Ranocha and Giesselmann
Computational Science and Engineering (2024) 1:1
https://doi.org/10.1007/s44207-024-00001-0

Computational Science and Engineering

RESEARCH Open Access

Stability of step size control based on a posteriori error estimates

Hendrik Ranocha[1*] and Jan Giesselmann[2]

*Correspondence:
hendrik.ranocha@uni-mainz.de

[1] Institute of Mathematics, Johannes Gutenberg University Mainz, Staudingerweg 9, Mainz 55128, Germany
[2] Numerical Analysis and Scientific Computing, Technical University of Darmstadt, Dolivostr. 15, Darmstadt 64293, Germany

Abstract

A posteriori error estimates based on residuals can be used for reliable error control of numerical methods. Here, we consider them in the context of ordinary differential equations and Runge-Kutta methods. In particular, we take the approach of Dedner & Giesselmann (2016) and investigate it when used to select the time step size. We focus on step size control stability when combined with explicit Runge-Kutta methods and demonstrate that a standard I controller is unstable while more advanced PI and PID controllers can be designed to be stable. We compare the stability properties of residual-based estimators and classical error estimators based on an embedded Runge-Kutta method both analytically and in numerical experiments.

Keywords: Runge-Kutta methods, Step size control, Step size control stability, PID controller, A posteriori error estimates, Residual-based error estimates

MSC Classification: 65L06, 65M20.

1 Introduction

A posteriori error estimators come in different varieties; reliability, efficiency and asymptotic exactness being frequent quality criteria. The diversity in error estimators reflects the fact that they can be used for two different purposes: *error control* and *step size selection*. Both purposes are connected but place an emphasis on different properties. When it comes to step size selection in adaptive numerical methods, low computational costs are paramount; this goal is achieved for explicit Runge-Kutta schemes by embedded schemes used in extrapolation mode [1]. However, this methodology provides no error control, i.e., the user cannot certify whether a given numerical simulation is compatible with some error tolerance or not. Reliable (and efficient) error estimators have as their primary objective to control the error but can also be used to compute provably quasi-optimal meshes in elliptic and parabolic problems [2, 3]. Such methods are, commonly, not used in step size control of (explicit) schemes for ordinary differential equations (ODEs) due to the larger computational costs. In particular, they usually require to measure how much the numerical solution fails to satisfy the ODE, by the so called residual, and to relate the residual to the error by a suitable stability theory which might be based on energy or duality arguments [4].

Nevertheless, if one decides to compute residuals in order to ensure error control, it makes sense to also use this information for choosing step sizes and it is desirable that this leads to stable step size control. Typically, the most simple error-based step size selection uses an I controller that multiplies the current time step size by a factor derived from an error estimate using asymptotic arguments. By construction, it usually works well in this asymptotic regime of small time step sizes. However, explicit Runge-Kutta methods also need to operate well when the time step size is limited by stability instead of accuracy. In this situation, step size control stability is important. The study of these properties has been initiated by Hall [5, 6] with further refinements and applications together with Higham [7, 8].

One option to obtain step size control stability when using embedded Runge-Kutta methods such as the classical schemes of Bogacki and Shampine [9, 10] or Dormand and Prince [1] is to design the methods specifically to allow step size control stability with the classical I controller [8]. However, most schemes of this class used nowadays make use of more advanced controllers such as PI and PID controllers developed for example in [11–14]. As demonstrated in [15, 16], these controllers can be used together with embedded Runge-Kutta method for efficient and robust time step size control in the context of compressible computational fluid dynamics where the stability limited regime is crucial due to the Courant-Friedrichs-Lewy step size restriction [17].

In the following Section 2, we introduce the notation and basic ideas of the methods. Next, we investigate the step size control stability of methods derived from the residual-based a posteriori error estimators of [18] analytically in Section 3 and numerically in Section 4. Finally, we summarize and discuss our results in Section 5. All source code required to reproduce the numerical experiments is available online in our reproducibility repository [19].

2 Basic ideas of step size control and a posteriori error estimates

Consider a system of ODEs

$$u'(t) = f\big(t, u(t)\big), \quad u(0) = u^0 \in \mathbb{R}^m. \tag{1}$$

One step of an explicit Runge-Kutta method can be written as [20, 21]

$$y^i = u^n + \Delta t_n \sum_{j=1}^{i-1} a_{ij} f(t^n + c_j \Delta t_n, y^j), \qquad i \in \{1, \dots, s\}, \tag{2}$$

$$u^{n+1} = u^n + \Delta t_n \sum_{i=1}^{s} b_i f(t^n + c_i \Delta t_n, y^i), \tag{3}$$

where y^i are the stage values, u^n is the numerical solution approximating u at time t^n, and $t^{n+1} = t^n + \Delta t_n$. As usual, we assume that the row-sum condition $\forall i : c_i = \sum_j a_{ij}$ is satisfied so that it suffices to consider autonomous problems.

Given an approach to estimate the error made in one step and a tolerance, a common approach is to compute a weighted error estimator w_{n+1} of the form "error estimate divided by tolerance" [20, Section II.4]. Let $|e_{n+1}|$ be an error estimate. If only an absolute tolerance

τ is used (and no relative tolerance), the weighted error estimator would simply be $w_{n+1} = |e_{n+1}|/\tau$. Thus, $w_{n+1} \leq 1$ means that the desired tolerance is achieved. Setting $\varepsilon_{n+1} = \frac{1}{w_{n+1}}$, classical methods for choosing time step sizes are based on I, PI, and PID controllers, e.g., [13, 14, 22]

$$\Delta t_{n+1} = \kappa\left(\varepsilon_{n+1}^{\beta_1/k}\varepsilon_n^{\beta_2/k}\varepsilon_{n-1}^{\beta_3/k}\right)\Delta t_n, \tag{4}$$

where k is chosen such that w_{n+1} is expected to be of order Δt_n^k. The function κ is a step size limiter, which we choose as $\kappa(a) = 1 + \arctan(a-1)$ [14]. The real numbers β_i are the controller parameters. The general form (4) of a PID controller reduces to a PI controller for $\beta_3 = 0$ and to a classical I controller for $\beta_2 = \beta_3 = 0$. There are different ways in which the estimator w_{n+1} can be obtained.

A classical approach to step size control is to obtain an error estimate via an embedded method that consists of (2) and

$$\widehat{u}^{n+1} = u^n + \Delta t_n \sum_{i=1}^{s} \widehat{b}_i f(t^n + c_i\Delta t_n, y^i) + \Delta t_n \widehat{b}_{s+1} f(t^{n+1}, u^{n+1}). \tag{5}$$

Typically, these methods are used in local extrapolation mode, i.e., the main method is of order p and the embedded method is of order $\widehat{p} = p - 1$. Then, k is chosen as $k = \min(p, \widehat{p}) + 1$, i.e., $k = p$ due to the local extrapolation mode. If $\widehat{b}_{s+1} \neq 0$, the right-hand side of the new approximation u^{n+1} is used. This first-same-as-last (FSAL) technique was introduced to improve the performance of the error estimator $u^{n+1} - \widehat{u}^{n+1}$ [1]. Overall, one obtains

$$w_{n+1} = \left(\frac{1}{m}\sum_{i=1}^{m}\left(\frac{u_i^{n+1} - \widehat{u}_i^{n+1}}{\tau_a + \tau_r \max\{|u_i^{n+1}|, |u_i^n|\}}\right)^2\right)^{1/2}, \tag{6}$$

where $\tau_a, \tau_r > 0$ are absolute and relative tolerances and m is the total number of degrees of freedom in u, cf. [20, Section II.4].

While step size control based on embedded Runge-Kutta schemes is highly efficient (for explicit schemes) and works very well in practice it has the (theoretical) drawback that it does not provide any rigorous upper bounds for the error.

In contrast, the error estimators described in the next section lead to provable upper bounds for the error but are more expensive to compute.

2.1 Energy based a posteriori error estimates

The basic idea of this type of error estimators is to compute a sufficiently regular reconstruction $\widehat{u}$ from the numerical solution, to compute the residual

$$R := \frac{\mathrm{d}}{\mathrm{d}t}\widehat{u} - f(\widehat{u}), \tag{7}$$

and to use a suitable stability theory of the ODE to bound the difference between u and $\widehat{u}$ in terms of R. Indeed, if f satisfies a one-sided Lipschitz condition, i.e. there exists $L \in \mathbb{R}$ such that for all $u, v \in \mathbb{R}^m$

$$\langle f(u) - f(v), u - v \rangle \leq L\|u - v\|^2, \tag{8}$$

then Gronwall's lemma implies, in the (scaled) Euclidean norm on $\mathbb{R}^m$,

Lemma 1 *Let u be a solution of (1) and let $\widehat{u}$ solve (7). Then, for all $0 \leq t \leq T$*

$$\|u(t) - \widehat{u}(t)\| \leq \left(\|u(0) - \widehat{u}(0)\| + \|R\exp(-L\cdot)\|_{L^1(0,T)}\right)\mathrm{e}^{Lt} \tag{9}$$

where L is such that (8) holds. If $\widehat{u}$ is the reconstruction of a numerical solution satisfying $\widehat{u}(t^n) = u^n$, then as an immediate consequence for all $0 \leq t^n \leq T$

$$\|u(t^n) - u^n\| \leq \left(\|u(0) - \widehat{u}(0)\| + \|R\exp(-L\cdot)\|_{L^1(0,T)}\right)\mathrm{e}^{LT}.$$

This provides an a posteriori error bound once it is clear how $\widehat{u}$ is computed from the numerical solution since R can be computed by evaluating (7).

