Journal of Scientific Computing (2024) 99:28
https://doi.org/10.1007/s10915-024-02478-5

Many-Stage Optimal Stabilized Runge–Kutta Methods for Hyperbolic Partial Differential Equations

Daniel Doehring[1] · **Gregor J. Gassner**[2] · **Manuel Torrilhon**[1]

Received: 7 June 2023 / Revised: 13 October 2023 / Accepted: 30 January 2024 /
Published online: 16 March 2024

Abstract
A novel optimization procedure for the generation of stability polynomials of stabilized explicit Runge–Kutta methods is devised. Intended for semidiscretizations of hyperbolic partial differential equations, the herein developed approach allows the optimization of stability polynomials with more than hundred stages. A potential application of these high degree stability polynomials are problems with locally varying characteristic speeds as found for non-uniformly refined meshes and spatially varying wave speeds. To demonstrate the applicability of the stability polynomials we construct $2N$-storage many-stage Runge–Kutta methods that match their designed second order of accuracy when applied to a range of linear and nonlinear hyperbolic PDEs with smooth solutions. These methods are constructed to reduce the amplification of round off errors which becomes a significant concern for these many-stage methods.

Keywords Runge–Kutta methods · Absolute stability · Method of lines · Initial value problems

Mathematics Subject Classification 65L06 · 65M20

1 Stabilized Explicit Runge–Kutta Methods

Explicit Runge–Kutta methods are commonly considered the default choice for the integration of hyperbolic partial differential equations (PDEs). In contrast to implicit methods, explicit methods require only evaluations of the right-hand side instead of solving a potentially nonlinear system of equations. In fact, in the context of hyperbolic PDEs typically nonlinear fluxes are of interest. Explicit methods come with the drawback that the maximum

✉ Daniel Doehring
doehring@acom.rwth-aachen.de

[1] Applied and Computational Mathematics, RWTH Aachen University, Schinkelstrasse 2, 52062 Aachen, North Rhine-Westphalia, Germany

[2] Department of Mathematics and Computer Science, Center for Data and Simulation Science, University of Cologne, Weyertal 86-90, 50931 Cologne, North Rhine-Westphalia, Germany

stable timestep needs to be significantly reduced which is in the context of (hyperbolic) PDEs commonly referred to as the CFL condition [1].

To increase computational efficiency, *stabilized* explicit Runge–Kutta methods have been introduced already in the 1960s [2–4] targeting semidiscretizations

$$\boldsymbol{U}(t_0) = \boldsymbol{U}_0 \tag{1.1a}$$

$$\boldsymbol{U}'(t) = \boldsymbol{F}\big(\boldsymbol{U}(t)\big) \tag{1.1b}$$

of parabolic PDEs. The approach of discretizing the spatial derivatives and leaving time continuous at first is commonly referred to as the *Method of Lines* approach. Although originally termed for finite difference based spatial discretizations [5] the name persisted and is used nowadays to comprise different discretization techniques like finite elements, finite volumes, and discontinuous Galerkin.

The central idea of stabilized explicit Runge–Kutta methods is to use additional stages primarily to improve the stability properties of the scheme, i.e., allow for larger timesteps. In particular, one usually settles for a moderate order of accuracy and uses the additional degrees of freedom to improve the stability properties of the scheme for a certain spectrum [6–8]. The fact that the eigenvalues of the Jacobian

$$J(\boldsymbol{U}) := \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{U}} \tag{1.2}$$

are for parabolic PDES distributed on the negative real axis enables a successful construction of stabilized explicit Runge–Kutta methods optimized for this special case, see for instance the reviews [9, 10]. The fact that the optimal stability polynomial $P_{S,1}(z)$ of first order accurate methods with S stages is given by the shifted Chebyshev polynomial of first kind $T_S(z)$ [2, 3, 11, 12] serves as a valuable guidance to construct higher order approximately optimal stability polynomials [13–18] which stand on solid ground due to proven existence and uniqueness of optimal stability polynomials with maximum (negative) real axis inclusion [19].

For the purely hyperbolic case, i.e., for eigenvalues $\lambda \in \sigma(J)$ exclusively on the imaginary axis also results for first order accurate methods in terms of Chebyshev polynomials of first kind are known [16, 20–22]. Approximations to higher order accurate stability polynomials can be found in [23].

For the more general case, i.e., where not only either the real or complex line are of interest, but rather a two-dimensional part of the complex plane, concrete results for stability polynomials are rare. To the best of our knowledge, only for disks results for first [24] and second [25, 26] order accurate optimal stability polynomials of variable degree S are available. In contrast, one usually has to resort to numerical optimization of the stability polynomials. A nonextensive list includes optimized methods for certain geometrically defined spectra [27, 28], particular equations [29–32] and spatial discretization techniques [33–35].

A general framework for maximizing the region of absolute stability of an explicit Runge–Kutta methods for a particular spectrum has been developed in [36] which has been extensively used [33, 34, 37–44]. The approach presented in [36] optimizes the coefficients $\boldsymbol{\alpha} \in \mathbb{R}$ of the stability polynomial

$$P(z; \boldsymbol{\alpha}) = \sum_{j=0}^{S} \alpha_j q_j(z) \tag{1.3}$$

where $\{q_j(z)\}_{j=0,\dots,S}$ forms a basis for the complex-valued polynomials with real coefficients. In the simplest case, q_j are given by the monomials z^j which lead for higher degrees S to an increasingly ill-conditioned optimization problem. This choice of basis and optimization in monomial coefficients is also employed in [27, 30, 31] where polynomials are optimized up to maximum degrees 14, 7, and 8, respectively. If a suitable basis $\{q_j(z)\}_{j=0,\dots,S}$ for the spectrum of interest can be found, stability polynomials of high degrees (up to $S = 80$ are reported in [36]) can be constructed. In [36], suitable bases are proposed for parabolic (negative real-line), purely hyperbolic (imaginary axis) and disk-spectra. In general, however, finding such a basis for a complicated real-world spectrum is a challenging task. Consequently, the works [33, 34, 37–44] employ the monomial basis which limits the maximum degree of the stability polynomials to 16-20.

In this work, we develop a formulation that avoids the search of a suitable basis. In particular, instead of parametrizing the polynomial in terms of the basis-coefficients, the stability polynomial $P(z)$ is characterized by the roots of $\big(P(z) - 1\big)/z$. This can be seen as a generalization of the approach employed in [28] wherein the stability polynomial is described by its extrema. Although this approach leads to a highly nonlinear formulation, it will be shown that for a relevant class of spectra an excellent initial guess can be supplied allowing the successful optimization of the highly nonlinear problem. By doing so, we are able to optimize stability polynomials of degrees larger than 100, which to the best of our knowledge is an unprecedented success for spectra of hyperbolic semidiscretizations. We emphasize that our approach involves for an efficient optimization the application of the coefficient-based method developed in [36] which forms thus an integral building block of our algorithm.

A potential application of these high degree stability polynomials are the recently published Paired-Explicit Runge–Kutta (P-ERK) methods [39, 40, 45] which achieve local time stepping effects by combining a set of stabilized methods. In particular, due to their construction as partitioned Runge–Kutta methods [46] they ensure consistency and conservation which are not trivially satisfied by other classical multirate methods [47]. Being based on the optimization approach developed in [36], the stability polynomials and corresponding P-ERK methods are currently limited to 16 stages. The availability of higher degree polynomials allows for a potentially even more efficient treatment of e.g. locally refined meshes or varying characteristic speeds. In this work, however, we focus on the optimization of high degree stability polynomials and methods that can be directly constructed thereof. The application to P-ERK methods is left for future work.

The paper is organized as follows. The motivation for the new optimization approach is presented in Sects. 2 and 3, based on findings for the proven optimal stability polynomials for disk-like spectra. The necessary details required for successful optimization of the problem are discussed in Sect. 4 which is followed by a discussion of the implementation in Sect. 5. In Sect. 6 a couple of high degree optimized stability polynomials are presented. Extensions of the original approach to non-convex spectra are covered in Sect. 7 and applied in Sect. 8. The construction of actual Runge–Kutta methods from the high-degree stability polynomials follows in Sect. 9 with a special focus on internal stability. Section 10 presents the application of the many-stage methods to linear and nonlinear problems. Section 11 concludes the paper.

Note on Terminology

For the sake of readability, we will refer to the shifted Chebyshev polynomials of first kind simply as Chebyshev polynomials. As the extreme points of the (shifted) Chebyshev polynomials (of first kind) play a crucial role in this work, we will call them in the interest of brevity simply Chebyshev extreme points. Furthermore, we will refer to the the Chebyshev extreme points with extremal value $+1$ as positive Chebyshev extreme points.

Since a significant part of the paper centers around stability polynomials from which Runge–Kutta methods of certain order may be constructed, we will say that a polynomial of degree S is of order/accuracy p whenever the first $i = 0, \dots, p$ coefficients match the first $i = 0, \dots, p$ coefficients of the Taylor series of the exponential.

2 Preliminaries

Runge–Kutta methods are single step methods, i.e., when applied to the constant coefficient linear initial value problem (IVP)

$$\boldsymbol{u}(t_0 = 0) = \boldsymbol{u}_0 \tag{2.1a}$$

$$\boldsymbol{u}'(t) = A\boldsymbol{u}(t) \tag{2.1b}$$

the new approximation $\boldsymbol{u}_{n+1}$ can be computed from the previous iterate $\boldsymbol{u}_n$ as

$$\boldsymbol{u}_{n+1} = R(\Delta t A)\boldsymbol{u}_n. \tag{2.2}$$

Here, $R(z)$ denotes the *stability function* [6] of the Runge–Kutta method. For implicit methods $R(z)$ is a rational function, while for explicit methods $R(z)$ is a polynomial with real coefficients

$$P_S(z; \boldsymbol{\alpha}) = \sum_{j=0}^{S} \alpha_j z^j, \quad \boldsymbol{\alpha} \in \mathbb{R}^{S+1}. \tag{2.3}$$

By comparing (2.2) to the solution $\boldsymbol{u}(t_{n+1}) = \exp(A\Delta t)\boldsymbol{u}(t_n)$ of (2.1b) it follows from the definition of the exponential that for a p'th order (linearly) consistent approximation the coefficients α_j need to satisfy

$$\alpha_j \overset{!}{=} \frac{1}{j!}, \quad j = 0, \dots, p. \tag{2.4}$$

It is customary to define the family of polynomials with real coefficients over the complex numbers of degree S and corresponding order of accuracy p as $\mathcal{P}_{S,p}$.

Due to the fact that the stability function $R(z)$ is a polynomial $P(z)$ for explicit methods it follows that the *region of absolute stability*

$$\mathcal{S} := \{z \in \mathbb{C} : |R(z)| \leq 1\} \tag{2.5}$$

is necessarily bounded. We further define the *boundary of the region of absolute stability* as

$$\partial\mathcal{S} := \{z \in \mathbb{C} : |R(z)| = 1\} \tag{2.6}$$

which will also be called in brief *stability boundary*.

2.1 Optimization Objective

The optimization objective is now to find the maximum possible timestep $\Delta t^\star_{S,p}$ for a polynomial of degree S corresponding to a method with order of accuracy p:

$$\max_{P_{S,p}\in\mathcal{P}_{S,p}} \Delta t \text{ such that } \left|P_{S,p}(\Delta t\lambda^{(m)})\right| \leq 1, \quad m = 1,\dots,M. \tag{2.7}$$

Here, $\{\lambda^{(m)}\}_{m=1,\dots,M} = \sigma(J)$ are the eigenvalues of the Jacobian (1.2) where the ordinary differential equation (ODE) system (1.1) corresponds in this work to the semidiscretization of PDEs describing typically physical processes. As a consequence, we assume that there are no amplifiying modes among the eigenvalues $\lambda^{(m)}$ corresponding to generation of energy, thus all eigenvalues should have non-positive real part:

$$\mathrm{Re}\big(\lambda\big) \leq 0 \qquad \forall\, \lambda \in \sigma(J). \tag{2.8}$$

It should be stressed that the maximum possible timestep $\Delta t^\star$ is the single optimization target in this work. In particular, we do not focus on the reduction of dispersion or dissipation errors as done for instance in [48–50] or other objectives like maximum strong stability preserving (SSP) coefficients [34, 51, 52].

2.1.1 Convex Problem Formulation

As noted in [36] (2.7) is for fixed timestep Δt a convex optimization problem when parametrizing the stability polynomial $P_{S,p}$ in terms of the coefficients $\boldsymbol{\alpha}$:

$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^{S-p}} \Delta t \text{ such that } \left|P_{S,p}(\Delta t\lambda^{(m)};\boldsymbol{\alpha})\right| \leq 1, \quad m = 1,\dots,M. \tag{2.9}$$

Paired with an outer bisection to determine the timestep, (2.9) leads to an efficient optimization routine that can be solved with standard software such as `SeDuMi` [53] or `ECOS` [54] for second order cone programs. The downside of formulation (2.9) is that, unless a suitable basis $\{q_j(z)\}_{j=0,\dots,S}$ for the stability polynomial can be found, one has to resort to simple choices such as the monomials. For these, the problem becomes increasingly ill-conditioned for higher degrees since the monomial coefficients scale essentially as $\alpha \sim 1/(j!)$, i.e., they become diminishingly small. Rescaling the coefficients as $\widetilde{\alpha_j} := \alpha_j/j!$ improves this issue slightly, but ultimately cannot solve this difficulty fundamentally. This is due to the limited precision of floating point numbers which cannot meet the high sensitivity of higher degree polynomials with respect to the coefficients. This limits the monomial-based version of (2.9) in standard double precision to about 16 to 20 stages for general spectra. We remark that the usage of higher precision datatypes can mitigate this issue.

Avoiding the difficulty of finding a suitable basis that leads to a well-conditioned problem is the key achievement of our approach motivated in the next subsection.

2.2 Central Observation

The linear consistency requirement (2.4) implies that every at least first order linearly consistent stability polynomial $P_{S,1}(z)$ can be written as

$$P_{S,1}(z;\widetilde{\boldsymbol{r}}) \;= 1 + z\prod_{j=1}^{S-1}\left(1-\frac{z}{\widetilde{r}_j}\right)$$

$$=: 1 + z\widetilde{P}_{S-1}(z;\widetilde{\boldsymbol{r}}) \tag{2.10}$$

where the *lower degree polynomial* $\widetilde{P}_{S-1}$ is parametrized by its complex-conjugated roots $\widetilde{\boldsymbol{r}} \in \mathbb{C}^{S-1}$. Clearly, the roots of the lower-degree polynomial (called from now on *pseudo-extrema* of the original polynomial) form a subset of the stability boundary $\partial\mathcal{S}$:

$$|P_{S,1}(\widetilde{r}_j;\widetilde{\boldsymbol{r}})| = |1 + \widetilde{r}_j \underbrace{\widetilde{P}_{S-1}(\widetilde{r}_j;\widetilde{\boldsymbol{r}})}_{=0}| = 1 \quad \Rightarrow \{\widetilde{r}_j\}_{j=1,\dots,S-1} \subset \partial\mathcal{S}. \tag{2.11}$$

This observation is particularly useful when the shape of the stability boundary $\partial\mathcal{S}$ can be a-priori roughly estimated. For optimized polynomials of higher degrees, $\partial\mathcal{S}$ will follow the spectrum closely, thus one can use an envelope of the spectrum itself as a reasonable approximation to $\partial\mathcal{S}$. This is discussed in Sects. 3.2 and 7 in more detail.

We remark that the formulation (2.10) bears some similarity to the basis $q_j = (1 + z/S)^j$ used in [36] for generating stability polynomials for circular spectra.

3 Motivating the Optimization in Pseudo-Extrema

The optimization problem (2.7) for stability polynomials in terms of complex-conjugated pseudo-extrema reads

$$\max_{\widetilde{\boldsymbol{r}}\in\mathbb{C}^{S-1}} \Delta t \text{ such that } \left|P_{S,p}(\Delta t\lambda^{(m)};\widetilde{\boldsymbol{r}})\right| \leq 1, \quad m = 1,\dots,M. \tag{3.1}$$

This is a highly nonlinear optimization problem for which we need an excellent initial guess in order to have a reasonable chance to reach the global optimum. In the next section, we will thoroughly motivate a suitable assumption on the distribution of the pseudo-extrema based on results of proven optimal stability polynomials.

3.1 Pseudo-Extrema of Proven Optimal Stability Polynomials for Disks

We direct our attention to examining the pseudo-extrema of known optimal stability polynomials $P_{S,p}(z)$ for circular spectra. For the sake of completeness, the case for parabolic spectra, i.e., eigenvalues on the negative real line, is provided in "Appendix A.1".

