefficients (A, b), we have $\mathcal{C}(A, \gamma b) \geq \mathcal{C}(A, b)$ for $0 \leq 1$ $\gamma \leq \gamma_*, where$

$$\gamma_* = -\frac{1}{\sum_k \alpha_k (-\mathcal{C}(A, b))^k} = \frac{-1}{R(-\mathcal{C}) - 1} \ge 1.$$

Proof. Combining Lemma 8 with a standard result on absolute monotonicity (see e.g. [16, Lemma 3.1) we have that $(A, \gamma b)$ is absolutely monotonic on $[-\mathcal{C}(A, b), 0]$ as long as $R_{\gamma}(-\mathcal{C}(A, b)) \geq 0$. Since $R_{\gamma}(z) - 1 = \gamma(R(z) - 1)$, we obtain the condition stated in the theorem.

Values of γ_* are given in Table 1 for some well-known SSP methods. For all of these methods, direct computation shows that $\mathcal{C}(A, \gamma b) = \mathcal{C}(A, b)$ for $0 \leq \gamma \leq \gamma_*$, while taking $\gamma > \gamma_*$ leads to a decrease in the SSP coefficient.

4 Numerical examples

For conservative systems, stability is guaranteed under any step size. However, one expects the accuracy to deteriorate significantly if γ is not close to unity. In the dissipative case, it is possible that γ becomes negative and stability is lost. A careful examination of the analysis in the previous sections suggests that step sizes on the order of what linearized stability analysis predicts to be stable should be acceptable.

The following Runge–Kutta methods will be used in the numerical experiments.

- SSPRK(2,2): Two-stage, second-order SSP method of [23].
- SSPRK(3,3): Three-stage, third-order SSP method of [23].
- SSPRK(10,4): Ten-stage, fourth-order SSP method of [15].
- RK(4,4): Classical four-stage, fourth-order method.
- BSRK(8,5): Eight-stage, fifth-order method of [2]. A fixed step size Δt is used and the embedded method is not used.

All of these methods have non-negative weights. We refer to the relaxation version of a method by replacing "RK" with "RRK"; e.g. RRK(4,4).

4.1 Linear, skew-Hermitian system

With this example we investigate the behavior of RRK methods for linear skew-Hermitian problems. For concreteness, we consider the advection equation

$$U_t = U_x$$
 $U(x, 0) = U^0(x)$ (23)

with periodic boundary conditions over a spatial interval of length 2π discretized in space by a Fourier spectral collocation method with m points. This results in a linear, constant-coefficient system of ODEs:

$$u'(t) = Du(t)$$

where $u_j(t) \approx U(x_j, t)$ at evenly spaced points x_j and D is the $m \times m$ skew-Hermitian Fourier spectral differentiation matrix. Since D is normal, the behavior of any RK method on this problem can be characterized simply in terms of the eigenvalues of D and the corresponding eigenvectors, which are discrete Fourier modes:

$$\lambda_{\xi} = i\xi \qquad \xi = -\frac{m}{2}, -\frac{m}{2} + 1, \dots, \frac{m}{2} - 1$$
$$v_{\xi} = [\exp(i\xi x_1), \exp(i\xi x_2), \dots, \exp(i\xi x_m)]^T.$$

Let us express the initial data in terms of these modes:

$$u^0 = \sum_{\xi} \hat{u}^0_{\xi} v_{\xi},$$

where \hat{u} is the discrete Fourier transform (DFT) of u. Then the exact solution of the semi-discrete system is given by

$$u(t) = \sum_{\xi} e^{i\xi t} \hat{u}^0_{\xi} v_{\xi}.$$

Thus the energy associated with each mode is constant in time. Applying a Runge–Kutta method we instead obtain the solution

$$u^n = \sum_{\xi} R(i\xi\Delta t)^n \hat{u}^0_{\xi} v_{\xi}.$$

where R(z) is again the stability function of the method. The energy associated with mode ξ is modified by the factor $|R(i\xi\Delta t)|$ at each step. The maximum stable step size is the value that guarantees $i\xi\Delta t \in S(A, b)$ for all ξ . This is just

$$\Delta t_{\max} = \frac{I(A,b)}{\max_{\xi} |\lambda_{\xi}|} = \frac{2}{m} I(A,b),$$