1 *Proof*

We multiply (1) minus (7) by $u - \widehat{u}$ and obtain

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|u - \widehat{u}\|^2 = \langle f(u) - f(\widehat{u}), u - \widehat{u}\rangle - \langle R, u - \widehat{u}\rangle \leq L\|u - \widehat{u}\|^2 + \|R\|\|u - \widehat{u}\|.$$

Thus, setting $y_1(t) := \exp(-Lt)\|u(t) - \widehat{u}(t)\|$ we get

$$\frac{\mathrm{d}}{\mathrm{d}t}y_1^2(t) \leq 2\|R(t)\|\exp(-2Lt)\|u(t) - \widehat{u}(t)\| = 2\|R(t)\|\exp(-Lt)y_1(t)$$

such that integrating in time leads to

$$\frac{1}{2}y_1^2(t) \leq \frac{1}{2}y_1^2(0) + \int_0^t \|R(s)\|\exp(-Ls)y_1(s)\mathrm{d}s.$$

Invoking [23, Theorem 5] leads to

$$y_1(t) \leq y_1(0) + \int_0^t \|R(s)\|\exp(-Ls)\mathrm{d}s.$$

Inserting the definition of y_1 and multiplying by $\exp(Lt)$ gives the desired result. □

If $\|R\exp(-L\cdot)\|_{L^1(0,T)}$ is computed for error control, it seems reasonable to choose step sizes based on $\|R\|_{L^1(t^n,t^{n+1})}$. However, certain care is needed in this approach: First of all, it needs to be ensured by a suitable reconstruction strategy that $\|R\|_{L^1(t^n,t^{n+1})}$ is indeed of order Δt_n^{p+1} for a p-th order scheme and sufficiently regular solutions. Defining such a reconstruction is not trivial since the stage values of RK schemes do not contain high order information directly and computing the residual involves taking a derivative which might lead to the loss of one order of convergence. Indeed, it turns out that, in general, for any numerical scheme a dedicated reconstruction method needs to be derived in order to ensure that $\|R\|_{L^1(t^n,t^{n+1})}$ is of order Δt_n^{p+1}.

It should also be kept in mind that there are many interesting scenarios where the step from (7) to (9) is far less straightforward. One scenario is that no L exists such that (8) holds. Another scenario that frequently arises when f stems from the spatial discretization of a PDE is that L is positive and depends on an inverse power of the spatial mesh width. In such cases a more sophisticated stability analysis is needed in order to connect errors and residuals in an optimal way.

This line of thought can be traced back, at least, to [24] and was expunged in detail in [25]. These works address discretizations of parabolic PDEs and implicit time discretizations. It turns out that in case explicit RK schemes are applied to ODEs or semi-discretizations of first order hyperbolic PDEs a more generic method based on Hermite interpolation can be used [18]. Indeed, inspection of the proof of [18, Theorem 2.1], which only covers equidistant time steps, shows that the reconstructions of Hermite-type considered here satisfy

$$\|R\|_{L^\infty(t^n,t^{n+1})} \leq C\Delta t_n^p \tag{10}$$

provided the Runge-Kutta scheme and f are such that the consistency error is bounded by $\tilde{C}\Delta t_n^{p+1}$. Detailed formulas of these reconstructions for generic cases can be found in [18]. Note that [18] also considers reconstructions using values from previous time steps and for those (10) will not hold, since the left-hand side also depends on sizes of previous time steps. All reconstructions considered in this paper only use values from the current time step and we will provide explicit formulas for the reconstructions we investigate in this paper, e.g., (38).

A key observation of [18] is that $f(t^n, u^n)$ is not only readily available in building the reconstruction since it is anyway computed during the time step but also known to be accurate enough in this scenario. Step size control for explicit RK schemes and the estimators from [18] is what we are going to investigate in this paper.

In the context of these methods we evaluate the weighted error estimator either in the L^1 norm as

$$w_{n+1} = \frac{\|R\|_{L^1(t^n,t^{n+1})}}{\tau_a + \tau_r \max\{\|u^n\|, \|u^{n+1}\|\}} \tag{11}$$

or in the L^2 norm as

$$w_{n+1} = \frac{\sqrt{\Delta t_n}\|R\|_{L^2(t^n,t^{n+1})}}{\tau_a + \tau_r \max\{\|u^n\|, \|u^{n+1}\|\}}. \tag{12}$$

Note the multiplication by $\sqrt{\Delta t_n}$ in (12) ensures the correct scaling in terms of the time step size Δt_n. Since the estimator is of the same order as the main method, we use $k = p + 1$.

2.2 Error per step versus error per unit step

There are several interesting properties of step size controllers. While the main focus of this paper is *step size control stability*, discussed in Section 3, the current section discusses two other important properties: *tolerance convergence* and the stronger property *tolerance proportionality*. *Tolerance convergence* means $\tau \to 0 \implies \|E\| \to 0$, i.e., to

obtain a global error E converging to zero when the tolerance τ goes to zero. *Tolerance proportionality* means $\exists c, C > 0 : c\tau \leq \|E\| \leq C\tau$ for all sufficiently small $\tau > 0$, i.e., to obtain a global error that is roughly proportional to the given tolerance. In this section, we discuss these properties for *error per step* (EPS) and *error per unit step* (EPUS) control based on (11) or (12).

The formulas (11) and (12) estimate the *error per step* and adaptation as in (4) aims at keeping it below a given tolerance. An alternative is to control the *error per unit step*. Then, the tolerance should control the error per step estimate $|e_{n+1}|$ divided by the step size Δt_n, i.e., $\tau \geq |e_{n+1}|/\Delta t_n$ instead of $\tau \geq |e_{n+1}|$ for EPS control. In this case, the controllers still have the same form as above but use $\varepsilon_{n+1} = \Delta t_n / w_{n+1}$ instead of $\varepsilon_{n+1} = 1/w_{n+1}$.

Higham [26] analyzed tolerance proportionality for different control strategies. He proved that tolerance proportionality is obtained for a pth-order method if a local error estimate scaling as $\mathcal{O}(\Delta t_n^p)$ is chosen as control objective [26, Corollary 2.1]. In particular, this is the case if the local error per step is controlled for an explicit Runge-Kutta pair in local extrapolation mode, i.e., where a main method of order p is coupled with an embedded method of order $\widehat{p} = p - 1$. This is also the case when the local error per unit step is controlled based on an error estimate $|e_{n+1}| = \mathcal{O}(\Delta t_n^{p+1})$ — which is the scaling of the residual-based error estimators described in Section 2.1.

Butcher [21, Sections 371–373] argues that an EPS control is close to producing "optimal" step size sequences (in a sense discussed there). Thus, an EPS control can be considered to be better than an EPUS control, even if the EPUS control provides tolerance proportionality and the EPS does not.

Moreover, tolerance convergence can still be expected from an EPS control if the corresponding EPUS control leads to tolerance proportionality. To see this, let $|e_{n+1}|$ be the local error estimate and τ the tolerance. The EPUS control goal is to achieve $|e_{n+1}|/\Delta t \leq \tau_{\text{EPUS}}$. The EPS control has the goal $|e_{n+1}| \leq \tau_{\text{EPS}}$. Thus, EPS and EPUS control have the same goal if $\tau_{\text{EPS}} = \tau_{\text{EPUS}} \Delta t$. For a pth-order method with tolerance proportionality, the step size will scale as $\Delta t \propto \tau_{\text{EPUS}}^{1/p}$. Thus, tolerance proportionality can be achieved for EPS control with rescaled tolerance $\tau_{\text{EPS}} = \tau_{\text{EPUS}} \Delta t \propto \tau_{\text{EPUS}} \tau_{\text{EPUS}}^{1/p} = \tau_{\text{EPUS}}^{(p+1)/p}$. Re-arranging, we can expect to obtain a global error scaling as $\tau_{\text{EPS}}^{p/(p+1)}$, still leading to tolerance convergence. Butcher [21, Section 373] argues that this is fine in practice — due to the link of EPS control to "optimal" step size sequences. Such a behavior does indeed occur in practice for well-known numerical integrators such as DASSL [14]. Since EPS control is common for the classical approach of using an embedded method with local extrapolation, we concentrate on this mode in the following section.

3 Step size control stability

In this section, we analyze step size control stability for residual error estimators and compare it to the situation for embedded methods. We follow the presentation of [27, Section IV.2] to introduce the concept of step size control stability. Consider the scalar test problem

$$u'(t) = \lambda u(t), \quad u(0) = u^0, \tag{13}$$

for $\lambda \in \mathbb{C}$ and a simplified I controller[1] of the form

$$\Delta t_{n+1} = \varepsilon_{n+1}^{1/k} \Delta t_n, \qquad \varepsilon_{n+1} = \frac{\tau}{|e_{n+1}|}, \tag{14}$$

where τ is a fixed tolerance and $|e_{n+1}|$ is the error estimate, e.g., $|e_{n+1}| = |u^{n+1} - \widehat{u}^{n+1}|$ when an embedded method is used. This yields the dynamical system

$$\begin{aligned} u^{n+1} &= R(\Delta t_n \lambda) u^n, \\ \Delta t_{n+1} &= \Delta t_n \left(\frac{\tau}{|e_{n+1}|} \right)^{1/k}, \end{aligned} \tag{15}$$

where R is the stability function of the (main) Runge-Kutta method. Here, we have assumed that the error estimate $|e_{n+1}|$ depends only on u^n and Δt_n so that no additional equation for $|e_{n+1}|$ is required. This assumption holds for all error estimates considered in this article. The analysis can be simplified by introducing logarithms

$$\eta_n = \log |u^n|, \quad \chi_n = \log \Delta t_n, \tag{16}$$

resulting in

$$\begin{aligned} \eta_{n+1} &= \log |R(\mathrm{e}^{\chi_n} \lambda)| + \eta_n, \\ \chi_{n+1} &= \chi_n + \frac{1}{k} \big(\log(\tau) - \log |e_{n+1}| \big). \end{aligned} \tag{17}$$

To study the step size control stability, we investigate the stability properties of fixed points defined by

$$|R(\mathrm{e}^{\chi_n} \lambda)| = 1, \quad \log |e_{n+1}| = \log(\tau). \tag{18}$$

The first equation states that the step size $\Delta t_n = \mathrm{e}^{\chi_n}$ is chosen such that $z = \Delta t_n \lambda$ is on the boundary of the stability region of the Runge-Kutta method. A stable behavior requires that the spectral radius of the Jacobian

$$J = \begin{pmatrix} 1 & \mu \\ -\frac{1}{k} \partial_{\eta_n} \log |e_{n+1}| & 1 - \frac{1}{k} \partial_{\chi_n} \log |e_{n+1}| \end{pmatrix}, \qquad \mu = \operatorname{Re}\left(\frac{R'(z)}{R(z)} z \right), \tag{19}$$

does not exceed unity.