3.1.1 First Order Consistent Stability Polynomial

As shown in [24], the optimal first order stability polynomial with largest disk inclusion is given by a sequence of Forward Euler steps:

$$P_{S,1}^{\text{Disk}}(z) = \left(1 + \frac{z}{S}\right)^S. \tag{3.2}$$

For this stability polynomial it holds that the disk with radius S

$$D_S := \{z \in \mathbb{C} : |z + S| \leq S\} \tag{3.3}$$

is contained in the region of absolute stability $\mathcal{S}$. Consider the positive Chebyshev extreme points $x_j \in \mathbb{R} : T_S(x_j) = 1$ of the S degree polynomial on the here relevant $[-2S, 0]$ interval

which are given by

$$x_j = S\left(\cos\left(\frac{2j\pi}{S}\right) - 1\right), \quad j = 0, \ldots, \begin{cases} S/2 & S \text{ even} \\ (S-1)/2 & S \text{ odd} \end{cases}. \tag{3.4}$$

We give two motivations for considering the Chebyshev extreme points in connection with the pseudo-extrema. First, as outlined in "Appendix A" the pseudo-extrema of (shifted) Chebyshev polynomials are trivially given by the (shifted) positive Chebyshev extreme points. This is of significance since the first order optimal stability polynomial for parabolic spectra is precisely given by the shifted Chebyshev polynomial of first kind. Second, we recall that the Chebyshev extreme points x_j can be perceived as the real part of the set of points z_j which partition the circle into segments with equal arc length [55]. In that case, the imaginary part y_j of z_j can be computed as

$$y_j = \pm\sqrt{S^2 - (x_j + S)^2} = \pm S\sqrt{1 - \cos^2\left(\frac{2j\pi}{S}\right)} = \pm S \sin\left(\frac{2j\pi}{S}\right), \tag{3.5}$$

since $\sin\left(\frac{2j\pi}{S}\right) \geq 0$, $j = 0, \ldots, S/2$. Consequently, the points $z_j = x_j \pm iy_j$ with x_j, y_j defined as above lie on the stability boundary $\partial\mathcal{S}$ and are thus a valid candidate for the pseudo-extrema. In fact, one can readily show that these points are indeed the pseudo-extrema of $P_{S,1}^{\text{Disk}}(z)$, as stated in the following theorem.

Theorem 1 *The $S-1$ pseudo-extrema of $P_{S,1}^{Disk}(z)$ are given by the positive Chebyshev extreme points with $x_j \neq 0$ of $T_S(1 + z/S)$ with projection onto the circle with radius S centered at $(-S, 0)$.*

Proof We compute

$$P_{S,1}^{\text{Disk}}(\widetilde{r}_j) = \left(1 + \frac{\widetilde{r}_j}{S}\right)^S = \left(1 + \frac{x_j \pm iy_j}{S}\right)^S \tag{3.6}$$

$$= \left(1 + \frac{S\left(\cos\left(\frac{2j\pi}{S}\right) - 1\right) \pm iS\sin\left(\frac{2j\pi}{S}\right)}{S}\right)^S \tag{3.7}$$

$$= \left(\cos\left(\frac{2j\pi}{S}\right) \pm i\sin\left(\frac{2j\pi}{S}\right)\right)^S \tag{3.8}$$

$$= \exp(\pm i\, 2j\, \pi) \overset{j\in\mathbb{N}}{=} 1. \tag{3.9}$$

As a consequence, we have that

$$\widetilde{P}_{S-1}(\widetilde{r}_j; \widetilde{\boldsymbol{r}}) = \frac{P_{S,1}^{\text{Disk}}(\widetilde{\boldsymbol{r}}_j; \widetilde{\boldsymbol{r}}) - 1}{\widetilde{r}_j} = 0 \quad j = 1, \ldots \begin{cases} S/2 & S \text{ even} \\ (S-1)/2 & S \text{ odd} \end{cases} \tag{3.10}$$

where we use the fact that $P_{S,1}(0, \widetilde{\boldsymbol{r}}) = 1$ which excludes $j = 0$ from the set of pseudo-extrema.

Remark 1 Note that for even stability polynomials we have one purely real pseudo extremum $\widetilde{r}_0 = -S$ and $S - 2$ complex conjugated pseudo-extrema. For odd stability polynomials, all pseudo-extrema are complex conjugated. This is in accordance with the motivating observation (2.11) where we established that the pseudo-extrema lie on the boundary of the region of

absolute stability $\partial\mathcal{S}$. We also note that the complex-conjugated pseudo-extrema of $P_{S,1}^{\text{Disk}}(z)$ correspond to the real pseudo-extrema with root-multiplicity two of the Chebyshev polynomials $T_S(1 + z/S^2)$, see "Appendix A".

3.1.2 Second Order Consistent Stability Polynomial

For disks/circular spectra there is also an explicit formula for the optimal second order accurate polynomial known [25]. The polynomial

$$P_{S,2}^{\text{Disk}}(z) = \frac{S-1}{S}\left(1 + \frac{z}{S-1}\right)^S + \frac{1}{S} \tag{3.11}$$

contains the maximum disk D_{S-1} in its region of absolute stability [25, 26]. As a side note, we realize that $P_{S,2}(z)$ is identical to the second order accurate Runge–Kutta Chebyshev method [56, 57] with infinite damping [58]. Furthermore, as mentioned in [58] $P_{S,2}(z)$ can be written as a sequence of forward Euler steps and belongs consequently to the family of total varitation diminishing (TVD)/SSP integrators [59, 60] with optimal SSP coefficient [61].

Theorem 2 *The $S-1$ pseudo-extrema of $P_{S,2}^{Disk}$ are given by the positive Chebyshev extreme points $x_j \neq 0$ of $T_S\big(1+z/(S-1)\big)$ when projected onto the circle with radius $S-1$ centered at $\big(-(S-1), 0\big)$.*

Proof Analogous to proof of Theorem 1.

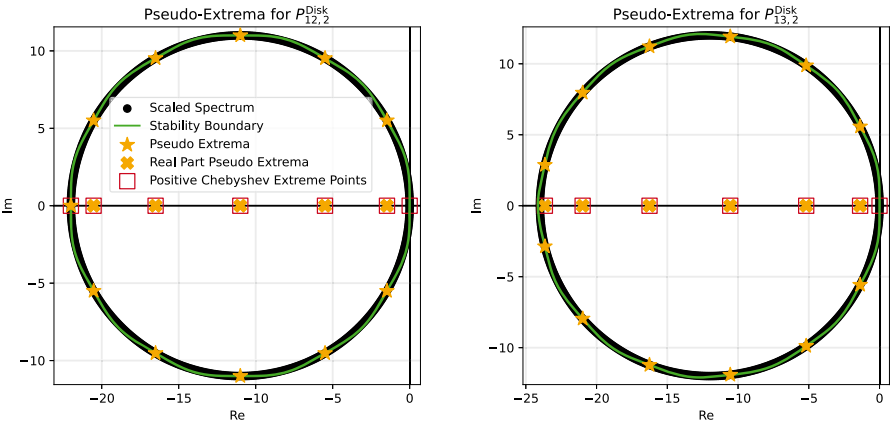
Remark 2 As for the first order accurate stability polynomial, we have for even S one real pseudo-extremum $\widetilde{r}_0 = -(S-1)$ and $S-2$ complex conjugated pseudo-extrema and for odd S a set of $(S-1)/2$ exclusively complex conjugated pseudo-extrema.

The significance of these results is that the pseudo-extrema are distributed with exact equal arc length on the hull of the spectrum, and consequently with approximately equal arc length on the stability boundary for a finite number of stages S. In fact, in the limit $S \to \infty$ the stability boundaries converge to the circle and thus the pseudo-extrema are asymptotically distributed with exact equal arc length on the stability boundary.

Especially higher degree stability polynomials have enough flexibility to fully adapt the stability boundary to the envelope of the spectrum. This is the case for even and odd polynomial degrees S as well as first and second order accurate polynomials. To illustrate this, we present Fig. 1 where the pseudo-extrema of second order accurate polynomials $P_{S,2}^{\text{Disk}}$ with degrees $S = 12$ and $S = 13$ are shown.

3.2 Central Assumption

Based on the findings for the circle we make a central assumption regarding the pseudo-extrema of strictly convex spectra, i.e., where the eigenvalues define a strictly convex set of points in the complex plane. To establish an intuition for this type of spectra we present four exemplarily representatives falling into this category in Fig. 2. We assume that the pseudo-extrema of optimal stability polynomials are distributed with approximately equal arc length on the convex hull of strictly convex spectra.



(a) Pseudo-Extrema $\tilde{r}_j$ and boundary of region of absolute stability $\partial\mathcal{S}$ of $P_{12,2}^{\text{Disk}}(z)$ alongside positive Chebyshev extreme points of $T_{12}(1 + z/S)$.

(b) Pseudo-Extrema $\tilde{r}_j$ and boundary of region of absolute stability $\partial\mathcal{S}$ of $P_{13,2}^{\text{Disk}}(z)$ alongside positive Chebyshev extreme points of $T_{13}(1 + z/S)$.

Fig. 1 Optimal second order accurate stability polynomials $P_{S,2}(z)$ given by (3.11) for even (**a**) and odd degree (**b**) polynomial. The spectrum corresponds to the canonical first order Finite Volume Upwind/Godunov discretization [62] of the advection equation $u_t + u_x = 0$ on the periodic $[-1, 1]$ domain discretized with 500 cells. In the even-degree case (**a**), the lower degree polynomial is odd and thus there is one pure real pseudo extremum at the left end of the spectrum. For odd polynomial degrees (**b**), the lower degree polynomial is of even-degree and we have only complex-conjugated pseudo-extrema. Note also that the segment crossing 0 is twice the length of the others, which follows from the fact that $P_{S,p}(0) \equiv 1 \; \forall \; S, \; p \geq 1$, cf. (2.10)

3.2.1 Strictly Convex Spectra

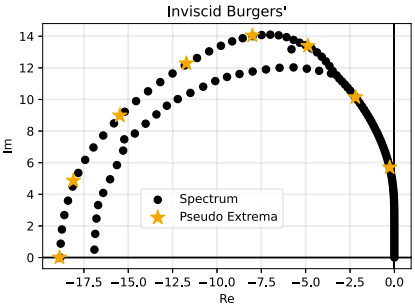
To motivate this assumption extending the findings for proven optimal spectra of the circle, we display the pseudo-extrema of optimized $P_{16,2}(z)$ polynomials for different spectra in Fig. 2. The optimal stability polynomials and optimal timesteps $\Delta t_{16,2}$ have been computed using the approach developed in [36] and the pseudo-extrema are computed numerically.

We present the spectra of four nonlinear hyperbolic PDEs

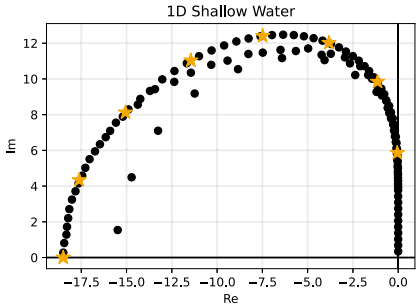
1. Burgers' equation
2. 1D Shallow Water equations
3. 1D Ideal Compressible Magnetohydrodynamics (MHD) equations
4. 2D Compressible Euler equations

when discretized with the discontinuous Galerkin spectral element method (DGSEM) [63, 64] using `Trixi.jl` [65–67]. We use local polynomials of degrees $1, 2, 3$ and a range of numerical fluxes and initial conditions, corresponding to both smooth and discontinuous solutions. For the sake of reproducibility we give the precise setups of the semidiscretizations corresponding to the displayed spectra in "Appendix B".

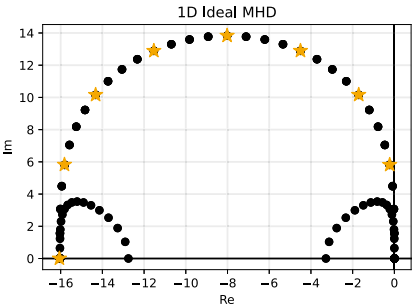
As seen from Fig. 2 the pseudo-extrema are distributed with approximately equal arc length on the spectrum enclosing curve. Based on this observation we will initialize the pseudo-extrema such that they are distributed with equal arc length on the spectrum enclosing curve, which is assumed to be very close to the optimal stability boundary of higher degree polynomials. The deviations from the equal arc-length initialization are then courtesy of the optimizer.



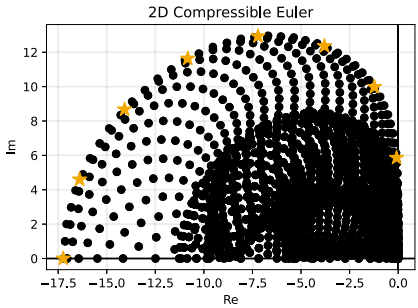
(a) Burgers' Equation corresponding to the semidiscretization defined by item 1.



(b) 1D Shallow Water Equations corresponding to the semidiscretization defined by item 2.



(c) 1D Compressible Ideal MHD corresponding to the semidiscretization defined by item 3.



(d) 2D Compressible Euler equations corresponding to the semidiscretization defined by item 4.

Fig. 2 Collection of strictly convex spectra of nonlinear hyperbolic PDEs and optimized pseudo-extrema $\widetilde{r}_j$. The spectra are scaled with the optimal timestep $\Delta t_{16,2}$ in each case

The significance of the strictly convex spectra stems from the observation that the optimal stability boundary $\partial\mathcal{S}^\star$ can be, for a sufficiently high polynomial degree S, reasonably well approximated with the convex hull of the eigenvalues.

This implies, in turn, that we can use the convex hull of the spectrum as an Ansatz for the initial placement of the pseudo-extrema $\widetilde{\boldsymbol{r}}$. Generalizing the proven results for the circle, the pseudo-extrema are then placed with equal arc length on the convex hull of the spectrum.

4 Detailed Formulation of the Optimization Problem

In this section we discuss the required details to successfully optimize a stability polynomial when parametrized by pseudo-extrema (3.1).

4.1 Constructing Complex-Conjugated Pseudo-Extrema

Since the lower-degree polynomial $\widetilde{P}_{S-1}$ has real coefficients, one has to ensure that the pseudo-extrema $\widetilde{\boldsymbol{r}}$ are complex-conjugated. In principle, this is a difficult task since throughout the optimization it might happen that a previously real pseudo-extremum is moved away from the real axis into the complex plane. Consequently, another real pseudo-extremum has to be set as the corresponding complex-conjugate, effectively changing the number of

free optimization variables by introducing an additional constraint. To avoid this switch-like behaviour we set a-priori a number of real and complex-conjugated pseudo-extrema. Based on the previous results, we restrict ourselves for even-degree stability polynomials to one real and $S-2$ complex conjugated pseudo-extrema and for odd stability polynomials to $S-1$ complex conjugated ones.

4.2 Restriction to Second Quadrant

Due to symmetry around the real axis it suffices to consider either the second or third quadrant in the complex plane only. In this work, we choose the second quadrant, which implies that we consider only complex numbers with non-negative imaginary part. This implies that for real Jacobians (1.2) with complex-conjugated eigenvalues $\lambda_j = \overline{\lambda_{j+1}}$ the stability requirement $|P_S(\Delta t \lambda^{(m)})| \leq 1$ needs only to be checked for the $m = 1, \dots, \widetilde{M}$ eigenvalues with non-negative imaginary part, thereby also reducing the number of constraints.

4.3 Scaling of the Optimal Timestep

We briefly comment on the scaling of the optimal timestep $\Delta t^\star$ with the degree of the stability polynomial S. It is well-known that the maximum admissible timestep scales for parabolic spectra covering the real-axis quadratic in S [9, 10]. In contrast, the one-sided imaginary axis inclusion of polynomials with real coefficients is bound by $S-1$ as shown in [26]. In [23] the tighter maximum imaginary stability limit $\sqrt{(S-1)^2-1}$ was conjectured and proven for special cases in [68]. The proven optimal stability polynomials for the disk scale both linearly in S, see [24, 25]. As a consequence, we consider for general hyperbolic-parabolic spectra an asymptotically linear scaling

$$\Delta t_{S,p} = \frac{S}{S_{\text{Ref}}} \Delta t_{S_{\text{Ref}},p}, \tag{4.1}$$

which is observed for instance in [33, 35, 36, 39, 42] as optimal. It should be mentioned that for $S \sim \mathcal{O}(1)$ better than linear scalings are possible, as for these degrees qualitative changes in the shape of the stability boundary still occur. Once the stability boundary is closely adapted to the spectrum, the linear scaling is recovered.

It is immediately clear that once the timestep increases only linearly with more stages S there is no additional efficiency gained. In particular, we observe for the methods with a large number of stages additional complications due to internal stability, see Sect. 9.1 which render the application of an *individual* many-stage scheme unattractive. Nevertheless, many-stage stability polynomials are of great value in the context of multirate partitioned Runge–Kutta methods where an *ensemble* of methods is used. As mentioned in the introduction, the P-ERK schemes [39, 40, 45] would benefit from high degree stability polynomials which enable the efficient integration of systems even in the presence of locally restricted CFL numbers.

4.4 Constraints for Higher Order

For higher order p-consistent methods $p-1$ equality constraints have to be added to the stability inequality constraints. To obtain second order consistent methods, for instance, the pseudo-extrema have to satisfy

$$\frac{1}{2} \overset{!}{=} -\sum_{j=1}^{S-1} \frac{1}{\widetilde{r}_j} \tag{4.2}$$

which follows from Vieta's formulas which establish a link between the roots and coefficients of a polynomial. More generally, for a stability polynomial that matches the first $k = 0, \ldots, p$ coefficients of the exponential $1/(j!)$ we have the additional constraints

$$\frac{1}{k!} \overset{!}{=} (-1)^{k-1} \sum_{\substack{1=j_1, j_2 \cdots j_{k-1} \\ j_1 \neq j_2 \neq \cdots \neq j_{k-1}}}^{S-1} \frac{1}{\widetilde{\boldsymbol{r}}_{j_1} \widetilde{\boldsymbol{r}}_{j_2} \ldots \widetilde{\boldsymbol{r}}_{j_{k-1}}}, \quad k = 2, \ldots, p. \tag{4.3}$$

In this work we focus on second order accurate polynomials since for these the linear order constraints (4.3) imply second order convergence also for nonlinear system, without any additional constraints placed on the coefficients of the method [6, 46]. Nevertheless, we also constructed third order accurate stability polynomials for some selected cases.

4.5 Focus on Even-Degree Stability Polynomials

From now on we focus on even-degree polynomials since they lead to a slightly simpler formulation of the optimization problem. As we are interested in the many-stage case, focusing on either even or odd degree case is no severe restriction.

