Relaxation Runge-Kutta Methods and Entropy Stability

Thomas Renström

April 26, 2023

1 Introduction

For certain simulations it is of utmost importance to use methods that follow conservation laws, making sure that the model stays true to physical laws. While there are already established methods for this, they are not as well known nor as simple to set up as the Runge-Kutta method. We will present the idea put forward by Ranocha et al. in [4] and attempt to replicate their result. We shall see that not only is the modification suggested in the article by both simple and cheap, it is also accurate. As this report is based on [4] we concern ourself mainly with the statements made there. A few of their references, those deemed as significant by this author, will be repeated here.

2 Preliminaries

2.1 ODE

We consider a time-dependent ODE of the form

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}u(t) &= f(t,u(t)), \quad t \in (0,T) \\ u(0) &= u^0 \end{aligned}$$

for $u \in \mathcal{H}$ being a real Hilbert space with the inner product $\langle \cdot, \cdot \rangle$, inducing the norm $\|\cdot\|$.

2.2 Entropy

We denote by

$$\eta : \mathcal{H} \to \mathbb{R}$$

a smooth convex function in time. We call this entropy, while in other application it might instead represent some other form of non-increasing quantity, e.g. energy or momentum.

The change in entropy over time is given by

$$\frac{\mathrm{d}}{\mathrm{d}t}\eta(u(t)) = \langle \eta'(u(t)), f(t, u(t))\rangle .$$

A entropy dissipative system will satisfy

$$\langle \eta'(u(t)), f(t, u(t))\rangle \leq 0, \quad \forall u \in \mathcal{H}, t \in [0, T],$$

and a entropy conservative system will satisfy

$$\langle \eta'(u(t)), f(t, u(t))\rangle = 0, \quad \forall u \in \mathcal{H}, t \in [0, T].$$

2.3 Classic Runge-Kutta

A general Runge-Kutta method with s stages is represented with the Butcher tableau

$$\begin{array}{c|c} c & A \\ \hline & b^{\mathrm{T}} \end{array},$$

with $A \in \mathbb{R}^{s\times s}$ and $b, c \in \mathbb{R}^s$.

The scheme of the method is

$$y_i = u^n + \Delta t \sum_{j=1}^{s} a_{i,j} f(t_n + c_j\Delta t, y_j), \quad i = 1, \ldots, s \tag{1}$$

$$u^{n+1} = u^n + \Delta t \sum_{i=1}^{s} b_i f(t_n + c_i\Delta t, y_i). \tag{2}$$

where y_i are the stage values.

For brevity we will use the notation

$$f_i := f(t_n + c_i\Delta t, y_i), \qquad f_0 := f(t_n, u^n).$$

3 Relaxation Runge-Kutta Methods

3.1 Relaxation Runge-Kutta

The basic idea of the Relaxed Runge-Kutta method is to introduce a scaling factor to the weights b_i. We call this scaling factor $\gamma_n \in \mathbb{R}$ and construct a new scheme

$$u_\gamma^{n+1} = u^n + \gamma_n \Delta t \sum_{i=1}^{s} b_i f_i,$$

with $\gamma_n \in \mathbb{R}$.

Relaxed Runge-Kutta, RRK, interprets $u_\gamma^{n+1} \approx u(t_n + \gamma \Delta t)$. This means that using this method we will not be using a uniform step length and the number of steps might not be the same as with the regular RK method.

The incremental direction technique, or IDT-method, interprets $u_\gamma^{n+1} \approx u(t_n + \Delta t)$. This method retains the original step length, and takes as many steps to complete as the regular RK method it is based on.

In [3] it was proposed choosing γ_n such that

$$\frac{\left\|u_\gamma^{n+1}\right\|^2 - \|u^n\|^2}{2} = \gamma_n \Delta t \sum_{i=1}^{s} b_i \langle y_i, f_i \rangle .$$

In this article however the suggestion is to instead use γ_n fulfilling the condition

$$\eta(u_\gamma^{n+1}) - \eta(u^n) = \gamma \Delta t \sum_{i=1}^{s} b_i \langle \eta'(y_i), f_i \rangle .$$

This is done by finding a root of

$$r(\gamma) = \eta(u^n + \gamma_n \Delta t \sum_{i=1}^{s} b_i f_i) - \eta(u^n) - \gamma \Delta t \sum_{i=1}^{s} b_i \langle \eta'(y_i), f_i \rangle . \tag{3}$$

The direction and entropy change

$$\begin{aligned} d^n &:= \sum_{i=1}^{s} b_i f_i \\ e &:= \Delta t \sum_{i=1}^{s} b_i \langle \eta'(y_i), f_i \rangle \end{aligned}$$

can both be computed on the fly during the RK method. This reduces finding the root of r to just a scalar root finding problem.