For a simplified PID controller of the form

$$\Delta t_{n+1} = \varepsilon_{n+1}^{\beta_1/k} \varepsilon_n^{\beta_2/k} \varepsilon_{n-1}^{\beta_3/k} \Delta t_n, \qquad \varepsilon_{n+1} = \frac{\tau}{|e_{n+1}|}, \tag{20}$$

the dynamical system becomes

[1] An I controller is a PID controller with $\beta_2 = \beta_3 = 0$. Here, we got rid of the step size limiter and set the first parameter $\beta_1 = 1$. This is a classical deadbeat controller.

$$\begin{aligned}
\eta_{n+1} &= \eta_n + \log|R(\mathrm{e}^{\chi_n}\lambda)|, \\
\chi_{n+1} &= \chi_n + \frac{\beta_1}{k}\big(\log(\tau) - \log|e_{n+1}|\big) + \frac{\beta_2}{k}\big(\log(\tau) - \log|e_n|\big) \\
&\quad + \frac{\beta_3}{k}\big(\log(\tau) - \log|e_{n-1}|\big).
\end{aligned} \tag{21}$$

This can be considered as a dynamical system mapping from the indices $(n, n-1, n-2)$ to the indices $(n+1, n, n-1)$. The corresponding Jacobian is

$$\begin{aligned}
J &= \begin{pmatrix}
1 & \mu & 0 & 0 & 0 & 0 \\
-\frac{\beta_1}{k}\frac{\partial l_{n+1}}{\partial \eta_n} & 1 - \frac{\beta_1}{k}\frac{\partial l_{n+1}}{\partial \chi_n} & -\frac{\beta_2}{k}\frac{\partial l_n}{\partial \eta_{n-1}} & -\frac{\beta_2}{k}\frac{\partial l_n}{\partial \chi_{n-1}} & -\frac{\beta_3}{k}\frac{\partial l_{n-1}}{\partial \eta_{n-2}} & -\frac{\beta_3}{k}\frac{\partial l_{n-1}}{\partial \chi_{n-2}} \\
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0
\end{pmatrix}, \\
l_j &= \log|e_j|, \quad \mu = \operatorname{Re}\left(\frac{R'(z)}{R(z)}z\right),
\end{aligned} \tag{22}$$

cf. [16, 22]. When an embedded method with stability function $\widehat{R}$ is used to estimate the error, we consider the polynomial $E(z) = R(z) - \widehat{R}(z)$. Then, the Jacobian (22) becomes [16, 22]

$$\begin{aligned}
J &= \begin{pmatrix}
1 & \mu & 0 & 0 & 0 & 0 \\
-\frac{\beta_1}{k} & 1 - \frac{\beta_1}{k}\nu & -\frac{\beta_2}{k} & -\frac{\beta_2}{k}\nu & -\frac{\beta_3}{k} & -\frac{\beta_3}{k}\nu \\
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0
\end{pmatrix}, \\
\mu &= \operatorname{Re}\left(\frac{R'(z)}{R(z)}z\right), \qquad \nu = \operatorname{Re}\left(\frac{E'(z)}{E(z)}z\right).
\end{aligned} \tag{23}$$

In the following, we will study step size control stability for several explicit Runge-Kutta methods from first to third order of accuracy. We begin with the explicit Euler method to illustrate the steps. The other calculations use Mathematica [28]; the corresponding notebooks are available in our reproducibility repository [19].

3.1 Explicit Euler method

The most simple explicit Runge-Kutta method is the explicit Euler method with stability function

$$R(z) = 1 + z. \tag{24}$$

We use the linear reconstruction polynomial

$$\widehat{u}(t) = u^n + \frac{t - t^n}{t^{n+1} - t^n}(u^{n+1} - u^n) \tag{25}$$

for the time interval $[t^n, t^{n+1}]$. Then, the weighted L^1 error estimate (11) is given by

$$\begin{aligned} |e_{n+1}| = \|R\|_{L^1(t^n,t^{n+1})} &= \int_{t^n}^{t^{n+1}} \left| \frac{\mathrm{d}}{\mathrm{d}t}\widehat{u}(t) - \lambda\widehat{u}(t) \right| \mathrm{d}t \\ &= \int_{t^n}^{t^{n+1}} (t-t^n)\mathrm{d}t \ |\lambda|^2|u^n| = \frac{1}{2}\Delta t_n^2|\lambda|^2|u^n|. \end{aligned} \tag{26}$$

The L^2 version (11) uses

$$|e_{n+1}| = \sqrt{\Delta t_n}\|R\|_{L^2(t^n,t^{n+1})} = \frac{1}{\sqrt{3}}\Delta t_n^2|\lambda|^2|u^n|. \tag{27}$$

Thus, the Jacobian (19) of the I controller becomes in both cases

$$J = \begin{pmatrix} 1 & \mu \\ -\frac{1}{k} & 1-\frac{2}{k} \end{pmatrix}, \qquad \mu = \mathrm{Re}\left(\frac{z}{1+z}\right). \tag{28}$$

3.2 Second-order, two-stage methods

All explicit second-order, two-stage Runge-Kutta methods have the stability function

$$R(z) = 1 + z + \frac{z^2}{2}. \tag{29}$$

We use the left-biased quadratic Hermite interpolation polynomial

$$\widehat{u}(t) = \left(1 - \frac{t^2}{\Delta t_n^2}\right)u^n + \left(t - \frac{t^2}{\Delta t_n}\right)f(t^n,u^n) + \frac{t^2}{\Delta t_n^2}u^{n+1} \tag{30}$$

normalized to $t \in [0, \Delta t_n]$. Then, the weighted L^1 error estimate (11) is given by $w_{n+1} = |e_{n+1}|/(\tau_a + \tau_r \max\{\|u^n\|, \|u^{n+1}\|\})$ with

$$\begin{aligned} |e_{n+1}| = \|R\|_{L^1(t^n,t^{n+1})} &= \int_{t^n}^{t^{n+1}} \left| \frac{\mathrm{d}}{\mathrm{d}t}\widehat{u}(t) - \lambda\widehat{u}(t) \right| \mathrm{d}t \\ &= \frac{1}{2}\int_0^{\Delta t_n} t^2\mathrm{d}t \ |\lambda|^3|u^n| = \frac{1}{6}\Delta t_n^3|\lambda|^3|u^n|. \end{aligned} \tag{31}$$

The corresponding L^2 version uses

$$|e_{n+1}| = \sqrt{\Delta t_n}\|R\|_{L^2(t^n,t^{n+1})} = \frac{1}{2\sqrt{5}}\Delta t_n^3|\lambda|^3|u^n|. \tag{32}$$

The expression of the Jacobian (19) of the I controller with residual error estimator becomes in both cases

$$J = \begin{pmatrix} 1 & \mu \\ -\frac{1}{k} & 1-\frac{3}{k} \end{pmatrix}, \quad \mu = \mathrm{Re}\left(\frac{z+z^2}{1+z+z^2/2}\right). \tag{33}$$

The corresponding Jacobian (23) based on an embedded explicit Euler method $\widehat{u}^{n+1} = u^n + \Delta t_n f(t^n,u^n)$ is

$$J = \begin{pmatrix} 1 & \mu \\ -\frac{1}{k} & 1 - \frac{2}{k} \end{pmatrix}, \qquad \mu = \operatorname{Re}\left(\frac{z + z^2}{1 + z + z^2/2} \right), \ k = 2. \tag{34}$$

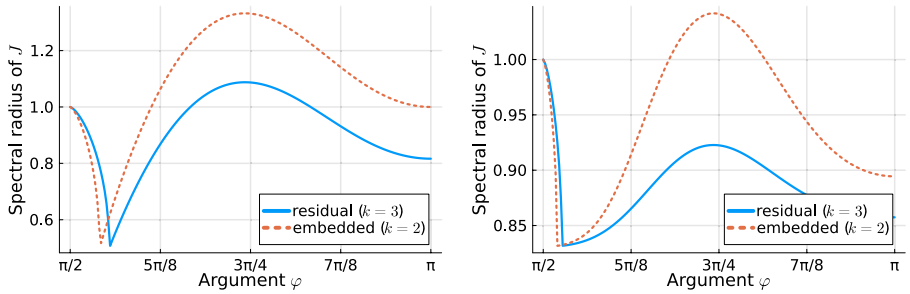
The eigenvalues of these Jacobians are complex expressions that do not lend themselves to an analytical investigation. Thus, we use a numerical approach to compute and visualize the spectral radii in Fig. 1. First, it is clear that the I controller does not lead to step size control stability for all methods. However, the instability is less severe for the residual-based approach — the spectral radius of the Jacobian is smaller in most regions and exceeds unity less compared to the version using an embedded Euler method.

To check the behavior in practice, we use the test problem (1) with

$$f(t,u) = -2000 \begin{pmatrix} \cos(t)u_1 + \sin(t)u_2 + 1 \\ -\sin(t)u_1 + \cos(t)u_2 + 1 \end{pmatrix}, \quad u^0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \tag{35}$$

in the time interval (0.0, 1.57) as suggested by Hairer and Wanner [27, Section IV.2]. With tolerances $\tau_a = \tau_r = 10^{-4}$ and $k = 2$, the embedded approach leads to 1877 accepted and 263 rejected steps. The L^1 residual-based approach with $k = 3$ leads to 1811 accepted and 27 rejected steps. The L^2 variant behaves similarly, see Table 1. Details of the implementation and further numerical experiments are discussed in Section 4.

The expression of the Jacobian (22) of the PI controller with residual error estimator becomes



(a) I controller with $\beta = (1, 0, 0)$, corresponding to the Jacobians (33) and (34).

(b) PI controller with $\beta = (0.6, -0.2, 0)$, corresponding to the Jacobians (36) and (23).