where I(A, b) is the length of the method's imaginary axis stability interval. In the following experiments, we use a step size $\Delta t = \mu \Delta t_{\text{max}}$. Using a given (standard) RK method and $0 \le \mu \le 1$, we have absolute stability and the energy of each mode decays. This is illustrated in Figure 3a, where we solve with m = 128 and plot the relative change in amplitude

$$\frac{|\hat{u}_{\xi}^{n}| - |\hat{u}_{\xi}^{0}|}{|\hat{u}_{\xi}^{0}|}$$

for each mode, for a range of time step sizes $\mu \Delta t_{\text{max}}$. Due to symmetry we plot only the positive wavenumbers. This figure does not depend on the initial data but only on $|R(iy)|^N$, where N is the total number of time steps taken. For larger values of μ , the high wavenumber modes are strongly damped.

Now let us consider what happens when applying a relaxation Runge–Kutta method with γ_n chosen according to (12) so that energy is conserved. At each step, the energy in mode ξ is modified by the factor $|R_{\gamma}(i\xi\Delta t)|$. If the initial data is chosen to consist of a single wavenumber k_{ξ} , then γ_n will take a value such that $|R_{\gamma}(i\xi\Delta t)| = 1$, and the same value of γ_n will be used at every step. For more general initial data, γ_n is chosen precisely so that the change in energy when summed over all modes is zero. This value depends on the data, so γ_n will be different at each step. Furthermore, at each step some modes will be diminished while others will grow. This is illustrated in Figure 3b, which is analogous to Figure 3a but for the energy-conserving RRK(4,4) method, with initial data taken as white noise; i.e. $\hat{u}_j^0 = e^{i\theta_j}$ where the phases θ_j are random. We see that high-wavenumber modes are again damped, especially for step sizes close to the stability limit. Meanwhile, some lower-wavenumber modes are amplified in order to preserve the total energy. If we instead take initial data that is reasonably well-resolved, such as

$$U^{0}(x) = \operatorname{sech}^{2}(7.5(x+1)).$$
(24)

the resulting amplification curves are nearly indistinguishable from those of the standard RK4 method (Figure 3a), as shown in Figure 3c. This is because most of the energy is in the low-wavenumber modes, which are propagated fairly accurately by the standard RK4 method, so little compensation is need in order to conserve energy. In the latter figure we also plot the amplification for $\mu = 1.02$, which is beyond the absolute stability limit. We see that in this case the RRK method greatly amplifies the highest-wavenumber mode.

In Figure 4, we compare the behavior of RK(4,4) and relaxation RK(4,4) (RRK(4,4)) for different values of μ . For $\mu < 1$, the methods give similar solutions. Unlike RK(4,4), RRK(4,4) remains stable for $\mu > 1$. However, taking $\mu = 1.016$ leads to highly inaccurate and oscillatory approximations.

Remark 8. Even for the linearly unstable value $\mu = 1.016$, we have found that γ remains within less than 10^{-2} of unity and the solution never blows up. For much larger values ($\mu \gtrsim 1.25$) the value of γ tends to zero after a few steps and the calculation is never completed.

4.2 A non-normal linear autonomous problem

Here we consider a linear autonomous problem u'(t) = Au(t) with non-normal right-hand side introduced by Sun & Shu [24]:

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}' = \begin{bmatrix} -1 & -2 & -2 \\ 0 & -1 & -2 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}.$$
 (25)



Figure 3: Relative amplification of each mode for the spectral semi-discretization of the advection equation, integrated up to t = 1, for step sizes $\Delta t = \mu \Delta t_{\text{max}}$.



(a) RK(4,4) vs. RRK(4,4); $\mu = 0.99$.

(b) RRK(4,4) with two slightly different step sizes

Figure 4: Solutions to the advection equation at $t = 400\pi$, using standard RK(4,4) and the energy-conserving relaxation modification of RK(4,4). The absolute stability limit corresponds to $\mu = 1$. The two methods give very similar solutions for all stable step sizes. The RK(4,4) solution blows up for $\mu > 1$; the RRK(4,4) solution is stable for all step sizes but becomes highly oscillatory for $\mu > 1.015$.



Figure 5: Standard RK(4,4) and its energy-conserving modification RRK(4,4) applied to the test problem of Sun & Shu[24]. The RK(4,4) solution gives an increased energy at the first step, while the RRK(4,4) solution is monotone.