Even-degree stability polynomials come with two main advantages. First, we observe for all considered spectra (not only strictly convex) that the left endpoint of the spectrum, in our cases a real eigenvalue, is always selected as one pseudo-extremum, allowing for an even improved initialization. For strictly convex spectra, this is the only real pseudo-extremum, cf. Figs. 1a, 2, and 3. The remaining pseudo-extrema can then be initialized with equal arc length starting from the left endpoint of the spectrum.

Second, under the assumption motivated in Sect. 3.2 and the linear scaling of the optimal timestep Sect. 4.3 one can readily re-use the optimal pseudo-extrema of the $S/2$ degree polynomial as an advanced initialization for the S degree polynomial. In particular, every second pseudo-extremum of the S stability polynomial is essentially known, cf. Fig. 3.

4.6 Construction of the Convex Hull Interpolant

For strictly convex spectra the convex hull defines in the second quadrant actually a strictly concave (strictly convex in the third quadrant) function, which allows to uniquely map the real part of the pseduo-extrema to some corresponding imaginary part. To relax the very steep section near the imaginary axis we add the origin $(0, 0)$ to the hull which helps to reduce the slope in that particular segment. This allows to carry out the optimization actually in terms of the real parts of the pseudo-extrema only, cutting the number of optimization variables in half.

Given the (strict) convex hull of the spectrum defined through a set of eigenvalues $\mu^{(j)} \in \sigma^{\text{Hull}}$ and the expected maximum timestep $\Delta t_{S,p}$, we define the scaled spectrum

$$\sigma_{S,p}^{\text{Hull}} := \left\{ \Delta t_{S,p} \cdot \lambda, \ \lambda \in \sigma^{\text{Hull}} \right\} \tag{4.4}$$

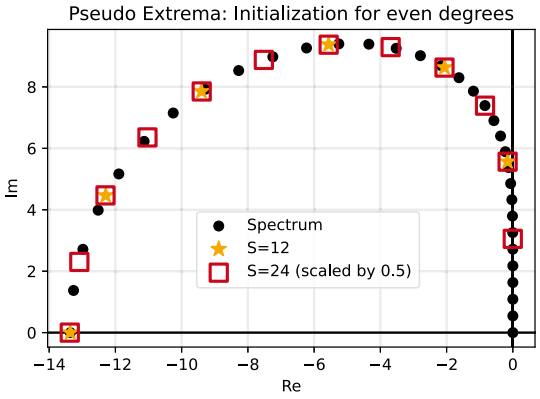


Fig. 3 Scaled spectrum $\sigma_{12,2}$ of the 1D linear advection equation $u_t + u_x = 0$ discretized through the DGSEM on [0, 1] using 16 cells/elements with DG polynomial degree 3 and Rusanov/Local Lax-Friedrichs flux. The pseudo-extrema of the 12 and 24 degree second order accurate polynomial are computed. To highlight that the relative positions of every second pseudo extremum $\tilde{r}^{2S}_{2j-1}$ agree with $\tilde{r}^{S}_{j}$, the former are scaled by 0.5 due to the linear scaling of the timestep according to (4.1)

$$\mu^{(j)}_{S,p} := \Delta t_{S,p} \cdot \mu^{(j)} \,. \tag{4.5}$$

Based on this we construct a piece-wise linear interpolant I as

$$\begin{aligned} I\left(x; \sigma^{\text{Hull}}_{S,p}\right) := \operatorname{Im}\left(\mu^{(j)}_{S,p}\right) + \frac{\operatorname{Im}\left(\mu^{(j+1)}_{S,p}\right) - \operatorname{Im}\left(\mu^{(j)}_{S,p}\right)}{\operatorname{Re}\left(\mu^{(j+1)}_{S,p}\right) - \operatorname{Re}\left(\mu^{(j)}_{S,p}\right)} \left(x - \operatorname{Re}\left(\mu^{(j)}_{S,p}\right)\right), \\ \operatorname{Re}\left(\mu^{(j)}_{S,p}\right) \le x < \operatorname{Re}\left(\mu^{(j+1)}_{S,p}\right). \end{aligned} \tag{4.6}$$

While quadratic or the classic cubic splines would result in a continuously differentiable interpolant I, they lead to numerical artifacts in the very steep sections near the imaginary axis that are a common feature of the higher order discontinuous Galerkin (DG) spectra. For these parts it is observed that linear splines capture the boundary of the stability domain $\partial\mathcal{S}$ best and should be used there. For the less steep regions quadratic and cubic splines were tested, but the increase in approximation quality compared to linear splines was for the majority of cases too small to justify the increased computational costs.

Global interpolation techniques were also tested but familiar problems like Runge's phenomenon and large over/undershoots in the intervals between the eigenvalues are observed for a variety of approaches, including Bernstein polynomials or mapping to Chebyshev points [69].

It should be mentioned that the spectrum close to the imaginary axis becomes arbitrarily steep and thus the interpolation becomes less reliable in this part. This becomes an issue for stability polynomials of degrees $\mathcal{O}(100)$ since then pseudo-extrema are placed in that region. To solve this problem, one can work with a spectrum enclosing curve $\gamma(\tau) : \mathbb{R} \to \mathbb{C}$ instead of a spectrum enclosing function $I : \mathbb{R} \to \mathbb{R}$. By identifying the eigenvalues on the hull $\mu^{(j)} \in \sigma^{\text{Hull}}$ with an arc length parameter $0 \le \tau^{(j)} \le 1$ the imaginary part can be computed

as

$$I_{\mathrm{Im}}\left(\tau;\sigma_{S,p}^{\mathrm{Hull}}\right)=\mathrm{Im}\left(\mu_{S,p}^{(j)}\right)+\frac{\mathrm{Im}\left(\mu_{S,p}^{(j+1)}\right)-\mathrm{Im}\left(\mu_{S,p}^{(j)}\right)}{\tau_{S,p}^{(j+1)}-\tau_{S,p}^{(j)}}\left(\tau-\tau_{S,p}^{(j)}\right), \quad \tau_{S,p}^{(j)}\le\tau<\tau_{S,p}^{(j+1)} \tag{4.7}$$

which removes the vanishingly small difference in real parts from the denominator. This, however, comes at the expense that we have to do the interpolation (4.7) also for the real part of the pseudo-extrema, thus doubling the computational work load. As discussed in Sect. 10, methods with more than hundred stages introduce additional complications related to internal stability. Consequently, we tailor the optimization approach around the more robust $S\sim\mathcal{O}(10)$ case and use the real parts of the pseudo-extrema as the optimization variables.

5 Implementation

In this section, we bundle the aforementioned details into the precise formulation of the solved problems.

5.1 Optimization Problems

Equipped with the interpolant $I(x)$ the optimization of lower-degree polynomial can now be efficiently conducted. The optimization problem (2.7) is first conducted in the real parts $x_j=\mathrm{Re}(\widetilde{r}_j)$ only, i.e.,

$$\max \Delta t \text{ over } \boldsymbol{x}\in\left[\min_m\left\{\mathrm{Re}\left(\lambda_{S,p}^{(m)}\right)\right\},0\right]^N \text{ such that} \tag{5.1a}$$

$$\left|1+\left(\Delta t\lambda^{(m)}\right)\widetilde{P}_{S-1}\left(\Delta t\lambda^{(m)};\boldsymbol{x}+iI\left(\boldsymbol{x};\sigma_{S,p}^{\mathrm{Hull}}\right)\right)\right|\le 1\quad m=1,\dots,\widetilde{M}. \tag{5.1b}$$

Note that the number of optimization variables N equals for even stability polynomials $1+\frac{S-2}{2}=\frac{S}{2}$ since we expect one real pseudo extremum and $S-2$ complex conjugated ones.

Due to the sensitive dependence of the stability and order constraints on the pseudo-extrema we conduct a second optimization run where we allow for small corrections y in the imaginary parts of the pseudo-extrema:

$$\boldsymbol{y}\in\left[-\varepsilon\max_m\left\{\mathrm{Im}\left(\lambda_{S,p}^{(m)}\right)\right\},\varepsilon\max_m\left\{\mathrm{Im}\left(\lambda_{S,p}^{(m)}\right)\right\}\right]^N. \tag{5.2}$$

Here, ε is the sole hyperparameter needed in our approach which could for all problems be set to $\mathcal{O}\left(10^{-2}\right)$ with default choice 0.02. The second stage optimization problem reads then

$$\max \Delta t \text{ over } \begin{pmatrix}\boldsymbol{x}\\ \boldsymbol{y}\end{pmatrix}\in\begin{pmatrix}\left[\min_m\left\{\mathrm{Re}\left(\lambda_{S,p}^{(m)}\right)\right\},0\right]^N\\ \left[-\varepsilon\max_m\left\{\mathrm{Im}\left(\lambda_{S,p}^{(m)}\right)\right\},\varepsilon\max_m\left\{\mathrm{Im}\left(\lambda_{S,p}^{(m)}\right)\right\}\right]^N\end{pmatrix} \tag{5.3a}$$

such that

$$\left|1+\left(\Delta t\lambda^{(m)}\right)\widetilde{P}_{S-1}\left(\Delta t\lambda^{(m)};\boldsymbol{x}+i\left[I\left(\boldsymbol{x};\sigma_{S,p}^{\mathrm{Hull}}\right)+\boldsymbol{y}\right]\right)\right|\le 1\quad m=1,\dots,\widetilde{M}. \tag{5.3b}$$

The entire algorithm is given in Algorithm 1. In principle, it is also possible to conduct the optimization in real parts and imaginary corrections (5.3) only, i.e., without solving (5.1) first. In all considered cases, however, optimizing the real parts of the pseudo-extrema first led not only to a more robust, but also overall faster optimization process.

Algorithm 1: Optimization problem in pseudo-extrema.

Perform Algorithm proposed in [36] to obtain $\Delta t_{S_{\text{Ref}},p}$ for certain S_{Ref}, p;
$\Delta t_{S,p} \leftarrow \Delta t_{S_{\text{Ref}},p} \frac{S}{S_{\text{Ref}}}$;
Scale spectrum with expected timestep: $\sigma^{(S,p)} := \{\lambda^{(m)} \cdot \Delta t_{S,p}\}_{m=1,\dots,\widetilde{M}}$;
Compute convex hull $\sigma_{S,p}^{\text{Hull}} \leftarrow \gamma_0\left(\sigma^{(S,p)}\right)$;
Compute Interpolant $I\left(x; \sigma_{S,p}^{\text{Hull}}\right)$;
if S, $S/2$ *are even* **and** *results* $\boldsymbol{x}^{(S/2)}$ *for the* $S/2$ *degree polynomial exist* **then**
  Set every second entry in $\boldsymbol{x}_0^{(S)}$ to $2\boldsymbol{x}^{(S/2)}$;
  Set remaining entries such that $\left(x_{0,j}, I\left(x_{0,j}; \sigma_{S,p}^{\text{Hull}}\right)\right)$, $j = 2, 4, \dots, N$ are equally distributed on $I\left(x; \sigma_{S,p}^{\text{Hull}}\right)$;
else
  Compute initial guess $\boldsymbol{x}_0$ such that $\left(x_{0,j}, I\left(x; \sigma_{S,p}^{\text{Hull}}\right)\right)$, $j = 1, \dots, N$ are equally distributed on $I\left(x; \sigma_{S,p}^{\text{Hull}}\right)$;
Initialize timestep with maximum possible value $\Delta t_0 \leftarrow \Delta t_{S,p}^{\star}$;
Find best candidate $\Delta t^{\star}, \boldsymbol{x}^*$ of optimization problem (5.1);
Update initial guess $\Delta t_0 \leftarrow \Delta t^{\star}, \boldsymbol{x}_0 \leftarrow \boldsymbol{x}^*, \boldsymbol{y}_0 = \boldsymbol{0}$;
$(\boldsymbol{x}^{\star}, \boldsymbol{y}^{\star}) \leftarrow$ Solve second stage optimization problem (5.3);
return $\widetilde{r} \leftarrow \boldsymbol{x}^{\star} \pm i\left[I\left(\boldsymbol{x}^{\star}; \sigma_{S,p}^{\text{Hull}}\right) + \boldsymbol{y}^{\star}\right]$;
return $\Delta t_{S,p}^{\star} \leftarrow \Delta t^{\star}$;

5.2 Feasibility Problems

Under assumption that we can indeed realize the optimal timestep $\Delta t_{S,p}$ one can formulate the optimization problems (5.1), (5.3) actually as feasibility problems by fixing $\Delta t = \Delta t_{S,p}$ and search for $\boldsymbol{x}, \boldsymbol{y}$ satisfying the stability and order constraints. The feasibility problem corresponding to (5.1) reads then

$$\text{Find } \boldsymbol{x} \in \left[\min_m \left\{\text{Re}\left(\lambda_{S,p}^{(m)}\right)\right\}, 0\right]^N \text{ such that} \tag{5.4a}$$

$$\left|1 + \left(\Delta t_{S,p}\lambda^{(m)}\right) \widetilde{P}_{S-1}\left(\Delta t_{S,p}\lambda^{(m)}; \boldsymbol{x} + iI\left(\boldsymbol{x}; \sigma_{S,p}^{\text{Hull}}\right)\right)\right| \leq 1 \quad m = 1, \dots, \widetilde{M}. \tag{5.4b}$$

and the analogy to (5.3) is given by

$$\text{Find } \begin{pmatrix}\boldsymbol{x}\\ \boldsymbol{y}\end{pmatrix} \in \begin{pmatrix}\left[\min_m \left\{\text{Re}\left(\lambda_{S,p}^{(m)}\right)\right\}, 0\right]^N \\ \left[-\varepsilon \max_m \left\{\text{Im}\left(\lambda_{S,p}^{(m)}\right)\right\}, \varepsilon \max_m \left\{\text{Im}\left(\lambda_{S,p}^{(m)}\right)\right\}\right]^N\end{pmatrix} \tag{5.5a}$$

such that

$$\left|1+\left(\Delta t_{S,p}\lambda^{(m)}\right)\widetilde{P}_{S-1}\left(\Delta t_{S,p}\lambda^{(m)};\boldsymbol{x}+i\left[I\left(\boldsymbol{x};\sigma_{S,p}^{\text{Hull}}\right)+\boldsymbol{y}\right]\right)\right|\leq 1 \quad m=1,\ldots,\widetilde{M}. \tag{5.5b}$$

In practice, the feasibility problems are solved significantly faster than their optimization counterparts. Note that no conceptual changes to Algorithm 1 are necessary, as the timestep handling in Lines Lines 11 and 16 can be omitted and (5.4) and (5.5) are solved in Lines 12 and 14, respectively.

5.3 Software

Both optimization and feasibility problem are solved via

`Ipopt: Interioir Point OPTimizer` [70], an optimization software designed for general non-linear problems. The herein used required linear solver is `MUMPS: MUltifrontal Massively Parallel sparse direct Solver` [71, 72] with optional package `METIS` [73]. Being a derivative based optimizer, `Ipopt` requires the Jacobian matrix and Hessian tensor of the constraints (and objective) to construct the Lagrangian of the problem. Here, the derivatives are computed algorithmically via `dco/c++` [74] which requires only the adaption of some boilerplate code. Besides being much more computationally efficient than approximating the derivatives using finite differences, algorithmic differentiation provides exact derivatives (up to machine precision).

This is especially relevant in this case, since the stability constraints are very sensitive with respect to the pseudo-extrema, i.e., small deviations in $\boldsymbol{x}$ can make the difference between a stable and an unstable method. This was observed when `Matlab`'s function for general nonlinear optimization, `fmincon` failed in finding the optimal solution. Therein, finite difference approximations of Jacobians and Hessians are used and consequently, the optimization is much less reliable than `Ipopt` combined with `dco/c++`.

In terms of complexity the proposed algorithms scales linearly in the number of constraints ($\widetilde{M}$ stability constraints at eigenvalues and $p-1$ order constraints) and quadratic in the number of unknowns (pseudo-extrema with distinct real part) N.

It should be mentioned that `Ipopt` provides many options which can significantly speed up the optimization/feasible point search. Most notably, one should specify `grad_f_constant yes` which ensures that the gradient of the objective is only evaluated once and the Hessian not at all. An option for a constant objective function as it is the case for the feasibility problems is not supported, but can actually be implemented by changing only two lines of the `Ipopt` source code [75]. Furthermore, for many problems the Hessian of the Lagrangian may be successfully approximated using the Limited-memory BFGS method [76] which is activated by specifying the option `hessian_approximation limited-memory`, resulting in significant speed up. In terms of tolerances, we demand satisfaction of the constraints to machine precision by setting `constr_viol_tol 1e-16`.

Furthermore, in most cases it suffices to supply only the convex hull and a small subset of the eigenvalues instead of the entire spectrum. This can be seen from e.g. Fig. 4a, d

where the stability constraint at the "interior" eigenvalues is usually satisfied without being explicitly enforced. One has to ensure, however, that there is always a larger number of constraints than pseudo-extrema to prevent degenerate cases where the roots of the stability polynomial are placed exactly on the eigenvalues, thus allowing in principle infinitely large timesteps.

The source code is made publicly available on `GitHub` [77].

