



Multirate time-integration based on dynamic ODE partitioning through adaptively refined meshes for compressible fluid dynamics

Daniel Doehring^{a,*}, Michael Schlottke-Lakemper^{a,b,c}, Gregor J. Gassner^d, Manuel Torrilhon^a

^a Applied and Computational Mathematics, RWTH Aachen University, Schinkelstrasse 2, Aachen, 52062, North Rhine-Westphalia, Germany

^b High-Performance Computing Center Stuttgart (HLRS), University of Stuttgart, Germany

^c High-Performance Scientific Computing, Center for Advanced Analytics and Predictive Sciences, University of Augsburg, Germany

^d Department of Mathematics and Computer Science, Center for Data and Simulation Science, University of Cologne, Germany

ARTICLE INFO

MSC:

65L06

65M20

76Mxx

76Nxx

Keywords:

Multirate time-integration

Runge-Kutta methods

Adaptive-mesh-refinement

ABSTRACT

In this paper, we apply the Paired-Explicit Runge-Kutta (P-ERK) schemes by Vermeire et al. [1,2] to dynamically partitioned systems arising from adaptive mesh refinement. The P-ERK schemes enable multirate time-integration with no changes in the spatial discretization methodology, making them readily implementable in existing codes that employ a method-of-lines approach.

We show that speedup compared to a range of state of the art Runge-Kutta methods can be realized, despite additional overhead due to the dynamic re-assignment of flagging variables and restricting nonlinear stability properties. The effectiveness of the approach is demonstrated for a range of simulation setups for viscous and inviscid convection-dominated compressible flows for which we provide a reproducibility repository.

In addition, we perform a thorough investigation of the nonlinear stability properties of the Paired-Explicit Runge-Kutta schemes regarding limitations due to the violation of monotonicity properties of the underlying spatial discretization. Furthermore, we present a novel approach for estimating the relevant eigenvalues of large Jacobians required for the optimization of stability polynomials.

1. Introduction

Unsteady convection-dominated flows are omnipresent in nature and engineering. Examples thereof include turbulent Navier-Stokes equations, magnetohydrodynamics, Euler equations of gas dynamics, and aeroacoustics besides many others. Despite describing notably different physical phenomena, these partial differential equations (PDEs) arise from conservation laws describing the balance of mass, momentum, and energy [3,4]. This allows for a unified treatment of these equations in the form of hyperbolic-parabolic balance laws. When written in conserved variables \mathbf{u} , the governing PDEs considered in this work are of the form

$$\partial_t \mathbf{u}(t, \mathbf{x}) + \nabla \cdot \mathbf{f}(\mathbf{u}(t, \mathbf{x}), \nabla \mathbf{u}(t, \mathbf{x})) = \mathbf{0}, \quad (1.1)$$

* Corresponding author.

E-mail address: doehring@acom.rwth-aachen.de (D. Doehring).

with flux function f , which may contain viscous terms function of ∇u .

A common feature of the aforementioned PDEs is that over the course of a numerical simulation, solution structures of significantly different spatial scales can be present simultaneously. Prominent examples thereof are hydrodynamic instabilities, shock waves and turbulent eddies. To resolve these structures efficiently it is for practical simulations desirable to use adaptive mesh refinement (AMR) [5] where the computational grid is refined precisely where a high resolution is required to resolve the relevant physical phenomena.

Given the inherently different nature of time (directed, one dimension) and space (non-directed, multiple dimensions) the temporal and spatial derivatives in (1.1) are conventionally treated separately. This approach is commonly referred to as the method of lines (MoL) [6], where the application of spatial discretization techniques such as Finite Differences/Volumes/Elements or Discontinuous Galerkin (DG) leads to a typically large system of ordinary differential equations (ODEs)

$$U(t_0) = U_0 \quad (1.2a)$$

$$U'(t) = F(t, U(t)). \quad (1.2b)$$

A benefit of the MoL approach is that the PDE can be solved in a modular fashion, i.e., the spatial discretization and the time-integration can be performed independently. Besides the ease of implementation, the MoL approach benefits from the availability of a large body of literature on both the spatial discretization techniques and time-integration methods. This enables an independent analysis of the building blocks of the overall fully discrete system, see for instance [6] for the DG method.

When using explicit time integration methods to solve the ODE system (1.2), the maximum stable timestep is restricted by some form of Courant-Friedrichs-Lewy (CFL) condition [7] which depends on the employed spatial discretization technique, the PDEs which are solved, and the smallest grid size in the domain. Although implicit time-integration methods remedy this problem, they require a careful implementation to solve the potentially huge ODE system (1.2b) in each Runge-Kutta stage efficiently. Additionally, if a Newton-based method is used to solve the nonlinear system, the Jacobian of the right-hand-side (RHS) F needs to be computed, stored, and inverted in each stage. To make this process feasible in large-scale simulations, storage of the Jacobian in sparse or matrix-free format is a must, increasing the complexity of the implementation. Nevertheless, we remark that efficient fully implicit time-integration methods for the simulation of compressible flows are available, see for instance [8,9]. Additionally, implicit-explicit (IMEX) methods [10] may be a viable middle ground balancing benefits and drawbacks of both implicit and explicit methods.

The presence of locally refined cells results for explicit methods in a reduction of the overall timestep, causing in principle unnecessary effort in the non-refined parts of the domain, thereby reducing the computational efficiency. This is a classic problem which has been tackled already in the 1980s when techniques for stiff ODEs, i.e., ODE problems with vastly varying time-scales where studied [11,12]. Since then, plenty of multirate time-integration techniques have been proposed, see for instance [13–20] and references therein. A common feature of such techniques is that they are in general non-trivial to implement and often require changes in the spatial discretization methodology. In particular, for some methods intermediate values need to be stored [15], interpolation of fine to coarse data may be required [15,16] or nontrivial flux exchange routines need to be devised [17,20]. In addition, such hybrid schemes combining spatial and temporal discretization require analysis starting from scratch, with little transferability of results from the individual spatial and temporal discretization techniques [14,15].

Recently, a class of stabilized Partitioned Runge-Kutta methods (PRKMs) have been proposed which achieve multirate effects while being easy to implement - in particular, there are no changes in the spatial discretization required. Termed Paired-Explicit Runge-Kutta (P-ERK) [1], these schemes amount only to application of different Runge-Kutta methods to the semidiscretization (1.2) with no need for specialized inter/extrapolation or flux updating strategies. In fact, they can be implemented similar to the well-known IMEX schemes [10], which apply different Runge-Kutta methods for the different operators present in the numerical scheme. A schematic illustration of the application of the P-ERK schemes to an adaptively refined mesh is provided in Fig. 1.

Furthermore, as the P-ERK schemes can be cast into the framework of PRKMs, one can utilize results on order conditions [21], conservation of linear invariants [22] and nonlinear stability properties [23–26]. In [1,2] the P-ERK schemes have been applied to static meshes for the simulation of the compressible Navier-Stokes equations. In this work, we apply the P-ERK schemes to dynamically refined meshes based on AMR which requires a dynamic re-assignment of the unknowns u to the corresponding integrators. We investigate whether speedup compared to other Runge-Kutta methods tailored to wave propagation and DG methods can be realized, despite the additional overhead due to the re-assignment. Besides considering viscous flows, we also apply the P-ERK schemes to purely hyperbolic problems.

This paper is structured as follows: First, we introduce the class of PRKMs by revisiting central properties such as order of consistency, linear/absolute stability, internal consistency, conservation properties and criteria for nonlinear stability. Next, we briefly introduce the P-ERK methods devised by Vermeire with a special focus on showcasing issues related to nonlinear stability. As the P-ERK schemes can be constructed from optimized stability polynomials, we discuss the construction of the stability polynomials in Section 4 with an emphasis on the construction of the spectrum for which we propose an efficient algorithm. In Section 5 we briefly compare the P-ERK schemes to the composing standalone methods. Next, we turn our attention in Section 6 onto describing the implementation aspects of the partitioning and devising a measure for estimation of the effectiveness of the P-ERK schemes. Then, we present a range of classical testcases for both viscous and inviscid flows in Section 7 for which we provide a reproducibility repository [27]. Finally, Section 8 concludes the paper.

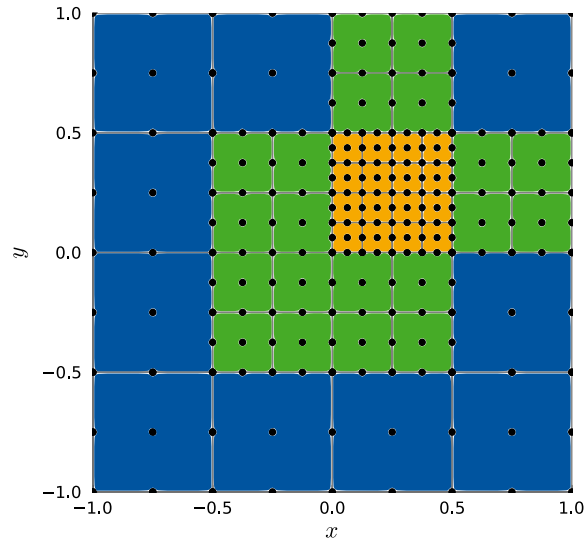


Fig. 1. Illustration of the assignment of grid cells (grey boundaries) to partitions $r = 1, 2, 3$ based on the minimum edge length h of the grid cells. Given a base number of stage-evaluations $E^{(1)}$ that is used to integrate the coarse blue cells, a $2E^{(1)} = E^{(2)}$ and a $4E^{(1)} = E^{(3)}$ stage-evaluation method for the green (medium) and orange (fine) cells, respectively, are required to perform the time-integration without reduction in CFL number (assuming constant wave speeds ρ_i across the domain). The Legendre-Gauss-Lobatto integration points of a second-order ($k = 2$) DG method are indicated by the black dots. Note that on each edge there are actually two points stacked on each other, one for each adjacent cell to allow for discontinuous solutions. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

2. Partitioned Runge-Kutta methods

In this work, we consider coefficient-based partitioned ODEs of the form

$$U(t_0) = U_0 \tag{2.1a}$$

$$U'(t) = \begin{pmatrix} U^{(1)}(t) \\ \vdots \\ U^{(R)}(t) \end{pmatrix}' = \begin{pmatrix} F^{(1)}(t, U^{(1)}(t), \dots, U^{(R)}(t)) \\ \vdots \\ F^{(R)}(t, U^{(1)}(t), \dots, U^{(R)}(t)) \end{pmatrix} = F(t, U(t)) \tag{2.1b}$$

where $R \in \mathbb{N}$ denotes the number of partitions which correspond to the levels of refinement in the context of quad/octree AMR, cf. Fig. 1. In the case of non-uniform meshes the partitioning of the semidiscretization (1.2) is based on the minimum edge length h of the grid cells. For the DG method employed in this work, the coefficients of the local solution polynomials are the unknowns of the ODE system (2.1) and can be uniquely assigned to a partition based on the minimum edge length h of the associated cell. In the following, the superscript $(\cdot)^{(r)}$ indicates quantities corresponding to the r 'th partition.

Systems of the form (2.1) motivate the usage of PRKMs, which are given in Butcher form by [28, Chapter II.15]

$$U_0 = U(t_0) \tag{2.2a}$$

$$K_i^{(r)} = F^{(r)} \left(t_n + c_i^{(r)} \Delta t, U_n + \Delta t \sum_{j=1}^S \sum_{k=1}^R a_{i,j}^{(k)} K_j^{(k)} \right), \quad r = 1, \dots, R, \quad i = 1, \dots, S \tag{2.2b}$$

$$U_{n+1} = U_n + \Delta t \sum_{i=1}^S \sum_{r=1}^R b_i^{(r)} K_i^{(r)}. \tag{2.2c}$$

Here, S denotes the number of stages K_i and $a_{i,j}^{(r)}$ form the Butcher array matrix $A^{(r)}$.

PRKMs have been introduced in the 1970s [29,30] and have received notable usage for the integration of Hamiltonian systems [31,32]. An essentially equivalent class of Runge-Kutta methods are Additive Runge-Kutta methods (ARKMs) [33,19,34], which have been developed in order to achieve efficient integration of multiscale and stiff systems [18,19]. Similar works have been performed with PRKMs [13,11,12]. Since the P-ERK methods we apply in this work are a special case of PRKMs [2], we briefly revisit conditions for convergence order, absolute stability, conservation of linear invariants, internal consistency, and nonlinear stability.

2.1. Order conditions

In addition to the classical conditions for convergence of order p , PRKMs need to satisfy additional conditions which have been derived by multiple authors, see [30,21,35] and for a compact summary [36]. For identical abscissae $c^{(r)} \equiv c$ and weights $b^{(r)} \equiv b$

(more on this restriction below), these in principle additional order constraints reduce for $p = 1, 2, 3$ conveniently to the classical ones:

$$p = 1 : \quad \mathbf{b}^T \mathbf{1} \stackrel{!}{=} 1 \tag{2.3a}$$

$$p = 2 : \quad \mathbf{b}^T \mathbf{c} \stackrel{!}{=} \frac{1}{2} \tag{2.3b}$$

$$p = 3 : \quad \mathbf{b}^T \mathbf{c}^2 \stackrel{!}{=} \frac{1}{3} \tag{2.3c}$$

$$\mathbf{b}^T A^{(r)} \mathbf{c} \stackrel{!}{=} \frac{1}{6} \quad \forall r = 1, \dots, R. \tag{2.3d}$$

Here, $\mathbf{1} \in \mathbb{R}^S$ denotes the vector of ones and the exponentiation of vectors is to be understood element-wise.