Note that since η is convex we have that r is convex in γ.

We note also that since

$$r(0) = \eta(u^n + 0) - \eta(u^n) - 0 = 0 \tag{4}$$

r has a root at 0.

Using a non-positive γ would not be feasible. A zero-valued γ would halt the scheme, while a negative γ would amount to steps backwards in time.

3.2 Existence of a solution

In order to show that r has a positive root we need two things. Firstly that r' is negative at the root $\gamma = 0$ and that r' is positive at some point $\gamma > 0$.

Lemma 3.1. *Let a Runge-Kutta method be given such that $\sum_{i=1}^{s} b_i a_{i,j} > 0$. If $n''(u^n)(f_0, f_0) > 0$ then $r'(0) < 0$ for sufficiently small $\Delta t > 0$.*

Note that $\sum_{i=1}^{s} b_i a_{i,j} > 0$ is a reasonable assumption, since $\sum_{i=1}^{s} b_i a_{i,j} = 1/2$ is a condition for second-order accuracy.

Proof. By the definition of $r(\gamma)$ in Equation 3,

$$\frac{dr}{d\gamma} = \eta'(u^n + \gamma \Delta t \sum_{i=1}^{s} b_i f_i) \cdot \left(\Delta t \sum_{i=1}^{s} b_i f_i \right) - \Delta t \sum_{i=1}^{s} b_i \left\langle \eta'(y_i), f_i \right\rangle .$$

Evaluating $r'(\gamma)$ at $\gamma = 0$ we get

$$\begin{aligned} r'(0) &= \eta'(u^n) \cdot \left(\Delta t \sum_{i=1}^{s} b_i f_i \right) - \Delta t \sum_{i=1}^{s} b_i \left\langle \eta'(y_i), f_i \right\rangle \\ &= \Delta t \sum_{i=1}^{s} b_i \left(\left\langle \eta'(u^n), f_i \right\rangle - \left\langle \eta'(y_i), f_i \right\rangle \right) \\ &= -\Delta t \sum_{i=1}^{s} b_i \left(\left\langle \eta'(y_i), f_i \right\rangle - \left\langle \eta'(u^n), f_i \right\rangle \right) . \end{aligned}$$

We expand $y_i = u^n + \Delta t \sum_{j=1}^{s} a_{i,j} f_j$ as defined in Equation 1 and get

$$r'(0) = -\Delta t \sum_{i=1}^{s} b_i \left(\left\langle \eta' \left(u^n + \Delta t \sum_{j=1}^{s} a_{i,j} f_j \right), f_i \right\rangle - \left\langle \eta'(u^n), f_i \right\rangle \right) .$$

Then, by the fundamental theorem of calculus,

$$\begin{aligned} r'(0) &= -\Delta t \sum_{i=1}^{s} b_i \int_0^1 \eta'' \left(u^n + v \Delta t \sum_{k=1}^{s} a_{i,k} f_k \right) \left(f_i, \Delta t \sum_{j=1}^{s} a_{i,j} f_j \right) \mathrm{d}v \\ &= -\Delta t^2 \sum_{i,j=1}^{s} b_i a_{i,j} \int_0^1 \eta'' \left(u^n + v \Delta t \sum_{k=1}^{s} a_{i,k} f_k \right) (f_i, f_j) \, \mathrm{d}v. \end{aligned}$$

With Taylor expansions of $f_i, f_j = f_0 + \mathcal{O}(\Delta t)$,

$$\begin{aligned} r'(0) &= -\Delta t^2 \sum_{i,j=1}^{s} b_i a_{i,j} \int_0^1 \eta'' \left(u^n + v \Delta t \sum_{k=1}^{s} a_{i,k} f_k \right) (f_0, f_0 + \mathcal{O}(\Delta t)) \, \mathrm{d}v \\ &= -\Delta t^2 \sum_{i,j=1}^{s} b_i a_{i,j} \int_0^1 \eta'' \left(u^n + v \Delta t \sum_{k=1}^{s} a_{i,k} f_k \right) (f_0, f_0) \, \mathrm{d}v + \mathcal{O}(\Delta t^3) \end{aligned}$$

For sufficiently small Δt we have that the integral is positive since $\eta''(u^n)(f_0, f_0) > 0$ and thus, with the other assumptions, $r'(0) < 0$. □