Table 1 Ratio of optimal timestep $\Delta t_{S,3}$ for circular spectra obtained from different optimization approaches

	$\Delta t/S$						
S	20	30	40	50	60	70	80
Coeff.-based [36]	0.881	0.890	0.939	0.941	0.952	0.965	0.958
Algorithm 1	0.809	0.901	0.914	0.951	0.959	0.935	0.969

6 Optimal Stability Polynomials for Strictly Convex Spectra

6.1 Comparison to Coefficient-Based Approach

Before coming to applications of Algorithm 1 to spectra of practical semidiscretizations, we seek to compare the herein developed approach to results obtained by applying the method developed in [36]. For this, we consider the problem of maximum disk inclusion, i.e., finding a stability polynomial such that the radius Δt of the disk

$$D_{\Delta t} = \{z : |1 + z/\Delta t| \leq 1\} \tag{6.1}$$

is contained in the region of absolute stability $\mathcal{S}$. In particular, we construct third-order accurate stability polynomials $P_{S,3}^{\text{Disk}}$ with Algorithm 1 and compare the maximum possible timestep of this approach to the timestep obtained when applying the coefficient-based approach from [36]. Using the basis recommended in [36] we construct stability polynomials and corresponding optimal timesteps for $S = 20, 30, \ldots, 80$. The timestep obtained in this way is then used twofold: First, we use it as the fixed, supplied timestep for the feasibility problems (5.4), (5.5) to check if Algorithm 1 is able to find a stability polynomial which is also A-stable for the supplied spectrum. This is indeed for every considered S the case, the corresponding pseudo-extrema are provided in the supplementary material.

Next, we test also the optimization problems (5.1), (5.3) whether they report a similar optimal timestep. Here, we supply the expected timestep based on $\Delta t_{20,3}$, cf. (4.1) instead of the optimal timestep found using the coefficient-based approach. Table 1 compares the ratio $\Delta t/S$ for the coefficient-based approach to the one corresponding to the obtained by applying Algorithm 1. For all considered values of S the ratios are quite similar and seem to approach a constant value, which is in accordance with the asymptotically linear scaling of the optimal timestep Sect. 4.3.

6.2 Applications

Here, we present many-stage optimal stability polynomials corresponding to first and second order for the spectra shown in Fig. 2. For each spectrum, we compute $\Delta t_{16,p}$ and $P_{16,p}(z)$ using the approach developed in [36]. Equipped with the optimal timestep, we supply the expected optimal timestep according to (4.1) to the feasibility problems (5.4), (5.5). In particular, we consider the constructed polynomial as optimal when the linear scaling of the maximum timestep is realized, which is the case for all examples mentioned in this section. For every spectrum, we start with a moderate number of stages $(26 - 32)$ and then double the number of stages to utilize the equal arc length property for even spectra discussed in Sect. 4.5.

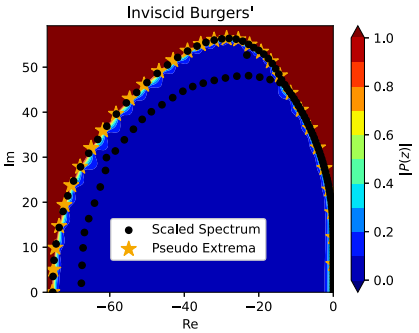
Table 2 Runtimes of feasibility problems for different degrees constrained to spectrum of Burgers equation

	$t[s]$		
Stages S	1st Order	2nd Order	3rd Order
16 (Algorithm from [36])	15.531	15.346	15.076
32 (Algorithm 1, lines `2-16`)	0.478	0.879	1.183
64 (Algorithm 1, lines `2-16`)	1.551	2.869	2.276
128 (Algorithm 1, lines `2-16`)	9.383	8.845	8.549
Cumulative t	26.942	27.939	27.084

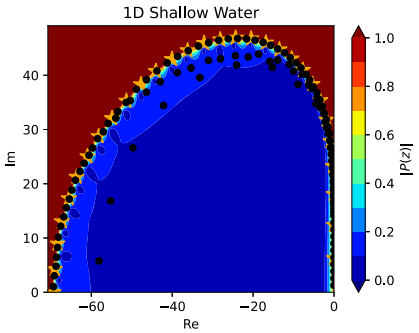
The times are obtained on a Dell Precision 5570 laptop equipped with an Intel i7-12800 H

The computed pseudo-extrema alongside the constraining list of eigenvalues are provided as supplementary material to this manuscript. Optimal stability polynomials are constructed for:

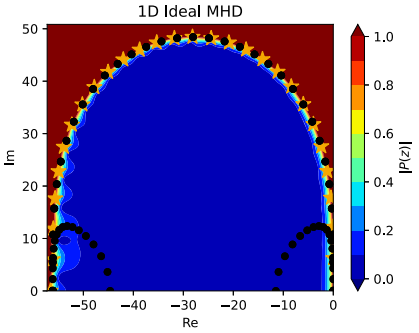
1. Burgers' Equation: We find the optimal stability polynomials of first, second, and third order with degrees 32, 64, 128 constrained by $\widetilde{M} = 129$ eigenvalues. In particular, we make use of the property discussed in Sect. 4.5 by conducting the optimization sequentially, i.e., first performing the optimization for $S = 32$, re-using this results for the initialization of the $S = 64$ polynomial and in turn using this results for the final $S = 128$ polynomial. The pseudo-extrema and contours of the optimal $P_{64,2}(z)$ stability polynomial are displayed in Fig. 4a.
   To highlight the efficiency of our approach, we provide the `Ipopt` runtimes for the feasibility problem runs in Table 2. We emphasize that we do not intend to compare the runtimes of Algorithm 1 to the optimization problem from [36]—instead, we would like to stress that equipped with an optimal timestep for a low degree polynomial, we can quickly compute the higher degree optimal ones.
2. 1D Shallow Water equations: Here we optimize polynomials of degrees 30, 60, and 120 constrained by $\widetilde{M} = 117$ eigenvalues, again in a sequential manner. The pseudo-extrema and contours of the optimal $P_{60,2}(z)$ stability polynomial are displayed in Fig. 4b. The CPU times are very similar to Burgers' equation with total $30.687s$ for the computation of the first order accurate stability polynomial and $28.726s$ for the second order case.
3. 1D Ideal compressible magnetohydrodynamics (MHD): Polynomials of degrees 28, 56, and 112 are found satisfying the $\widetilde{M} = 252$ stability constraints due to the eigenvalues. The pseudo-extrema and contours of the optimal $P_{56,2}(z)$ stability polynomial are displayed in Fig. 4c. The generation of the 112 degree stability polynomial takes $94.433s$ and $98.377s$ for first and second order case, respectively.
4. 2D Compressible Euler equations: To conclude this result section, we generate stability polynomials with degrees 26, 52, and 104 for a reduced set of $\widetilde{M} = 717$ eigenvalue constraints. The pseudo-extrema and contours of the optimal $P_{52,2}(z)$ stability polynomial are displayed in Fig. 4d. Due to the quadratic scaling in the number of constraints we have significantly longer runtimes then in the previous cases, with total 2.456 min for the first order and 8.55 min for the second order case. Note that these runtimes can be significantly reduced if only a small portion of the eigenvalues alongside the convex hull thereof is supplied as constraints.



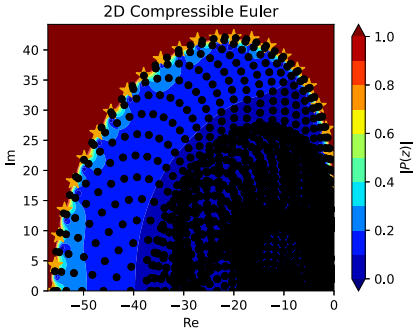
(a) Spectrum of Burgers' Equation computed for the semidiscretization defined by item 1.



(b) Spectrum of the 1D Shallow Water Equations computed for the semidiscretization defined by item 2.



(c) Spectrum of the 1D Compressible Ideal MHD computed for the semidiscretization defined by item 3.



(d) Spectrum of the 2D Compressible Euler equations computed for the semidiscretization defined by item 4.

Fig. 4 Collection of strictly convex spectra of nonlinear hyperbolic PDEs and optimized pseudo-extrema. The spectra are scaled with the optimal timestep $\Delta t_{16,2}$ in each case

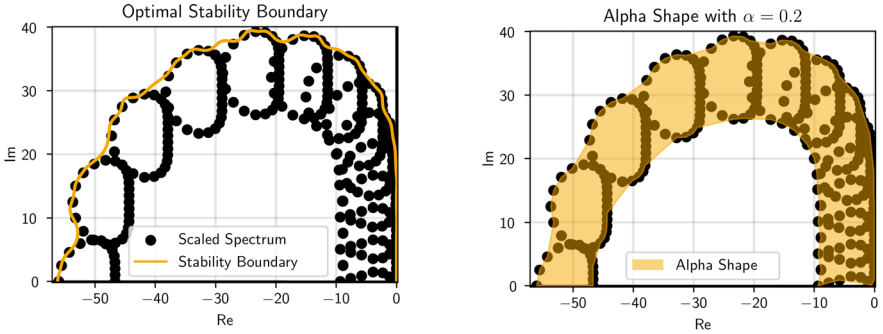
7 Non-Convex Spectra

While a rich set of spatial discretization, equation, boundary and initial condition combinations leads to strictly convex spectra, especially in one spatial dimension, at least equally many cases with non-convex spectra are observed. In this section, we propose two possible treatments thereof.

First, alpha shapes [78, 79] are proposed as a potentially accurate, yet more expensive candidate which come with an unknown hyperparameter. Second, we follow the previous sections and use a convex hull approach also for non-convex spectra.

7.1 Alpha Shapes

Alpha shapes [78, 79] provide a rigorous way of constructing a point enclosing shape that is potentially closer to the point set than the convex hull. Alpha shapes are parametrized by the real scalar α parameter that defines how close the alpha shape is shrinked to the points. For $\alpha > 0$, the construction of alpha shapes can intuitively described by rolling a disk with radius $1/\alpha$ around the points and drawing a line between two points of the set whenever they are on the edge of the disk and no points are in the interior of the disk. For $\alpha = 0$, the disk



(a) ∂S_{50} for the scaled spectrum $\Delta t_{50,2}\sigma$.

(b) Alpha shape with $\alpha = 0.2$ of the scaled spectrum $\Delta t_{50,2}\sigma$.

Fig. 5 Approximation of the optimal stability boundary by the contour of an alpha shape. The spectrum σ is obtained from a DGSEM discretization of the 2D linear advection equation with velocities $a_x = 0.5, a_y = -0.1$ on the periodic domain $\Omega = [-1, 1]^2$ discretized by a 6×6 mesh with local polynomials of degree 3 and Rusanov/Local Lax-Friedrichs flux

has infinite radius, i.e., degenerates to a straight line and the resulting shape is equivalent to the convex hull which is constructed by tilting the straight line around the points. While alpha shapes provide potentially a very accurate ansatz for the stability boundary, cf. Fig. 5 they come with two main drawbacks.

First, alpha shapes introduce a hyperparameter whose choice is not clear. While in general higher degree polynomials follow the stability boundary more closely, i.e., correspond to larger alpha values the best choice is highly dependent on the individual spectra. Furthermore, for high α corresponding to small radii there is always the danger of obtaining a disjoint alpha shape which is not useful in this context. Both of these issues can be in principle addressed through manual inspection, which is clearly not attractive.

Secondly, and more severely, alpha shapes provide in general only a spectrum enclosing curve $\gamma_\alpha(\tau) : \mathbb{R} \to \mathbb{C}$, i.e., no spectrum enclosing function. This implies that for both the imaginary part (cf. (4.7)) and real part

$$I_{\mathrm{Re}}\left(\tau; \sigma^{\alpha}_{S,p}\right) = \mathrm{Re}\left(\mu^{(j)}_{S,p}\right) + \frac{\mathrm{Re}\left(\mu^{(j+1)}_{S,p}\right) - \mathrm{Re}\left(\mu^{(j)}_{S,p}\right)}{\tau^{(j+1)}_{S,p} - \tau^{(j)}_{S,p}}\left(\tau - \tau^{(j)}_{S,p}\right), \quad \tau^{(j)}_{S,p} \le \tau < \tau^{(j+1)}_{S,p} \tag{7.1}$$

of the pseudo-extrema interpolation has to be performed. Here, $\mu^{(j)} \in \sigma^{\alpha}$ are the eigenvalues selected as the upper part of the alpha shape with corresponding arc length parameter $0 \le \tau^{(j)} \le 1$. This results in doubled computational costs and an overall more difficult optimization task.

Nevertheless, we stress that the general idea of distributing the pseudo-extrema with equal arc length, i.e., equal distances $\Delta\tau = \tau^{(j+1)} - \tau^{(j)}, \ \forall j$ is still applicable and Algorithm 1 can be used under the appropriate changes.

7.2 Convex Hull Ansatz

By constructing the convex hull for non-convex spectra the framework prepared in the previous sections is recovered. Precisely, even if the hull of spectrum itself may be nonconvex,

the convex hull is constructed and used as the ansatz for the initial placement of the pseudo-extrema. Using the convex hull difficulties with additional hyperparameters or the danger of disjoint alpha shapes are ruled out. This comes with the drawback that we can no longer expect to actually realize the maximum possible optimal timestep according to (4.1). Consequently, one has to indeed perform the optimization problems (5.1), (5.3) as the feasibility problems with fixed timestep will fail. Depending on the spectrum there are a priori no guarantees how efficient this approach will be. In practice, however, we find for many real-world spectra excellent results, as illustrated in the next section.

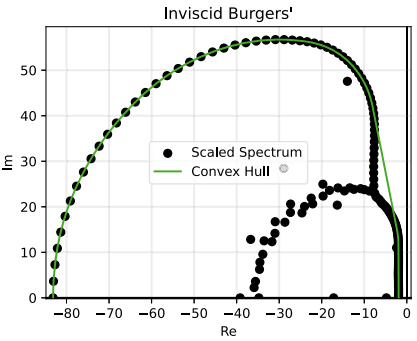
8 Optimal Stability Polynomials for Nonconvex Spectra

Here, we present results for two relevant, exemplary nonconvex spectra for which we use the convex hull-based approach.

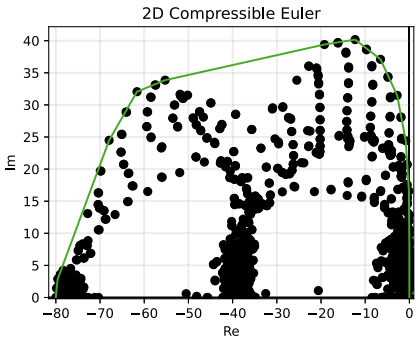
1. Burgers' Equation with Riemannian initial data $u_0(x) = \begin{cases} 1.5 & x < 0.5 \\ 0.5 & x \geq 0 \end{cases}$. As before, we discretize $\Omega = [0, 1]$ with 64 cells and use Godunov's flux. At $x = 0$ we fix $u(t, 0) = 1.5$ while at $x = 1$ an outflow boundary is used. The solution is reconstructed using fourth order local polynomials. The corresponding spectrum is displayed in Fig. 6a alongside its convex hull. The convex hull is in most parts extremely close the spectrum and we thus expect maximal timesteps very close to the optimal ones according to the linear scaling (4.1). Indeed, for 30, 60, and 120 degree polynomials we can realize the optimal timestep. As for the convex spectra, we display the $S = 60$ case in Fig. 6a.
2. 2D compressible Euler equations: Isentropic vortex advection. The advection of an isentropic vortex is a classic, fully nonlinear testcase with known analytical solution [59, 80]. In terms of the physical parameters we use the same setup as in [39–41] which is spelled out in Sect. 10.2.2. The numerical scheme is composed of Rusanov/Local Lax-Friedrichs flux, second order local polynomials and 8 cells in each direction. This little resolution is required to execute the eigenvalue decomposition in reasonable time. The corresponding spectrum is shown in Fig. 6b alongside its convex hull. Similar to Burgers' Equation, the spectrum is in most parts reasonably close to the spectrum. For $S = 28$ we can realize about 97.9% of the theoretically optimal timestep and for $S = 56$, $S = 112$ indeed the optimal timestep according to the linear timestep scaling (4.1).

9 Construction of Many-Stage Runge–Kutta Methods

As discussed before, the primary use case of the high degree stability polynomials is the possibility to construct many-stage methods for e.g. partitioned multirate Runge–Kutta methods such as the P-ERK [39, 40, 45] schemes. Nevertheless, we seek to demonstrate the capabilities of the optimized stability polynomials by constructing many-stage standalone methods. Here, we limit ourselves to second order accurate methods for which the linear order constraints (2.4) imply second order convergence even in the nonlinear case, without any additional constraints [6, 46]. Furthermore, we leave the construction of SSP methods for future work and focus here on hyperbolic PDEs with smooth solutions.

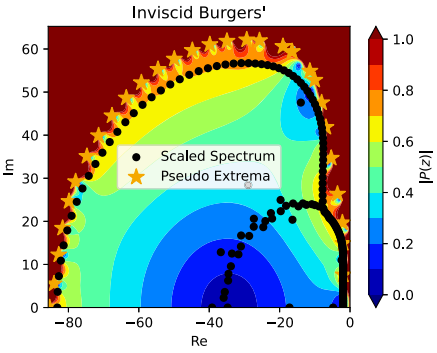


(a) Spectrum of Burgers' equation computed for the semidiscretization as specified by item 1.

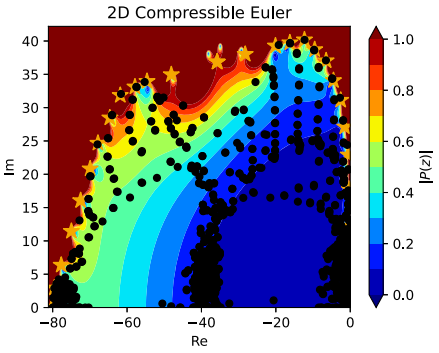


(b) Spectrum of the 2D Compressible Euler equations computed for the semidiscretization as specified by item 2.