For order $p = 4$, however, even for identical abscissae c and weights \mathbf{b} non-trivial coupling conditions between the Butcher arrays $A^{(r)}$ of the different methods arise [36], rendering the construction of an optimized fourth-order P-ERK scheme significantly more difficult. In the original publication [1] second-order P-ERK schemes are proposed which have then be extended to third-order in a subsequent work [2].

2.2. Absolute stability

Absolute stability, or linear stability, examines the asymptotic behavior of a time integration method applied to the constant-coefficient linear test system [37, Chapter IV.2]

$$\mathbf{U}'(t) = \mathbf{J} \mathbf{U}(t). \tag{2.4}$$

In this work, \mathbf{J} corresponds to the Jacobian of the RHS $\mathbf{F}(t, \mathbf{U}(t))$ (1.2)

$$\mathbf{J}(t, \mathbf{U}) := \frac{\partial \mathbf{F}(t, \mathbf{U}(t))}{\partial \mathbf{U}}. \tag{2.5}$$

A time-integration method is called absolutely stable if its associated stability function $P(z)$ (which is a polynomial for explicit methods) is in magnitude less or equal to one when evaluated at the scaled eigenvalues with non-positive real part

$$|P(\Delta t \lambda)| \stackrel{!}{\leq} 1 \quad \forall \lambda \in \sigma(\mathbf{J}) : \text{Re}(\lambda) \leq 0. \tag{2.6}$$

In other words, we demand that the scaled spectrum lies in the region of absolute stability of the method with stability polynomial $P(z)$. Since the RHS \mathbf{F} depends on the solution \mathbf{U} itself, we require (2.6) to hold for all (unknown) states $\mathbf{U}(t)$ reached during the simulation. This is in principle difficult to ensure, but in practice the spectra $\sigma(\mathbf{J}(\mathbf{U}))$ are observed to be relatively similar to the spectrum of the initial condition $\sigma(\mathbf{J}(\mathbf{U}_0))$. By ‘similar’ we are referring to the characteristic distribution of the eigenvalues in the complex plane, which remains (up to scaling due to mesh refinement) robust over the course of a simulation.

The linear stability properties of PRKMs have been studied in [38] with special focus on separable Hamiltonian systems. For ARKMs, a linear stability analysis is performed in [34] based on the test problem $\mathbf{U}' = \sum_{r=1}^R \lambda^{(r)} \mathbf{U}$.

For PRKMs it is customary to investigate the test equation

$$\mathbf{U}' = \Lambda \mathbf{U}, \quad \Lambda \in \mathbb{C}^{N \times N} \tag{2.7}$$

which can be naturally partitioned according to a set of mask-matrices $I^{(r)} \in \{0, 1\}^{N \times N}$:

$$\Lambda = \sum_{r=1}^R \Lambda^{(r)} = \sum_{r=1}^R I^{(r)} \Lambda. \tag{2.8}$$

The mask-matrices fulfill the property $\sum_{r=1}^R I^{(r)} = I$, with I being the identity matrix of dimensions $N \times N$. The mask matrices $I^{(r)}$ are required to specify the exact partitioning of (2.1b):

$$\mathbf{U}(t) = \underbrace{\sum_{r=1}^R I^{(r)} \mathbf{U}(t)}_{=: \mathbf{U}^{(r)}(t)}, \quad \mathbf{F}(t, \mathbf{U}(t)) = \underbrace{\sum_{r=1}^R I^{(r)} \mathbf{F}(t, \mathbf{U}(t))}_{=: \mathbf{F}^{(r)}(t, \mathbf{U}(t))}. \tag{2.9}$$

The matrix-valued stability function $P(Z) \in \mathbb{C}^{N \times N}$ is with $Z^{(r)} := \Delta t \Lambda^{(r)}$ given by [22]

$$P(Z) = I + \left(\sum_{r=1}^R (\mathbf{b}^{(r)} \otimes I)^T (Z^{(r)} \otimes I) \right) \left(I_{N \cdot S} - \sum_{r=1}^R (A^{(r)} \otimes I) (Z^{(r)} \otimes I) \right)^{-1} (\mathbf{1} \otimes I), \tag{2.10}$$

where \otimes denotes the Kronecker product and $I_{N \cdot S}$ the identity matrix of dimension $N \cdot S$. As discussed in [38], it is difficult to infer information on the linear stability of the overall partitioned scheme from (2.10) based on the stability properties of the individual

schemes $(A^{(r)}, b^{(r)})$ for arbitrary mask matrices $I^{(r)}$. In contrast to classic Runge-Kutta methods, one can for PRKMs no longer solely infer the action of the time-integration scheme from the stability functions of the individual methods [38]. This issue returns in Section 3.2 when the nonlinear stability properties of a special class of PRKMs are discussed.

In practice, however, it is observed that absolute stability of each method $(A^{(r)}, b^{(r)})$ for its associated spectrum $\lambda^{(r)}$ suffices for overall linear stability, irrespective of the concrete realization of the mask matrices $I^{(r)}$ [1,2]. To check this, one can either compute $P(Z)$ via (2.10) or set up the fully discrete system directly, i.e., apply the Runge-Kutta method to (2.4). Then, linear stability follows from applying stability criteria for discrete-time linear time-invariant systems [39, Chapter 8.6]. This is illustrated for a concrete example in Section 3.2.2.

2.3. Internal consistency

To ensure that the approximations of the partitioned stages $K_i^{(r)}$ approximate indeed the same timesteps c_i across partitions r , we require internal consistency [22,25,34], i.e.,

$$\sum_{j=1}^{i-1} a_{i,j}^{(r)} \stackrel{!}{=} c_i \quad \forall i = 1, \dots, S \quad \forall r = 1, \dots, R, \quad (2.11)$$

which is equivalent to all R Runge-Kutta methods having stage order one [22]. It has been demonstrated in [25,22] that internal consistency is required to avoid spurious oscillations at the interfaces between partitions.

2.4. Conservation of linear invariants

Since the underlying PDEs from which the semidiscretization (1.2) is constructed correspond oftentimes to conservation laws like (1.1) prescribing the conservation of mass, momentum, and energy it is natural to require that the application of a PRKM preserves this property for conservative spatial discretizations such as finite volume and DG schemes. For equation based partitioning according to (2.1b) this is ensured if the weights b are identical across partitions [14,22], i.e.,

$$b_i^{(r_1)} = b_i^{(r_2)} = b_i, \quad \forall i = 1, \dots, S, \quad \forall r_1, r_2 = 1, \dots, R. \quad (2.12)$$

For flux-based partitioning this restriction may be relaxed, see [26] for a detailed discussion.

2.5. Nonlinear stability

Nonlinear stability includes (among stronger results for scalar equations) positivity preservation of physical quantities such as pressure and density and the suppression of spurious oscillations around discontinuities. Time integration methods guaranteeing these properties are termed strong stability preserving (SSP) and have been well-studied over the past years [40,41]. The nonlinear stability properties of partitioned/additive Runge-Kutta methods have been investigated in [23–26]. In the aforementioned works it was shown that for an overall SSP PRKM the individual methods have to be SSP and the timestep Δt is furthermore restricted by

$$\Delta t = \min_{r=1, \dots, R} \{c_r^{\text{SSP}}\} \Delta t_{\text{Forward Euler}}. \quad (2.13)$$

Here, c_r^{SSP} denotes the SSP coefficient [41, Chapter 3] (radius of absolute monotonicity [23,42]) of the r 'th method. Except for some special cases where the timestep is restricted due to absolute stability (see, e.g., [43]) (2.13) renders the application of SSP-capable PRKMs in many cases ineffective, as the method with the smallest SSP coefficient determines the overall admissible timestep which guarantees nonlinear stability.

Having discussed the properties of PRKMs, we now turn our attention to a special case thereof, namely the Paired-Explicit Runge-Kutta (P-ERK) methods proposed by Vermeire [1].

3. Paired-explicit Runge-Kutta (P-ERK) methods

Paired-Explicit Runge-Kutta (P-ERK) methods have been originally proposed in [1] and were extended to third-order in [2]. These schemes target systems which exhibit locally varying characteristic speeds, most prominently due to non-uniform grids. In the context of the DG schemes considered in this work, the local time scales β_i depend both on the spectral radii $\rho_i, i = 1, \dots, N_D$ of the directional Jacobians $J_i = \partial_u f_i, i = 1, \dots, N_D$ of the flux function (1.1) and the local grid size h_i :

$$\beta \sim \min_{i=1, \dots, N_D} \frac{h_i}{(k+1) \cdot \rho_i}. \quad (3.1)$$

Here, $N_D \in \{1, 2, 3\}$ denotes the number of spatial dimensions and k the local polynomial degree of the DG approximation. For a more detailed discussion on the particular form of the CFL constraint for DG methods the interested reader is referred to [6,44,45]. When applying an explicit time-integration method to the semidiscretization (1.2b) the maximum timestep Δt is restricted according to

$$\Delta t \leq \text{CFL} \cdot \beta \tag{3.2}$$

where $\text{CFL} \in \mathbb{R}_+$ depends on the employed time-integration method. As discussed in Section 4 we seek to maximize this factor by optimizing the domain of absolute stability for the different methods.

Note that the estimate (3.1) is only a valid bound for the maximum timestep Δt in the convection-dominated case, i.e., when the spectral radius ρ of the Jacobian of the fully discrete system (1.2b) $J_F(\mathbf{U})$ scales as

$$\rho \sim \frac{|a|}{h} \gg \frac{|d|}{h^2} \tag{3.3}$$

where h denotes the smallest characteristic cell size and a, d are suitable quantifiers for the influence of convection and diffusion, respectively.

For such problems, one can partition the semidiscretization (1.2) according to the local CFL stability constraint. For each partition r , an optimized stability polynomial $P_{p;E^{(r)}}^{(r)}(z)$ of order p and degree $E^{(r)}$ is constructed which is then used to derive a specific Runge-Kutta method $(A^{(r)}, \mathbf{b}^{(r)})$. In particular, methods with more RHS evaluations E and larger CFL_E are used in regions with smaller CFL number, while methods with smaller E and smaller region of absolute stability (smaller CFL_E) are used for the elements of larger size and higher CFL. This is illustrated in Fig. 1 where a three times refined grid using a quadtree refinement strategy is shown. This enables a more efficient time integration since the evaluations E of the right-hand-side of (1.2b) are performed only in the regions where they are actually required, and saved in other parts of the domain. Most important, efficiency is gained only in the coarse regions of the mesh, where cheaper integrators are used instead of the expensive one used in the finest parts. This idea is conceptually similar to local time-stepping methods which perform additional operations only in the fine regions of the mesh by reducing the timestep locally [14,15].

3.1. Introduction

To illustrate the concept of the P-ERK schemes, we present the Butcher Tableaus of the second-order accurate P-ERK method obtained from a four-stage and a six-stage method:

i	c	$A^{(1)}$				$A^{(2)}$							
1	0												
2	$1/10$	$1/10$				$1/10$							
3	$2/10$	$2/10$	0			$2/10 - a_{3,2}^{(2)}$	$a_{3,2}^{(2)}$						
4	$3/10$	$3/10$	0	0		$3/10 - a_{4,3}^{(2)}$	0	$a_{4,3}^{(2)}$					
5	$4/10$	$4/10 - a_{5,4}^{(1)}$	0	0	$a_{5,4}^{(1)}$	$4/10 - a_{5,4}^{(2)}$	0	0	$a_{5,4}^{(2)}$				
6	$5/10$	$5/10 - a_{6,5}^{(1)}$	0	0	0	$a_{6,5}^{(1)}$	$5/10 - a_{6,4}^{(2)}$	0	0	0	$a_{6,4}^{(2)}$		
	\mathbf{b}^T	0	0	0	0	0	1	0	0	0	0	0	1

The first method $A^{(1)}$ requires only the computation of stages $\mathbf{K}_1^{(1)}, \mathbf{K}_4^{(1)}, \mathbf{K}_5^{(1)}, \mathbf{K}_6^{(1)}$, i.e., four evaluations of $F^{(1)}$ while the second method $A^{(2)}$ requires computation of all stages $\mathbf{K}_i^{(2)}, i = 1, \dots, 6$, i.e., six evaluations of $F^{(2)}$. When looking at the first method in isolation, it becomes a reducible [37, Chapter IV.12] method, i.e., the stages $\mathbf{K}_2^{(1)}, \mathbf{K}_3^{(1)}$ do not influence the final solution $U_{n+1}^{(1)}$ and the Butcher tableau could be truncated to a four-stage method. In the context of PRKMs, however, the second and third stage are required for an internally consistent cf., (2.11), update of the intermediate state $U_n + \Delta t \sum_{j=1}^S \sum_{r=1}^R a_{i,j}^{(r)} \mathbf{K}_j^{(r)}$, see (2.2b).

This particular form of the Butcher tableaux (3.4) comes with three advantages: First, the P-ERK methods are low-storage, requiring only storage of two Runge-Kutta stages at a time. Second, the computational costs per stage are not increasing if higher stage methods are used, as always at most two previous stages are used for computing the next stage. Third, the sparse structure of both $A^{(r)}$ and $\mathbf{b}^{(r)}$ allows for a simplified computation of the coefficients from a certain stability polynomial. The P-ERK methods are clearly internally consistent and conservative, i.e., methods which satisfy (2.11) and (2.12), respectively. The restriction to shared timesteps c_i and weights b_i comes also with the advantage of significantly simplified order constraints [36], [28, Chapter II.15], cf. (2.3).