Fig. 1 Spectral radius of the Jacobian J for error estimates based on the residual and an embedded Euler method for explicit second-order, two-stage Runge-Kutta methods. The Jacobian is evaluated at $z = re^{i\varphi}$ where the radius r is chosen such that z is on the boundary of the stability region of the (main) method

Table 1 Number of accepted and rejected time steps of Heun's second-order method with an embedded explicit Euler method for the test problem (35) with tolerances $\tau_a = \tau_r = 10^{-4}$

	I controller, $\beta = (1, 0, 0)$			**PI controller, $\beta = (0.6, -0.2, 0)$**		
	Residual Estimator		**Embedded**	**Residual Estimator**		**Embedded**
	L^1	L^2	**Method**	L^1	L^2	**Method**
k	3	3	2	3	3	2
Accepted	1811	1815	1877	1824	1828	1918
Rejected	27	26	263	0	0	55

$$J = \begin{pmatrix} 1 & \mu & 0 & 0 \\ -\frac{\beta_1}{k} & 1-\frac{3\beta_1}{k} & -\frac{\beta_2}{k} & -\frac{3\beta_2}{k} \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \quad \mu = \operatorname{Re}\left(\frac{z+z^2}{1+z+z^2/2} \right). \tag{36}$$

The spectral radii of the Jacobians for the PI controller with parameters $\beta = (0.6, -0.2, 0.0)$ are visualized in Fig. 1. Clearly, the more involved controller leads to step size control stability for the residual-based approach but not for the version using an embedded method. For the test problem (35), we get 1918 accepted and 55 rejected steps for the the embedded approach while the residual-based approach leads to 1824 accepted and no rejected steps.

3.3 Third-order, three-stage methods

All explicit third-order, three-stage Runge-Kutta methods have the stability function

$$R(z) = 1 + z + \frac{z^2}{2} + \frac{z^3}{6}. \tag{37}$$

We would like to use the central cubic Hermite interpolation polynomial

$$\begin{aligned} \widehat{u}(t) = &\left(1 - \frac{3t^2}{\Delta t_n^2} + \frac{2t^3}{\Delta t_n^3}\right) u^n + \left(t - \frac{2t^2}{\Delta t_n} + \frac{t^3}{\Delta t_n^2}\right) f(t^n, u^n) \\ &+ \left(\frac{3t^2}{\Delta t_n^2} - \frac{2t^3}{\Delta t_n^3}\right) u^{n+1} + \left(-\frac{t^2}{\Delta t_n} + \frac{t^3}{\Delta t_n^2}\right) f(t^{n+1}, u^{n+1}) \end{aligned} \tag{38}$$

normalized to $t \in [0, \Delta t_n]$. However, we have not been able to evaluate the integral $\|R\|_{L^1(t^n, t^{n+1})}$ analytically, even when using Mathematica [28]. Thus, we use the left-biased cubic Hermite interpolation polynomial

$$\begin{aligned} \widehat{u}(t) = &\left(1 - \frac{t^3}{\Delta t_n^3}\right) u^n + \left(t - \frac{t^3}{\Delta t_n^2}\right) f(t^n, u^n) \\ &+ \left(\frac{t^2}{2} - \frac{t^3}{2\Delta t_n}\right) (f_t + f_u f)(t^n, u^n) + \frac{t^3}{\Delta t_n^3} u^{n+1} \end{aligned} \tag{39}$$

normalized to $t \in [0, \Delta t_n]$ for a first analysis but the central version in the implementation and a more numerically supported analysis.

3.3.1 Left-biased cubic Hermite interpolation

The analysis proceeds with the weighted L^1 error estimate (11) given by

$$\begin{aligned} |e_{n+1}| = \|R\|_{L^1(t^n, t^{n+1})} &= \int_{t^n}^{t^{n+1}} \left| \frac{\mathrm{d}}{\mathrm{d}t} \widehat{u}(t) - \lambda \widehat{u}(t) \right| \mathrm{d}t \\ &= \frac{1}{6} \int_0^{\Delta t_n} t^3 \mathrm{d}t \; |\lambda|^4 |u^n| = \frac{1}{24} \Delta t_n^4 |\lambda|^4 |u^n|. \end{aligned} \tag{40}$$

The expression of the Jacobian (19) of the I controller with residual error estimator becomes

$$J = \begin{pmatrix} 1 & \mu \\ -\frac{1}{k} & 1 - \frac{4}{k} \end{pmatrix}, \quad \mu = \operatorname{Re}\left(\frac{z + z^2 + z^3/2}{1 + z + z^2/2 + z^3/6} \right). \tag{41}$$

If we use the L^2 error estimate (12) instead, we get

$$|e_{n+1}| = \sqrt{\Delta t_n} \|R\|_{L^2(t^n, t^{n+1})} = \frac{1}{6\sqrt{7}} \Delta t_n^4 |\lambda|^4 |u^n| \tag{42}$$

for the left-biased cubic Hermite interpolation (39) and thus the same Jacobian as for the L^1 error estimate discussed above.

The spectral radii of the Jacobians are visualized in Fig. 2. Again, all methods do not lead to step size control stability.

The expression of the Jacobian (22) of the PI controller with residual error estimator becomes

$$J = \begin{pmatrix} 1 & \mu & 0 & 0 \\ -\frac{\beta_1}{k} & 1 - \frac{4\beta_1}{k} & -\frac{\beta_2}{k} & -\frac{4\beta_2}{k} \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \quad \mu = \operatorname{Re}\left(\frac{z + z^2 + z^3/2}{1 + z + z^2/2 + z^3/6} \right). \tag{43}$$

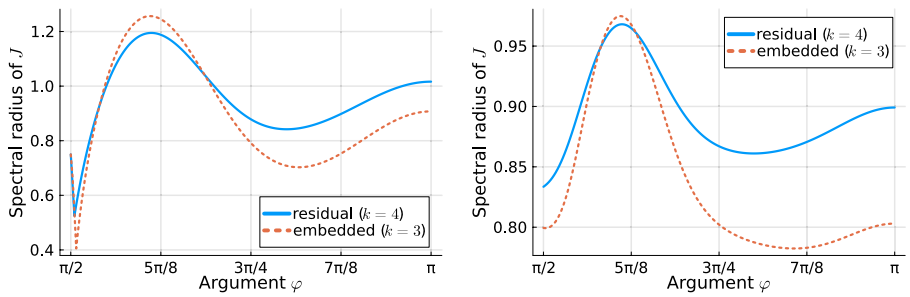
The spectral radii of the Jacobians for the PI controller with parameters $\beta = (0.6, -0.2, 0.0)$ recommended in [16] are visualized in Fig. 2. Clearly, the more involved controller leads to step size control stability for all approaches.

3.3.2 Central cubic Hermite interpolation

Recall the Jacobian (19) of the I controller system. For the L^1 residual error estimator with central cubic Hermite interpolation, the Jacobian is

$$J = \begin{pmatrix} 1 & \mu \\ -\frac{1}{k} & 1 - \frac{1}{k} \partial_{\chi_n} \log |e_{n+1}| \end{pmatrix}, \qquad \mu = \operatorname{Re}\left(\frac{R'(z)}{R(z)} z \right), \tag{44}$$

where $\eta_n = \log |u^n|$, $\chi_n = \log \Delta t_n$, and $z = \lambda \Delta t_n$. We have not been able to compute the expression $\partial_{\chi_n} |e_{n+1}|$ analytically in this case. However, we can evaluate it numerically by



(a) I controller with $\beta = (1, 0, 0)$, corresponding to the Jacobians (41) and (23).

(b) PI controller with $\beta = (0.6, -0.2, 0)$, corresponding to the Jacobians (43) and (23).

Fig. 2 Spectral radius of the Jacobian J for error estimates based on the L^1 residual and the second-order embedded method for the third-order method of Bogacki and Shampine [9] with left-biased cubic Hermite interpolation. The Jacobian is evaluated at $z = re^{i\varphi}$ where the radius r is chosen such that z is on the boundary of the stability region of the (main) method

using an adaptive Gauss-Kronrod quadrature with relative tolerance 10^{-8} and absolute tolerance 10^{-14} implemented in QuadGK.jl [29] for the integrals

$$\begin{aligned} |e_{n+1}| &= \int_0^{\Delta t_n} h(t)\mathrm{d}t, \qquad h(t) := \frac{1}{6} t(\Delta t_n - t)\left|t - 2\Delta t_n + t\Delta t_n \lambda\right| |\lambda|^4 |u^n|, \\ \partial_{\chi_n} |e_{n+1}| &= \left(h(\Delta t_n) + \int_0^{\Delta t_n} \frac{\partial h(t)}{\partial \Delta t_n} \mathrm{d}t \right) \Delta t_n, \end{aligned} \tag{45}$$

where the derivatives are evaluated using ForwardDiff.jl [30].

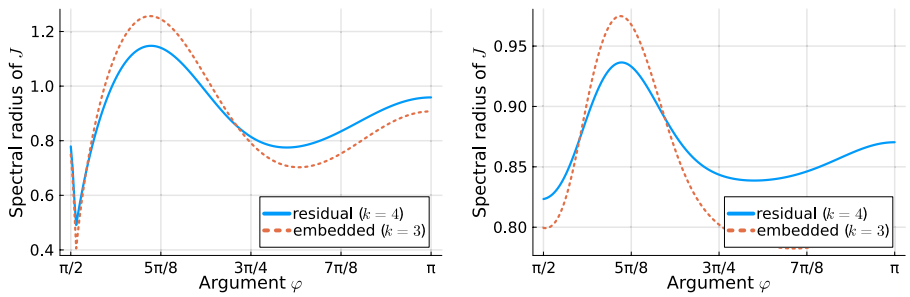
The spectral radii of the Jacobians are visualized in Fig. 3. Again, both methods do not lead to step size control stability for the simple I controller. It is interesting to see that the general trend of the spectral radius is similar to the one computed for the left-biased cubic Hermite interpolation shown in Fig. 2. However, the spectral radius does not exceed unity near $\varphi = \pi$ for the central interpolation (while still being close to unity). This is in accordance with numerical results presented later in Section 4.1.

Using the same test problem (35) as for second-order, two-stage methods results in 1318 accepted and 120 rejected steps for the the embedded approach while the residual-based approach leads to 1327 accepted and 21 rejected steps, see Table 2. This is in accordance with the spectral radius of the residual-based approach exceeding unity less than the embedded approach.