Lemma 3.2. *Let a Runge-Kutta method be given such that $\sum_{i,j=1}^s b_i(a_{i,j} - b_j) < 0$. If $\eta''(u^n)(f_0, f_0) > 0$ then $r'(1) > 0$ for sufficiently small $\Delta t > 0$.*

Note that the assumption $\sum_{i,j=1}^s b_i(a_{i,j} - b_j) < 0$ is reasonable. Since $\sum_{i,j=1}^s b_i b_j = 1$ and $\sum_{i=1}^s b_i a_{i,j} = 1/2$ is a condition for second-order accuracy we have that $\sum_{i,j=1}^s b_i(a_{i,j} - b_j) = -1/2$

Proof. By the definition of $r(\gamma)$ in Equation 3,

$$\frac{dr}{d\gamma} = \eta'(u^n + \gamma \Delta t \sum_{i=1}^s b_i f_i) \cdot \left(\Delta t \sum_{i=1}^s b_i f_i \right) - \Delta t \sum_{i=1}^s b_i \left\langle \eta'(y_i), f_i \right\rangle .$$

Evaluating $r'(\gamma)$ at $\gamma = 1$ we get

$$\begin{aligned} r'(1) &= \eta'(u^n + \Delta t \sum_{j=1}^s b_j f_j) \cdot \left(\Delta t \sum_{i=1}^s b_i f_i \right) - \Delta t \sum_{i=1}^s b_i \left\langle \eta'(y_i), f_i \right\rangle \\ &= \Delta t \sum_{i=1}^s b_i \left(\left\langle \eta'(u^n + \Delta t \sum_{j=1}^s b_j f_j), f_i \right\rangle - \left\langle \eta'(y_i), f_i \right\rangle \right) \\ &= -\Delta t \sum_{i=1}^s b_i \left(\left\langle \eta'(y_i), f_i \right\rangle - \left\langle \eta'(u^n + \Delta t \sum_{j=1}^s b_j f_j), f_i \right\rangle \right) \end{aligned}$$

Expanding $y_i = u^n + \Delta t \sum_{j=1}^s a_{i,j} f_j$ as defined in Equation 1 we get

$$r'(1) = -\Delta t \sum_{i=1}^s b_i \left(\left\langle \eta'(u^n + \Delta t \sum_{j=1}^s a_{i,j} f_j), f_i \right\rangle - \left\langle \eta'(u^n + \Delta t \sum_{j=1}^s b_j f_j), f_i \right\rangle \right) .$$

We then make substitutions according to Equation 2, $u^{n+1} = u^n + \Delta t \sum_{j=1}^s b_j f_j$,

$$r'(1) = -\Delta t \sum_{i=1}^s b_i \left(\left\langle \eta' \left(u^{n+1} + \Delta t \sum_{j=1}^s (a_{i,j} - b_j) f_j \right), f_i \right\rangle - \left\langle \eta'(u^{n+1}), f_i \right\rangle \right) .$$

Then, by the fundamental theorem of calculus,

$$r'(1) = -\Delta t \sum_{i=1}^s b_i \int_0^1 \eta'' \left(u^{n+1} + \Delta t \sum_{k=1}^s (a_{i,k} - b_k) f_k \right) \left(f_i, \Delta t \sum_{j=1}^s (a_{i,j} - b_j) f_j \right) \mathrm{d}v$$

$$r'(1) = -\Delta t^2 \sum_{i,j=1}^{s} b_i \left(a_{i,j} - b_j\right) \int_0^1 \eta'' \left(u^{n+1} + \Delta t \sum_{k=1}^{s} \left(a_{i,k} - b_k\right) f_k\right) \left(f_i, f_j\right) \mathrm{d}v$$

With Taylor expansions of $f_i, f_j = f_0 + \mathcal{O}(\Delta t)$,

$$r'(1) = -\Delta t^2 \sum_{i,j=1}^{s} b_i \left(a_{i,j} - b_j\right) \int_0^1 \eta'' \left(u^{n+1} + \Delta t \sum_{k=1}^{s} \left(a_{i,k} - b_k\right) f_k\right) \left(f_0, f_0 + \mathcal{O}(\Delta t)\right) \mathrm{d}v$$

$$r'(1) = -\Delta t^2 \sum_{i,j=1}^{s} b_i \left(a_{i,j} - b_j\right) \int_0^1 \eta'' \left(u^{n+1} + \Delta t \sum_{k=1}^{s} \left(a_{i,k} - b_k\right) f_k\right) \left(f_0, f_0\right) \mathrm{d}v + \mathcal{O}(\Delta t^3)$$