We remark that a different choices of abscissae c and weights b than presented in (3.4) following [1] are possible. One options is to use $c_S = 1$ and $b_1 = b_S = 1/2$ which in the $S = 2$ case boils down to Heun’s method. Another compelling option is given by $c_S = 2/3$ and $b_1 = 1/4, b_S = 3/4$ which are known in the $S = 2$ case as Ralston’s method which minimizes the truncation error [46]. The choice proposed in [1] corresponds for $S = 2$ to the explicit midpoint method.

The stability polynomials corresponding to (3.4) are given by

$$P_{2;4}^{(1)}(z) = 1 + z + \frac{1}{2}z^2 + \alpha_3^{(1)}z^3 + \alpha_4^{(1)}z^4 \tag{3.5a}$$

$$P_{2;6}^{(2)}(z) = 1 + z + \frac{1}{2}z^2 + \alpha_3^{(2)}z^3 + \alpha_4^{(2)}z^4 + \alpha_5^{(2)}z^5 + \alpha_6^{(2)}z^6. \tag{3.5b}$$

The first subscript of $P^{(r)}$ denotes the linear order of consistency p of the method, while the second subscript denotes the degree of the polynomial which is equal to the number of RHS evaluations $E^{(r)}$.

The free coefficients $\alpha^{(1)} \in \mathbb{R}^2, \alpha^{(2)} \in \mathbb{R}^4$ can be optimized for a certain spectrum $\sigma(J_F(U))$ of the Jacobian $J_F(U)$ (2.5). Using optimized stabilized Runge-Kutta methods further increases the efficiency of the scheme as the region of absolute stability can be closely adapted to the spectrum $\sigma(J_F(U))$ [47]. For convection-dominated problems, cf. (3.3), the spectra $\sigma(J_F(U))$ of the semidiscretization increases linearly with the smallest inverse grid size $1/h$. At the same time, the largest possible timestep $\Delta t_{p,E}$ (in the sense of absolute stability) of an optimized p 'th order, E degree stability polynomial scales also (asymptotically) linearly with the degree = number of stage evaluations E [48,49,47]. Consequently, the usage of an optimized $2E$ stage-evaluation Runge-Kutta method for the simulation of a convection-dominated PDE with smallest grid size $\frac{1}{2}h$ allows using the same timestep Δt as the simulation of the same problem on grid size h with an E stage-evaluation scheme.

In contrast, for diffusion-dominated PDEs, the spectral radius increases quadratically with the inverse of the grid size, cf. (3.3). Thus, for an efficient treatment of such problems the region of stability needs also to increase quadratically with the number of stages S , which is the case for Runge-Kutta-Chebyshev methods [50,51] targeting exactly such problems. These methods, however, do not extend without special treatment (see for instance [52,53]) significantly into the complex plane, thus bearing issues for the convection-induced complex eigenvalues. Consequently, we restrict ourselves in this work to convection-dominated problems for which the required number of stages increases only linearly while refining the mesh.

3.2. Nonlinear stability properties

The key element to the effectiveness of the P-ERK schemes is the fact that evaluations of the right-hand-side can be avoided in regions of the domain with accordingly increased characteristic time β . This, in turn, requires that the weight vector \mathbf{b} is relatively sparse in order to be able to omit as many stage computations as possible. In the context of the previous example (3.4), it is clear that the computation of $\mathbf{K}_2^{(1)}$ and $\mathbf{K}_3^{(1)}$ can only be avoided if $b_2 = b_3 = 0$.

The presence of zeros in the shared weight vector, however, renders the method non SSP as the requirement $\mathbf{b}^{(r)} > \mathbf{0}, r = 1, \dots, R$ [24,54,42] (the inequality is to be understood component-wise) is violated for the P-ERK schemes. In particular, this SSP requirement is always violated for the highest stage method with $E = S$, as for this scheme there are stage evaluations K_i with $b_i = 0$. This extends to the other (reducible) methods with $E < S$ as well, with only exception of the base method where the evaluated stages coincide with the non-zeros in the weight vector (which could be SSP as a standalone scheme). Consequently, the P-ERK schemes offer no guarantees in terms of nonlinear stability properties, and, in fact, we show in the remainder of this section that this leads to problems even in the simplest examples.

3.2.1. Linear advection with Godunov's scheme

To illustrate potential issues with nonlinear stability, we consider the canonical first-order Finite Volume Godunov scheme applied to the 1D advection equation with smooth initial data

$$u_t + u_x = 0, \tag{3.6a}$$

$$u_0(t_0 = 0, x) = 1 + \frac{1}{2} \sin(\pi x) \tag{3.6b}$$

on $\Omega = (-1, 1)$ equipped with periodic boundaries. For a potentially non-uniform spatial discretization the semidiscretization (1.2) reads for Upwind/Godunov flux

$$\mathbf{U}_0 = \mathbf{U}(t_0) \tag{3.7a}$$

$$\frac{d}{dt} \mathbf{U}(t) = \underbrace{\begin{pmatrix} -\frac{1}{\Delta x_1} & 0 & \dots & 0 & \frac{1}{\Delta x_1} \\ \frac{1}{\Delta x_2} & -\frac{1}{\Delta x_2} & 0 & \dots & 0 \\ & \ddots & \ddots & & \\ 0 & \dots & 0 & \frac{1}{\Delta x_N} & -\frac{1}{\Delta x_N} \end{pmatrix}}_{=:L} \mathbf{U}(t), \tag{3.7b}$$

where

$$U_i(t) := \frac{1}{\Delta x_i} \int_{x_{i-1/2}}^{x_{i+1/2}} u(t, x) dx, \quad i = 1, \dots, N \tag{3.8}$$

denote the cell-averages. Applying a two-level ($R = 2$) PRKM to (3.7b) leads to the fully discrete system

$$\mathbf{U}_{n+1} = \underbrace{\begin{pmatrix} D^{(1)}(\Delta t, L) \\ D^{(2)}(\Delta t, L) \end{pmatrix}}_{=:D} \mathbf{U}_n \tag{3.9}$$

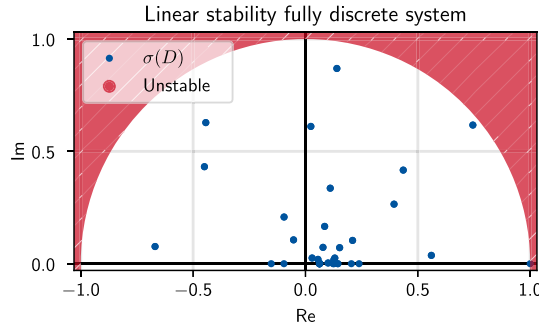


Fig. 2. Spectrum $\sigma(D)$ of the fully discrete matrix D of the P-ERK_{2;(8,16)} scheme applied to (3.7b). The unstable region with $|\lambda| > 1$ is shaded in red. Note that the purely real eigenvalue around (1, 0) is indeed stable with numeric value $1.0 - 28\epsilon$ and machine epsilon $\epsilon = \mathcal{O}(10^{-16})$.

which is equivalent to the matrix-valued stability function (2.10) with $R = 2$ and specified mask matrices $I^{(1)}, I^{(2)}$.

3.2.2. Linear stability of the fully discrete system

First, we consider an equidistant discretization with $N = 64, \Delta x = \Delta x^{(1)} = \Delta x^{(2)} = \Delta x/N$ where we use the optimized sixteen stage-evaluation method in the center of the domain, i.e., within $[-0.5, 0.5]$, and the optimized eight stage-evaluation method in the remainder of the domain, see Fig. 3a. For the Godunov scheme, the spectrum $\sigma(J(U))$ is circular with center at $-a/\Delta x$ and radius $\rho = a/\Delta x$. For this spectrum the optimal stability polynomial for second-order accurate methods is known analytically for all polynomial degrees [49]. The Butcher array defining values $a_{i,i-1}^{(r)}$ of the employed $E = \{8, 16\}$ stage methods are provided in compact form in (3.10). The other non-zero entry $a_{i,1}^{(r)}$ is determined by the internal consistency condition (2.11), i.e., $a_{i,1}^{(r)} = c_i - a_{i,i-1}^{(r)}$.

i	c	$a_{i,i-1}^{(1)}, E^{(1)} = 8$	$a_{i,i-1}^{(2)}, E^{(2)} = 16$
1	0	0	0
2	$1/30$	0	0
3	$2/30$	0	0.008333333333333333
4	$3/30$	0	0.013333333333333334
5	$4/30$	0	0.019047619047619042
6	$5/30$	0	0.025641025641025637
7	$6/30$	0	0.033333333333333354
8	$7/30$	0	0.042424242424242434
9	$8/30$	0	0.0533333333333333295
10	$9/30$	0	0.066666666666666667
11	$10/30$	0.019841269841269837	0.083333333333333337
12	$11/30$	0.04489795918367346	0.10476190476190472
13	$12/30$	0.07792207792207795	0.13333333333333336
14	$13/30$	0.12380952380952381	0.17333333333333337
15	$14/30$	0.19230769230769232	0.23333333333333323
16	$15/30$	0.3061224489795918	0.3333333333333334

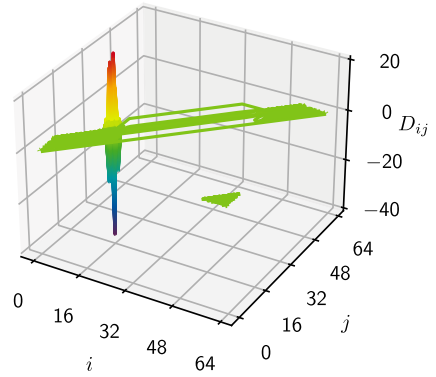
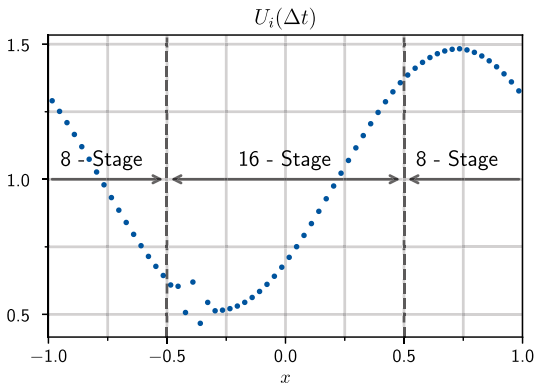
The solution $U_1 = U(\Delta t)$ after one step with maximum stable timestep $\Delta t = 0.21875$ is presented in Fig. 3a. Clearly, there are heavy oscillations at the interface between the two schemes. It is natural to ask whether these oscillations increase over time and cause a blow-up of the overall simulation. To answer this question, we consider the spectrum of D (or equivalently $P(Z)$) which is also displayed in Fig. 2. In the present case the spectral radius $\rho(D) < 1$ and thus the overall scheme is linearly stable [39, Chapter 8.6], which is confirmed by long time runs. This shows that the observed oscillations are no consequence of violated linear/absolute stability, but instead are due to the nonlinear stability properties of the scheme which are explored in the next subsection.

3.2.3. Nonlinear stability of the fully discrete system

The fully discrete system (3.9) is monotonicity-preserving in the sense of Harten [55] if and only if

$$D_{ij} \geq 0 \quad \forall i, j = 1, \dots, N. \tag{3.11}$$

The application of the P-ERK schemes leads in general to a violation of (3.11), even for uniform meshes as clearly seen in Fig. 3a. To shed some more light on this, we present a plot of the matrix entries D_{ij} in Fig. 3b. It is evident from Fig. 3b that monotonicity is lost at the intersection of the schemes, precisely where the higher-stage method retrieves only first-order data from the lower stage scheme. First-order information means in this case that the intermediate state $U_n + \Delta t \sum_{i=1}^S \sum_{r=1}^R b_i^{(r)} K_i^{(r)}$ (cf. (2.2b)) received on the $r = 1$ partition only information from one evaluation of the RHS. In particular, the rows $i = 17, \dots, 24$ of D have negative entries which originates from the fact that for these points the higher stage method still relies on first-order information only, precisely the



(a) Result U_1 of (3.9) after one step. The vertical dashed lines indicate the partitioning into $U^{(1)}$ and $U^{(2)}$.

(b) Non-zero entries D_{ij} of fully-discrete system matrix D , cf. (3.9).

Fig. 3. Cell averages $U_i(\Delta t)$ (Fig. 3a) after performing one timestep of the second-order 8-16 optimized P-ERK scheme applied to (3.7b) on a uniform mesh with $\Delta x = 2/64$. The entries of the fully discrete Matrix D (cf (3.9)) are shown in Fig. 3b.

previous iterate U_n plus the first stage K_1 . In particular, for the $S = 16, E = \{8, 16\}$ combination at stage $i = 8$ the only update to $U^{(1)}$ is due to $(c_8 - a_{8,1}^{(1)})K_8^{(1)}$, while $U^{(2)}$ receives an update including $a_{8,7}^{(2)}K_7^{(2)}$ which contains seven consecutive evaluations of the RHS.