Fig. 6 Collection of nonconvex convex spectra of nonlinear hyperbolic PDEs and corresponding convex hull. The spectra are scaled with the optimal timesteps $\Delta t_{S,2}$



(a) Spectrum of Burgers' Equation computed for the semidiscretization defined by item 1.



(b) Spectrum of the 2D Compressible Euler equations computed for the semidiscretization defined by item 2.

Fig. 7 Collection of strictly convex spectra of nonlinear hyperbolic PDEs and optimized pseudo-extrema. The spectra are scaled with the optimal timestep $\Delta t_{16,2}$ in each case

Due to the factorized form of the stability polynomial one can directly read-off a possible choice of the intermediate steps:

$$P_S(z) = 1 + z\widetilde{P}_{S-1} = 1 + z \underbrace{\left(\prod_{j=1}^{S_{\text{Real}}} \underbrace{\left(1 - \frac{z}{\widetilde{r}_j}\right)}_{\text{Forward Euler}} \cdot \prod_{j=1}^{S_{\text{Complex}}/2} \underbrace{\left(1 - \frac{z}{\widetilde{r}_j}\right)\left(1 - \frac{z}{\widetilde{r}_j^\star}\right)}_{\text{Two-Stage Submethod}} \right)}_{\text{Forward Euler}} \tag{9.1}$$

Note that we group the complex-conjugated pseudo-extrema to avoid complex-valued timesteps. We focus on real timesteps since they allow easy implementation into existing codes and spare us from technical complications. Readers interested in complex-valued timesteps are pointed to [81] and references therein.

Being equipped with a stability polynomial, one has now in principle infinitely many possible choices for the actual Runge–Kutta method. It is well-known that internal stability, i.e., the propagation of round-off errors is a relevant concern for many-stage explicit methods [56, 57, 82, 83]. This issue arises also here and seems to be the major difficulty in realizing the maximum timestep.

Given the factorized form of the stability polynomial it is natural to construct Runge–Kutta methods with a sequential structure. Here, we specify the methods in *modified Shu-Osher form* [82, 84]

$$\boldsymbol{Y}_k := v_k \boldsymbol{U}_n + \sum_{l=1}^{k-1} \left(\alpha_{k,l} \boldsymbol{Y}_l + \Delta t \beta_{k,l} \boldsymbol{F}(\boldsymbol{Y}_l) \right), \quad k = 1, \dots, S+1 \tag{9.2a}$$

$$\boldsymbol{U}_{n+1} = \boldsymbol{Y}_{S+1} \tag{9.2b}$$

and set $v_1 = 1, v_k = 0, k > 1$ which implies $\boldsymbol{Y}_1 = \boldsymbol{U}_n$. Following [82] we introduce the notation

$$\alpha_{1:S}|_{i,j} := \alpha_{i,j}, \quad \beta_{1:S}|_{i,j} := \beta_{i,j} \tag{9.3}$$

with $\alpha_{1:S}, \beta_{1:S} \in \mathbb{R}^{S \times S}$ resembling the coefficients of the intermediate stages. The parameters of the final stage are denoted via $\boldsymbol{\alpha}_{S+1}, \boldsymbol{\beta}_{S+1} \in \mathbb{R}^{1 \times S}$, i.e.,

$$\boldsymbol{\alpha}_{S+1} := \left(\alpha_{S+1,1} \dots \alpha_{S+1,S} \right), \quad \boldsymbol{\beta}_{S+1} := \left(\beta_{S+1,1} \dots \beta_{S+1,S} \right) \tag{9.4}$$

which are in the present case conveniently chosen as

$$\boldsymbol{\alpha}_{S+1} = \left(1 \dots 0 \right), \quad \boldsymbol{\beta}_{S+1} = \left(0 \dots 0 \; z \right) \tag{9.5}$$

due to the particular representation of the stability polynomial (9.1).

The two-stage submethod (cf. (9.1)) corresponding to pseudo-extrema $\widetilde{r}_j, \widetilde{r}_j^\star$ are naturally represented via the low-storage scheme

$$\boldsymbol{Y}_k = \boldsymbol{Y}_{k-1} + \beta_{k,k-1}^{(j)} \boldsymbol{F}\left(\boldsymbol{Y}_{k-1}\right) \tag{9.6a}$$

$$\boldsymbol{Y}_{k+1} = \alpha_{k+1,k-1}^{(j)} \boldsymbol{Y}_{k-1} + \alpha_{k+1,k}^{(j)} \boldsymbol{Y}_k + \Delta t \left(\beta_{k+1,k-1}^{(j)} \boldsymbol{F}(\boldsymbol{U}_{k-1}) + \beta_{k+1,k}^{(j)} \boldsymbol{F}(\boldsymbol{Y}_k) \right). \tag{9.6b}$$

The Forward Euler steps are readily reflected by $\alpha_{k,k-1}^{(j)} = 1, \beta_{k,k-1}^{(j)} = \frac{-1}{\widetilde{r}_j}$. To meet the stability polynomial (9.1) we have for the two-stage methods the constraints

$$\frac{-2\mathrm{Re}(\widetilde{r}_j)}{|\widetilde{r}_j|^2} \stackrel{!}{=} \alpha_{k+1,k}^{(j)} \beta_{k,k-1}^{(j)} + \beta_{k+1,k-1}^{(j)} + \beta_{k+1,k}^{(j)} \tag{9.7a}$$

$$\frac{1}{|\widetilde{r}_j|^2} \stackrel{!}{=} \beta_{k,k-1}^{(j)} \beta_{k+1,k}^{(j)} \tag{9.7b}$$

besides the usual requirement for consistency $\sum_{l=0}^{k-1} \alpha_{k,l} = 1$ [60]. The timestamps/abscissae are computed via $\boldsymbol{c} := (I - \alpha_{1:S})^{-1} \beta_{1:S} \mathbf{1}$ [82] where $\mathbf{1}$ denotes a column vector of ones. The ordering of the two-stage submethods, i.e., the relation of j to k is determined based on the accumulated β values

$$\left\| \boldsymbol{\beta}_k^{(j)} \right\|_1 = \left\| \left(\beta_{k,k-1}^{(j)}, \beta_{k+1,k-1}^{(j)}, \beta_{k+1,k}^{(j)} \right) \right\|_1. \tag{9.8}$$

The methods are then ordered such that $\left\| \boldsymbol{\beta}_k^{(j)} \right\|_1$ increases with k.

9.1 Internal Stability

For the discussion of internal stability we follow the main results derived in [82]. Under the influence of round-off errors $\boldsymbol{e}_k$ the modified Shu-Osher form (9.2a) reads

$$\widetilde{\boldsymbol{Y}}_k := v_k \widetilde{\boldsymbol{U}}_n + \sum_{l=1}^{k-1} \left(\alpha_{j,l} \widetilde{\boldsymbol{Y}}_l + \Delta t \beta_{k,l} \boldsymbol{F}(\widetilde{\boldsymbol{Y}}_l) \right) + \boldsymbol{e}_k, \quad k = 1, \ldots, S+1 \tag{9.9}$$

where we denote the perturbed, round-off error bearing stages by $\widetilde{\boldsymbol{Y}}_k$. In [82] it was shown that the defect from actually computed, perturbed solution $\widetilde{\boldsymbol{U}}_n$ to the true solution $\boldsymbol{U}(t_n)$ can for linear ordinary differential equations (ODEs) be estimated as

$$\left\| \widetilde{\boldsymbol{U}}_{n+1} - \boldsymbol{U}(t_{n+1}) \right\| \leq \left\| \widetilde{\boldsymbol{U}}_n - \boldsymbol{U}(t_n) \right\| + \sum_{k=1}^{S+1} \left\| Q_j(z) \right\| \left\| \boldsymbol{e}_k \right\| + \mathcal{O}(\Delta t^{p+1}). \tag{9.10}$$

In addition to the usual truncation error $\mathcal{O}(\Delta t^{p+1})$ we have the amplification of round-off errors $\sum_k \|Q_k(z)\| \|\boldsymbol{e}_k\|$. Internal stability becomes a concern when both are of similar magnitude [82].

In the estimate of the error of the perturbed iterates (9.10) $Q_j(z)$ denote the *internal stability polynomials* [56, 82] which are for a method in Shu-Osher form computed as [82]

$$\boldsymbol{Q}(z; \alpha, \beta) := (\boldsymbol{\alpha}_{S+1} + z\boldsymbol{\beta}_{S+1}) \left(I - \alpha_{1:S} - z\beta_{1:S} \right)^{-1}. \tag{9.11}$$

For explicit methods there is no initial round-off error $\boldsymbol{e}_1$ [82] and it suffices to consider the internal stability polynomials starting from second stage. To estimate the potential of round-off error amplification it is customary to investigate

$$\widetilde{\mathcal{M}}(\alpha, \beta) := \max_{z \in \mathcal{S}} \sum_{k=2}^{S+1} |Q_k(z; \alpha, \beta)| \tag{9.12}$$

which is a more precise variant of the *maximum internal amplification factor* proposed in [82]. For standard double precision floating point datatypes we expect the round-off errors to be $\varepsilon = \mathcal{O}\left(10^{-15}\right)$ at most, thus we expect the overall internal errors to be of order $\widetilde{\mathcal{M}} \cdot 10^{-15}$. This can then be compared to $\mathcal{O}(\Delta t^{p+1})$ yielding an approximate criterion to estimate the influence of round-off errors. Both metrics are reported for the examples considered in Sect. 10.

In principle, one could now set out to minimize $\widetilde{\mathcal{M}}$ over α, β under the constraints (9.7). Given that even for methods with two intermediate stages (9.6) we have $\frac{5}{2} S_{\text{Complex}} + S_{\text{Real}}$ free optimization variables and one is minimizing not only one, but S polynomials at M eigenvalues this is a significantly harder optimization problem than the original one (2.7). In addition, there are no immediate properties of the internal stability polynomials Q_j available, rendering this problem practically infeasible. As an alternative, we propose a significantly cheaper heuristically motivated approach in the next section.

Table 3 Internal error amplification $\widetilde{M}$ (9.12) for different method construction paradigms

	$\beta_{k,l} \in \mathbb{R}_0^+$	$\beta_{k,l} \in \mathbb{R}$
No Grouping	$2.39 \cdot 10^{22}$	$2.19 \cdot 10^{22}$
Grouping	$7.54 \cdot 10^{16}$	$1.63 \cdot 10^{9}$

9.2 Internal Stability Optimization: Heuristic Approach

We begin by recalling that $\alpha_{1:S}, \beta_{1:S}$ are for explicit methods strictly lower triangular, i.e., nilpotent matrices with index S. Consequently, the identity

$$(I - \alpha_{1:S} - z\beta_{1:S})^{-1} = \sum_{k=0}^{S-1} (\alpha_{1:S} + z\beta_{1:S})^k \tag{9.13}$$

holds and allows an alternative to the inversion of the typically ill-conditioned matrix. We make now the following heuristic argument: For $\alpha_{k,l} \in [0, 1]$ and $|z| \sim \mathcal{O}(100)$ for the many-stage methods we expect a significantly higher sensitivity of $\widetilde{M}$ with respect to $\beta_{k,l}$ than $\alpha_{k,l}$. In order to minimize $\widetilde{M}$ we thus minimize the accumulated $\beta_{k,l}$ values (9.8) for each two-stage submethod individually. For these optimization problems constraints (9.7) satisfying local minima are easily found. Note that negative $\beta_{k,l}$ are allowed here, which allow further reduction of $\widetilde{M}$ due to the increased flexibility. To illustrate this, consider the 104 degree stability polynomial optimized for the scalar advection equation (see Sect. 10.1.1) for which we construct Runge–Kutta methods with non-negative and unconstrained $\beta_{k,l}$. For this case, values of $\mathcal{M}$ for grouped and non grouped pseudo-extrema (see next section) with non-negative and unconstrained $\beta_{k,l}$ are tabulated in Table 3.

9.3 Generalized Lebedev's Idea

For the higher order DG spectra we observe a couple of pseudo-extrema near the imaginary axis with $\left|\mathrm{Im}(\widetilde{\boldsymbol{r}}_j)\right| \gg -\mathrm{Re}(\widetilde{\boldsymbol{r}}_j)$ which can lead to $\beta_{k,l} \gg 1$, which should be avoided. Similar to Lebedev's Idea/Realization [85, 86] we group the complex conjugated pseudo-extrema with $\mathrm{Re}(\widetilde{\boldsymbol{r}}_j) > -0.5$ with the pseudo-extrema of largest $\left|\mathrm{Re}(\widetilde{\boldsymbol{r}}_j)\right|$. By staying with the two register form (9.6) we have now 13 free parameters, three linear constraints due to $\sum_{k=0}^{j-1} \alpha_{j,k} = 1$ and four nonlinear constraints which are derived similar to (9.7) by multplying out polynomials. As they are quite lengthy but readily obtained we omit them here. As for the remaining two-stage methods, we optimize α, β to minimize $\left\|\boldsymbol{\beta}_k^{(j)}\right\|_1$.

The grouping of pseudo-extrema with small and large real part has a dramatic influence on the internal stability properties. For instance, consider again the 104 degree, second order accurate stability polynomial optimized for the scalar advection equation (see Sect. 10.1.1) for which we construct Runge–Kutta methods with and without grouping. Values for $\widetilde{\mathcal{M}}$ are tabulated in Table 3 highlighting the importance of the grouping. Due to this results all in the following constructed methods bear grouping of pseudo-extrema and may have negative $\beta_{k,l}$.

9.4 Note on SSP Properties

It is natural to ask for SSP properties of Runge–Kutta schemes intended for the integration of (nonlinear) hyperbolic systems. Furthermore, given the representation of the method in Shu-Osher form (9.2) one can readily tell whether the method is monotonicity preserving or not. For the methods resembling the factorized form of the stability polynomial the coefficients of the last stage are chosen according to (9.5). Consequently, the SSP cofficient [60]

$$c := \min_{k,l} \frac{\alpha_{k,l}}{\beta_{k,l}} \tag{9.14}$$

is due to $\alpha_{S+1,S} = 0, \beta_{S+1,S} = 1$ always zero, regardless of the parametrization of the intermediate stages. This implies that there exists no timestep $\Delta t > 0$ such that the Runge–Kutta method is guaranteed to be monotonicity preserving. One will also need to revisit whether negative $\beta_{k,l}$ should be allowed as they demand special treatment for SSP methods [60]. The construction of many-stage SSP methods is ongoing research.

10 Many-Stage Runge–Kutta Methods: Results

We present convergence studies of the many-stage Runge–Kutta methods which are constructed from the factorized form of the stability polynomial using the generalization of Lebedev's Idea and possibly negative $\beta_{k,l}$.

10.1 Linear Problems

10.1.1 1D Advection of a Gaussian Pulse

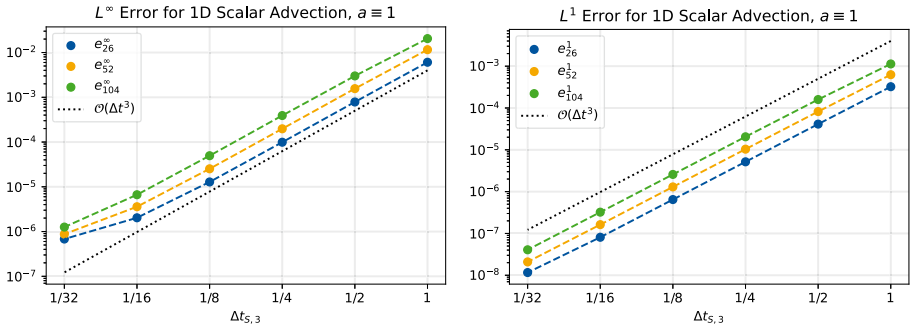
As a first example, we consider the linear advection equation with unit transport velocity and Gaussian initial data

$$u_t + u_x = 0, \quad u(x,0) = u_0(x) = \exp\left(\frac{-x^2}{0.1}\right) \tag{10.1}$$

on $\Omega = [-5, 5]$ equipped with periodic boundaries. We discretize Ω with 512 cells and (10.1) via the DGSEM with third order polynomials and Rusanov/Local Lax-Friedrichs flux implemented in `Trixi.jl` [65–67]. Using `Trixi.jl` we can generate the Jacobian using algorithmic differentiation and compute the corresponding spectrum. For this spectrum we compute the reference timestep $\Delta_{16,3} \approx 3.53 \cdot 10^{-2}$ using the algorithm proposed in [36] which is then supplied to the feasibility problems (5.4) and (5.5).

We compute ten passes through the domain corresponding to final time $t_f = 100$ with 26, 52, and 104 stage, third order accurate methods using the Runge–Kutta parametrization with negative β and generalized Lebedev's idea. As both the timestep as the computational cost (due to the low-storage formulation of the Runge–Kutta method) scale linearly with the number of stages S, we obtain constant runtimes for this examples. The used timestep Δt, number of taken timesteps N_t, internal amplification factors $\widetilde{\mathcal{M}}$ and the order of the truncation error Δt^4 are reported in Table 4. The third order convergence in L^∞-norm

$$e_S^\infty := \left\| u^{(h,S)}(t_f, x) - u(t_f, x) \right\|_\infty \tag{10.2}$$



(a) L^∞ errors for scalar advection equation. (b) Weighted L^1 errors for scalar advection equation.