The expression of the Jacobian (22) of the PI controller with residual error estimator becomes

$$J = \begin{pmatrix} 1 & \mu & 0 & 0 \\ -\frac{\beta_1}{k} & 1 - \frac{\beta_1}{k} \frac{\partial |e_{n+1}|}{\partial \chi_n} & -\frac{\beta_2}{k} & -\frac{\beta_2}{k} \frac{\partial |e_n|}{\partial \chi_{n-1}} \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \quad \mu = \operatorname{Re}\left(\frac{z + z^2 + z^3/2}{1 + z + z^2/2 + z^3/6} \right). \tag{46}$$

The spectral radii of the Jacobians for the PI controller with parameters $\beta = (0.6, -0.2, 0.0)$ recommended in [16] are visualized in Fig. 3. Clearly, the more involved controller leads to step size control stability for all approaches. For the



(a) I controller with $\beta = (1, 0, 0)$, corresponding to the Jacobians (44) and (23).

(b) PI controller with $\beta = (0.6, -0.2, 0)$, corresponding to the Jacobians (46) and (23).

Fig. 3 Spectral radius of the Jacobian J for error estimates based on the L^1 residual and the second-order embedded method for the third-order method of Bogacki and Shampine [9] with central cubic Hermite interpolation. The Jacobian is evaluated at $z = re^{i\varphi}$ where the radius r is chosen such that z is on the boundary of the stability region of the (main) method

test problem (35), we get 1330 accepted and 1 rejected steps for the the embedded approach while the residual-based approach leads to 1333 accepted and 1 rejected steps.

Finally, we are able to compute the L^2 error estimate for the central cubic Hermite interpolation (38) analytically, resulting in

$$|e_{n+1}| = \sqrt{\Delta t_n}\|R\|_{L^2(t^n,t^{n+1})} = \frac{1}{6\sqrt{105}}\Delta t_n^4|\lambda|^4|u^n|\sqrt{8+\Delta t^2|\lambda|^2-5\Delta t_n\mathrm{Re}(\lambda)}. \quad (47)$$

Then, the expression of the Jacobian (19) of the I controller with L^2 residual error estimator becomes

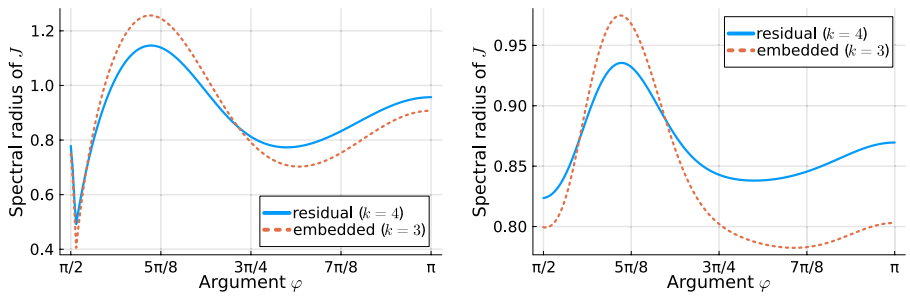
$$\begin{aligned} J &= \begin{pmatrix} 1 & \mu \\ -\frac{1}{k} & J_{22} \end{pmatrix}, \\ \mu = \mathrm{Re}\left(\frac{z+z^2+z^3/2}{1+z+z^2/2+z^3/6}\right),\ J_{22} &= 1-\frac{64+10|z|^2-45\mathrm{Re}(z)}{2k\left(8+|z|^2-5\mathrm{Re}(z)\right)}. \end{aligned} \quad (48)$$

The spectral radii of the Jacobians are visualized in Fig. 4. The spectral radii of the L^2 error estimator with central cubic Hermite interpolation (38) exceed unity less than the spectral radii of the L^1 error estimator (11) with left-biased cubic Hermite interpolation (39). However, they still do not lead to step size control stability.

The expression of the Jacobian (22) of the PI controller with L^2 residual error estimator becomes

$$\begin{aligned} J &= \begin{pmatrix} 1 & \mu & 0 & 0 \\ -\frac{\beta_1}{k} & 1-\beta_1\alpha & -\frac{\beta_2}{k} & -\beta_2\alpha \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \\ \mu = \mathrm{Re}\left(\frac{z+z^2+z^3/2}{1+z+z^2/2+z^3/6}\right),\ \alpha &= \frac{64+10|z|^2-45\mathrm{Re}(z)}{2k(8+|z|^2-5\mathrm{Re}(z)}. \end{aligned} \quad (49)$$

The spectral radii of the Jacobians for the PI controller with parameters $\beta = (0.6, -0.2, 0.0)$ recommended in [16] are visualized in Fig. 4. Clearly, the more involved



(a) I controller with $\beta = (1, 0, 0)$, corresponding to the Jacobians (48) and (23).

(b) PI controller with $\beta = (0.6, -0.2, 0)$, corresponding to the Jacobians (49) and (23).

Fig. 4 Spectral radius of the Jacobian J for error estimates based on the L^2 residual and the second-order embedded method for the third-order method of Bogacki and Shampine [9] with central cubic Hermite interpolation. The Jacobian is evaluated at $z = re^{i\varphi}$ where the radius r is chosen such that z is on the boundary of the stability region of the (main) method

controller leads to step size control stability for all approaches. Again, the L^2 error estimator with central cubic Hermite interpolation leads to more damping around $\varphi = 5\pi/8$ than the L^1 version with left-biased interpolation.

Using the L^2 error estimator with central cubic Hermite interpolation but otherwise the same setup as before, we get 1326 accepted and 25 rejected steps for the I controller and 1333 accepted and 1 rejected steps for the PI controller, see Table 2.

4 Numerical experiments

We have implemented all methods in Julia [31] and use OrdinaryDiffEq.jl [32] for the classical schemes with embedded methods. We use an adaptive Gauss-Kronrod quadrature implemented in QuadGK.jl [29] to compute the integrals appearing in the residual error estimators; we set the relative error tolerance to 10^{-8} for the ODE tests and to 10^{-6} for the tests involving partial differential equations (PDEs). The absolute and relative tolerances of the step size controller are equal, $\tau_a = \tau_r = \tau$. We use an adaptation of the algorithm of [20, Section II.4] to determine the initial time step size. We use FFTW.jl [33] via the interface provided by SummationByPartsOperators.jl [34] for Fourier collocation methods and Trixi.jl [35, 36] for discontinuous Galerkin discretizations of conservation laws. Finally, we use Plots.jl [37] to visualize the results. We use EPS control for all experiments. All source code required to reproduce the numerical experiments is available in our reproducibility repository [19].

First, we test the step size control stability theory with a nonlinear ODE in Section 4.1. Thereafter, we study the methods in the two regimes important for step size control of explicit time integration schemes: the asymptotic regime of small time step sizes and the stability-limited regime. We choose a classical ODE problem (Section 4.2) and a non-stiff PDE (Section 4.3) for the asymptotic regime. Afterwards, we consider hyperbolic conservation laws to study the stability-limited regime in Sections 4.5 and 4.6.

In all cases, we just show numerical results for the L^1 residual error estimates. The corresponding results based on L^2 error estimates are very similar (and can also be reproduced using the code of our reproducibility repository [19]).

4.1 A nonlinear ODE

First, we follow [8] and consider the nonlinear test problem

$$u' = -Bu + U^T(z_1^2/2 - z_2^2/2, z_1z_2, z_3^2, z_4^2)^T, \quad u(0) = (0, -2, -1, -1)^T, \tag{50}$$

Table 2 Number of accepted and rejected time steps of the Runge-Kutta pair of Bogacki and Shampine [9] for the test problem (35) with tolerances $\tau_a = \tau_r = 10^{-4}$ using the central cubic Hermite interpolation for the residual error estimators

	I controller, $\beta = (1, 0, 0)$			**PI controller, $\beta = (0.6, -0.2, 0)$**		
	Residual Estimator		**Embedded**	**Residual Estimator**		**Embedded**
	L^1	L^2	**Method**	L^1	L^2	**Method**
k	4	4	3	4	4	3
Accepted	1327	1326	1318	1333	1333	1330
Rejected	21	25	120	1	1	1

of Krogh [38] with

$$z = Uu, \quad B = U^T \begin{pmatrix} -10\cos(\varphi) & -10\sin(\varphi) & 0 & 0 \\ 10\sin(\varphi) & -10\cos(\varphi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1/2 \end{pmatrix} U, \tag{51}$$

where $U \in \mathbb{R}^{4\times 4}$ contains on the diagonal and $+1/2$ in all other components. For a fixed parameter φ, the dominant eigenvalues of the Jacobian for $t \to \infty$ become $-10|\cos(\varphi)| \pm \mathrm{i}10\sin(\varphi)$.

We integrate the problem in the time interval $[0, 100]$ using the third-order method of Bogacki and Shampine [9] with both the embedded method and the L^1 residual error estimator. The number of rejected steps for the simple I controller and the PI controller given by $\beta = (0.6, -0.2)$ are shown in Fig. 5. Clearly, a significant number of steps is rejected when the parameter φ is in the region around $\varphi = 5\pi/8$ where the simple I controller is not stable. In contrast, the advanced PI controller leads to at most one or two rejected steps for all values of φ.

4.2 Euler equations of a rigid body

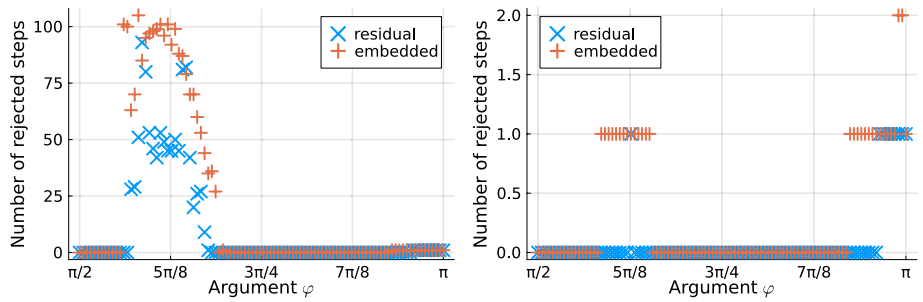
We consider the Euler equations of a rigid body with parameters used by Krogh [38], i.e.,

$$u'(t) = \begin{pmatrix} u_2 u_3 \\ -u_1 u_3 \\ -0.51 u_1 u_2 \end{pmatrix}, \quad u(0) = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}. \tag{52}$$

The solution is periodic with periodic given by Krogh [38]. We compute the ℓ^2 error after one period.