The influence of the negative entries can be clearly seen in Fig. 3a, where oscillations arise behind the transition from one scheme to the other. Since the initial condition is transported from the left to the right, the usage of the upwind fluxes corresponds to using information from the left. This is why there are oscillations in this case at $x = -0.5$, since the higher stage-evaluation method receives only first-order information based upon $K_1^{(1)}$. At $x = 0.5$ there are for positive advection velocity $a = 1$ no oscillations, as the lower stage-evaluation method retrieves high order information at any time. In turn, for a negative transportation velocity, i.e., $u_i - u_x = 0$ the oscillations arise at $x = 0.5$.

We mention that each row sum $d_i := \sum_{j=1}^N D_{ij}$ equals one, reflecting that the scheme is conservative, agreeing with the criteria (2.12). Furthermore, the P-ERK scheme converges with second-order once the asymptotic regime is reached.

This loss of monotonicity is noteworthy as it foreshadows possible problems for more complex scenarios. The first-order Godunov finite volume scheme is known to be very diffusive, yet already for this system and uniform meshes, artificial oscillations are introduced. There are several ways how the oscillations can be mitigated. First of all, the oscillations become less severe if the timestep is reduced since this leads to a reduction in magnitude of the negative entries D_{ij} . Next, for a higher spatial resolution N the influence of the negative entries is reduced as neighboring values U_i are of more similar value (note that no oscillations can occur for a constant solution since $d_i := \sum_{j=1}^N D_{ij} = 1$). Finally, the oscillations can also be reduced if the difference in the number of stage evaluations E is reduced, with monotonicity recovered for the same method in each part of the domain. These observations are quantified in Section 3.2.5 for the intended application of the P-ERK schemes to non-uniform grids.

3.2.4. Alternative P-ERK schemes

It is also instructive to consider a different stage evaluation pattern of the P-ERK methods. The original P-ERK methods perform all shared evaluations at the end, cf. (3.4). In principle, also different stage evaluation patterns are possible while preserving the key features of the P-ERK schemes. For instance, one can construct “alternating” schemes such as

i	c	$A^{(1)}$				
1	0					
2	$1/10$	$1/10$				
3	$2/10$	$2/10$	0			
4	$3/10$	$3/10 - a_{4,3}^{(1)}$	0	$a_{4,3}^{(1)}$		
5	$4/10$	$4/10$	0	0	0	
6	$5/10$	$5/10 - a_{6,4}^{(1)}$	0	0	$a_{6,4}^{(1)}$	0
\mathbf{b}^T		0	0	0	0	1

(3.12)

from the same stability polynomial $P_{2;4}^{(1)}(z)$ (3.5) that would be used for the traditional P-ERK scheme (3.4). For the first scheme, specified according to (3.12), the stages $K_1^{(1)}, K_3^{(1)}, K_4^{(1)}, K_6^{(1)}$ are evaluated, while for the classic P-ERK scheme (3.4) $K_1^{(1)}, K_4^{(1)}, K_5^{(1)}, K_6^{(1)}$ are evaluated.

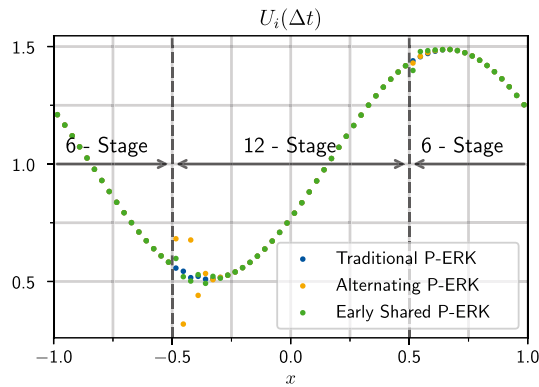


Fig. 4. Result U^1 to (3.9) after one timestep of two different 6-12 P-ERK schemes with different stage evaluation patterns on a uniform mesh. The solution colored in blue correspond to the standard P-ERK scheme (3.4) while the solution obtained with the alternating P-ERK scheme (3.12) is colored in orange.

Another possible design pattern for P-ERK schemes is to perform the shared evaluations at the early stages of the method, rather than at the end as for the traditional P-ERK schemes. The Butcher Tableau for the $p = 2, E = 4, S = 6$ scheme with this stage evaluation pattern is given by

i	c	$A^{(1)}$			
1	0				
2	$1/10$	$1/10$			
3	$2/10$	$2/10 - a_{3,2}^{(1)}$	$a_{3,2}^{(1)}$		
4	$3/10$	$3/10$	0	0	
5	$4/10$	$4/10$	0	0	0
6	$5/10$	$5/10 - a_{6,3}^{(1)}$	0	$a_{6,3}^{(1)}$	0
\mathbf{b}^T		0	0	0	0

(3.13)

where the free coefficients $a_{3,2}^{(1)}, a_{6,3}^{(1)}$ are again obtained from the identical stability polynomial $P_{2,4}^{(1)}(z)$ that may be used for the other methods. In this case, stages $\mathbf{K}_1^{(1)}, \mathbf{K}_2^{(1)}, \mathbf{K}_3^{(1)}, \mathbf{K}_6^{(1)}$ are evaluated.

To illustrate the influence of the stage evaluation pattern Fig. 4 compares the results after one timestep for the standard (3.4), alternating (3.12), and shared early stage (3.13) P-ERK scheme. Clearly, the schemes yield different results at the interfaces at the schemes, with the strongest oscillations observed for the alternating scheme. Notably, the scheme with shared early stage evaluations causes the largest errors at the fine-coarse interface at $x = 0.5$.

Analogous to the linear stability properties discussed in Section 2.2, the action of partitioned schemes is no longer determined by the stability polynomials of the individual schemes only, even for linear ODEs. In fact, due to the partitioning there is a discontinuity in the time-integration method introduced and consequently, the final outcome does depend on the design of the stage evaluation pattern.

This feature is similar in spirit to SSP time integration schemes. Consider for instance the two-stage second-order stability polynomial, for which infinitely many Runge-Kutta methods exist, for instance the midpoint method and Heun’s method mentioned earlier. When applied to a linear problem, both give the identical result, as the overall outcome is governed by the same stability polynomial. In contrast, when applied to a nonlinear, potentially discontinuous problem, the implementation of the intermediate stage is of crucial importance as this determines the nonlinear stability properties, see the instructive example given in [56].

3.2.5. Non-uniform meshes

Coming to the intended application case of the P-ERK schemes to non-uniform meshes, it is observed that the presence of a discontinuity in the system matrix L (3.7b) further impairs nonlinear stability. To give evidence for this, we consider again the model problem (3.6) and refine the grid in $\Omega^{(2)} := [-0.5, 0.5]$ by a factor $\alpha > 1$, i.e., $\Delta x_2 = \Delta x_1 / \alpha$. In the non-refined part of the mesh with $\Delta x_1 = 2/64$ an $E^{(1)} = 8$ stage-evaluation method is used, while in the refined part the number of stage evaluations is determined based on α such that the timestep $\Delta t_8 = 0.21875$ can be kept constant. For all configurations presented in this section linear stability was checked by means of the spectral radius of the fully discrete system matrix D (3.9) and long-time simulations.

To measure the oscillations, we consider the relative total variation (TV) increase after one timestep

$$e^{\text{TV}} := \frac{\|U_1\|_{\text{TV}} - \|U_0\|_{\text{TV}}}{\|U_0\|_{\text{TV}}} \tag{3.14}$$

with total variation seminorm $\|U\|_{\text{TV}} := \sum_i |U_{i+1} - U_i|$. e^{TV} is reported in Table 1 for a range of stage evaluations $E^{(2)}$ in the refined region. Only for the most modestly refined grid the total variation actually decreases, preserving the diffusive nature of

the Godunov scheme. For higher refinement ratios, exponential increase in total variation is observed, with dramatic values for the highest refinements.

Table 1

Increase in relative total variation e_{TV} for non-uniform meshes with grid sizes $\Delta x, \frac{\Delta x}{\alpha}$. The $E^{(1)} = 8$ method is used in the non-refined region for all α .

$E^{(2)}$	9	10	11	12	13	14	15	16
α	1.125	1.25	1.375	1.5	1.625	1.75	1.875	2.0
e^{TV}	-0.03	0.11	0.55	1.65	3.71	7.85	15.4	26.0

Similar to the case for uniform grid with partitioned schemes, the oscillations decrease with a reduction in timestep, increase in spatial resolution, and smaller difference in the number of stage evaluations $E^{(2)} - E^{(1)}$. To illustrate this, we reduce the maximum stable timestep Δt_8 by a factor $CFL \leq 1$ and provide the increase in total variation in Table 2 Next, the base resolution $N^{(1)}$, i.e., the

Table 2

Increase in relative total variation e_{TV} for non-uniform meshes with varying base grid size $\Delta x^{(1)} = 2/64$ and two times ($\alpha = 2$) finer grid in $\Omega^{(2)} = [-0.5, 0.5]$ under reduction of timestep $\Delta t = CFL \cdot \Delta t_8$. The coarse cells are integrated with the $E^{(1)} = 8$ stage evaluation method, while the fine cells are integrated using the $E^{(2)} = 16$ method.

CFL	0.4	0.5	0.6	0.7	0.8	0.9	1.0
e^{TV}	-0.01	0.03	0.28	1.21	4.04	11.5	26.0

resolution in $\Omega^{(1)} := (-1, -0.5] \cup [0.5, 1)$ is varied from $N^{(1)} = 64$ to $N^{(1)} = 4095$ for the $E^{(1)} = 8, E^{(2)} = 16$ scheme. As tabulated in Table 3 the increased resolution causes the increase in total variation to diminish on finer grids.

Table 3

Increase in relative total variation e_{TV} for non-uniform meshes with varying base grid size $\Delta x_{base} = 2/N^{(1)}$ and two times ($\alpha = 2$) finer grid in $\Omega^{(2)} = [-0.5, 0.5]$. The coarse cells are integrated with the $E^{(1)} = 8$ stage evaluation method, while the fine cells are integrated using the $E^{(2)} = 16$ method.

$N^{(1)}$	64	128	256	512	1024	2048	4096
e^{TV}	26.0	7.27	1.82	0.45	0.11	0.02	0.01

Finally, keeping cell size $\Delta x^{(1)} = \frac{2}{64}$ and ratio $\alpha = 2$, the difference in stage evaluations $\Delta E = E^{(2)} - E^{(1)}$ is varied. Here, we always set $E^{(2)} = 2E^{(1)}$ which implies that $\Delta E = E^{(2)} - E^{(1)} = 2E^{(1)} - E^{(1)} = E^{(1)}$. For smaller ΔE the increase in total variation is reduced, as shown in Table 4. This can be intuitively explained by the fact that for smaller ΔE the higher stage method does less stages with only first-order information based on $K_1^{(1)}$ from the lower stage method.

Table 4

Increase in relative total variation e^{TV} for non-uniform meshes with constant base grid size $\Delta x^{(1)} = 2/64$ and two times ($\alpha = 2$) finer mesh in $\Omega^{(2)} = [-0.5, 0.5]$ under variation of the number of base stage evaluations $E^{(1)} = \{2, 3, 4, \dots, 8\}$. The fine cells are integrated using the $E^{(2)} = 2E^{(1)}$ stage-evaluation method.

$\Delta E = E^{(1)}$	2	3	4	5	6	7	8
e^{TV}	-0.00	-0.01	-0.00	0.06	0.61	4.25	26.0

Since these issues are encountered for the Godunov scheme which evolves the cell averages, it is immediately clear that conventional limiting procedures are not applicable to remedy the loss of monotonicity. In particular, derivatives of standard procedures such as MUSCL-type limiters [4, Chapter 6] or related techniques to enforce, e.g., positivity [57] do not work as the mean state is already corrupted and therefore cannot be used as a fallback solution. In contrast, one would need to devise some form of flux limiting (or in this case stage-limiting) similar to techniques employed for the Lax-Wendroff scheme [4, Chapter 6.3][Chapter III.1] [3], which also evolves cell averages.

3.2.6. Relation of nonlinear stability analysis to published results

We would like to classify the previous observations in the context of the published results on P-ERK schemes. In [1,2] the P-ERK schemes have been successfully applied to the simulation of the compressible Navier-Stokes equations for applications such as flows around airfoils. The meshes employed in the aforementioned works are characterized by a high resolution and small refinement ratios between neighboring elements. For instance, the mesh for the simulation of a turbulent flow around the SD7003 airfoil employed in [2] has an average refinement ratio of $\bar{\alpha} \approx 1.15$. As illustrated by Table 1 moderate refinement ratios are much less vulnerable to monotonicity violations. As the refinement ration of neighboring cells is typically small, the difference in stage evaluations $E^{(r+1)} - E^{(r)}$ may also be relatively small. In fact, in [1,2] the authors use P-ERK families with a maximum difference in

stage evaluations of $\Delta E = 4$. Again, we demonstrated by means of Table 4 that the monotonicity violations are reduced for smaller ΔE . In conjunction with the fact that the simulated equations bear a diffusive nature, the potential loss of monotonicity is not observed in the aforementioned works.

Besides presenting application to the compressible Navier-Stokes equations, the inviscid isentropic vortex advection test case [40,58] is also studied in [1,2] for the purpose of validating the P-ERK schemes. Notably, they employ a random distribution of P-ERK schemes on a uniform grid to test the order of convergence of the schemes. For this test case it is in principle possible that the $E^{(R)} = 16$ scheme is placed next to an $E^{(1)} = 2$ (for the second-order accurate case) or $E^{(1)} = 3$ (for the third-order accurate case) scheme. Based on the findings in the previous section one might expect that this could reveal the loss of monotonicity described in the previous section. This is not the case, however, as the timestep has to be adapted to the scheme with the smallest number of stage evaluations which requires a significant reduction in timestep. For instance, the maximum stable timestep for the $p = 3, E = 3$ scheme is about 8 times smaller than the one for the $p = 3, E = 16$ scheme. As tabulated in Table 2 there are no artificial oscillations observed if the timestep is chosen small enough.