Fig. 8 Third order convergence in L^∞, L^1 for methods with 26, 52, and 104 stages optimized for the 1D scalar advection equation (10.1)

Table 4 Results for the simulation of the 1D scalar advection equation (10.1) with three many-stage optimized methods constructed using Algorithm 1

S	Δt	N_t	$\widetilde{\mathcal{M}} \cdot 10^{-15}$	Δt^4
26	$5.72 \cdot 10^{-2}$	1748	$3.59 \cdot 10^{-12}$	$1.07 \cdot 10^{-5}$
52	$1.14 \cdot 10^{-1}$	874	$1.46 \cdot 10^{-9}$	$1.72 \cdot 10^{-4}$
104	$2.29 \cdot 10^{-1}$	437	$5.39 \cdot 10^{-5}$	$2.75 \cdot 10^{-3}$

The timestep Δt is kept constant over the N_t timesteps, with the possible exemption of the last timestep which may be reduced in order to match the desired final time $t_f = 100$

is observed until spatial discretization errors are becoming relevant, cf. Fig. 8a. In weighted L^1-norm

$$e_S^1 := \frac{1}{|\Omega|} \left\| u^{(h,S)}(t_f, x) - u(t_f, x) \right\|_1 \tag{10.3}$$

third order convergence is observed except for the smallest timestep, as displayed in Fig. 8b. We observe that the error constant increases with stage number, which seems to be a general phenomenon for the herein constructed methods.

10.1.2 2D Linearized Euler Equations

The linearized Euler equations with source term read in two spatial dimensions, primitive variables

$$\partial_t \begin{pmatrix} \rho' \\ u' \\ v' \\ p' \end{pmatrix} + \partial_x \begin{pmatrix} \bar{\rho} u' + \bar{u} \rho' \\ \bar{u} u' + \frac{p'}{\bar{\rho}} \\ \bar{u} v' \\ \bar{u} p' + c^2 \bar{\rho} u' \end{pmatrix} + \partial_y \begin{pmatrix} \bar{\rho} v' + \bar{v} \rho' \\ \bar{v} u' \\ \bar{v} v' + \frac{p'}{\bar{\rho}} \\ \bar{v} p' + c^2 \bar{\rho} v' \end{pmatrix} = \boldsymbol{s} \tag{10.4}$$

where $(\bar{\rho}, \bar{u}, \bar{v}, c) = (1, 1, 1, 1)$ denote the base state. By the method of *manufactured solutions* [87, 88] we set the solution to

$$\begin{pmatrix} \rho' \\ u' \\ v' \\ p' \end{pmatrix} = \begin{pmatrix} -\cos(2\pi t)\,[\sin(2\pi x) - \sin(2\pi y)] \\ \sin(2\pi t)\cos(2\pi x) \\ \sin(2\pi t)\cos(2\pi y) \\ -\cos(2\pi t)\,[\sin(2\pi x) - \sin(2\pi y)] \end{pmatrix} \tag{10.5}$$

Table 5 Results for the simulation of the 2D linearized Euler equations (10.4) with three many-stage optimized methods constructed using Algorithm 1

S	Δt	CFL	N_t	$\widetilde{\mathcal{M}} \cdot 10^{-15}$	Δt^3
32	$1.13 \cdot 10^{-2}$	1.0	928	$1.91 \cdot 10^{-11}$	$1.45 \cdot 10^{-6}$
64	$2.17 \cdot 10^{-2}$	0.96	484	$6.30 \cdot 10^{-9}$	$1.03 \cdot 10^{-5}$
128	$3.67 \cdot 10^{-2}$	0.81	287	$5.63 \cdot 10^{-5}$	$4.94 \cdot 10^{-5}$

The timestep Δt is kept constant over the N_t timesteps, with the possible exemption of the last timestep which may be reduced in order to match the desired final time $t_f = 10.5$

with corresponding source terms

$$\boldsymbol{s}(t, x, y) = 2\pi \begin{pmatrix} -\cos(2\pi t)\,[\cos(2\pi x) - \cos(2\pi y)] \\ \sin(2\pi t)\sin(2\pi x) \\ \sin(2\pi t)\sin(2\pi y) \\ -\cos(2\pi t)\,[\cos(2\pi x) - \cos(2\pi y)] \end{pmatrix}. \tag{10.6}$$

The linearized Euler equations (10.4) are discretized on $\Omega = [0, 1]^2$ with 64 elements in each direction. We employ the DGSEM with fourth order local polynomials and Harten-Lax-Van Leer (HLL) [89] flux on a periodic domain. To leverage the computational costs of the eigenvalue decomposition, we compute the spectrum for a discretization with 8 cells in each coordinate direction. For the resulting set of eigenvalues we compute the reference timestep $\Delta t_{16,2} \approx 4.53 \cdot 10^{-2}$ using the algorithm proposed in [36] which is then supplied to the feasibility problems (5.4) and (5.5).

For the actual simulation on the 8 times finer mesh we reduce the obtained timesteps accordingly. A convergence study is carried out in terms of the error in density fluctuation ρ at final time $t_f = 10.5$ and displayed in Fig. 9. We use $t_f = 10.5$ since we observed for $t_f = 10$ spurious convergence of higher than second order, which is a peculiarity of the employed testcase rather than a general phenomenon. Again, second order convergence is observed in L^∞ for all timesteps. Here, we construct polynomials of degrees 32, 64, and 128 and the corresponding methods with parametrization using potentially negative β and the generalized Lebedev's idea. Since the right-hand-side of the ODE system (1.1b) depends also explicitly on time this example also showcases convergence of the many-stage methods for non-homogeneous problems.

In contrast to the previous example, we observe that the theoretically possible maximum stable timestep has to be reduced by factor CFL (see Table 5) for a stable simulation which we attribute to issues with internal stability.

Consider for instance the 128 stage method with CFL = 1.0 (corresponding to an unstable simulation) with $\widetilde{\mathcal{M}} \cdot 10^{-15} \approx 7.60 \cdot 10^{-2}$ while $\Delta t^3 \approx 9.29 \cdot 10^{-4}$. As discussed in [82], we observe in this case stability problems when error terms due to round off are of larger magnitude than the truncation error.

For the $S = 64$ stability polynomial the reduction in optimal timestep is a consequence of the optimization of the stability polynomials for the reduced spectrum, rather than related to internal stability.

10.2 Nonlinear Problems

10.2.1 Burgers' Equation with Source Term

We consider Burgers' Equation as the classic prototype equation for nonlinear hyperbolic PDEs. With source term $s(t, x)$, Burgers' Equation reads

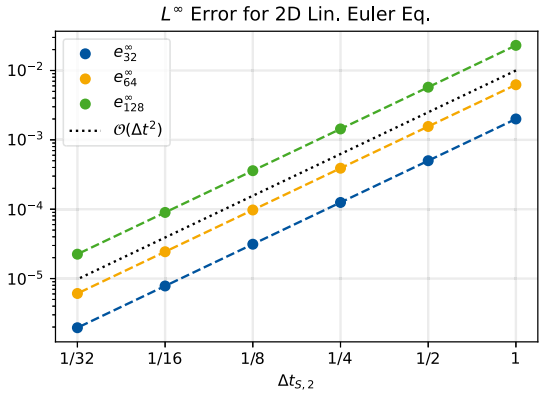


Fig. 9 Second order convergence of the optimized methods with 32, 64, and 128 stages optimized for the 2D linearized Euler equations. For each method, we reduce from the largest stable timestep CFL $\cdot\, \Delta t_{S,2}$

$$\partial_t u + \frac{1}{2}\partial_x u^2 = s. \tag{10.7}$$

The presence of the source term allows a convenient construction of a continuously differentiable solution $u(t, x) \in C^1$ via the method of manufactured solutions. Here, the solution is set to

$$u(t, x) := 2 + \sin\left(2\pi(x - t)\right) \tag{10.8}$$

which corresponds to a simple advection of the initial condition, although this time through a nonlinear partial differential equation (PDE). Note that $u(t, x)$ is periodic on $[0, 1]$ for all times t. The source term is then computed from the governing PDE (10.7) as

$$s(t, x) := 2\pi \cos\left(2\pi(x - t)\right)\Big(1 + \sin\left(2\pi(x - t)\right)\Big). \tag{10.9}$$

The spatial discretization is realized using the DGSEM with third order polynomials on a 256 element mesh $\Omega^{(h)}$ with periodic boundaries and Godunov flux [90].

Despite being effectively also only the simple advection of the initial data $u_0(x)$, it is now observed that the effectively stable timestep cannot be kept constant as in the case of the advection equation discussed previously. This is attributed to the fact that the perturbations $\boldsymbol{e}_j$ do not only corrupt the stage updates (cf. (9.9)), but thereby also change the point of evaluation of the Jacobian and thus the spectrum. This can have dramatic effects since the stability polynomial is only optimized for the spectrum corresponding to the unperturbed solution. This is further amplified by the lack of the SSP property of the constructed schemes. Thus, as the time integration schemes used here have no oscillation-surpressing guarantees, these do arise and further perturb the spectrum. To measure this, we compute the increase of the *total variation*

$$e_{\mathrm{TV}} := \left\|\boldsymbol{u}^{(h)}\left(t_f\right)\right\|_{\mathrm{TV}} - \left\|\boldsymbol{u}^{(h)}\left(t_0\right)\right\|_{\mathrm{TV}} \tag{10.10}$$

where the total variation semi-norm is defined as usual

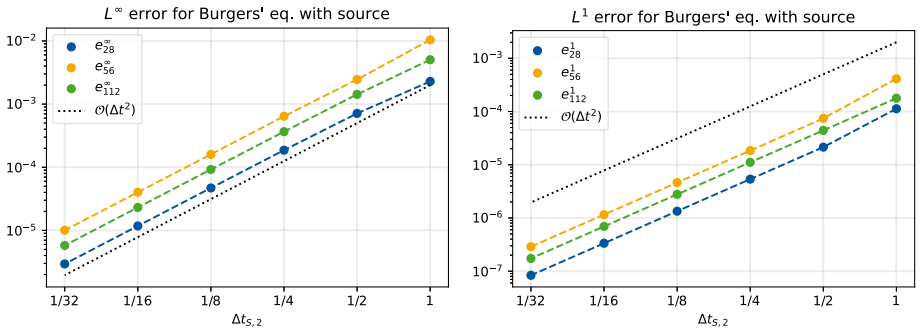
$$\|\boldsymbol{u}(t)\|_{\mathrm{TV}} := \sum_j \left|u_{j+1}(t) - u_j(t)\right|. \tag{10.11}$$

For methods with $S = 28$, 56, and 112 stages we observe indeed increase in total variation, cf. Table 6. Here, the significant decrease of the optimal timestep even for methods with moderate stages is not expected to follow from issues with internal stability. Although the

Table 6 Results for the simulation of Burgers equation (10.7) with three many-stage optimized methods constructed using Algorithm 1

S	Δt	CFL	N_t	$\widetilde{\mathcal{M}} \cdot 10^{-15}$	Δt^3	e_{TV}
28	$3.67 \cdot 10^{-3}$	0.72	1363	$1.41 \cdot 10^{-12}$	$4.95 \cdot 10^{-8}$	$1.30 \cdot 10^{-2}$
56	$7.04 \cdot 10^{-3}$	0.69	711	$1.39 \cdot 10^{-10}$	$3.48 \cdot 10^{-7}$	$5.18 \cdot 10^{-2}$
112	$3.99 \cdot 10^{-3}$	0.28	876	$4.79 \cdot 10^{-10}$	$1.86 \cdot 10^{-7}$	$6.09 \cdot 10^{-3}$

The timestep Δt is kept constant over the N_t timesteps, with the possible exemption of the last timestep which may be reduced in order to match the desired final time $t_f = 5$



(a) L^∞ errors for Burgers' equation. (b) Weighted L^1 errors for Burgers' equation.

Fig. 10 Second order convergence in L^∞, L^1 for methods optimized for Burgers' equation with source (10.7)

estimate for the perturbed approximation (9.10) is only valid for linear systems, we consider it also here as a crude estimate for the influence of internal stability. As note in table Table 6 the maximum possible timestep $\Delta t_{28,2}$ has to be decreased to 72% of its theoretically possible value. As $\widetilde{M} \cdot 10^{-15}$ is still 4 orders of magnitude smaller than Δt^3 we do not expect the round off errors to limit the stability.

Instead, we believe that the lack of the SSP property causes the limitations in timestep. To provide reason for this, we consider SSP the third order accurate methods with variable number of stages $S_n = n^2$ proposed in [91]. For the $S = 15^2 = 225$ method we have $\widetilde{M} \cdot 10^{-15} \approx 2.13 \cdot 10^{-13}$ and $\Delta t^4 \approx 5.42 \cdot 10^{-8}$, i.e., similar values to the 28 stage method constructed here. In that case, however, we have $e_{\mathrm{TV}} \equiv 0$, as expected for the TVD time integration when applied to a smooth solution. For the schemes constructed here the spurious oscillations can even produce amplifying modes in the spectrum of the semidiscretization, i.e., move the system into an unphysical, energy-generating state. In that case, the true solution itself is unstable - this is a qualitatively different scenario to the lack of absolute stability where the numerical scheme is unstable for a certain eigenvalue. For $S = 28$ and Δt from Table 6 we have for instance at $t = 0.5$ an eigenvalue with $\mathrm{Re}(\lambda) = 8.57 \cdot 10^{-12}$ which bears the possibility to cause the whole computation to diverge. This effect becomes especially severe for the 112 stage method where the CFL number has to be dramatically decreased for a stable simulation.

Nevertheless, all methods achieve second order convergence in L^∞-norm once the oscillations vanished which is the case for $\Delta t \leq \frac{1}{4}\Delta t_{S,2}$, cf. Fig. 10a. In weighted L^1-norm (10.3) second order convergence is observed for every timestep as displayed in Fig. 10b.

10.2.2 2D Compressible Euler Equations: Isentropic Vortex Advection

To conclude the examples, we consider the 2D compressible Euler equations

$$\partial_t \rho + \nabla \cdot \begin{pmatrix} \rho v_x \\ \rho v_y \end{pmatrix} = 0 \tag{10.12a}$$

$$\partial_t \begin{pmatrix} \rho v_x \\ \rho v_y \end{pmatrix} + \nabla \cdot \begin{pmatrix} \rho v_x^2 & \rho v_x v_y \\ \rho v_y v_x & \rho v_y^2 \end{pmatrix} + \nabla p = \mathbf{0} \tag{10.12b}$$

$$\partial_t E + \nabla \cdot \begin{pmatrix} (E+p) v_x \\ (E+p) v_y \end{pmatrix} = 0 \tag{10.12c}$$

with total energy $E = E(\rho, \boldsymbol{v}, p) = \rho \left(\frac{p}{\gamma - 1} + \frac{1}{2} \left(v_x^2 + v_y^2 \right) \right)$. As a testcase we consider the advection of an isentropic vortex [80, 92]. Here, we use similar parameters to the ones used in [39, 40]. In particular, the base state is set to

$$\rho_\infty = 1, \quad \boldsymbol{v}_\infty = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad p_\infty := \frac{\rho_\infty^\gamma}{\gamma \mathrm{Ma}_\infty^2} \tag{10.13}$$

with $\mathrm{Ma}_\infty = 0.4$. To localize the effect of the vortex centered at $\boldsymbol{c}(t, x, y) := \begin{pmatrix} x \\ y \end{pmatrix} - \boldsymbol{v}_\infty t$ the perturbations are weighted with the Gaussian $g(t, x, y) := \exp\left(\frac{1 - \|\boldsymbol{c}(t,x,y)\|_2^2}{2R^2} \right)$ where $R = 1.5$. While the size of the vortex is governed by R, its intensity/strength is quantified by I which is set here to 13.5 following [39, 40]. The density is given by

$$\rho(t, x, y) = \rho_\infty \left(1 - \frac{I^2 M^2 (\gamma - 1) g^2(t, x, y)}{8\pi^2} \right)^{\frac{1}{\gamma - 1}} \tag{10.14}$$

and the corresponding perturbed velocities are

$$\boldsymbol{v}(t, x, y) = \boldsymbol{v}_\infty + \frac{I g(t, x, y)}{2\pi R} \boldsymbol{c}(t, x, y) \tag{10.15}$$

while the pressure is computed analogous to the base pressure (10.13) as

$$p(t, x, y) = \frac{\rho^\gamma(t, x, y)}{\gamma \mathrm{Ma}_\infty^2}. \tag{10.16}$$

The advection of the vortex is simulated on $\Omega = [-10, 10]^2$ discretized with 64×64 elements, sixth order local polynomials, and HLLC Flux [93]. Final time t_f is set to 20 which corresponds to a full traversion of the vortex through the periodic domain. For this setup 30, 60, and 120 degree methods of second order accuracy are constructed. As for the previous example, we observe a reduction of the theoretically stable timestep as given in Table 7 which we attribute again to the lack of SSP guarantees, which cause the rise of oscillations that move the system into an amplifying state.

For the L^∞ error in density $\|\rho^{(h)}(t_f, x, y) - \rho(t_f, x, y)\|_\infty$ second order convergence is observed, until spatial discretization errors start to become significant, cf. Fig. 11.