This problem is dissipative, but — as shown in [24] — the classical 4th-order Runge–Kutta method RK(4,4) is not monotone for this problem, no matter how small one takes Δt . To provide a concrete example, we compute $R(\Delta tA)$, where R(z) is the stability polynomial of RK(4,4) and we choose step size $\Delta t = 0.5$. The first singular value of the matrix $R(\Delta tA)$ is approximately 1.001. Taking a single step with RK(4,4) and initial condition equal to the first right singular vector thus leads to an increase in the energy, as shown in Figure 5. The increase is even larger for the step size 0.7. In contrast, RRK(4,4) preserves monotonicity with either step size.

4.3 Nonlinear oscillator

Here we consider the problem

$$\begin{bmatrix} u_1\\u_2 \end{bmatrix}' = \frac{1}{\|u\|^2} \begin{bmatrix} -u_2\\u_1 \end{bmatrix}, \quad \begin{bmatrix} u_1(0)\\u_2(0) \end{bmatrix} = \begin{bmatrix} 1\\0 \end{bmatrix},$$
(26)

with analytical solution

$$\begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} = \begin{bmatrix} \cos(t) \\ \sin(t) \end{bmatrix}.$$

Although energy is conserved in the exact solution, the widely used SSPRK(3,3) method of Shu & Osher [23] produces a solution whose energy is monotonically increasing for every positive step size [19]. Similar behavior is observed for several other explicit RK methods we have tested, as shown in Figure 6a. By applying the modification described in the present work, we obtain instead Figure 6b, showing that energy is conserved up to roundoff error for all methods.

Figure 7 shows the convergence for each of the standard RK methods (solid lines) and its energy-conserving modification (dashed lines). We see that the relaxation method is in each case on par with or more accurate than the original method.

4.4 Burgers' equation

We solve the inviscid Burgers' equation

$$U_t + \frac{1}{2}(U^2)_x = 0 \tag{27}$$



Figure 6: Evolution of the energy in the numerical solution of (26). Standard RK methods exhibit energy growth or decay, while relaxation methods conserve energy to within roundoff error. Note that the scale of the left figure is logarithmic, while the scale of the right figure is linear.



Figure 7: Convergence of several methods and their energy-conserving relaxation variants on the test problem (26). Solid lines are standard RK methods; dashed lines are relaxation RK methods.



Figure 8: Evolution of the energy in the numerical solution of Burgers' equation (27) with energyconservative flux (28), using $\Delta t = 0.3\Delta x$. Standard RK methods exhibit energy growth or decay, while relaxation methods conserve energy to within roundoff error. Note that the vertical scale of the left figure is logarithmic, while the scale of the right figure is linear.

on the interval $-1 \le x \le 1$ with periodic boundary conditions and initial data

$$U(x,0) = \exp(-30x^2)$$

with the flux-differencing discretization

$$u_i'(t) = -\frac{1}{\Delta x}(F_{i+1/2} - F_{i-1/2})$$

where $F_{i\pm 1/2}$ is the numerical flux, defined below. The spatial domain is discretized with 50 equally-spaced points. In the convergence tests below, this spatial discretization is held fixed while the time step is varied, in order to investigate only the temporal convergence.

4.4.1 Energy-conservative semi-discretization

We take the second-order accurate symmetric flux [26]

$$F_{i+1/2} = \frac{u_i^2 + u_i u_{i+1} + u_{i+1}^2}{6} \tag{28}$$

which yields a conservative semi-discrete system. Because of the lack of numerical viscosity, the semi-discrete solution is not the vanishing-viscosity solution, but instead develops a dispersive shock.

Energy evolution up to t = 0.2 is shown in Figures 8. As expected, standard RK methods all exhibit significant growth or dissipation of energy, while RRK methods preserve energy up to roundoff error. Convergence results at t = 0.03 (just before shock formation) are shown in Figure 9, where we compare the IDT approach with the full relaxation approach. All RRK methods achieve the order of accuracy of the corresponding RK method, whereas for IDT methods the convergence rate is reduced by one, as predicted by Theorem 3.