To further strengthen the assessment discussed above, we created a test case based on a structured, smoothly refined, high-resolution mesh where we use a second-order, 16-stage P-ERK scheme with stage evaluations $E = \{4, 6, 8, \dots, 16\}$. In agreement with the analysis above, this testcase can be run without reduction of the timestep for increased stage evaluations $E^{(R)}$, i.e., the stage normalized timestep $\Delta t/E^{(R)}$ can be kept almost constant for all schemes. This application is presented in more detail in Appendix A. However, we would like to stress that the use case considered in this work is different from the one considered in [1,2] and in Appendix A, as for quad/octree AMR the mesh refinement ratio is always a factor of two and thus the difference in stage evaluations $E^{(r)} - E^{(r-1)}$ grows exponentially with the number of refinement levels.

3.3. Higher-order P-ERK methods

In the original paper [1] P-ERK schemes of order $p = 2$ are presented which are extended to order $p = 3$ in [2]. A particular feature of Runge-Kutta methods is that, given a certain stability polynomial, infinitely many Runge-Kutta methods with corresponding Butcher arrays can be constructed. For the third-order P-ERK methods, the proposed archetype of the Butcher tableau is [2]

$$\begin{array}{c|ccc} 0 & & & \\ \frac{1}{3} & \frac{1}{3} & & \\ \frac{1}{3} & & & \\ \hline 1 & -1 & 2 & \\ \hline & 0 & \frac{3}{4} & \frac{1}{4} \end{array} \tag{3.15}$$

which corresponds to the third-order Runge-Kutta method presented in [59] with parameter $c_2 = \frac{1}{3}$. The presence of the negative entry $a_{3,1} = -1$ renders the incorporation of the $E = 3, p = 3$ in a P-ERK ensemble unattractive as the negative entry corresponds essentially to downwinding the numerical fluxes of stage \mathbf{K}_1 [56]. Alternatively, we choose the Butcher array of the canonical three-stage, third-order method by Shu and Osher [40] as the base scheme for our $p = 3$ P-ERK schemes, i.e.,

$$\begin{array}{c|ccc} 0 & & & \\ 1 & 1 & & \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4} & \\ \hline \frac{1}{2} & \frac{1}{6} & \frac{1}{6} & \frac{4}{6} \end{array} \tag{3.16}$$

We emphasize that the usage of the first Runge-Kutta stage \mathbf{K}_1 in the final update step does not increase storage requirements of the scheme as \mathbf{K}_1 is used anyways to compute the higher stages, cf. (3.4). Also, the computational costs per stage stay at two scalar-vector multiplications and one vector-vector addition.

We have not yet discussed how to choose the free abscissae $c_i, i = 2, \dots, S - 2$. In this work, we set

$$c_i = \frac{i-1}{S-3}, \quad i = 1, \dots, S-2 \tag{3.17}$$

which corresponds to a linear distribution of timesteps between $c_1 = 0$ and $c_{S-2} = 1$, similar to the second-order case [1]. We mention that more sophisticated choices of the abscissae are in principle possible, e.g., to improve the internal stability properties of the scheme [60]. Since this is not easily done and becomes also more relevant for many-stage methods which are not the main focus of this work, we restrict ourselves in this work to the linear distribution according to (3.17). Additionally, in contrast to the second-order accurate method, a nonlinear solver is required to obtain the $a_{i,i-1}$ from the coefficients α_j of the stability polynomial. Although in principle not guaranteed, with the choice of abscissae (3.17) we obtain positive Butcher coefficients $a_{i,i-1}$ for all optimized stability polynomials considered in this work.

As mentioned in Section 2.1 the construction of a fourth order P-ERK scheme is significantly more complex due to the coupling conditions involving the different Butcher arrays $A^{(r)}$ and is left for future research.

4. Construction of the stability polynomials

Besides their multirate property an additional advantage of the P-ERK schemes is that the stability polynomials of the constituting schemes can be optimized according to the intended application to maximize the admissible timestep Δt . In particular, we can

construct stability polynomials that are tailored to the problem at hand, i.e., the spatial discretization, boundary and initial conditions. The optimization problem for the p 'th order, $E^{(r)}$ degree stability polynomial thus reads

$$\max \Delta t \quad \text{s.t.} \quad \left| P_{p;E^{(r)}}^{(r)}(\Delta t \lambda) \right| \leq 1, \forall \lambda \in \sigma. \tag{4.1}$$

To solve this problem we resort to the approach developed in [47] which has also been applied in [1,2]. In the following, we first describe how the relevant eigenvalues of the spectrum σ are obtained and then discuss the solution of the optimization problem (4.1).

4.1. Generating Jacobian and spectrum

Throughout this work we use `Trixi.jl` [61–63], a high-order DG code written in `Julia` for the numerical simulation of conservation laws. `Trixi.jl` supports algorithmic differentiation (AD) which allows exact and efficient computation of the Jacobian $J(U)$ (2.5). As long as memory requirements are no concern, the Jacobian can be computed quickly even for relatively large ODE systems (1.2). In contrast, the eigenvalue decomposition to obtain the spectrum $\sigma(J(U))$ is computationally infeasible for practical simulation setups with millions of degrees of freedom (DoFs). At the same time, for the optimization of a stability polynomial the entire spectrum is not required. Instead, it suffices to supply a subset of the eigenvalues forming the boundary of the spectrum [62,64]. To obtain this subset of eigenvalues lying on the boundary of the spectrum efficiently, we employ the following routine:

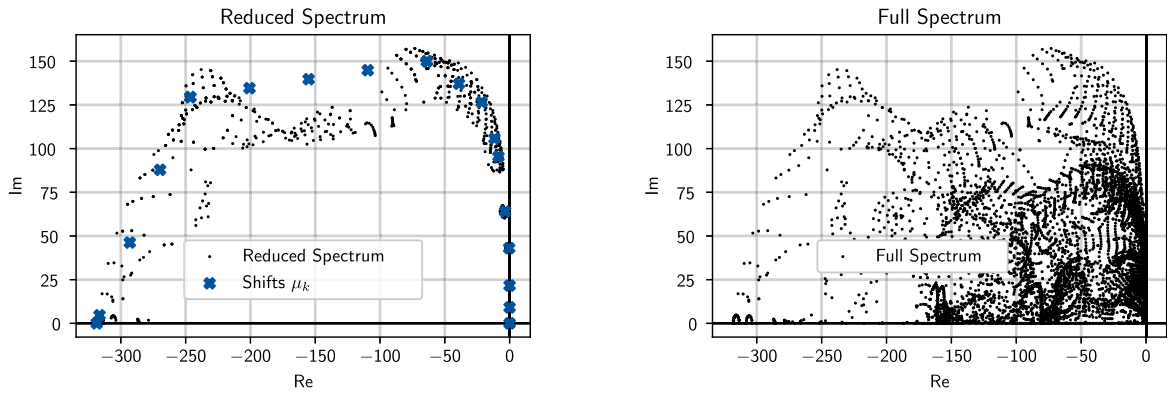
1. Compute $\tilde{\sigma}(\tilde{J}(U_0))$ for a semidiscretization $U' = \tilde{F}(U)$ with reduced minimal spatial resolution \tilde{h} such that the complete eigenvalue decomposition is still computational feasible. In this work, we use reduced systems with dimensionality $\mathcal{O}(10^3 \times 10^3)$.
2. Compute the convex hull $\tilde{\mu}(\tilde{\sigma})$ of the reduced spectrum $\tilde{\sigma}$. As the convex hull points can be quite distant from each other, additional points on the polygons are added such that the distance between two points is below the average distance δ between hull-points $\delta := \frac{\sum_{k=1}^{K-1} \|\tilde{\mu}_{k+1} - \tilde{\mu}_k\|_2}{K-1}$. It suffices to consider either the second or third quadrant of the complex plane due to symmetry of σ with respect to the real axis. This also assumes that there are no amplifying modes with $\text{Re}(\lambda) > 0$ in the spectrum, which is the case for the semidiscretizations arising from conservation laws considered in this work.
3. Increase the spatial resolution to the actual value used in the simulation, or the maximum value for which $J(U_0)$ can still be stored in main memory. Convert $J(U_0)$ to sparse format to save memory and computational costs for the subsequent Arnoldi iterations. For the DG discretizations considered here, $J(U_0)$ is typically extremely sparse due to the small DG stencils. As a side-note, ideally one would compute $J(U_0)$ already in sparse format. This, however, requires special techniques to identify the nonzeros (matrix-coloring) in the Jacobian which are not yet supported in `Trixi.jl`.
4. As discussed in Section 3.1 the spectrum of convection-dominated PDEs scales increases linearly with decreasing characteristic grid size h . Consequently, the convex hull $\tilde{\mu} \in \mathbb{C}^K$ of the reduced spectrum is scaled by \tilde{h}/h to agree with the discretization $U' = F(U)$.

Then, the ‘outer’ eigenvalues of $J(U_0)$ can be estimated by performing Arnoldi iterations with shift $\mu_k := \frac{\tilde{h}}{h} \tilde{\mu}_k$ to compute estimates to the eigenvalues close to the convex hull point μ_k . This can be done efficiently as the Arnoldi iterations for the different hull points $\mu_k, k = 1, \dots, K$ can be parallelized. Furthermore, as the Arnoldi iteration involves only matrix-vector products, this approach heavily benefits from storing $J(U_0)$ in sparse format. Moreover, the stopping tolerance can be set relatively high as an estimate to the spectrum $\sigma(J(U_0))$ is not only sufficient, but in fact beneficial since $\tilde{\sigma}(J(U))$ changes over the course of the simulation. This leads to an overall more robust stability polynomial. In terms of software we use `Arpack.jl`, a `Julia` wrapper around `arpack-ng` [65].

In this work, we try to estimate the outer shape of the spectrum with 1000 eigenvalues, hence supplying to each Arnoldi iteration a target number of $M := 1000/k$ eigenvalues. In general, after combining the estimates λ_k one will have less than 1000 distinct eigenvalues as some eigenvalues are found from multiple different shifts μ_k . In practice, this still gives a sufficient large set of eigenvalues $\lambda = \cup_{k=1}^K \lambda_k$.

5. If also in step 3 a reduced system had to be considered (for instance due to memory restrictions), scale the obtained eigenvalues λ accordingly.

To illustrate the advantages of the presented routine, the reduced spectrum and the complete spectrum computed through a standard eigendecomposition are presented in Fig. 5. The spectrum shown in this case corresponds to the initial condition leading to a Kelvin-Helmholtz instability (see [66] for the precise setup) with the discontinuous Galerkin spectral element method (DGSEM) [67,68] with flux-differencing [69,70] using solution polynomials of degree $k = 3$, Harten-Lax-Van Leer-Einfeldt (HLL) surface-flux [71], subcell shock-capturing [72] to suppress spurious oscillations, and entropy-conserving volume flux [73,74]. Clearly, the shape of the spectrum is captured very well by the reduced spectrum while being significantly faster to compute. Furthermore, by estimating the spectrum using the procedure outlined above, one automatically obtains a reduced spectrum that captures the essential information while speeding up the subsequent optimization of the stability polynomial (discussed in the next section) as there are fewer redundant constraints to be checked. In this case, the shifts $\tilde{\mu}_k$ are obtained from a reduced system with $\tilde{J} \in \mathbb{R}^{1024 \times 1024}$ for which the eigendecomposition can be performed in 0.4 seconds. The large system is for the purpose of this study chosen such that the full eigendecomposition can still be performed in somewhat reasonable time. Here, we consider $J \in \mathbb{R}^{16384 \times 16384}$ where the full



(a) Reduced spectrum obtained from the procedure described in Section 4.1.

(b) Full spectrum obtained from standard eigendecomposition.

Fig. 5. Reduced spectrum obtained from Arnoldi iterations around shifts (Fig. 5a) and full spectrum obtained from standard eigendecomposition (Fig. 5b). The spectrum corresponds to a DGSEM discretization of the compressible Euler equations with an initial condition leading to a Kelvin-Helmholtz instability.

eigendecomposition takes more than 11 minutes to complete. In contrast, for $K = 20$ scaled shifts the Arnoldi iterations require in total only about 30 seconds to complete. Here, $|\lambda| = 930$ distinct eigenvalues were found which are displayed in Fig. 5a.

We use this spectra estimation for the applications in Section 7 where the problem setup requires already for the initial condition a relatively fine grid resolution. Precisely, these are the visco-resistive Orszag-Tang vortex (7.1.2), the Kelvin-Helmholtz instability (7.2.1), and Rayleigh-Taylor instability (7.2.2).

4.2. Optimization of the stability polynomial

Equipped with a set of constraining eigenvalues, it remains to find stability polynomials of prescribed degrees $E^{(r)}$ and linear order p which maximize the stable timestep Δt . For this, we use the procedure developed in [47] which allows (in standard double precision) for an efficient and reliable optimization of general stability polynomials up to, say, degree 16. In principle, also the optimization of higher-degree polynomials is possible if a suitable basis, i.e., different from the monomials z^j can be found for the spectrum of interest. While this is possible for simple cases like parabolic (negative real axis), purely hyperbolic (imaginary axis) and circle/disk spectra, the choice of the basis for the general case is unclear [47]. The computational cost of the optimization procedure depends on the number of constraints, i.e., the number of supplied eigenvalues σ . For moderate numbers of eigenvalues, say, up to 2000 (which are in practice sufficient to estimate the spectrum of the operator) the optimization can be done in less than a minute.