Table 7 Results for the simulation of the 2D compressible Euler equations (10.12) with three many-stage optimized methods constructed using Algorithm 1

S	Δt	CFL	N_t	$\widetilde{\mathcal{M}} \cdot 10^{-15}$	Δt^3
30	$5.92 \cdot 10^{-2}$	0.79	338	$1.80 \cdot 10^{-12}$	$2.08 \cdot 10^{-4}$
60	$9.75 \cdot 10^{-2}$	0.65	206	$3.21 \cdot 10^{-11}$	$9.26 \cdot 10^{-4}$
120	$1.20 \cdot 10^{-1}$	0.40	167	$6.50 \cdot 10^{-10}$	$1.72 \cdot 10^{-3}$

The timestep Δt is kept constant over the N_t timesteps, with the possible exemption of the last timestep which may be reduced in order to match the desired final time $t_f = 20$

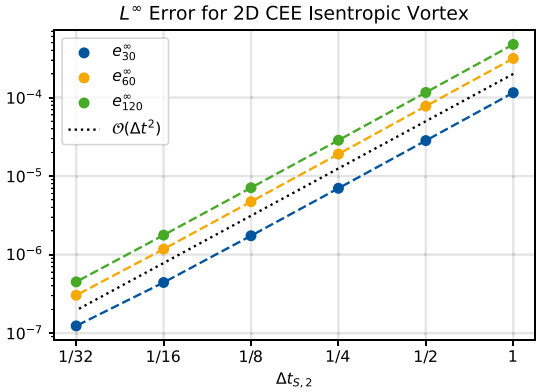


Fig. 11 Second order convergence of the optimized methods with 30, 60, and 120 stages optimized for the 2D compressible Euler equations. For each method, we reduce from the largest stable timestep CFL $\cdot \Delta t_{S,2}$

11 Conclusions

In this work, a novel optimization approach for the generation of optimal stability polynomials for spectra of hyperbolic PDEs is devised. Parametrizing the stability polynomials in terms of the herein introduced pseudo-extrema offers both a numerically stable and readily interpretable representation. The optimization approach is motivated by examining the properties of the pseudo-extrema for the proven optimal stability polynomials of first and second order for disks. These findings are directly extended to strictly convex spectra for which an optimization procedure is developed in detail. For non-convex spectra, possible remedies are discussed and optimal stability polynomials for both convex and nonconvex spectra are presented. Stability polynomials with degrees larger than 100 are constructed for a range of classical hyperbolic PDEs that match the linear consistency requirements up to order three.

Equipped with high degree stability polynomials in factorized form a possible choice for the actual Runge–Kutta methods is proposed. The construction of the numerical schemes centers around minimizing the propagation and amplification of round-off errors, which can become a concern for many-stage methods. Here, methods of second order are constructed which match their designed order of accuracy for both linear and nonlinear problems. For linear problems, only internal stability might limit the theoretically possible maximum timestep while for nonlinear problems the lack of the SSP property is much more severe which spoils the effectiveness of the very high-stage methods.

As mentioned earlier, the application of *individual* optimized many-stage methods do not provide efficiency gains compared to medium- or few-stage methods as both the stable timestep and the computational cost scale linearly in the number of stages. In fact, the main motivation for the construction of such high degree stability polynomials are multiscale problems where a *composition* of stabilized Runge–Kutta methods is applied to integrate the

system more efficiently. In this scenario, one would adjust the number of stages according to the characteristic speeds in a part of the domain with the aim to keep the overall timestep constant. For this case, the relatively robust medium-stage methods with up to, say, 32 stages are a promising candidate to be included in the family of Runge–Kutta methods.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10915-024-02478-5.

Acknowledgements We thank the reviewers for taking the time and effort necessary to review this manuscript. We appreciate the valuable comments and suggestions which improved the quality of the manuscript.

Funding Open Access funding enabled and organized by Projekt DEAL. Funding by German Research Foundation (DFG) under grant number DFG-FOR5409.

Data availibility All data generated or analysed during this study are included in this published article and its supplementary information files.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

Appendix A. Pseudo-Extrema of Shifted Chebyshev Polynomials

Consider the shifted Chebyshev polynomials of first kind

$$T_S\left(1+z/S^2\right) = \cos\left(S \arccos\left(1+z/S^2\right)\right) \tag{A.1}$$

providing the optimal first order stability polynomial for parabolic spectra [2, 3, 11]. It is well-known that the $S+1$ extrema of the shifted Chebyshev polynomials are on the $[-2S^2, 0]$ interval given by

$$x_j := S^2\left(\cos\left(\frac{j\pi}{S}\right) - 1\right), \quad j = 0, \ldots, S \tag{A.2}$$

where $T_S\left(1+x_j/S^2\right)$ attains either $+1$ or -1. In particular, (A.2) includes the $S-1$ critical points corresponding to $j = 1, \ldots, S-1$ besides the extrema at the ends of the domain which correspond to $j = 0$ (right end) and $j = S$ (left end).

We can rewrite the stability polynomial according to (2.10) as

$$\widetilde{P}_{S-1}(z;\widetilde{\boldsymbol{r}}) = \frac{P_{S,1}(z;\widetilde{\boldsymbol{r}}) - 1}{z} \tag{A.3}$$

and seek out to determine the pseudo-extrema $\widetilde{\boldsymbol{r}}$, i.e., the roots of $\widetilde{P}_{S-1}$. Apart from $z = 0$ the roots of $\widetilde{P}_{S-1}$ are given by the roots of the nominator, i.e.,

$$P_{S,1}(z;\widetilde{\boldsymbol{r}}) - 1 = 0. \tag{A.4}$$

For even degree S, $S/2+1$ roots of (A.4) are given by the Chebyshev extreme points where $T_S\left(1+x_j/S^2\right)$ takes value $+1$:

$$\widetilde{r}_j = S^2\left(\cos\left(\frac{2j\pi}{S}\right)-1\right), \quad j=0,\dots,S/2. \tag{A.5}$$

We recall that the nominator of (A.3) $P_{S,1}(z)-1 = T_S\left(1+z/S^2\right)-1$ is a polynomial of degree S in real coefficients and thus has, by the fundamental theorem of algebra, S (possibly complex-conjugated) roots. As we have found already $S/2+1$ real roots it follows that the remaining $S/2-1$ roots have to be multiples of the already found ones. In particular, we have that the "outer" roots $\widetilde{r}_0 = -2S^2$ and $\widetilde{r}_{S/2} = 0$ have multiplicity one, and the "interior" roots $\widetilde{r}_j$, $j=1,\dots S/2-1$ have each multiplicity two. This follows from the fact that the $\widetilde{r}_j$, $j=1,\dots S/2-1$ are a subset of the critical points of $P_{S,1}(z)-1$, i.e., $\frac{\mathrm{d}}{\mathrm{d}z}\left(T_S\left(1+z/S^2\right)-1\right)|_{\widetilde{r}_j}=0,\ j=1,\dots S/2-1$.

Considering the roots of (A.3) we have to be careful when it comes to $j=0$, i.e., $x_j=0$ since we divide by $\widetilde{r}_j=0$. Recalling the introduction of the lower degree polynomial (2.10) we have by construction $\widetilde{P}_{S-1}(0;\widetilde{\boldsymbol{r}})=1$. As a consequence, we have that the $S-1$ roots of the lower-degree polynomial $\widetilde{P}_{S-1}(z;\widetilde{\boldsymbol{r}})$ are given by

$$\widetilde{r}_j = S^2\left(\cos\left(\frac{2j\pi}{S}\right)-1\right), \quad j=1,\dots,S/2 \tag{A.6}$$

where $\widetilde{r}_{S/2}$ has multiplicity one and the remaining $S/2-1$ roots have multiplicity two.

For odd degree S, we can follow a similar reasoning which yields that the pseudo-extrema in this case are given by

$$\widetilde{r}_j = S^2\left(\cos\left(\frac{2j\pi}{S}\right)-1\right), \quad j=1,\dots,\frac{S-1}{2} \tag{A.7}$$

which all have multiplicity two. This follows from the fact that for odd S, the extremum at the left end of the $\left[-2S^2,0\right]$ interval of the Chebyshev polynomial is -1 while for even S, the left extremum takes value $+1$.

While these findings are for the shifted Chebyshev polynomials admittedly trivial, they generalize neatly to the known first and second order accurate optimal stability polynomials of circular/disk-like spectra, as shown in Sect. 3.1.

Appendix B. Simulation Configurations with Strictly Convex Spectra

1. Burgers' Equation $\partial_t u + \partial_x \frac{1}{2}u^2 = 0$ is discretized using the DGSEM on $\Omega=[0,1]$ and Godunov flux [90]. The periodic domain is discretized with 64 elements on which the solution is reconstructed using third order local polynomials. As initial condition we supply $u(t_0,x) = 2+\sin(2\pi x)$ leading to a multi-valued solution at $t=\frac{1}{2\pi}$. The spectrum is computed using the algorithmic differentiation capabilities of `Trixi.jl` and is displayed in Fig. 2a.
2. The Shallow Water equations with variable bottom topography $b(x)$ read in 1D

$$\partial_t\begin{pmatrix}h\\hv\end{pmatrix} + \partial_x\begin{pmatrix}hv\\hv^2+\frac{g}{2}h^2\end{pmatrix} + \begin{pmatrix}0\\gh\partial_x b\end{pmatrix} = \mathbf{0}. \tag{B.1}$$

We discretize (B.1) on $\Omega=[0,\sqrt{2}]$ using the DGSEM with flux differencing [94, 95]. In particular, for the first component of (B.1) the surface flux is approximated using the

Rusanov/Local Lax-Friedrichs flux and the second component via the flux proposed in [96]. For the volume term we used the fluxes presented in [97]. Ω is discretized with 32 cells where we use third order local polynomials to reconstruct the solution. The initial condition is in this case a moderate discontinuity:

$$\begin{pmatrix} h(t_0,x) \\ v(t_0,x) \end{pmatrix} = \begin{pmatrix} -b(x) + \begin{cases} 3.25 & |x-0.7| > 0.5 \\ 4 & \text{else} \end{cases} \\ \begin{cases} 0 & |x-0.7| > 0.5 \\ 0.1882 & \text{else} \end{cases} \end{pmatrix} \tag{B.2}$$

with sinusoidal bottom topography $b(x) = \sin(x)$. The corresponding spectrum is displayed in Fig. 2b.

3. The ideal compressible MHD equations considered here are of form

$$\partial_t \begin{pmatrix} \rho \\ \rho v_1 \\ \rho v_2 \\ \rho v_3 \\ \rho e \\ B_1 \\ B_2 \\ B_3 \end{pmatrix} + \partial_x \begin{pmatrix} \rho v_1 \\ \rho v_1^2 + p + e_B - B_1^2 \\ \rho v_1 v_2 - B_1 B_2 \\ \rho v_1 v_3 - B_1 B_3 \\ v_1(e_v + \gamma(\rho e - e_v - e_B) + 2e_B) - B_1(v_1 B_1 + v_2 B_2 + v_3 B_3) \\ 0 \\ v_1 B_2 - v_2 B_1 \\ v_1 B_3 - v_3 B_1 \end{pmatrix} = \mathbf{0} \tag{B.3}$$

with $e_B := 0.5(B_1^2 + B_2^2 + B_3^2)$, $e_v := \rho(v_1^2 + v_2^2 + v_3^2)$, $p = (\gamma - 1)(\rho e - e_v - e_B)$. The solution is represented with $p = 1$ local polynomials and we employ the Rusanov/Local Lax-Friedrichs flux [98]. The periodic domain $\Omega = [0, 1]$ is again discretized using 32 cells. For the initial condition we choose the Alfvén wave [99]

$$\begin{pmatrix} \rho(t_0,x) \\ v_1(t_0,x) \\ v_2(t_0,x) \\ v_3(t_0,x) \\ p(t_0,x) \\ B_1(t_0,x) \\ B_2(t_0,x) \\ B_3(t_0,x) \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0.1\sin(2\pi x) \\ 0.1\cos(2\pi x) \\ 0.1 \\ 1 \\ 0.1\sin(2\pi x) \\ 0.1\cos(2\pi x) \end{pmatrix} \tag{B.4}$$

and set $\gamma = \frac{5}{3}$. The spectrum for this particular configuration is shown in Fig. 2c.

4. The compressible Euler equations in two spatial dimensions are given by

$$\partial_t \begin{pmatrix} \rho \\ \rho v_1 \\ \rho v_2 \\ \rho e \end{pmatrix} + \partial_x \begin{pmatrix} \rho v_1 \\ \rho v_1^2 + p \\ \rho v_1 v_2 \\ (\rho e + p) v_1 \end{pmatrix} + \partial_y \begin{pmatrix} \rho v_2 \\ \rho v_1 v_2 \\ \rho v_2^2 + p \\ (\rho e + p) v_2 \end{pmatrix} = \mathbf{0} \tag{B.5}$$

which are represented on $\Omega = [-1, 1]^2$ with 8 cells per direction using second order local polynomials. Here, $p = (\gamma - 1)(\rho e - 0.5\rho(v_1^2 + v_2^2))$ and $\gamma = 1.4$. For the flux we use the Harten-Lax-Van Leer Contact (HLLC) flux [93] and supply again periodic boundaries.

The fields are initialized as

$$\begin{pmatrix} \rho(t_0, x) \\ v_1(t_0, x) \\ v_2(t_0, x) \\ p(t_0, x) \end{pmatrix} = \begin{pmatrix} 2 + 0.1 \sin\left(\pi(x + y)\right) \\ 1 \\ 1 \\ \left[2 + 0.1 \sin\left(\pi(x + y)\right)\right]^2 \end{pmatrix}. \tag{B.6}$$

The spectrum corresponding to this initial state and semidiscretization is displayed in Fig. 2d.

References

1. Courant, R., Friedrichs, K., Lewy, H.: Über die partiellen Differenzengleichungen der mathematischen Physik. Mathematische Annalen **100**, 32–74 (1928). https://doi.org/10.1007/BF01448839
2. Franklin, J.: Numerical stability in digital and analog computation for diffusion problems. J. Math. Phys. **37**, 305–315 (1958)
3. Guillou, A., Lago, B.: Domaine de stabilité associé aux formules d'intégration numérique d'équations différentielles, a pas séparés et a pas liés. recherche de formules a grand rayon de stabilité, Ier Congr. Ass. Fran. Calcul., AFCAL, pp. 43–56 (1960)
4. Saul'ev, V.: Integration of parabolic equations by the grid method. Fizmatgiz Moscow **13**, 14–19 (1960)
5. Forsythe, G.E., Wasow, W.R.: Finite-Difference Methods for Partial Differential Equations. Applied Mathematics Series, Wiley, New York (1960)
6. Wanner, G., Hairer, E.: Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems, vol. 375. Springer, Berlin (1996). https://doi.org/10.1007/978-3-642-05221-7
7. Hundsdorfer, W.H., Verwer, J.G.: Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations. Springer Series in Computational Mathematics, vol. 375, 1st edn. Springer, Berlin (2010)
8. Verwer, J.G.: Explicit Runge–Kutta methods for parabolic partial differential equations. Appl. Numer. Math. **22**, 359–379 (1996). https://doi.org/10.1016/S0168-9274(96)00022-0
9. Abdulle, A.: Explicit stabilized Runge–Kutta methods, Technical Report. Mathematics Institute of Computational Science and Engineering, School of Basic Sciences, Section of Mathematics EPFL Lausanne (2011)
10. Van der Houwen, P.: The development of Runge–Kutta methods for partial differential equations. Appl. Numer. Math. **20**, 261–272 (1996). https://doi.org/10.1016/0168-9274(95)00109-3
11. Chzao-Din, Y.: Some difference schemes for the solution of the first boundary value problem for linear differential equations with partial derivatives. Moscow Stage University, Thesis (1958)
12. Burrage, K.: Order and stability properties of explicit multivalue methods. Appl. Numer. Math. **1**, 363–379 (1985). https://doi.org/10.1016/0168-9274(85)90001-7
13. Lomax, H.: On the Construction of Highly Stable, Explicit, Numerical Methods for Integrating Coupled Ordinary Differential Equations with Parasitic Eigenvalues. National Aeronautics and Space Administration (1968)
14. Lebedev, V.: A new method for determining the roots of polynomials of least deviation on a segment with weight and subject to additional conditions. Part I. Russ. J. Numer. Anal. Math. Model. **8**, 195–222 (1993a). https://doi.org/10.1515/rnam.1993.8.3.195
15. Lebedev, V.: A new method for determining the roots of polynomials of least deviation on a segment with weight and subject to additional conditions. Part II, Russ. J. Numer. Anal. Math. Model. **8**, 397–426 (1993b). https://doi.org/10.1515/rnam.1993.8.5.397
16. Van der Houwen, P.J.: Construction of integration formulas for initial value problems. Technical Report, Stichting Mathematisch Centrum, Amsterdam (1977)
17. Van der Houwen, P., Kok, J.: Numerical Solution of a Minimax Problem. Stichting Mathematisch Centrum, Toegepaste Wiskunde (1971)
18. Abdulle, A., Medovikov, A.A.: Second order Chebyshev methods based on orthogonal polynomials. Numer. Math. **90**, 1–18 (2001). https://doi.org/10.1007/s002110100292
19. Riha, W.: Optimal stability polynomials. Computing **9**, 37–43 (1972). https://doi.org/10.1007/BF02236374
20. Van der Houwen, P.: Explicit Runge–Kutta formulas with increased stability boundaries. Numer. Math. **20**, 149–164 (1972). https://doi.org/10.1007/BF01404404
21. Kinnmark, I.P., Gray, W.G.: One step integration methods with maximum stability regions. Math. Comput. Simul. **26**, 87–92 (1984). https://doi.org/10.1016/0378-4754(84)90039-9