Figure 9: Convergence for Burgers' equation with energy-conservative flux, using several standard RK methods (solid lines) and their energy-conserving modifications (dashed lines). The value of Δx is fixed and the solution is computed at t = 0.03, just before the time of shock formation. Without step size rescaling, the rate of convergence is reduced by one in some cases as indicated in Theorem 3. With step size rescaling, in most cases the standard and relaxed RK methods give almost exactly the same accuracy.



Figure 10: Evolution of the energy in the numerical solution of Burgers' equation (27) with energy-dissipative flux (29), with a $\Delta t = 0.2\Delta x$. Standard RK methods (left) exhibit excessive dissipation, due to numerical errors. Relaxation methods (right) approximate the correct energy evolution much more accurately.

4.4.2 Energy-dissipative semi-discretization

We obtain a dissipative system by adding a centered difference to the flux:

$$F_{i+1/2} = \frac{u_i^2 + u_i u_{i+1} + u_{i+1}^2}{6} - \epsilon(u_{i+1} - u_i).$$
⁽²⁹⁾

The amount of dissipation is controlled by $\epsilon > 0$. The scheme is still consistent with (27) since the amount of dissipation is proportional to Δx . We take $\epsilon = 1/100$. With this dissipative flux, the solution develops a viscous shock.

Results are shown in Figures 10 and 11. What is most interesting is that applying the relaxation approach dramatically improves the numerical approximation of the global dissipation. Convergence results are similar to those obtained with the conservative flux.

5 Conclusions

The relaxation approach we have proposed seems to be a simple and effective way to make any Runge-Kutta method preserve conservation or dissipativity with respect to an inner-product norm. This can be extended to more general convex functionals; see [22]. While we have focused here on explicit methods, the technique applies to implicit methods as well. Like Runge-Kutta methods, relaxation Runge-Kutta methods automatically preserve linear first integrals. If the original method is equipped with an embedded error estimator or dense output formula, these can be used without modification (other than the rescaling of the step size when determining the dense output times). As we have shown, many SSP RK methods retain the same SSP coefficient when used as RRK methods. The linear stability properties of a method are slightly modified depending on the choice of γ , but in general the allowable step size for an RRK method is essentially the same as that allowed for the original RK method.

Together these properties make RRK methods an attractive choice for symmetric hyperbolic systems; in combination with entropy-stable spatial discretizations they give a fully-discrete, explicit scheme that is provably entropy stable (for quadratic entropies).



Figure 11: Convergence for Burgers' equation with energy-dissipative flux, using several standard RK methods (solid lines) and their energy-conserving modifications (dashed lines). The value of Δx is fixed. The IDT methods (i.e., relaxation but without step size rescaling) give a rate of convergence that is reduced by one in some cases as indicated in Theorem 3. With step size rescaling, in most cases the standard and relaxed RK methods give similar accuracy.

In several experiments we have compared results obtained by viewing u_{γ}^{n+1} as an approximation to $u(t_n + \Delta t)$ (the so-called incremental direction technique, or IDT) versus those obtained by viewing it as an approximation to $u(t_n + \gamma_n \Delta t)$ (the relaxation approach). The main purpose of this comparison is to illustrate our theoretical convergence estimates; it is clear that in practice one should always use the latter interpretation. Comparison with embedded projection methods is planned as future work.

For an RK method with s stages, determination of γ_n via the formula (12) requires the evaluation of s + 1 inner products (see Remark 4). For typical high-order discretizations of nonlinear PDEs this cost is negligible compared to the s evaluations of f required for each step. For simpler applications (such as linear wave equations) the cost may be an important factor. A careful comparison of RRK schemes relative to other time discretizations for linear wave equations is the subject of ongoing work.

6 Acknowledgments

The author is grateful to Hendrik Ranocha for helpful comments on drafts of this work. This work was supported by funding from King Abdullah University of Science & Technology.