For sufficiently small Δt we have that the integral is positive since $\eta''(u^n)(f_0, f_0) > 0$ and thus with the other assumptions, $r'(1) > 0$. □

Theorem 3.3. *Assume that the Runge-Kutta method satisfies $\sum_{i,j=1}^{s} b_i a_{i,j} > 0$ and $\sum_{i,j=1}^{s} b_i(a_{i,j} - b_j) < 0$. If $\eta''(u^n)(f_0, f_0) > 0$ then r has a positive root for sufficiently small $\Delta t > 0$.*

Proof. Since $r(0) = 0$ and $r'(0) < 0$ we have that $r(\gamma) < 0$ for small $\gamma > 0$. Because $r'(1) > 0$ and r is convex we have that r' is monotone. Thus, there must be a positive root of r. □

3.3 Accuracy

The following theorem is presented in [4] without a proof with a reference to [3]. We will proceed, accepting it at face value.

Theorem 3.4. *Let a given RK-method be of order p. Consider the IDT and RRK methods based on them and suppose that $\gamma_n = 1 + \mathcal{O}(\Delta t^{p-1})$, then*

1. *The IDT method interpreting $u_\gamma^{n+1} \approx u(t_n + \Delta t)$ has order $p - 1$.*
2. *The RRK method interpreting $u_\gamma^{n+1} \approx u(t_n + \gamma\Delta t)$ has order p.*

Thus, if we can show that $\gamma_n = 1 + \mathcal{O}(\Delta t^{p-1})$, then our method is accurate.

Setting $u^n = u(t_n)$ a Runge-Kutta method of order p gives

$$\begin{aligned}\eta(u^{n+1}) - \eta(u^n) &= \eta(u(t_n + \Delta t)) - \eta(u^n) + \mathcal{O}(\Delta t^{p+1}) \\ &= \int_{t_n}^{t_n+\Delta t} \langle \eta'(u(t)), f(t, u(t)) \rangle \, \mathrm{d}t + \mathcal{O}(\Delta t^{p+1}) \\ &= \Delta t \sum_{i=1}^{s} \langle \eta'(u(t_n + c_i\Delta t)), f(t_n + c_i\Delta t, u(t_n + c_i\Delta t)) \rangle + \mathcal{O}(\Delta t^{p+1}).\end{aligned} \tag{5}$$

Furthermore, we have by the local truncation error of a Runge-Kutta method of order p that

$$\sum_{i=1}^{s} b_i f(t_n + c_i \Delta t, y_i) = \sum_{i=1}^{s} b_i f(t_n + c_i \Delta t, u(t_n + c_i \Delta t)) + \mathcal{O}(\Delta t^p),$$

and we know that the elements of the Butcher's tableau pertaining to a Runge-Kutta method is not dependent on f we test whether we can replace f by any smooth function.

Theorem 3.5. *Let $\mathcal{W}$ be a Banach space, $\psi : [0, T] \times \mathcal{H} \to \mathcal{W}$ a smooth function and b_i, c_i coefficients of a Runge-Kutta method of order p. Then*

$$\sum_{i=1}^{s} b_i \psi(t_n + c_i \Delta t, y_i) = \sum_{i=1}^{s} b_i \psi(t_n + c_i \Delta t, u(t_n + c_i \Delta t)) + \mathcal{O}(\Delta t^p).$$

Proof. We consider the function $\phi = \int_{t_n}^{t} \psi(\tau, u(\tau)) \mathrm{d}\tau$ with $\phi(t_n) = 0$ and construct the ODE

$$\frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} \phi(t) \\ u(t) \end{bmatrix} = \begin{bmatrix} \psi(t, u(t)) \\ f(t, u(t)) \end{bmatrix}, t \in (t_n, T), \quad \begin{bmatrix} \phi(t_n) \\ u(t_n) \end{bmatrix} = \begin{bmatrix} 0 \\ u^n \end{bmatrix} \tag{6}$$

Applying a pth order accurate Runge-Kutta method to Equation 6 we get

$$\begin{bmatrix} \phi^{n+1} \\ u^{n+1} \end{bmatrix} = \begin{bmatrix} \phi^{n} \\ u^{n} \end{bmatrix} + \begin{bmatrix} \Delta t \sum_{i=1}^{s} b_i \psi(t_n + c_i \Delta t, y_i) \\ \Delta t \sum_{i=1}^{s} b_i f(t_n + c_i \Delta t, y_i) \end{bmatrix}$$