22. Sonneveld, P., Van Leer, B.: A minimax problem along the imaginary axis, Technical Report 4. Technische Hogeschool Delft, Onderafdeling der Wiskunde en Informatica (1984)
23. Kinnmark, I.P., Gray, W.G.: One step integration methods of third-fourth order accuracy with large hyperbolic stability limits. Math. Comput. Simul. **26**, 181–188 (1984). https://doi.org/10.1016/0378-4754(84)90056-9
24. Jeltsch, R., Nevanlinna, O.: Largest disk of stability of explicit Runge–Kutta methods. BIT Numer. Math. **18**, 500–502 (1978)
25. Owren, B., Seip, K.: Some stability results for explicit Runge–Kutta methods. BIT Numer. Math. **30**, 700–706 (1990). https://doi.org/10.1007/BF01933217
26. Vichnevetsky, R.: New stability theorems concerning one-step numerical methods for ordinary differential equations. Math. Comput. Simul. **25**, 199–205 (1983). https://doi.org/10.1016/0378-4754(83)90092-7
27. Niegemann, J., Diehl, R., Busch, K.: Efficient low-storage Runge–Kutta schemes with optimized stability regions. J. Comput. Phys. **231**, 364–372 (2012). https://doi.org/10.1016/j.jcp.2011.09.003
28. Torrilhon, M., Jeltsch, R.: Essentially optimal explicit Runge–Kutta methods with application to hyperbolic-parabolic equations. Numer. Math. **106**, 303–334 (2007). https://doi.org/10.1007/s00211-006-0059-5
29. Kennedy, C.A., Carpenter, M.H., Lewis, R.: Low-storage, explicit Runge–Kutta schemes for the compressible Navier–Stokes equations. Appl. Numer. Math. **35**, 177–219 (2000). https://doi.org/10.1016/S0168-9274(99)00141-5
30. Allampalli, V., Hixon, R., Nallasamy, M., Sawyer, S.D.: High-accuracy large-step explicit Runge–Kutta (HALE-RK) schemes for computational aeroacoustics. J. Comput. Phys. **228**, 3837–3850 (2009). https://doi.org/10.1016/j.jcp.2009.02.015
31. Toulorge, T., Desmet, W.: Optimal Runge–Kutta schemes for discontinuous Galerkin space discretizations applied to wave propagation problems. J. Comput. Phys. **231**, 2067–2091 (2012). https://doi.org/10.1016/j.jcp.2011.11.024
32. Mead, J., Renaut, R.: Optimal Runge–Kutta methods for first order pseudospectral operators. J. Comput. Phys. **152**, 404–419 (1999). https://doi.org/10.1006/jcph.1999.6260
33. Al Jahdali, R., Boukharfane, R, Dalcin, L., Parsani, M.: Optimized explicit Runge–Kutta schemes for entropy stable discontinuous collocated methods applied to the Euler and Navier–Stokes equations. In: AIAA Scitech 2021 Forum, 2021, p. 0633. https://doi.org/10.2514/6.2021-0633
34. Kubatko, E.J., Yeager, B.A., Ketcheson, D.I.: Optimal strong-stability-preserving Runge–Kutta time discretizations for discontinuous Galerkin methods. J. Sci. Comput. **60**, 313–344 (2014). https://doi.org/10.1007/s10915-013-9796-7
35. Nasab, S.H., Cagnone, J.-S., Vermeire, B.C.: Optimal explicit Runge–Kutta time stepping for density-based finite-volume solver. In: AIAA SCITECH 2022 Forum, 2022, p. 1049. https://doi.org/10.2514/6.2022-1049
36. Ketcheson, D., Ahmadia, A.: Optimal stability polynomials for numerical integration of initial value problems. Commun. Appl. Math. Comput. Sci. **7**, 247–271 (2013). https://doi.org/10.2140/camcos.2012.7.247
37. Vermeire, B., Loppi, N., Vincent, P.: Optimal Runge–Kutta schemes for pseudo time-stepping with high-order unstructured methods. J. Comput. Phys. **383**, 55–71 (2019). https://doi.org/10.1016/j.jcp.2019.01.003
38. Schlottke-Lakemper, M., Winters, A.R., Ranocha, H., Gassner, G.J.: A purely hyperbolic discontinuous Galerkin approach for self-gravitating gas dynamics. J. Comput. Phys. **442**, 110467 (2021). https://doi.org/10.1016/j.jcp.2021.110467
39. Vermeire, B.C.: Paired explicit Runge–Kutta schemes for stiff systems of equations. J. Comput. Phys. **393**, 465–483 (2019). https://doi.org/10.1016/j.jcp.2019.05.014
40. Hedayati Nasab, S., Vermeire, B.C.: Third-order paired explicit Runge–Kutta schemes for stiff systems of equations. J. Comput. Phys. **468**, 111470 (2022). https://doi.org/10.1016/j.jcp.2022.111470
41. Vermeire, B.C., Hedayati Nasab, S.: Accelerated implicit-explicit Runge–Kutta schemes for locally stiff systems. J. Comput. Phys. **429**, 110022 (2021). https://doi.org/10.1016/j.jcp.2020.110022
42. Hedayati Nasab, S., Pereira, C.A., Vermeire, B.C.: Optimal Runge–Kutta stability polynomials for multidimensional high-order methods. J. Sci. Comput. **89**, 11 (2021). https://doi.org/10.1007/s10915-021-01620-x
43. Klinge, M., Weiner, R.: Strong stability preserving explicit peer methods for discontinuous Galerkin discretizations. J. Sci. Comput. **75**, 1057–1078 (2018). https://doi.org/10.1007/s10915-017-0573-x
44. Ellison, A.C., Fornberg, B.: A parallel-in-time approach for wave-type PDEs. Numer. Math. **148**, 79–98 (2021). https://doi.org/10.1007/s00211-021-01197-5

45. Vermeire, B.C.: Embedded paired explicit Runge–Kutta schemes. J. Comput. Phys. **487**, 112159 (2023). https://doi.org/10.1016/j.jcp.2023.112159. (https://www.sciencedirect.com/science/article/pii/S0021999123002541.)
46. Hairer, E., Wanner, G., Nørsett, S.P.: Solving Ordinary Differential Equations I: Nonstiff Problems, 2nd edn. Springer, Berlin (1993). https://doi.org/10.1007/978-3-540-78862-1
47. Hundsdorfer, W., Mozartova, A., Savcenco, V.M.: Vonotonicity Conditions for Multirate and Partitioned Explicit Runge–Kutta Schemes, pp. 177–195. Springer, Berlin (2013). https://doi.org/10.1007/978-3-642-33221-0_11
48. Hu, F., Hussaini, M.Y., Manthey, J.: Low-dissipation and low-dispersion Runge–Kutta schemes for computational acoustics. J. Comput. Phys. **124**, 177–191 (1996). https://doi.org/10.1006/jcph.1996.0052
49. Berland, J., Bogey, C., Bailly, C.: Low-dissipation and low-dispersion fourth-order Runge–Kutta algorithm. Comput. Fluids **35**, 1459–1463 (2006). https://doi.org/10.1016/j.compfluid.2005.04.003
50. Bernardini, M., Pirozzoli, S.: A general strategy for the optimization of Runge–Kutta schemes for wave propagation phenomena. J. Comput. Phys. **228**, 4182–4199 (2009). https://doi.org/10.1016/j.jcp.2009.02.032
51. Spiteri, R.J., Ruuth, S.J.: A new class of optimal high-order strong-stability-preserving time discretization methods. SIAM J. Numer. Anal. **40**, 469–491 (2002). https://doi.org/10.1137/S0036142901389025
52. Ruuth, S.: Global optimization of explicit strong-stability-preserving Runge–Kutta methods. Math. Comput. **75**, 183–207 (2006). https://doi.org/10.1090/S0025-5718-05-01772-2
53. Sturm, J.F.: Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. Optim. Methods Softw. **11**, 625–653 (1999). https://doi.org/10.1080/10556789908805766
54. Domahidi, A., Chu, E., Boyd, S.: ECOS: an SOCP solver for embedded systems. In: 2013 European Control Conference (ECC), pp. 3071–3076. https://doi.org/10.23919/ECC.2013.6669541
55. Trefethen, L.N.: Approximation Theory and Approximation Practice, Other Titles in Applied Mathematics, Extended SIAM, London (2019). https://doi.org/10.1137/1.9781611975949
56. Verwer, J.G., Hundsdorfer, W.H., Sommeijer, B.P.: Convergence properties of the Runge–Kutta–Chebyshev method. Numer. Math. **57**, 157–178 (1990). https://doi.org/10.1007/BF01386405
57. van Der Houwen, P.J., Sommeijer, B.P.: On the internal stability of explicit, m-stage Runge–Kutta methods for large m-values. ZAMM J. Appl. Math. Mech. **60**, 479–485 (1980). https://doi.org/10.1002/zamm.19800601005
58. Verwer, J.G., Sommeijer, B.P., Hundsdorfer, W.: RKC time-stepping for advection–diffusion–reaction problems. J. Comput. Phys. **201**, 61–79 (2004). https://doi.org/10.1016/j.jcp.2004.05.002
59. Shu, C.-W., Osher, S.: Efficient implementation of essentially non-oscillatory shock-capturing schemes. J. Comput. Phys. **77**, 439–471 (1988). https://doi.org/10.1016/0021-9991(88)90177-5
60. Gottlieb, S., Shu, C.-W., Tadmor, E.: Strong stability-preserving high-order time discretization methods. SIAM Rev. **43**, 89–112 (2001). https://doi.org/10.1137/S003614450036757X
61. Gottlieb, S., Ketcheson, D., Shu, C.-W.: Strong Stability Preserving Runge–Kutta and Multistep Time Discretizations. World Scientific, Singapore (2011). https://doi.org/10.1142/7498
62. Godunov, S.K., Bohachevsky, I.: Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics. Matematičeskij sbornik **47**(89), 271–306 (1959)
63. Kopriva, D.A., Gassner, G.: On the quadrature and weak form choices in collocation type discontinuous Galerkin spectral element methods. J. Sci. Comput. **44**, 136–155 (2010). https://doi.org/10.1007/s10915-010-9372-3
64. Kopriva, D.A.: Implementing Spectral Methods for Partial Differential Equations: Algorithms for Scientists and Engineers, Scientific Computation (SCIENTCOMP), 1st edn. Springer, Berlin (2009). https://doi.org/10.1007/978-90-481-2261-5
65. Ranocha, H., Schlottke-Lakemper, M., Winters, A.R., Faulhaber, E., Chan, J., Gassner, G.: Adaptive numerical simulations with Trixijl: a case study of Julia for scientific computing. Proc. JuliaCon. Conf. **1**, 77 (2022). https://doi.org/10.21105/jcon.00077
66. Schlottke-Lakemper, M., Winters, A.R., Ranocha, H., Gassner, G.: A purely hyperbolic discontinuous Galerkin approach for self-gravitating gas dynamics. J. Comput. Phys. **442**, 110467 (2021). https://doi.org/10.1016/j.jcp.2021.110467
67. Schlottke-Lakemper, M., Gassner, G., Ranocha, H., Winters, A.R., Chan, J.: Trixi.jl: adaptive high-order numerical simulations of hyperbolic PDEs in Julia (2021b). https://github.com/trixi-framework/Trixi.jl. https://doi.org/10.5281/zenodo.3996439
68. Kinnmark, I.P.: A principle for construction of one-step integration methods with maximum imaginary stability limits. Math. Comput. Simul. **29**, 87–106 (1987). https://doi.org/10.1016/0378-4754(87)90100-5

69. De Marchi, S., Marchetti, F., Perracchione, E., Poggiali, D.: Polynomial interpolation via mapped bases without resampling. J. Comput. Appl. Math. **364**, 112347 (2020). https://doi.org/10.1016/j.cam.2019.112347
70. Wächter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math. Program. **106**, 25–57 (2006). https://doi.org/10.1007/s10107-004-0559-y
71. Amestoy, P., Duff, I.S., Koster, J., L'Excellent, J.-Y.: A fully asynchronous multifrontal solver using distributed dynamic scheduling. SIAM J. Matrix Anal. Appl. **23**, 15–41 (2001). https://doi.org/10.1137/S0895479899358194
72. Amestoy, P., Buttari, A., L'Excellent, J.-Y., Mary, T.: Performance and scalability of the block low-rank multifrontal factorization on multicore architectures. ACM Trans. Math. Softw. **45**, 2:1-2:26 (2019). https://doi.org/10.1145/3242094
73. Karypis, G., Kumar, V.: A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM J. Sci. Comput. **20**, 359–392 (1998). https://doi.org/10.1137/S1064827595287997
74. Naumann, U., Leppkes, K., Lotz, J.: dco/C++ user guide. Technical Report, RWTH Aachen, Department of Computer Science (2014)
75. Doehring, D., Vigerske, S.: Tell ipopt it is a feasibility problem #597, GitHub issue (2022). https://github.com/coin-or/Ipopt/issues/597
76. Liu, D.C., Nocedal, J.: On the limited memory BFGS method for large scale optimization. Math. Program. **45**, 503–528 (1989). https://doi.org/10.1007/BF01589116
77. Doehring, D.: OSPREI: optimal stability polynomials in roots for explicit time integration (2023). https://github.com/DanielDoehring/OSPREI, https://doi.org/10.5281/zenodo.8009493
78. Edelsbrunner, H., Kirkpatrick, D., Seidel, R.: On the shape of a set of points in the plane. IEEE Trans. Inf. Theory **29**, 551–559 (1983). https://doi.org/10.1109/TIT.1983.1056714
79. Edelsbrunner, H.: Alpha shapes—a survey, Technical Report 1, IST Austria (Institute of Science and Technology Austria) (2011)
80. Wang, Z.J., Fidkowski, K., Abgrall, R., Bassi, F., Caraeni, D., Cary, A., Deconinck, H., Hartmann, R., Hillewaert, K., Huynh, H.T., et al.: High-order CFD methods: current status and perspective. Int. J. Numer. Methods Fluids **72**, 811–845 (2013). https://doi.org/10.1002/fld.3767
81. George, J.D. , Jung, S.Y., Mangan, N.M.: Walking into the complex plane to 'order' better time integrators (2021). arXiv preprint arXiv:2110.04402. https://doi.org/10.48550/arXiv.2110.04402
82. Ketcheson, D.I., Loczi, L., Parsani, M.: Internal error propagation in explicit Runge–Kutta methods. J. Numer. Anal. **52**, 2227–2249 (2014). https://doi.org/10.1137/130936245
83. O'Sullivan, S.: Factorized Runge–Kutta–Chebyshev methods. J. Phys. Conf. Ser. **837**, 012020 (2017). https://doi.org/10.1088/1742-6596/837/1/012020
84. Ferracina, L., Spijker, M.: An extension and analysis of the Shu–Osher representation of Runge–Kutta methods. Math. Comput. **74**, 201–219 (2005). https://doi.org/10.1090/S0025-5718-04-01664-3
85. Lebedev, V.: Explicit difference schemes with time-variable steps for solving stiff systems of equations. Soviet J. Numer. Anal. Math. Modell. **4**, 111–135 (1989). https://doi.org/10.1515/rnam.1989.4.2.111
86. Lebedev, V.: How to Solve Stiff Systems of Differential Equations by Explicit Methods, 1st edn. CRC Press, Boca Raton (1994)
87. Salari, K., Knupp, P.: Code verification by the method of manufactured solutions, Technical Report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States) (2000). https://doi.org/10.2172/759450
88. Roache, P.J.: Code verification by the method of manufactured solutions. J. Fluids Eng. **124**, 4–10 (2002). https://doi.org/10.1115/1.1436090
89. Harten, A., Lax, P.D., Leer, B.V.: On upstream differencing and Godunov-type schemes for hyperbolic conservation laws. SIAM Rev. **25**, 35–61 (1983)
90. Osher, S.: Riemann solvers, the entropy condition, and difference. SIAM J. Numer. Anal. **21**, 217–235 (1984)
91. Ketcheson, D.I.: Highly efficient strong stability-preserving Runge–Kutta methods with low-storage implementations. SIAM J. Sci. Comput. **30**, 2113–2136 (2008). https://doi.org/10.1137/07070485X
92. Shu, C.W.: Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. In: Quarteroni, A. (ed.) Advanced Numerical Approximation of Nonlinear Hyperbolic Equations. Lecture Notes in Mathematics, pp. 325–432. Springer, Berlin (1998). https://doi.org/10.1007/BFb0096351
93. Toro, E.F., Spruce, M., Speares, W.: Restoration of the contact surface in the HLL-Riemann solver. Shock Waves **4**, 25–34 (1994). https://doi.org/10.1007/BF01414629
94. LeFloch, P.G., Mercier, J.M., Rohde, C.: Fully discrete, entropy conservative schemes of arbitraryorder. SIAM J. Numer. Anal. **40**, 1968–1992 (2002). https://doi.org/10.1137/S003614290240069X

95. Chen, T., Shu, C.-W.: Entropy stable high order discontinuous Galerkin methods with suitable quadrature rules for hyperbolic conservation laws. J. Comput. Phys. **345**, 427–461 (2017). https://doi.org/10.1016/j.jcp.2017.05.025
96. Fjordholm, U.S., Mishra, S., Tadmor, E.: Well-balanced and energy stable schemes for the shallow water equations with discontinuous topography. J. Comput. Phys. **230**, 5587–5609 (2011). https://doi.org/10.1016/j.jcp.2011.03.042
97. Wintermeyer, N., Winters, A.R., Gassner, G.J., Kopriva, D.A.: An entropy stable nodal discontinuous Galerkin method for the two dimensional shallow water equations on unstructured curvilinear meshes with discontinuous bathymetry. J. Comput. Phys. **340**, 200–242 (2017). https://doi.org/10.1016/j.jcp.2017.03.036
98. Rusanov, V.: The calculation of the interaction of non-stationary shock waves and obstacles. USSR Comput. Math. Math. Phys. **1**, 304–320 (1962). https://doi.org/10.1016/0041-5553(62)90062-9
99. Tóth, G.: The div(b)=0 constraint in shock-capturing magnetohydrodynamics codes. J. Comput. Phys. **161**, 605–652 (2000). https://doi.org/10.1006/jcph.2000.6519