In the case of quad/octree-based AMR the characteristic cell size is halved with every level of refinement. This implies in the case of convection-dominated spectra (3.3) that the stable timestep needs to be reduced by a factor two, or, in the case of stabilized Runge-Kutta methods, that the number of stage-evaluation needs to be doubled.

Hence, for an R -times refined grid, one needs to construct stability polynomials with degrees up to $S^{(R+1)} = S^{(1)} \cdot 2^R$, i.e., the polynomial degree scales exponentially in the number of grid refinements R . As mentioned above, the optimization approach developed in [47] is for general spectra (for which no special suited basis can be found) not capable of reliably optimizing such high degree polynomials. Through employing a different ansatz for the optimization, one can optimize stability polynomials for convection dominated spectra with degrees larger than 100 [64].

Thus, we have in principle the possibility to integrate four to five times refined meshes with the same timestep as used for the base discretization. However, as discussed in Section 3.2 the nonlinear stability properties of P-ERK schemes worsen quickly for an increasing difference in stage evaluations $\Delta E^{(r+1)} := E^{(r+1)} - E^{(r)}$, cf. Table 4. This renders the application of P-ERK schemes with stage evaluations $E^{(r)} = E^{(1)} \cdot \{1, 2, 4, 8, 16, \dots\}$ very inefficient, as the overall timestep needs to be reduced due to the poor nonlinear stability properties. As a consequence, we will restrict ourselves for practical cases presented in Section 7 to few-stage P-ERK families.

4.3. Constructing P-ERK methods from stability polynomials

Equipped with an optimized stability polynomial $P_{p,E^{(r)}}(z)$ it remains to construct the corresponding Runge-Kutta method. As the weights b^T and abscissae c are shared for all members of the family, these are fixed a priori. Thus, it remains to construct the free parameters of the Butcher arrays $A^{(r)}$, i.e., the entries $a_{i,i-1}^{(r)}$. For the sake of readability, we will truncate $a_{i,i-1}^{(r)}$ to a_i in this section. The central relation between the stability polynomial and the Butcher array is given by [37, Chapter IV.7]

$$P(z) = 1 + z b^T (I - zA)^{-1} \mathbf{1} \tag{4.2}$$

where $\mathbf{1}$ is the column-vector of ones. Following [47] the optimal stability polynomials are represented in monomial form, i.e.,

$$P_{p;E}(z) = \sum_{i=0}^p \frac{z^i}{i!} + \sum_{i=p+1}^E \alpha_i z^i, \quad \alpha \in \mathbb{R}^{E-p}, \tag{4.3}$$

with optimization variables α .

4.3.1. Second-order P-ERK methods

For the second-order P-ERK methods, the general form of the stability polynomial from an S -stage method is given by

$$P_{2;E}(z) = 1 + z + \frac{1}{2}z^2 + c_{S-1}a_S z^3 + c_{S-2}a_S a_{S-1} z^4 + \dots + c_{S-E+2} z^{S-E+2} \prod_{i=S-E+3}^S a_i. \tag{4.4}$$

Equipped with the optimized monomial coefficients α we can now construct the $a_i = a_{i,j-1}$ sequentially via

$$a_{S-i} = \frac{\alpha_{3+i}}{c_{S-i-1}} \prod_{j=0}^{i-1} \frac{1}{a_{S-j}}, \quad i = 0, \dots, S-3. \tag{4.5}$$

4.3.2. Third-order P-ERK methods

For the third-order P-ERK methods, the general form of the stability polynomial from an S -stage method is given by

$$P_{3;E}(z) = 1 + z + \frac{1}{2}z^2 + \frac{1}{6}z^3 + \sum_{i=1}^{S-4} \left(c_{S-1-i} a_S b_S \prod_{j=1}^i a_{S-j} + c_{S-2-i} b_{S-1} a_{S-1} \prod_{j=1}^i a_{S-1-j} \right) z^{i+3} + c_2 b_S a_S \prod_{j=1}^i a_{S-j} z^S. \tag{4.6}$$

Besides matching the monomial coefficients α we have a constraint arising from the third-order consistency conditions, cf. (2.3), see also [2].

$$\frac{1}{6} \stackrel{!}{=} b_S a_S + b_{S-1} a_{S-1}. \tag{4.7}$$

In contrast to the second-order case, the coefficients a_i cannot be directly computed from the monomial coefficients, but instead a nonlinear system of equations needs to be solved. Additionally, we demand that there are no negative entries in the Butcher arrays $A^{(r)}$ corresponding to downwinding [56] by requiring that the a_i and also $a_{i,1} = c_i - a_i$ are positive. In practice, we find such a solution relatively quick using a standard trust-region method with randomly initialized starting points.

5. Validation

Before coming to applications we seek to compare the combined P-ERK schemes to standalone optimized schemes with regard to errors and conservation properties. For this, we consider the classic isentropic vortex testcase [40,58] with the same parameters as used in [1,2], except that here we use a more compact domain $\Omega = [0, 10]^2$. In addition to the convergence studies presented in [1,2] we compare the errors of the P-ERK schemes to the errors of an individual scheme. This information is essential for assessing the effectiveness of the P-ERK schemes since one is ultimately interested in accuracy per computational cost. For this testcase, the domain is initially discretized with 64 elements per direction, on which the solution is reconstructed using polynomials with degrees two (for second-order P-ERK scheme) and three (for third-order P-ERK scheme). The mesh is dynamically refined around the vortex based on the distance to the center of the vortex with two additional levels of refinement, i.e., the minimal grid size is in this case $h = \frac{10}{256}$. The numerical fluxes are computed using the Harten-Lax-Van Leer-Contact (HLLC) flux [75]. For solution polynomials of degree $k = 2$ we construct $p = 2$ optimal stability polynomials with degrees 3, 6, and 12, while for solution polynomials of degree $k = 3$, $p = 3$ stability polynomials of degrees 4, 8, and 16 are optimized. We choose this particular P-ERK families since the presence of the additional free parameter allows the optimization of the lowest-degree stability polynomials, i.e., $P_{2,3}(z), P_{3,4}(z)$, to the spectrum.

The density errors at final time $t_f = 20$

$$e_\rho(x, y) := \rho(t_f, x, y) - \rho_h(t_f, x, y) \tag{5.1}$$

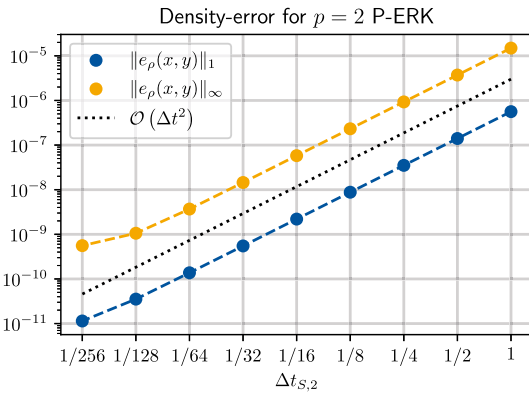
are tabulated in Table 5 for each individual optimized Runge-Kutta method and the composed P-ERK scheme. Both L^∞ and domain-normalized L^1 errors are for the joint P-ERK schemes of similar size as the errors of the individual schemes.

In [22] it is shown that at the interfaces of different schemes order reduction [76, Chapter II.2] may be encountered. As argued in [22], the interfaces between methods act like additional time-dependent internal boundary conditions, thus potentially suffering from effects also encountered for non-periodic problems [76, Chapter II.2]. Nevertheless, for both L^1 and L^∞ error, both second

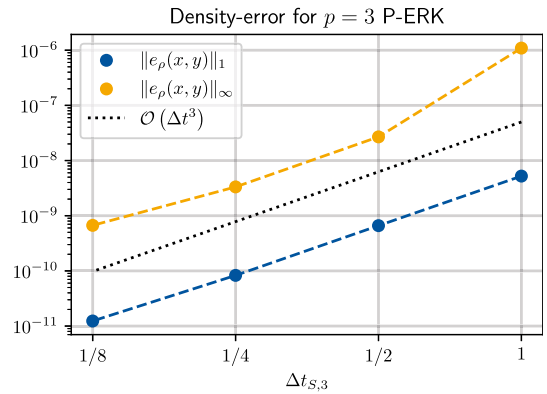
Table 5

Density-errors for isentropic vortex advection testcase after one pass through the domain $\Omega = [0, 10]^2$ at $t_f = 20$. The compressible Euler equations are discretized using the DGSEM with HLLC flux [75] on a dynamically refined mesh with cell sizes ranging from $\Delta x^{(1)} = 10/64$ to $\Delta x^{(3)} = 2/256$. For Runge-Kutta order $p = 2$, the local polynomials are of degree $k = 2$ while for Runge-Kutta order $p = 3$ the solution is constructed using $k = 3$ polynomials.

Method	$\frac{1}{ \Omega } \ e_\rho(x, y)\ _{L^1(\Omega)}$	$\ e_\rho(x, y)\ _{L^\infty(\Omega)}$
P-ERK _{2,3}	$1.77 \cdot 10^{-5}$	$6.43 \cdot 10^{-4}$
P-ERK _{2,6}	$1.77 \cdot 10^{-5}$	$6.40 \cdot 10^{-4}$
P-ERK _{2,12}	$1.74 \cdot 10^{-5}$	$6.30 \cdot 10^{-4}$
P-ERK _{2,{3,6,12}}	$2.09 \cdot 10^{-5}$	$5.86 \cdot 10^{-4}$
P-ERK _{3,4}	$4.05 \cdot 10^{-7}$	$1.12 \cdot 10^{-5}$
P-ERK _{3,8}	$4.05 \cdot 10^{-7}$	$1.12 \cdot 10^{-5}$
P-ERK _{3,16}	$4.05 \cdot 10^{-7}$	$1.11 \cdot 10^{-5}$
P-ERK _{3,{4,8,16}}	$3.73 \cdot 10^{-7}$	$1.36 \cdot 10^{-5}$



(a) Density errors for second-order $E = \{3, 6, 12\}$ P-ERK family.



(b) Density errors for third-order $E = \{4, 8, 16\}$ P-ERK family.

Fig. 6. L^1 and L^∞ density errors for second (Fig. 6a) and third-order (Fig. 6b) P-ERK schemes applied to the isentropic vortex advection testcase. We observe the expected order of convergence for both second and third-order P-ERK schemes until the spatial discretization error dominates the overall error. This becomes especially visible for the L^∞ errors plotted in orange.

Table 6

Conservation-errors for isentropic vortex advection testcase after one pass through the domain at $t_f = 20$ for the third-order P-ERK scheme and the three-stage, third-order Shu-Osher scheme.

	P-ERK _{3,{4,8,16}}	SSP _{3,3}
$e_\rho^{\text{Cons}}(t_f)$	$3.94 \cdot 10^{-13}$	$7.82 \cdot 10^{-13}$
$e_{\rho v_x}^{\text{Cons}}(t_f)$	$4.79 \cdot 10^{-14}$	$4.89 \cdot 10^{-13}$
$e_{\rho v_y}^{\text{Cons}}(t_f)$	$7.11 \cdot 10^{-14}$	$5.00 \cdot 10^{-13}$
$e_{\rho e}^{\text{Cons}}(t_f)$	$1.06 \cdot 10^{-12}$	$3.64 \cdot 10^{-12}$

and third-order P-ERK schemes display the expected order of convergence, as illustrated in Fig. 6. As in [1,2], we employ local polynomials of degree $k = 6$ for the convergence tests for both second and third-order P-ERK to minimize spatial errors.

It remains to examine whether the application of P-ERK schemes causes additional violation of the conserved fields compared to a standard Runge-Kutta scheme. For this purpose the third-order P-ERK scheme is compared against the three stage, third-order SSP method by Shu and Osher. As for the comparison of the density-errors of the P-ERK schemes to standalone schemes, the local polynomials are of degree $k = 3$. The conservation error

$$e_u^{\text{Cons}}(t_f) := \left| \int_{\Omega} u(t_0, \mathbf{x}) d\mathbf{x} - \int_{\Omega} u(t_f, \mathbf{x}) d\mathbf{x} \right| \tag{5.2}$$

is computed for mass, momentum and energy after one pass of the vortex through the domain and tabulated in Table 6. Clearly, the


```

function calc_volume_integral!(du, u, mesh, nonconservative_terms, equations,
                             volume_integral::VolumeIntegralWeakForm,
                             dg::DGSEM, cache)
    # Loop over all elements in the cache
    @threaded for element in eachelement(dg, cache)
        weak_form_kernel!(du, u, element, mesh, nonconservative_terms,
                          equations, dg, cache)
    end
    return nothing
end

```

Listing 1: Volume integral computation for standard time integration schemes in `Trixi.jl` [61–63].

```

function calc_volume_integral!(du, u, mesh, nonconservative_terms, equations,
                             volume_integral::VolumeIntegralWeakForm,
                             dg::DGSEM, cache,
                             elements_r::Vector{Int64})
    # Loop over elements of the r'th level
    @threaded for element in elements_r
        weak_form_kernel!(du, u, element, mesh, nonconservative_terms,
                          equations, dg, cache)
    end
    return nothing
end

```

Listing 2: Volume integral computation for P-ERK schemes in `Trixi.jl` [27]. In contrast to the standard volume integral computation given by Listing 1, the P-ERK version takes an additional argument `elements_r` which contains the elements bundling the $k + 1$ polynomial coefficients from $U^{(r)}$, cf. (2.1b).

errors of the $p = 3, E = \{4, 8, 16\}$ P-ERK family are on par with the errors for the Shu-Osher scheme. In particular, these conservation violations are dominated by the spatial accuracy, i.e., reducing the timestep has very little influence. We also compared the $p = 2, E = \{3, 6, 12\}$ P-ERK family to the two-stage second-order SSP scheme (Heun’s method) for a discretization with $k = 2$ solution polynomials. For this configuration the conservation errors are negligible, both for the P-ERK and SSP_{2,2} scheme.