Since $\phi^n = 0$ we get that

$$\Delta t \sum_{i=1}^{s} b_i \psi(t_n + c_i \Delta t, y_i) = \phi^{n+1} = \phi(t_n + \Delta t) + \mathcal{O}(\Delta t^{p+1}) \tag{7}$$

We also have that

$$\begin{aligned} \phi(t_n + \Delta t) &= \int_{t_n}^{t_n + \Delta t} \psi(t, u(t)) \mathrm{d}t \\ &= \Delta t \sum_{i=1}^{s} b_i \psi(t_n + c_i \Delta t, u(t_n + c_i \Delta t)) + \mathcal{O}(\Delta t^{p+1}) \end{aligned} \tag{8}$$

Equations 7 and 8 together gives

$$\sum_{i=1}^{s} b_i \psi(t_n + c_i \Delta t, y_i) = \sum_{i=1}^{s} b_i \psi(t_n + c_i \Delta t, u(t_n + c_i \Delta t)) + \mathcal{O}(\Delta t^{p+1}).$$

□

From Theorem 3.5 we get the following corollary.

Corollary 3.6. *If η is smooth and the given Runge-Kutta method is p-order accurate, $r(\gamma = 1) = \mathcal{O}(\Delta t^{p+1})$.*

Proof. Inserting $\phi(t,u) = \langle \eta'(u), f(t,u)\rangle$ into Theorem 3.5 and applying Equation 5 we get

$$\eta(u^{n+1}) - \eta(u^n) = \Delta t \sum_{i=1}^{s} b_i \langle \eta'(y_i), f(t_n + c_i\Delta t, y_i)\rangle + \mathcal{O}(\Delta t^{p+1}),$$

or equivalently

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

□

With these results we can now construct the main theorem.

Theorem 3.7. *Assume there exists a positive root γ_n of r. Consider the IDT/RRK methods based on a given Runge-Kutta method that is p-order accurate. Then*

1. *The IDT method interpreting $u_\gamma^{n+1} \approx u(t_n + \Delta t)$ has order $p - 1$.*
2. *The RRK method interpreting $u_\gamma^{n+1} \approx u(t_n + \gamma\Delta t)$ has order p.*

Proof. We take $1+k$ to be a root of r with $k < \Delta t \in \mathbb{R}$. By the Taylor expansion

$$r(1+k) = r(1) + r'(1)k + \mathcal{O}(k^2)$$

We have from Corollary 3.6 that $r(1) = \mathcal{O}(\Delta t^{p+1})$ and from Lemma 3.2 that $r'(1) = c\Delta t^2 + \mathcal{O}(\Delta t^3)$ for some $c \in \mathbb{R}^+$. Inserting this we get

$$0 = \mathcal{O}(\Delta t^{p+1}) + (c\Delta t^2 + \mathcal{O}(\Delta t^3))k + \mathcal{O}(k^2).$$

Which is equivalent to

$$k = \frac{\mathcal{O}(\Delta t^{p+1})}{c\Delta t^2 + \mathcal{O}(\Delta t^3)} = \mathcal{O}(\Delta t^{p-1})$$

Thus there is a root $\gamma_n = 1 + \mathcal{O}(\Delta t^{p-1})$ of r. Applying to Theorem 3.4 yields the desired result. □

4 Numerical Examples

Since the stepsize of RRK is unknown a priori, the last step of the method is always made using the ordinary RK method. This was applied to the IDT methods as well, although not necessary. Because of this we are in our relevant examples regarding the second to last value instead of the last.

We are studying the effects of values of Δt close to 1 in the examples. These are not reasonable choices for Δt, as the roots of r become less predictable. For the scalar root finding the method `scipy.optimize.root_scalar` was used within the hardcoded range $[0.5, 2]$ was used.

Note that fewer RK-methods were used for Problem 1 than in Problem 2. This is because the higher order Runge-Kutta methods resulted in an error from the method `isClose` used somewhere in `Assimulo`, a library of solvers that ours were written to inherit from.

The problems presented in this section are two of the same used in the article [4], with the only change being that the source had swapped the first and second position of the analytic solution.

The Butcher tableaus for SSPRK(3,3) and RK(4,4) were sourced online from [5] while the ones for VRK(9,6) and VRK(13,8) were taken from [1] and [2] respectively.