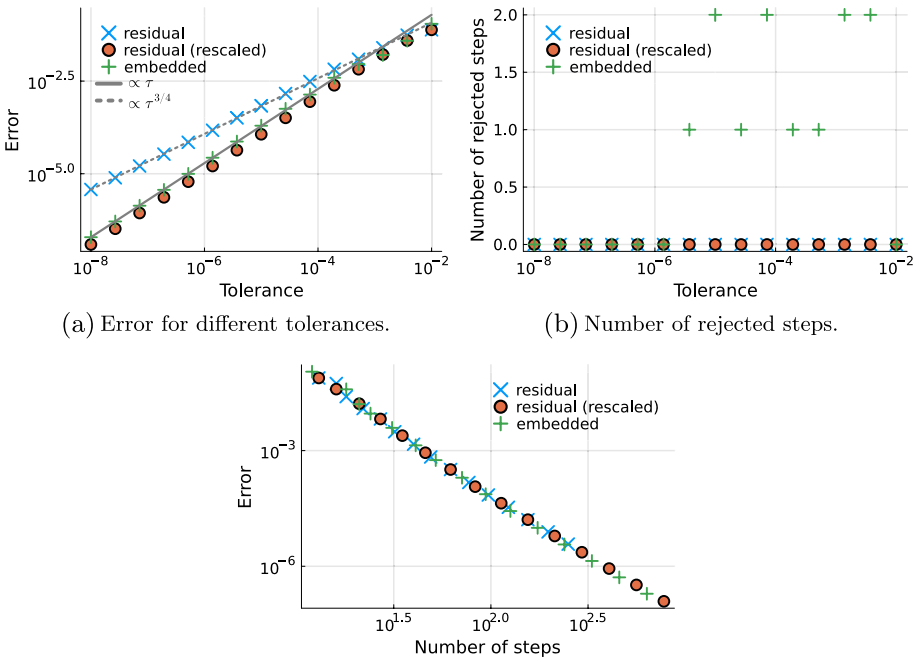
Numerical results obtained by the third-order method of Bogacki and Shampine [9] with both the embedded method and the L^1 residual error estimator are shown in Fig. 6. First, there are no issues with step rejections for this problem. Indeed, the residual-based approach leads to no step rejections and the embedded method rejects at most one or two steps for a few tolerances.

Next, we can observe the expected tolerance proportionality for the embedded method with EPS control. We also observe the expected scaling of the global error as $\tau^{p/(p+1)} = \tau^{3/4}$ for a given tolerance τ and the residual-based approach with EPS



(a) I controller, $\beta = (1, 0)$. (b) PI controller, $\beta = (0.6, -0.2)$.

Fig. 5 Number of rejected steps for the nonlinear ODE (50) using the the third-order method of Bogacki and Shampine [9] with tolerance $\tau = 10^{-4}$



(a) Error for different tolerances.

(b) Number of rejected steps.

(c) Work precision diagram.

Fig. 6 Results for the Euler equations of a rigid body using the third-order method of Bogacki and Shampine [9] with PI controller parameters $\beta = (0.6, -0.2)$. Since all results use an EPS control, we also rescale the tolerance to achieve tolerance proportionality for the residual-based approach

control. Thus, we can achieve tolerance proportionality by rescaling the tolerances as demonstrated in Fig. 6. Nevertheless, the residual-based approach with EPS control leads to tolerance convergence, even without rescaling.

Furthermore, the efficiency measured as the error for a fixed number of (accepted plus rejected) steps is the same for all variants. Indeed, all approaches lead to the same behavior in a classical work precision diagram. Thus, both approaches lead to *tolerance convergence* (the error goes to zero for $\tau \to 0$) and *computational stability* (small changes in the tolerance lead to small changes of the numerical results), see [39] for a discussion of these properties.

4.3 1D Benjamin-Bona-Mahony equation

Next, we consider the Benjamin-Bona-Mahony (BBM) equation [40] (also known as regularized long wave equation)

$$\begin{aligned} (\mathrm{I} - \partial_x^2)\partial_t u(t,x) + \partial_x \frac{u(t,x)^2}{2} + \partial_x u(t,x) = 0, \\ u(0,x) = u^0(x), \end{aligned} \tag{53}$$

with periodic boundary conditions as a model of nonlinear dispersive wave equations. We use the Fourier collocation semidiscretization of [41] conserving discrete versions of the linear and quadratic invariants

$$\int u(t,x)\mathrm{d}x, \quad \int \left(u(t,x)^2 + \left(\partial_x u(t,x)\right)^2\right)\mathrm{d}x. \tag{54}$$

Due to the dispersive term $\partial_x^2 \partial_t u$ with mixed space and time derivatives, the semidiscretization yields a non-stiff ODE with CFL restriction of the form $\Delta t \lesssim \text{const}$.

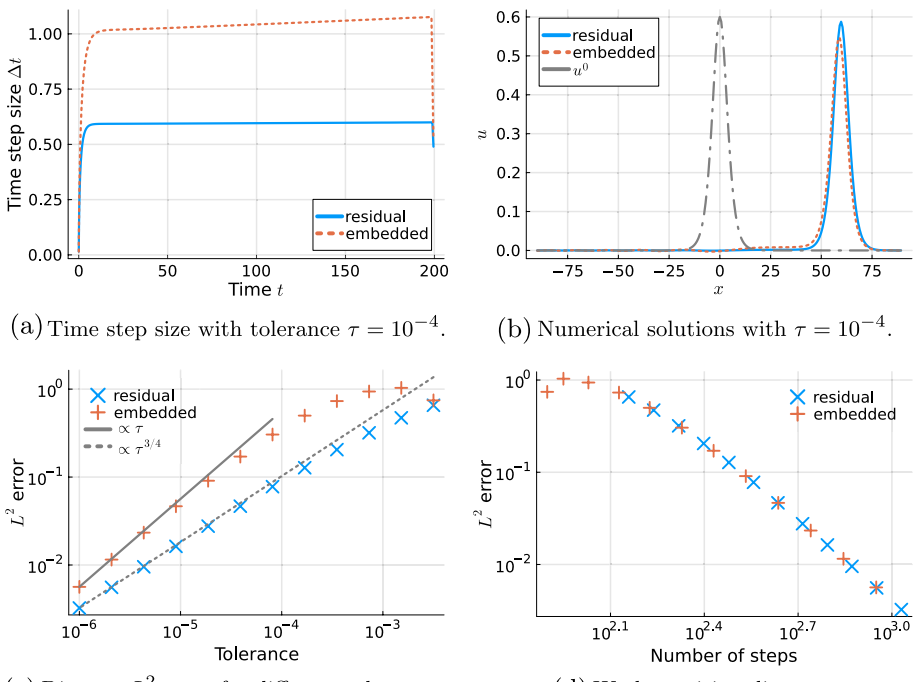
We consider the traveling wave solution

$$u(t,x) = A\cosh(K(x-ct)), \quad A = 3(c-1), \quad K = \frac{1}{2}\sqrt{1-1/c}, \tag{55}$$

with speed $c = 1.2$ in the periodic domain $[-90, 90]$ and integrate the semidiscretization with the third-order method of Bogacki and Shampine [9] in a time interval big enough so that the wave traverses the domain a bit more than once.

The results are visualized in Fig. 7. The residual-based approach leads to the expected behavior of a short transient period and a constant time step size afterwards. The embedded method behaves similarly but leads to a slowly growing time step size. The reason for this appears to be that it takes slightly bigger time steps, leading to more dissipation of the numerical solution. This in turn allows to take even bigger time steps, leading to a visibly more dissipated numerical solution.

Both methods lead to an expected convergence behavior for stricter tolerances. In particular, we get the tolerance proportionality for the embedded method and a scaling of the global error as $\tau^{p/(p+1)} = \tau^{3/4}$ for the residual-based approach with EPS control. Moreover, both methods lead to the same behavior in a work precision diagram measuring the discrete L^2 error at the final time for a given number of steps — while both methods lead to no rejected steps in this case.



(a) Time step size with tolerance $\tau = 10^{-4}$.

(b) Numerical solutions with $\tau = 10^{-4}$.

(c) Discrete L^2 error for different tolerances.

(d) Work precision diagram.

Fig. 7 Results for the BBM equation with Fourier semidiscretization in space and the third-order method of Bogacki and Shampine [9] in time with PI controller parameters $\beta = (0.6, -0.2)$

4.4 Reliability of the global error estimate

Next, we study the reliability of the global a posteriori error estimate. We consider the linear equation

$$u'(t) = u(t),\ t \in (0,1), \qquad u(0) = 1, \tag{56}$$

with (one-sided) Lipschitz constant $L = 1$. Furthermore, we consider the nonlinear problem

$$u'(t) = \exp\big(-u(t)\big),\ t \in (0,100), \qquad u(0) = 1, \tag{57}$$

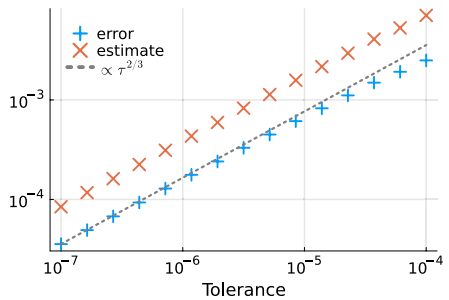
with one-sided Lipschitz constant $L = 0$ (since $u \mapsto \mathrm{e}^{-u}$ is monotonically decreasing).

We measure the error of numerical solutions at the final time and use the residual-based L^2 error estimate for EPS control of the time step size with the PID controller given by $\beta = (0.6, -0.2)$. The results for Heun's second-order method and the third-order method of Bogacki and Shampine [9] are shown in Fig. 8. First, we observe the expected scaling of the global error as $\tau^{p/(p+1)}$ in all cases. Moreover, we see that the global error estimate of Lemma 1 is reliable since it yields an upper bound on the error in all cases.