References

- Peter Albrecht. The Runge-Kutta Theory in a Nutshell. SIAM Journal on Numerical Analysis, 33(5):1712 – 1735, 1996.
- [2] P Bogacki and Lawrence F Shampine. An efficient Runge-Kutta (4, 5) pair. Computers & Mathematics with Applications, 32(6):15–28, 1996.
- [3] Kevin Burrage and John C Butcher. Stability criteria for implicit Runge-Kutta methods. SIAM Journal on Numerical Analysis, 16(1):46-57, 1979.
- [4] M Calvo, MP Laburta, JI Montijano, and L Rández. Projection methods preserving Lyapunov functions. BIT Numerical Mathematics, 50(2):223-241, 2010.
- [5] Manuel Calvo, D Hernández-Abreu, Juan I Montijano, and Luis Rández. On the preservation of invariants by explicit Runge-Kutta methods. SIAM Journal on Scientific Computing, 28(3):868–885, 2006.
- [6] Germund Dahlquist and Rolf Jeltsch. Reducibility and contractivity of Runge-Kutta methods revisited. *Bit Numerical Mathematics*, 46:567–587, 2006.
- [7] K. Dekker and J. G. Verwer. Stability of Runge-Kutta methods for stiff nonlinear differential equations, volume 2 of CWI Monographs. North-Holland Publishing Co., 1984.
- [8] N Del Buono and C Mastroserio. Explicit methods based on a class of four stage fourth order Runge-Kutta methods for preserving quadratic laws. *Journal of Computational and Applied Mathematics*, 140(1-2):231–243, 2002.
- [9] Luca Ferracina and M. N. Spijker. An extension and analysis of the Shu-Osher representation of Runge-Kutta methods. *Mathematics of Computation*, 249:201–219, 2005.
- [10] V. Grimm and G. R. W. Quispel. Geometric integration methods that preserve lyapunov functions. BIT Numerical Mathematics, 45(4):709–723, Dec 2005.

- [11] Ernst Hairer. Unconditionally stable explicit methods for parabolic equations. Numerische Mathematik, 35(1):57–68, 1980.
- [12] Ernst Hairer, Christian Lubich, and Gerhard Wanner. Geometric numerical integration: structure-preserving algorithms for ordinary differential equations, volume 31. Springer Science & Business Media, 2006.
- [13] Ernst Hairer, Syvert P. Nø rsett, and G. Wanner. Solving ordinary differential equations {I}: Nonstiff Problems. Springer Series in Computational Mathematics. Springer, Berlin, second edition, 1993.
- [14] Inmaculada Higueras. Monotonicity for Runge-Kutta methods: inner product norms. Journal of Scientific Computing, 24:97–117, 2005.
- [15] David I Ketcheson. Highly efficient strong stability-preserving Runge-Kutta methods with low-storage implementations. SIAM Journal on Scientific Computing, 30(4):2113–2136, 2008.
- [16] J. F. B. M. Kraaijevanger. Contractivity of Runge-Kutta Methods. BIT Numerical Mathematics, 31:482–528, 1991.
- [17] MP Laburta, Juan I Montijano, Luis Rández, and Manuel Calvo. Numerical methods for non conservative perturbations of conservative problems. *Computer Physics Communications*, 187:72–82, 2015.
- [18] Carlos Lozano. Entropy production by explicit Runge-Kutta schemes. Journal of Scientific Computing, 76(1):521–565, 2018.
- [19] Hendrik Ranocha. On strong stability of explicit Runge-Kutta methods for nonlinear semibounded operators, Nov. 2018.
- [20] Hendrik Ranocha. Private communication, 2019.
- [21] Hendrik Ranocha and Philipp Öffner. L₂ stability of explicit Runge–Kutta schemes. Journal of Scientific Computing, pages 1–17, 2018.
- [22] Hendrik Ranocha, Mohammed Sayyari, Lisandro Dalcin, Matteo Parsani, and David I. Ketcheson. Relaxation Runge–Kutta methods: Fully-discrete explicit entropy-stable schemes for the Euler and Navier–Stokes equations, 05 2019. To appear on arXiv.
- [23] Chi-Wang Shu and Stanley Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics*, 77(2):439–471, August 1988.
- [24] Zheng Sun and Chi-Wang Shu. Stability of the fourth order Runge–Kutta method for timedependent partial differential equations. Annals of Mathematical Sciences and Applications, 2(2):255–284, 2017.
- [25] Zheng Sun and Chi-Wang Shu. Strong stability of explicit Runge-Kutta time discretizations. arXiv preprint arXiv:1811.10680, 2018.
- [26] Eitan Tadmor. Entropy stability theory for difference approximations of nonlinear conservation laws and related time-dependent problems. Acta Numerica, 12:451–512, 2003.
- [27] Albert Wambecq. Rational Runge-Kutta methods for solving systems of ordinary differential equations. Computing, 20(4):333–342, 1978.