All presented results from this section can be reproduced using the publicly available repository [27].

6. Methodology

In this section, we outline the procedure for the application of P-ERK schemes to simulations with quad/octree-based AMR in `Trixi.jl`.

6.1. Implementation

For obtaining the spatial semidiscretization (1.2) we use the DGSEM [67,68] with flux-differencing [69,70] as implemented in `Trixi.jl`. To demonstrate that indeed no changes to the spatial discretization are required, we exemplarily provide the standard implementation of the volume integral computation in `Trixi.jl` in Listing 1 and the implementation used for the P-ERK schemes in Listing 2. The only difference is that the P-ERK version takes an additional argument `elements_r` which contains the indices of the elements of partition r .

6.2. Dynamic partitioning based on AMR

In this work we employ two types of quad-octree based meshes with AMR. First, we consider a `TreeMesh` based on a hierarchical tree data structure with uniform cell sizes obtained from subsequent refinements of a square/cubic domain. For these types of meshes, the assignment of cells to partitions is trivial as the cell sizes correspond to the position of the nodes in the tree, which makes it very efficient to query the characteristic grid size of each cell. Additionally, we also employ in principle unstructured meshes based on the `p4est` library [77]. In that case, we cannot simply use the position of a cell in the tree data structure as a measure for the characteristic grid size, since cell sizes can vary even though they are formally on the same level of refinement. This implies that the computation of a criterion (e.g., shortest edge length, inner/outer cell diameter, determinant of the coordinate-transformation matrix,...) is required for the assignment of each cell which introduces additional computational work. Most examples considered in this work employ the first mesh type but we also present one application of the latter, where the shortest edge length of each cell is recomputed in the partition assignment step.

`Trixi.jl` requires four data structures to be assigned to the partitions. First, `elements` and its associated `boundaries` are trivially assigned according to the size of the associated cell. Second, `interfaces` are by construction only between conforming cells of thus identical size and are hence also uniquely associated with a level. Finally, `mortars` (which take care of hanging nodes at the non-conforming interface of cells of different sizes [78]) are associated to the partition corresponding to the fine neighbor cells in order to receive updates whenever the fine neighboring cell is updated.

6.3. Aptitude for massively-parallel, distributed-memory architectures

We would like to emphasize that the presented P-ERK schemes are particularly well-suited for massively-parallel, distributed-memory architectures. This is due to the fact that the partitioning of the cells belonging to a shared-memory unit can be done completely independently of the other units, assuming a unique partitioning criterion is available. In the case of distributed storage of the mesh data, a functionality provided by `p4est`, the index arrays of the `elements`, `boundaries`, `interfaces`, and `mortars` would then also be stored on each memory unit only. In particular, no additional data exchange is needed if the P-ERK method is used over a standard Runge-Kutta method.

6.4. Assessing performance

It is clear that the potential speedup of simulation using P-ERK is strongly depending on the ratio of coarse to fine cells. In case of two levels of refinement, we can expect almost a speedup of factor two if there is only one refined cell and the remainder of the mesh is coarse. Conversely, there is effectively no speedup to be gained if there is only one coarse cell left with the rest of the mesh refined.

6.4.1. Actual number of right-hand-side evaluations

In order to accurately assess the performance of a given P-ERK scheme when applied to a specific problem we require a way to compute the optimal, minimal runtime. In the ideal scenario, one would have an optimized scheme for every cell size present in the mesh such that the timestep does not need to be reduced if the cells are further refined throughout the simulation. As argued in Section 3.2 this is in practice not possible due to the loss of monotonicity that occurs for the usage of schemes with increasing difference in stage evaluations $\Delta E^{(r+1)} := E^{(r+1)} - E^{(r)}$. Consequently, we restrict ourselves in practice to, e.g., a three-level ($R = 3$) P-ERK family with moderate number of stage evaluations such as $E = \{3, 6, 12\}$. If, however, there is now a mesh with more than three levels L of refinement, i.e., $L > R$ optimality is inevitably lost since either the lowest or highest level is integrated non-optimal. In this work, we always use the highest stage-evaluation method for the finest level and fill the lower levels with the lowest stage-evaluation method. For instance, if we have a mesh with six levels of refinement and a P-ERK scheme with $E = \{3, 6, 12\}$, we would use for the finest cells the $E = 12$ scheme, for the second finest cells the $E = 6$ scheme, and for the remaining levels the $E = 3$ scheme. For this case, the number of scalar, i.e., per DG solution coefficient, RHS evaluations $N_{\text{RHS}}^{\text{Actual}}$ can be computed as

$$N_{\text{RHS}}^{\text{Actual}} = \sum_{i=1}^{\lceil N_t/N_{\text{AMR}} \rceil} N_{\text{AMR}} N (k+1)^{N_D} \left[\sum_{l=1}^R E^{(l)} N_C(i, l) + \sum_{l=R+1}^L E^{(1)} N_C(i, l) \right] \tag{6.1}$$

In the formula above, N_t denotes the overall number of timesteps to progress the simulation from initial time t_0 to final time t_f . N_{AMR} is the number of timesteps taken until the next (potential) change of the mesh, N denotes the number of fields of the PDE (1.1), k is the DG solution polynomial degree, and $N_D \in \{1, 2, 3\}$ denotes the number of spatial dimensions. $N_C(i, l)$ denotes the number of cells associated with the l 'th level assigned at the i 'th AMR call and $E^{(l)}$ denotes the number of stage evaluations for the l 'th level.

6.4.2. Minimal number of right-hand-side evaluations

For an optimal scheme one would have a method $E^{(l)}$ for every level $l = 1, \dots, L$ of refinement. In this case, the second sum in (6.1) is not necessary as the first sum would already cover all levels of refinement, i.e., $R = L$. Recalling that for the quad/octree meshes considered here the cell sizes vary from level to level by a factor of 2 it is clear that for a coarser cell, in principle, a timestep twice as large can be used. For the optimized Runge-Kutta schemes considered here, this translates to employing a method with half the number of stage evaluations E . Consequently, the total number of additional scalar, i.e., per degree of freedom (DoF), right-hand-side evaluations $N_{\text{RHS}+} := N_{\text{RHS}}^{\text{Actual}} - N_{\text{RHS}}^{R=L}$ due to the lack of a full range of optimized schemes can be computed as

$$N_{\text{RHS}+} = \sum_{i=1}^{\lceil N_t/N_{\text{AMR}} \rceil} N_{\text{AMR}} N (k+1)^{N_D} \sum_{l=R+1}^L E^{(1)} \left(1 - \frac{1}{2^{l-R}} \right) N_C(i, l). \tag{6.2}$$

The term in parentheses computes the difference in actual stage evaluations $E^{(1)}$ to the number of in principal required stage evaluations $E^{(l)}/2^{l-R}$ for the $l = R + 1, \dots, L$ coarser levels.

For most examples considered in this work, a three-member P-ERK family involving the non-optimized $p = 3, E^{(1)} = 3$ schemes is performing best since this leads to a P-ERK scheme with minimal difference in stage evaluations ΔE which is advantageous when it comes to avoiding artificial oscillations at the interfaces, see Table 4. Then, the linearly stable timestep is usually doubling already for the $E^{(2)} = 4$ method and doubling a second time for, say, the $E^{(3)} = 7$ scheme. While the timestep Δt decreases by a factor of four from the $E^{(3)} = 7$ to the $E^{(1)} = 3$ scheme, the number of stage evaluations decreases only by a factor of $3/7 > \frac{1}{4}$. In other words, such a P-ERK scheme is non-optimal on every but the finest level $r = R$. Hence, to quantify the additional computational effort due to the lack of a full range of optimized schemes we need to compute the number of additional RHS evaluations not only for those levels $l = R + 1, \dots, L$ that are not integrated with the optimal scheme but also for the levels $l = 1, \dots, R - 1$ for which methods are present. Thus, the optimal, gold-standard P-ERK method has stage evaluations $E_{\star} = E_{\star}^{(1)} \cdot \{1, 2, 4, \dots, 2^{L-1}\}$, or, equivalently, $E_{\star} = E^{(R)} \cdot \{2^{-(L-1)}, 2^{-(L-2)}, \dots, 1\}$. Consequently, the overall number of additional RHS evaluations $N_{\text{RHS}+}$ compared

to a hypothetical optimal P-ERK scheme for which both timestep and number of stage evaluations halve for every member method the number of additional RHS evaluations

$$N_{\text{RHS}+}^{\text{Hyp}} := N_{\text{RHS}}^{\text{Actual}} - N_{\text{RHS}}^{\text{Opt}} \quad (6.3)$$

which can be computed as

$$N_{\text{RHS}+}^{\text{Hyp}} = \sum_{i=1}^{\lceil N_i/N_{\text{AMR}} \rceil} N_{\text{AMR}} N(k+1)^{N_D} \left[\sum_{l=1}^{R-1} \left(E^{(l)} - \frac{E^{(R)}}{2^{l-1}} \right) N_{\text{C}}(i, l) + \sum_{l=R+1}^L \left(E^{(1)} - \frac{E^{(R)}}{2^{l-1}} \right) N_{\text{C}}(i, l) \right]. \quad (6.4)$$

Note that this assumes the same maximal linearly stable timestep is achieved for both the optimal and actual P-ERK method, i.e., $E^{(R)} = E_{\star}^{(R)}$.

It is worthwhile to discuss (6.4) for a concrete example. In this work, the method with highest stage-evaluations is often $E^{(3)} = 6$ and it is thus not immediately clear how one would interpret e.g. $\frac{6}{2^2} = 1.5$ stage evaluations for the hypothetical optimal scheme. To resolve this, recall that for optimized stabilized Runge-Kutta methods the maximum linearly stable timestep increases linearly with the number of stage evaluations. Thus, the computational effort in the ideal case (i.e., without loss of monotonicity) of a P-ERK scheme with base method of $E^{(1)} = 3$ is equivalent to a P-ERK scheme with base method $E^{(1)} = 6$ since (again, in the ideal case) the timestep of the latter family can be doubled, hence cutting the number of timesteps N_i and the AMR interval N_{AMR} in half. Thus, up to rounding, the number of scalar RHS evaluations $N_{\text{RHS}}^{\text{Opt}}$ for two P-ERK families with different number of base evaluations $E^{(1)}$ is the same:

$$N_{\text{RHS}}^{\text{Opt}} = \sum_{i=1}^{\lceil N_i/N_{\text{AMR}} \rceil} N_{\text{AMR}} N(k+1)^{N_D} \sum_{l=1}^L E^{(l)} N_{\text{C}}(i, l) \quad (6.5)$$

$$= \sum_{i=1}^{\lceil (N_i/2)/(N_{\text{AMR}}/2) \rceil} \frac{N_{\text{AMR}}}{2} N(k+1)^{N_D} \sum_{l=1}^L 2E^{(l)} N_{\text{C}}(i, l) \quad (6.6)$$

Now, one can in principal choose $E^{(R)}$ high enough such that $E^{(R)}/2^{L-1}$ is still an integer.

6.4.3. Degree of optimality

Equipped with the number of additional RHS evaluations $N_{\text{RHS}+}^{\text{Hyp}}$ and the average cost of evaluating the RHS per cell $\tau_{\text{RHS}/\text{C}}$, which is provided by `Trixi.jl`, one can compute the potential savings of a fully-optimal scheme. Neglecting possible additional overhead due to the usage of more levels, the hypothetical optimal runtime τ_{Opt} can then be computed as

$$\tau_{\text{Opt}} := N_{\text{RHS}}^{\text{Opt}} \cdot \tau_{\text{RHS}/\text{C}} \quad (6.7a)$$

$$= \tau - N_{\text{RHS}+}^{\text{Hyp}} \cdot \tau_{\text{RHS}/\text{C}}, \quad (6.7b)$$

where τ denotes the measured wallclock runtime of an actual simulation with a P-ERK scheme and $\tau_{\text{RHS}/\text{C}}$ quantifies the average cost of evaluating the RHS F (1.2b) per cell.

We remark that (6.7a) is a simplified model which assumes that the measured wallclock runtime τ is solely governed by the number of RHS evaluations. In practice, we see that `Trixi.jl` may be more a memory- than compute-bound code and thus the actual runtime τ may not be directly proportional to the number of RHS evaluations, but may be affected by other factors such as cache misses due to non-coalesced memory accesses.