4.1 Problem 1 - Conserved exponential entropy

For this problem we consider the system

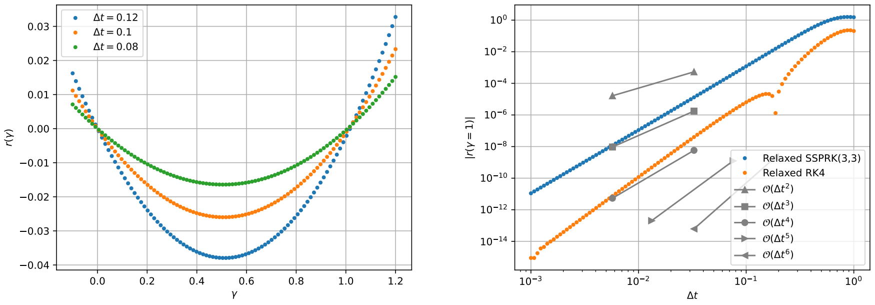
$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} = \begin{bmatrix} -\exp(u_2(t)) \\ \exp(u_1(t)) \end{bmatrix}, \qquad u^0 = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix},$$

with exponential entropy

$$\eta(u) = \exp(u_1) + \exp(u_2), \qquad \eta'(u) = \begin{bmatrix} \exp(u_1) \\ \exp(u_2) \end{bmatrix},$$

and analytic solution

$$u(t) = \left(\log\left(\frac{\mathrm{e} + \mathrm{e}^{3/2}}{\sqrt{\mathrm{e}} + \mathrm{e}^{(\sqrt{\mathrm{e}}+\mathrm{e})t}} \right), \log\left(\frac{\mathrm{e}^{(\sqrt{\mathrm{e}}+\mathrm{e})t}(\sqrt{\mathrm{e}} + \mathrm{e})}{\sqrt{\mathrm{e}} + \mathrm{e}^{(\sqrt{\mathrm{e}}+\mathrm{e})t}} \right) \right)^{\mathrm{T}}.$$

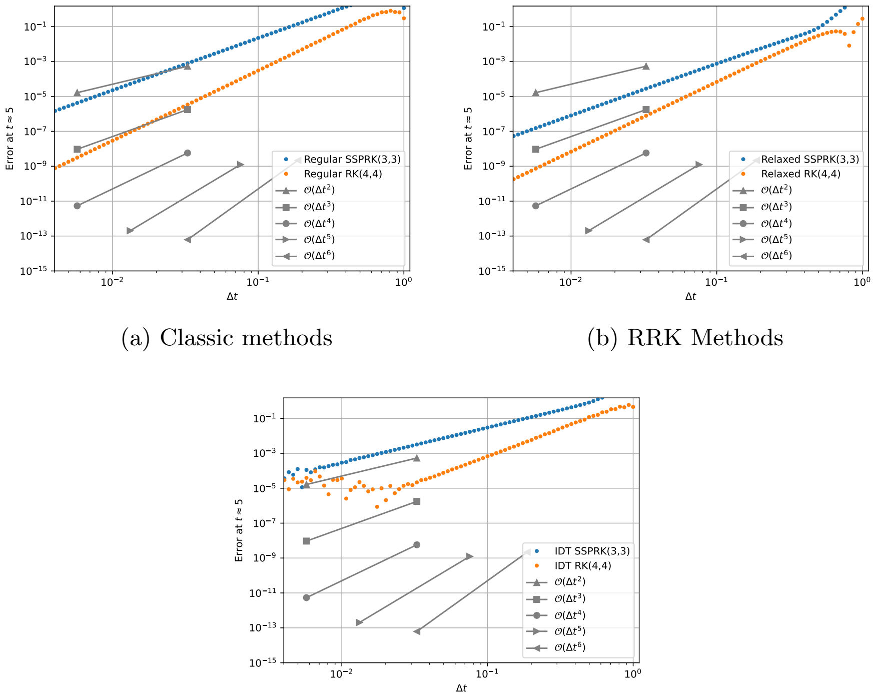


(a) $r(\gamma)$ for SSPRK(3,3)

(b) $|r(\gamma = 1)|$ for some RK methods

Figure 1: Numerical results for r at the first time step of problem 1.

From Figure 1a we can see that the shape of $r(\gamma)$ behaves as expected, with roots at 0 and around 1. In Figure 1b we see that $r(\gamma = 1) = \mathcal{O}(\Delta t^{p+1})$ as predicted in Corollary 3.6.