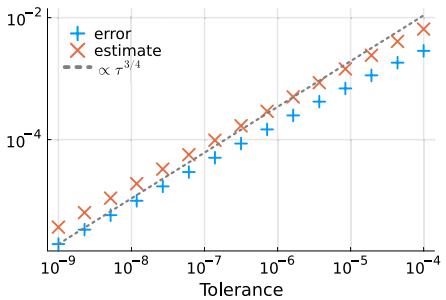
4.5 2D linear advection

We consider the 2D linear advection equation

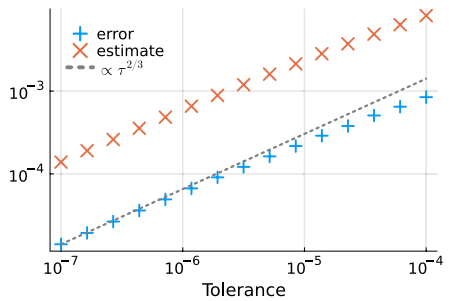
$$\partial_t u + \operatorname{div}(au) = 0 \tag{58}$$



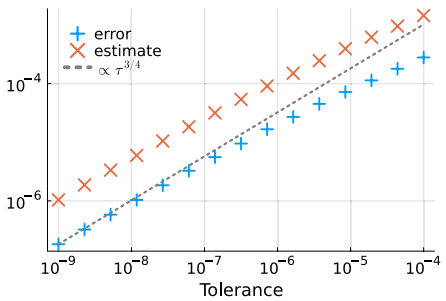
(a) Linear problem (56), second-order method of Heun.



(b) Linear problem (56), third-order method of Bogacki and Shampine [9].



(c) Nonlinear problem (57), second-order method of Heun.



(d) Nonlinear problem (57), third-order method of Bogacki and Shampine [9].

Fig. 8 Comparison of the global error estimate at the final time obtained from Lemma 1 with residual-based L^2 estimates and one-sided Lipschitz constants. The time step size is adapted using the same error estimates and the PI controller with parameters $\beta = (0.6, -0.2)$

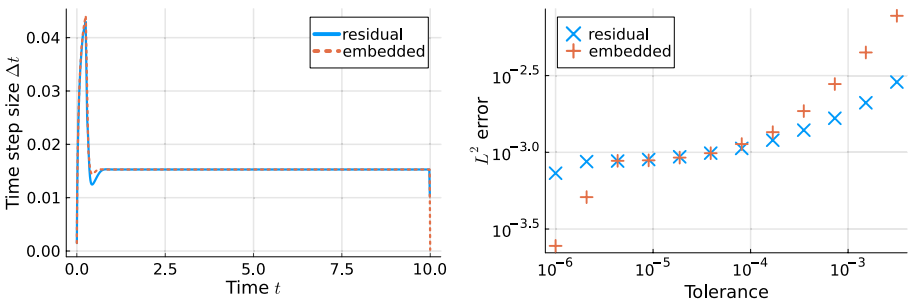
with periodic boundary conditions in $[-1,1]^2$, the advection velocity $a = (1,1)^T$, and a sinusoidal initial condition. Using the method of lines approach, we discretize the PDE first in space with a discontinuous Galerkin spectral element method (DGSEM) on Gauss-Lobatto-Legendre nodes representing polynomials of degree $p = 4$; an introduction to such nodal DG methods is given in the textbooks [42, 43]. We divide the domain into 8^2 uniform elements and apply the local Lax-Friedrichs/Rusanov flux at interfaces. Finally, we integrate the resulting ODE in time using the third-order method of Bogacki and Shampine [9] with PI controller parameters $\beta = (0.6, -0.2)$.

As discussed in [15, 16], a typical behavior for such problems is as follows. Initially, the time step size varies a bit and quickly converges to a constant. There is usually a range of tolerances where the time step size Δt is restricted by stability. In this regime, error-based step size control with stable controllers results in optimal time step sizes that can also be obtained by manually optimizing a Courant-Friedrichs-Lewy (CFL) factor. This behavior is shown in Fig. 9; the embedded method and the residual-based error estimator behave very similarly. In particular, both methods lead to at most three rejected steps for loose tolerances.

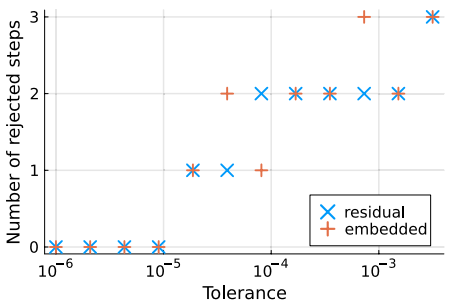
4.6 3D inviscid Taylor-Green vortex

Next, we consider the ideal 3D compressible Euler equations to simulate an inviscid Taylor-Green vortex. We choose the initial condition

$$\begin{aligned} &\varrho = 1,\ v_1 = \sin(x_1)\cos(x_2)\cos(x_3),\ v_2 = -\cos(x_1)\sin(x_2)\cos(x_3),\ v_3 = 0, \\ &p = \frac{\varrho}{\mathrm{Ma}^2\gamma} + \varrho\frac{\cos(2x_1)\cos(2x_3) + 2\cos(2x_2) + 2\cos(2x_1) + \cos(2x_2)\cos(2x_3)}{16}, \end{aligned} \tag{59}$$



(a) Time step size with tolerance $\tau = 10^{-4}$. (b) Discrete L^2 error for different tolerances.



(c) Number of rejected steps.

Fig. 9 Results for the linear advection problem with DGSEM semidiscretization in space and the third-order method of Bogacki and Shampine [9] in time with PI controller parameters $\beta = (0.6, -0.2)$

where Ma = 0.1 is the Mach number, ϱ the density, v the velocity, and p the pressure. We consider the domain $[-\pi, \pi]^3$ with periodic boundary conditions and a time interval [0, 10]. We use the entropy-stable semidiscretization with flux differencing DGSEM with polynomials of degree $p = 3$, the entropy-conservative flux of [44–46] in the volume and a local Lax-Friedrichs/Rusanov numerical flux at interfaces. This kind of flux differencing discretizations is described in [47, 48]; see also [49].

The results of a simulation with 8^3 elements and a time integration tolerance $\tau = 10^{-5}$ are shown in Fig. 10. In this case, the time step size Δt is again restricted by stability constraints. The time step sizes chosen by the different estimators — the embedded method and the L^1 residual estimate — are visually indistinguishable. The residual-based approach leads to 3 step rejections while the embedded method leads to 2 rejected steps.

5 Summary and conclusions

We have analyzed stability of step size control of explicit Runge-Kutta methods for ODEs using residual-based a posteriori error estimators of [18] when step sizes are dictated by stability and not accuracy. It turned out that the situation is comparable to the case of embedded methods, i.e., the classical I controller does not lead to step size control stability while more advanced PI and PID controllers can be designed to be stable. We have analyzed the situation for ODEs and demonstrated that the findings extend to some PDEs discretized using the method of lines. In particular, we have considered the nonlinear dispersive Benjamin-Bona-Mahony equation as well as discontinuous Galerkin semidiscretizations of the 2D linear advection and 3D compressible Euler equations.

The general behavior of the methods using estimates obtained from the residual-based approach and embedded methods was comparable in the numerical experiments. Thus, the main difference is that the residual-based approach leads additionally to a rigorous a posteriori estimate of the global error while the embedded methods are computationally cheaper.

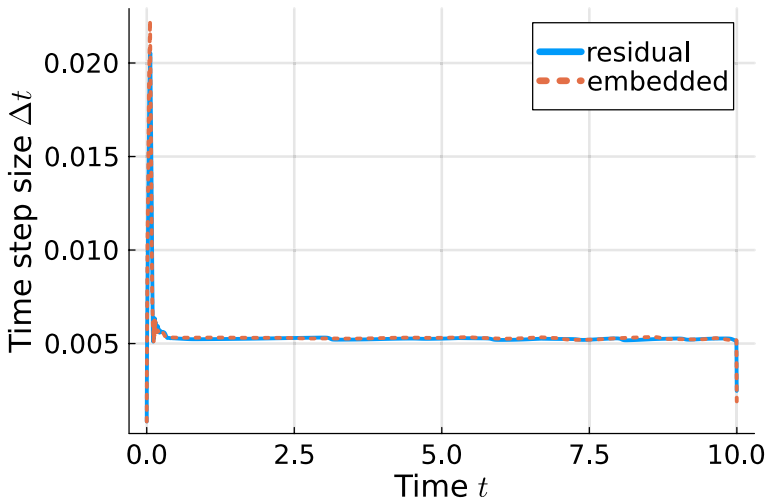


Fig. 10 Results for the 3D compressible Euler equations setup for an inviscid Taylor-Green vortex with entropy-stable flux differencing DGSEM in space and the third-order method of Bogacki and Shampine [9] in time with PI controller parameters $\beta = (0.6, -0.2)$ and tolerance $\tau = 10^{-5}$

Authors' contributions
Conceptualization: H.R., J.G.; Data curation: H.R.; Formal analysis and investigation: H.R.; Funding acquisition: H.R.; Visualization: H.R.; Writing - original draft preparation: H.R.; Writing - review and editing: H.R., J.G.

Funding
Open Access funding enabled and organized by Projekt DEAL. HR was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation, project number 513301895) and the Daimler und Benz Stiftung (Daimler and Benz foundation, project number 32-10/22).

Code Availability
The source code required to reproduce the numerical experiments presented in this article is archived in our reproducibility repository available at https://doi.org/10.5281/zenodo.8177157 and https://github.com/ranocha/2023_RK_error_estimate.

Availability of data and materials
The source code and datasets generated and analyzed for this study are available in our reproducibility repository available at https://doi.org/10.5281/zenodo.8177157 and https://github.com/ranocha/2023_RK_error_estimate.

Declarations

Competing interests
The authors declare no competing interests.