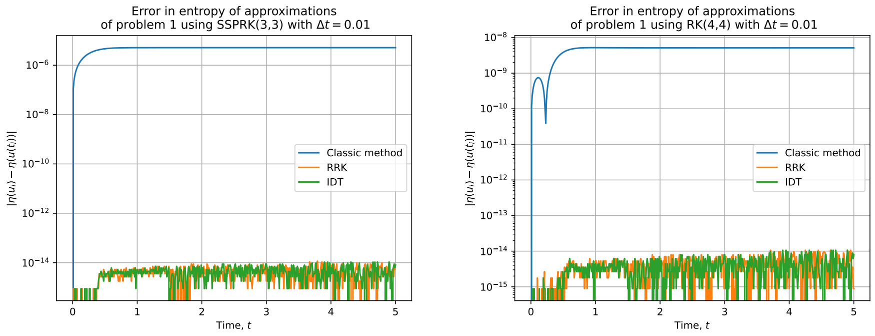
(a) Classic methods

(b) RRK Methods

(c) IDT methods

Figure 2: Convergence study for problem 1.

From Figure 2b we see that the RRK methods are of order p just like the unmodified methods in Figure 2a, meanwhile the IDT methods pictured in Figure 2c are of order $p - 1$, all in accordance with Theorem 3.7.



(a) SSPK(3,3)

(b) RK(4,4)

Figure 3: Error of entropy for problem 1.

In Figures 3a and 3b we see the error of the entropy η in the approximated points compared to the entropy of the analytic solution. Both RRK and IDT seem to perform equally well being zero withing floating point accuracy, which is considerably better than the unmodified method.

4.2 Problem 2 - Dissipated exponential entropy

Consider the ODE

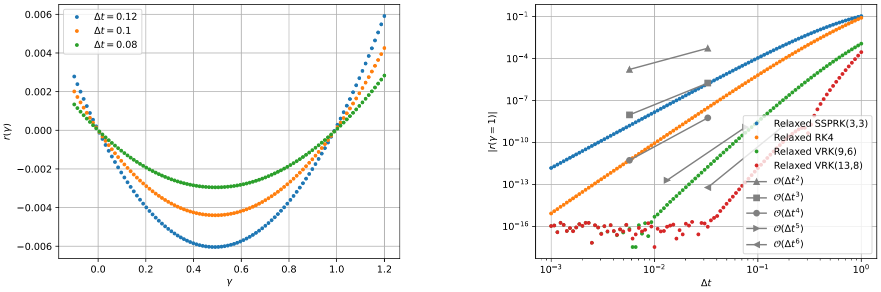
$$\frac{\mathrm{d}}{\mathrm{d}t}u(t) = -\exp(u(t)), \qquad u^0 = 0.5,$$

with the exponential entropy

$$\eta(u) = \exp(u), \qquad \eta'(u) = \exp(u),$$

and analytical solution

$$u(t) = -\log(\mathrm{e}^{-1/2} + t).$$

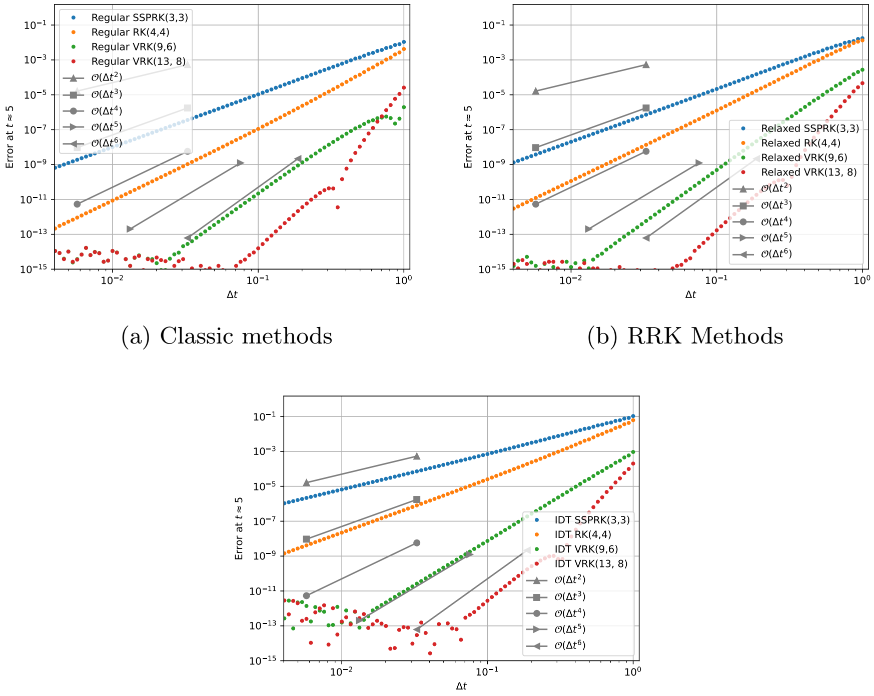


(a) $r(\gamma)$ for SSPRK(3,3)

(b) $|r(\gamma = 1)|$ for some RK methods

Figure 4: Numerical results for r at the first time step of problem 2.

From Figures 4a and 4b we once more see the expected behavior of r and its roots.



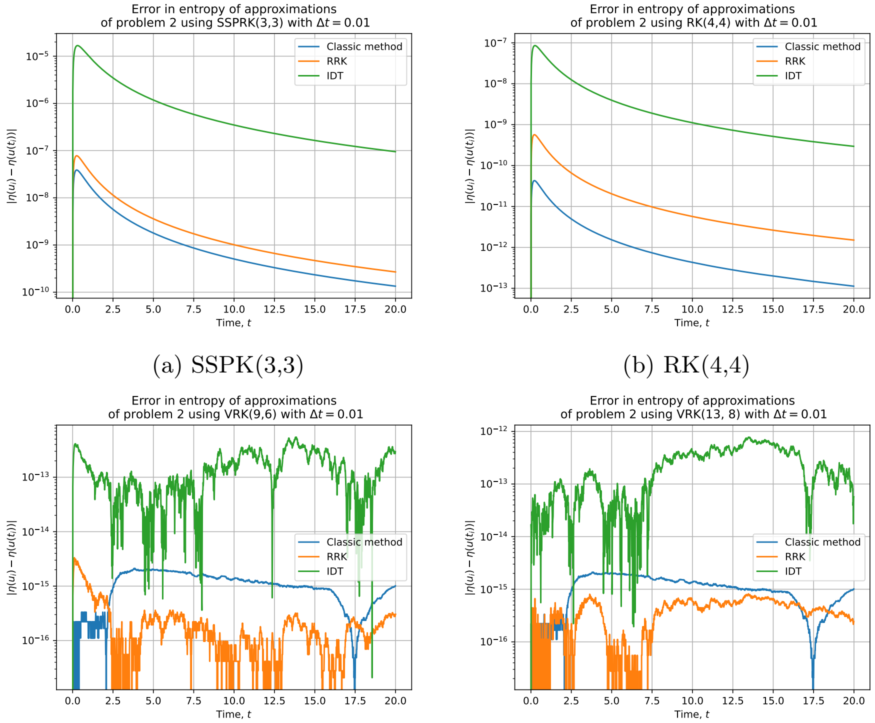
(a) Classic methods

(b) RRK Methods

(c) IDT methods

Figure 5: Convergence study for problem 2.

Once more, the behavior of the methods are as proposed in Theorem 3.7.



(a) SSPK(3,3)

(b) RK(4,4)

(c) VRK(9,6)

(d) VRK(13,8)

Figure 6: Error of entropy for problem 2.

When we compare the errors of entropy for problem 2 we get the strange result that in Figures 6a and 6b the unmodified methods seem to ourperform RRK and both outperform IDT. In Figures 6c and 6d the RRK and the unmodified methods perform similarly being zero withing floating point accuracy, and once again the are both better than the IDT methods.

5 Conclusion

As we have seen the method is simple to set up, being just a few additional lines of code added to the classic and well known Runge-Kutta method. Furthermore, as the most computation heavy part of it is a scalar root finding problem, it is cheap to perform. Thus, since the entropy stability makes the model physically correct there is no reason not to implement it. One drawback is that the method requires prior knowledge of η. This is however a reasonable assumption as someone interested in conserving the entropy of their simulation should have some insight into the entropy of their problem. A practice, an adaptation in code could take the entropy as an optional argument and when not supplied default into the classic Runge-Kutta. This requires little effort from the coder and leaves the decision in the hands of the user.

References

[1] Jim Verner. Jim verner's refuge for runge-kutta pairs, 2022. [Online; accessed 26-April-2023].

[2] Jim Verner. Jim verner's refuge for runge-kutta pairs, 2022. [Online; accessed 26-April-2023].

[3] David I. Ketcheson. Relaxation runge-kutta methods: Conservation and stability for inner-product norms, 2019.

[4] Hendrik Ranocha, Mohammed Sayyari, Lisandro Dalcin, Matteo Parsani, and David I. Ketcheson. Relaxation runge–kutta methods: Fully discrete explicit entropy-stable schemes for the compressible euler and navier–stokes equations. *SIAM Journal on Scientific Computing*, 42(2):A612–A638, jan 2020.

[5] Wikipedia contributors. List of runge–kutta methods — Wikipedia, the free encyclopedia, 2023. [Online; accessed 26-April-2023].