Received: 11 December 2023 Accepted: 8 March 2024
Published online: 23 September 2024

References

1. Dormand JR, Prince PJ. A family of embedded Runge-Kutta formulae. J Comput Appl Math. 1980;6(1):19–26. https://doi.org/10.1016/0771-050X(80)90013-3.
2. Becker R, Gantner G, Innerberger M, Praetorius D. Goal-oriented adaptive finite element methods with optimal computational complexity. Numer Math. 2023;153(1):111–40. https://doi.org/10.1007/s00211-022-01334-8.
3. Kreuzer C, Möller CA, Schmidt A, Siebert KG. Design and convergence analysis for an adaptive discretization of the heat equation. IMA J Numer Anal. 2012;32(4):1375–403. https://doi.org/10.1093/imanum/drr026.
4. Lakkis O, Makridakis C, Pryer T. A comparison of duality and energy a posteriori estimates for $L_\infty(0,T;L_2(\Omega))$ in parabolic problems. Math Comput. 2015;84(294):1537–69. https://doi.org/10.1090/S0025-5718-2014-02912-8.
5. Hall G. Equilibrium states of Runge Kutta schemes. ACM Trans Math Softw (TOMS). 1985;11(3):289–301. https://doi.org/10.1145/214408.214424.
6. Hall G. Equilibrium states of Runge-Kutta schemes: part II. ACM Trans Math Softw (TOMS). 1986;12(3):183–92. https://doi.org/10.1145/7921.7922.
7. Hall G, Higham DJ. Analysis of stepsize selection schemes for Runge-Kutta codes. IMA J Numer Anal. 1988;8(3):305–10. https://doi.org/10.1093/imanum/8.3.305.
8. Higham DJ, Hall G. Embedded Runge-Kutta formulae with stable equilibrium states. J Comput Appl Math. 1990;29(1):25–33. https://doi.org/10.1016/0377-0427(90)90192-3.
9. Bogacki P, Shampine LF. A 3(2) pair of Runge-Kutta formulas. Appl Math Lett. 1989;2(4):321–5. https://doi.org/10.1016/0893-9659(89)90079-7.
10. Bogacki P, Shampine LF. An efficient Runge-Kutta (4,5) pair. Comput Math Appl. 1996;32(6):15–28. https://doi.org/10.1016/0898-1221(96)00141-1.
11. Gustafsson K, Lundh M, Söderlind G. A PI stepsize control for the numerical solution of ordinary differential equations. BIT Numer Math. 1988;28(2):270–87. https://doi.org/10.1007/BF01934091.
12. Gustafsson K. Control theoretic techniques for stepsize selection in explicit Runge-Kutta methods. ACM Trans Math Softw (TOMS). 1991;17(4):533–54. https://doi.org/10.1145/210232.210242.
13. Söderlind G. Time-step selection algorithms: Adaptivity, control, and signal processing. Appl Numer Math. 2006;56(3–4):488–502. https://doi.org/10.1016/j.apnum.2005.04.026.
14. Söderlind G, Wang L. Adaptive time-stepping and computational stability. J Comput Appl Math. 2006;185(2):225–43. https://doi.org/10.1016/j.cam.2005.03.008.
15. Ranocha H, Winters AR, Castro HG, Dalcin L, Schlottke-Lakemper M, Gassner GJ, et al. On error-based step size control for discontinuous Galerkin methods for compressible fluid dynamics. Commun Appl Math Comput. 2023. https://doi.org/10.1007/s42967-023-00264-y.
16. Ranocha H, Dalcin L, Parsani M, Ketcheson DI. Optimized Runge-Kutta Methods with Automatic Step Size Control for Compressible Computational Fluid Dynamics. Commun Appl Math Comput. 2021;4:1191–228. https://doi.org/10.1007/s42967-021-00159-w.
17. Courant R, Friedrichs KO, Lewy H. On the partial difference equations of mathematical physics. IBM J Res Dev. 1967;11(2):215–34.
18. Dedner A, Giesselmann J. A posteriori analysis of fully discrete method of lines discontinuous Galerkin schemes for systems of conservation laws. SIAM J Numer Anal. 2016;54(6):3523–49. https://doi.org/10.1137/15M1046265.
19. Ranocha H, Giesselmann J. Reproducibility repository for "Stability of step size control based on a posteriori error estimates". 2023. https://doi.org/10.5281/zenodo.8177157.
20. Hairer E, Nørsett SP, Wanner G. Solving Ordinary Differential Equations I: Nonstiff Problems. vol. 8 of Springer Series in Computational Mathematics. Berlin Heidelberg: Springer-Verlag; 2008. https://doi.org/10.1007/978-3-540-78862-1.

21. Butcher JC. Numerical Methods for Ordinary Differential Equations. Chichester: Wiley; 2016. https://doi.org/10.1002/9781119121534.
22. Kennedy CA, Carpenter MH, Lewis RM. Low-storage, explicit Runge-Kutta schemes for the compressible Navier-Stokes equations. Appl Numer Math. 2000;35(3):177–219. https://doi.org/10.1016/S0168-9274(99)00141-5.
23. Dragomir SS. Some Gronwall type inequalities and applications. Hauppauge: Nova Science Publishers; 2003.
24. Makridakis C, Nochetto RH. A posteriori error analysis for higher order dissipative methods for evolution problems. Numer Math. 2006;104(4):489–514. https://doi.org/10.1007/s00211-006-0013-6.
25. Makridakis C. Space and time reconstructions in a posteriori analysis of evolution problems. ESAIM Proc. 2007;21:31–44. https://doi.org/10.1051/proc:072104.
26. Higham DJ. Global error versus tolerance for explicit Runge-Kutta methods. IMA J Numer Anal. 1991;11(4):457–80. https://doi.org/10.1093/imanum/11.4.457.
27. Hairer E, Wanner G. Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems. vol. 14 of Springer Series in Computational Mathematics. Berlin Heidelberg: Springer-Verlag; 2010. https://doi.org/10.1007/978-3-642-05221-7.
28. Wolfram Research, Inc. Mathematica. 2019. https://www.wolfram.com.
29. Johnson SG. QuadGK.jl: Gauss–Kronrod integration in Julia. 2013. https://github.com/JuliaMath/QuadGK.jl. Accessed 20 July 2023.
30. Revels J, Lubin M, Papamarkou T. Forward-Mode Automatic Differentiation in Julia. 2016. https://doi.org/10.48550/arXiv.1607.07892. Accessed 20 July 2023.
31. Bezanson J, Edelman A, Karpinski S, Shah VB. Julia: A Fresh Approach to Numerical Computing. SIAM Rev. 2017;59(1):65–98. https://doi.org/10.1137/141000671.
32. Rackauckas C, Nie Q. DifferentialEquations.jl – A Performant and Feature-Rich Ecosystem for Solving Differential Equations in Julia. J Open Res Softw. 2017;5(1):15. https://doi.org/10.5334/jors.151.
33. Frigo M, Johnson SG. The design and implementation of FFTW3. Proc IEEE. 2005;93(2):216–31. https://doi.org/10.1109/JPROC.2004.840301.
34. Ranocha H. SummationByPartsOperators.jl: A Julia library of provably stable semidiscretization techniques with mimetic properties. J Open Source Softw. 2021;6(64):3454. https://doi.org/10.21105/joss.03454.
35. Ranocha H, Schlottke-Lakemper M, Winters AR, Faulhaber E, Chan J, Gassner GJ. Adaptive numerical simulations with Trixi.jl: A case study of Julia for scientific computing. Proc JuliaCon Conf. 2022;1(1):77. https://doi.org/10.21105/jcon.00077.
36. Schlottke-Lakemper M, Winters AR, Ranocha H, Gassner GJ. A purely hyperbolic discontinuous Galerkin approach for self-gravitating gas dynamics. J Comput Phys. 2021;442:110467. https://doi.org/10.1016/j.jcp.2021.110467.
37. Christ S, Schwabeneder D, Rackauckas C, Borregaard MK, Breloff T. Plots.jl — a user extendable plotting API for the Julia programming language. J Open Res Softw. 2023. https://doi.org/10.5334/jors.431.
38. Krogh FT. On testing a subroutine for the numerical integration of ordinary differential equations. J ACM. 1973;20(4):545–62. https://doi.org/10.1145/321784.321786.
39. Söderlind G. Automatic control and adaptive time-stepping. Numer Algoritm. 2002;31(1–4):281–310. https://doi.org/10.1023/A:1021160023092.
40. Benjamin TB, Bona JL, Mahony JJ. Model equations for long waves in nonlinear dispersive systems. Philos Trans R Soc Lond Ser A Math Phys Sci. 1972;272(1220):47–78. https://doi.org/10.1098/rsta.1972.0032.
41. Ranocha H, Mitsotakis D, Ketcheson DI. A Broad Class of Conservative Numerical Methods for Dispersive Wave Equations. Commun Comput Phys. 2021;29(4):979–1029. https://doi.org/10.4208/cicp.OA-2020-0119.
42. Hesthaven JS, Warburton T. Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications. vol. 54 of Texts in Applied Mathematics. New York: Springer Science & Business Media; 2007. https://doi.org/10.1007/978-0-387-72067-8.
43. Kopriva DA. Implementing Spectral Methods for Partial Differential Equations: Algorithms for Scientists and Engineers. New York: Springer Science & Business Media; 2009. https://doi.org/10.1007/978-90-481-2261-5.
44. Ranocha H. Comparison of Some Entropy Conservative Numerical Fluxes for the Euler Equations. J Sci Comput. 2018;76(1):216–42. https://doi.org/10.1007/s10915-017-0618-1.
45. Ranocha H. Entropy Conserving and Kinetic Energy Preserving Numerical Methods for the Euler Equations Using Summation-by-Parts Operators. In: Sherwin SJ, Moxey D, Peiró J, Vincent PE, Schwab C, editors. Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2018. vol. 134 of Lecture Notes in Computational Science and Engineering. Cham: Springer; 2020. pp. 525–35. https://doi.org/10.1007/978-3-030-39647-3_42.
46. Ranocha H, Gassner GJ. Preventing pressure oscillations does not fix local linear stability issues of entropy-based split-form high-order schemes. Commun Appl Math Comput. 2021. https://doi.org/10.1007/s42967-021-00148-z.
47. Fisher TC, Carpenter MH. High-order entropy stable finite difference schemes for nonlinear conservation laws: Finite domains. J Comput Phys. 2013;252:518–57. https://doi.org/10.1016/j.jcp.2013.06.014.
48. Gassner GJ, Winters AR, Kopriva DA. Split Form Nodal Discontinuous Galerkin Schemes with Summation-By-Parts Property for the Compressible Euler Equations. J Comput Phys. 2016;327:39–66. https://doi.org/10.1016/j.jcp.2016.09.013.
49. Ranocha H, Schlottke-Lakemper M, Chan J, Rueda-Ramirez AM, Winters AR, Hindenlang F, et al. Efficient implementation of modern entropy stable and kinetic energy preserving discontinuous Galerkin methods for conservation laws. 2021. https://doi.org/10.48550/arXiv.2112.10517.

Publisher's Note