It is important to note that (6.7b) is only valid if the CFL number of the actual simulation using a P-ERK scheme does not need to be reduced below the value obtained during optimization for absolute stability. While this is the case for the hyperbolic-parabolic examples considered in this work, the practical CFL number used in the simulation of hyperbolic PDEs needs to be reduced. In this case, the number of additional RHS evaluations $N_{\text{RHS}+}^{\text{Hyp}}$ is given by

$$N_{\text{RHS}+}^{\text{Hyp}} := N_{\text{RHS}}^{\text{Actual}} - \frac{\text{CFL}^{\text{Actual}}}{\text{CFL}^{\text{Opt}}} N_{\text{RHS}}^{\text{Opt}}. \quad (6.8)$$

To examine the performance of the non-optimal schemes we will provide the ratio

$$\kappa := \frac{\tau_{\text{Opt}}}{\tau} \leq 1 \quad (6.9)$$

for the applications discussed in Section 7. This is a measure of optimality in the sense that this quantifies how much of the theoretically possible speedup is actually achieved in practice.

6.4.4. Memory footprint

The additional data structures due to the partitioning increase the memory footprint of the method. This is a classic example of storage vs. computations where we save RHS evaluations at the cost of storing additional information. In particular, for the

herein employed `TreeMesh` and `p4est` mesh for each element, interface, boundary, and mortar an integer specifying the corresponding method r is stored. For other mesh types, such as the structured mesh used in Appendix A there are only cells required and thus only for those additional data structures indicating the partitioning are required. For the considered examples the partitioning variables cause on average a 10 – 30% increase in memory footprint.

7. Applications

In this section, we present a number of applications of the P-ERK schemes to classic test problems with quad/octree-based AMR. All presented cases are made publicly available in the reproducibility repository [27]. We begin by studying hyperbolic-parabolic PDEs, i.e., systems with viscous fluxes for which we expect in general a better performance of the P-ERK schemes due to inherent damping of potential high-frequency oscillations at the interfaces of methods. Additionally, we also consider hyperbolic PDEs to assess whether the P-ERK schemes can be used efficiently in practice for such systems without the need for additional stabilization. In particular, this is a novel contribution since the P-ERK schemes have so far only been applied to hyperbolic-parabolic systems [1,2].

For all considered examples, three-level ($R = 3$) P-ERK families are observed to be the most efficient P-ERK schemes. In particular, the methods are always chosen such that the admissible timestep doubles between levels, yielding values of $\text{CFL} = \{1, 2, 4\}$, cf. (3.2). Schemes with more family members, i.e., $R > 3$ are observed to suffer from the issues discussed in Section 3.2 which, in turn, demand a reduction in the CFL number, rendering them ultimately less efficient than the three-level families. In this work, we compare the multi-level P-ERK schemes always to a single optimized P-ERK method and four third-order integrators optimized for PDEs implemented in `OrdinaryDiffEq.jl` [79]. These are a seven-stage low-dissipation, low-dispersion scheme optimized for wave propagation problems with DG methods [80], a five-stage method optimized for high-order spectral difference methods applied to wave-propagation problems [81], a four stage method optimized for compressible Navier-Stokes equations [82], and a recently published five-stage method with timestep computation based on error-control [83]. Furthermore, we compare also to canonical third-order Shu-Osher SSP scheme [40], which is the considered benchmark method in [1,2].

For all simulations we aim for cells as coarse as possible that still allow for a meaningful simulation. This is important since we could generate artificial speedup by e.g. allowing only four instead of six levels of refinement which would lead to many more lowest-level cells on which the speedup is gained. Additionally, we employ robust AMR indicator setups to avoid a too strong influence of the inevitably different solutions of the different methods on the AMR process and thus on the mesh and computational cost.

7.1. Hyperbolic-parabolic equations

The presence of viscous fluxes makes the P-ERK schemes less vulnerable to oscillations due to monotonicity violations and are thus, at least in the convection-dominated regime, a suitable choice for the integration of such systems. In fact, for the hyperbolic-parabolic equations considered here, the $R = 3$ P-ERK schemes can be run with the same CFL number as the optimized single P-ERK method. The hyperbolic-parabolic problems are discretized using the DGSEM [67,68] where the viscous fluxes are treated using the Bassi-Rebay 1 (BR1) scheme [84,85]. The admissible timestep is in every case computed based on the inviscid CFL condition (3.1), (3.2) where we ensured that the inviscid terms are indeed limiting the timestep, i.e., not the viscous terms, cf. (3.3).

7.1.1. Periodic shear-layer

This example is selected to showcase the potential speed up from using the P-ERK schemes. In contrast to the majority of the other examples shown in this work, the finest cells make up only for a small fraction of the entire grid. In particular, the mesh is refined only in confined regions where the roll-up of emerging vortices is observed, keeping the majority of the domain relatively coarse.

Following the configuration described in [86], we set

$$\mathbf{u}_{\text{primal}}(t_0 = 0, x, y) = \begin{pmatrix} \rho \\ v_x \\ v_y \\ p \end{pmatrix} = \begin{pmatrix} 1 \\ \begin{cases} \tanh(\kappa(y - 0.25)) & y \leq 0.5 \\ \tanh(\kappa(0.75 - y)) & y > 0.5 \end{cases} \\ \delta \sin(2\pi(x + 0.25)) \\ c^2 \rho / \gamma \end{pmatrix} \quad (7.1)$$

with flow parameters $\kappa = 80$, $\delta = 0.05$, $c_0 = 10$, $\text{Ma}_0 = 0.1$, $\text{Re}_0 = 2.5 \cdot 10^4$, $\text{Pr} = 0.72$ and run the simulation until $t_f = 1.2$ as in [86]. In terms of spatial discretization, local polynomials of degree $k = 3$ with HLLC surface-flux and flux differencing for the entropy-stable volume flux by Ranocha [87] are employed to discretize the spatial derivatives. In contrast to the subsequent examples the subcell shock-capturing [72] is not required to suppress any spurious oscillations. The domain $\Omega = [0, 1]^2$ is initially heavily underresolved with only $2^3 = 8$ cells per coordinate direction. To allow for a physically meaningful simulation that resolves the vortex build-up, AMR is employed based on the indicator by Löhner [88] with indicator variable v_x . The implementation in `Trixi.jl` of the indicator by Löhner follows the adaptations to Legendre-Gauss-Lobatto elements developed in the FLASH code [89]. In this simulation, cells are allowed to be refined up to six times where the finest level corresponds to a discretization with $2^9 = 512$ cells per dimension. The velocity component v_x and the computational mesh at $t = 0.8$ are displayed in Fig. 7.

For this setup, a $p = 3$ P-ERK scheme with $E = \{3, 4, 7\}$ stage evaluations is constructed, where the admissible timestep doubles for each family member. The mesh is allowed to change every $N_{\text{AMR}} = 20$ timesteps, which involves the recomputation of the

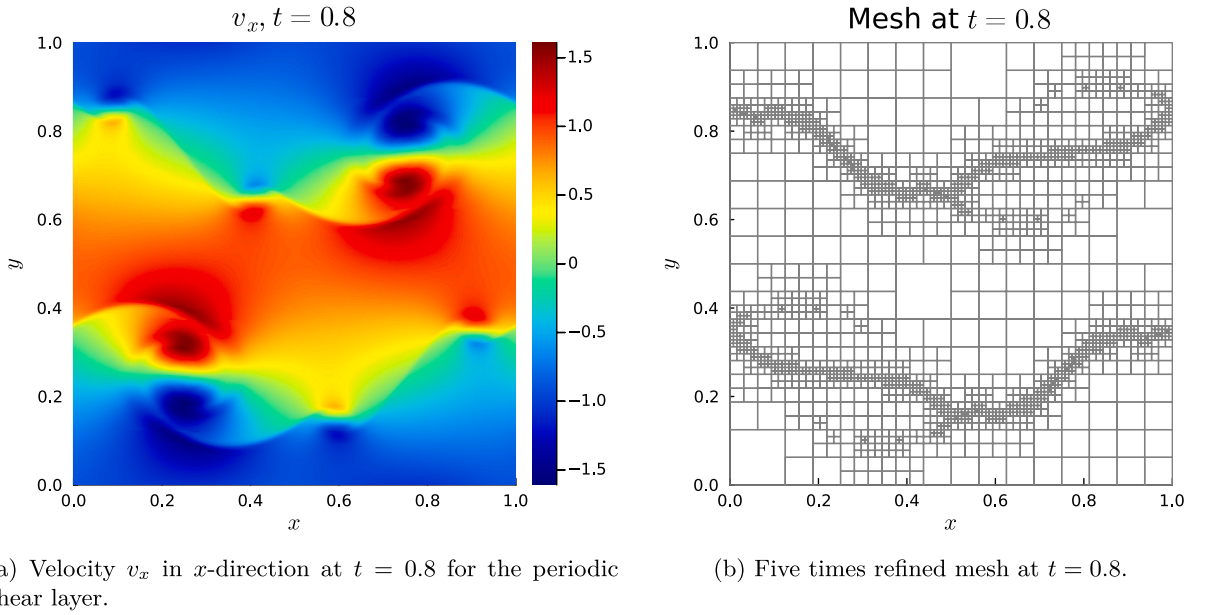


Fig. 7. Velocity v_x in x -direction at $t = 0.8$ for the doubly periodic double shear layer testcase (7.1) (Fig. 7a) and corresponding adaptively refined mesh Fig. 7b.

flagging-variables for the P-ERK integrator. In this case, the latter consumes 2.5% of the total runtime with is the overhead due to the partitioning. Comparing this to the speedup of the stand-alone P-ERK scheme, this overhead is well invested additional computational effort. Comparing the simulation with P-ERK_{3;{3,4,7}} to the standalone method P-ERK_{3;7}, the memory footprint increases in the mean by 20%.

The ratio of over ten runs averaged relative runtimes $\bar{\tau}/\bar{\tau}_{\text{P-ERK}_{3;{3,4,7}}}$ of the different schemes are tabulated in Table 7. Also, we provide the degree of optimality $\kappa = \tau_{\text{Opt}}/\bar{\tau}$ cf. (6.9) and the number of performed scalar RHS evaluations $N_{\text{RHS}}^{\text{Actual}}$ (6.1) for the different schemes. For each method we compare to, the AMR interval is adjusted such that approximately the same number of grid refinements is performed as for the simulation with the P-ERK family. This is roughly equivalent to performing the mesh refinement at the same physical timestamps. When comparing the different schemes it is important to note that, while the simulations are equipped with identical parameters (except for CFL number and AMR interval N_{AMR}), the AMR indicator which governs local mesh refinement or coarsening is based on the solution itself, which is inevitably different for the different methods. This becomes already apparent for this example as the DGLDD_{3;7} [80] scheme is slightly faster than the optimized P-ERK_{3;7} scheme which can be traced back to the fact that the finest level is reached later for the simulation with the DGLDD_{3;7} scheme, hence imposing the timestep restriction later.

Overall, the P-ERK_{3;{3,4,7}} is the fastest method for this testcase with all other methods being at least 40% slower. For this example we have a four times as large overall timestep for the P-ERK_{3;{3,4,7}} scheme compared to the SSP_{3;3} method while observing a speedup of factor 2.356. We observe that the saved number of RHS evaluations due to the optimized P-ERK scheme does not directly translate into a speedup of the same factor. For instance, the standalone P-ERK scheme requires roughly 76.2% more scalar RHS evaluations, but is only 46.8% slower. This is due to fact that Trixi.jl is for the standard DGSEM with a simple numerical flux memory-bound and thus saved computations do not translate proportional into runtime savings. In particular, data locality is in general lost for the partitioned approach, i.e., the coefficients $U^{(r)}$ are in general not stored in a contiguous memory block. Also, the data structures containing elements and interfaces are accessed in a non-sequential manner which leads to cache misses and thus reduced performance. In contrast, for the standard methods everything is processed as present in memory which explains the non-ideal performance of the partitioned approach.

Comparing this to the examples given in [2] where the authors observe speedups of about 4.3 across examples for an eight times as large timestep (compared to the SSP_{3;3}) we conclude that we have essentially the same ratio of speedup per timestep increase.

The four-stage method CKLLS_{3;4} [82] is not significantly faster than the canonical Shu-Osher SSP scheme, and is thus not considered in the following examples. This is possibly due to optimization for Navier-Stokes equations where diffusion plays a more dominant role than in the convection-dominated flows considered here.

7.1.2. Visco-resistive Orszag-Tang vortex

We consider the classic Orszag-Tang vortex [90] in the context of the visco-resistive Magnetohydrodynamics (VRMHD) equations [91]. The ideal magnetohydrodynamics (MHD) equations are implemented with divergence cleaning through a generalized Lagrangian multiplier (GLM) approach [92,93] and extended by viscous and resistive terms by means of the BR1 scheme [84].

As in [91] the domain is $\Omega = [0, 2\pi]^2$ and the gas parameters are set to $\gamma = \frac{5}{3}$, $\text{Pr} = 1$. Compared to [91] we reduce viscosity and resistivity by one order of magnitude to $\mu = \nu = 10^{-3}$ to stay convection-dominated (cf. (3.3)) even in the finest parts of the mesh. The initial condition is given by [91]

Table 7
 Runtimes, optimality fraction (6.9), and number of scalar RHS evaluations (6.1) of different third-order ODE solvers compared to the optimized $p = 3, S = \{3, 4, 7\}$ P-ERK scheme for the periodic shear-layer.

Method	$\bar{\tau}/\tau_{\text{P-ERK}(3,4,7)}$	τ_{opt}/τ	$N_{\text{RHS}}^{\text{Actual}}$
P-ERK _{3;{3,4,7}}	1.0	68.8%	$5.26 \cdot 10^9$
P-ERK _{3;7}	1.468	46.9%	$9.27 \cdot 10^9$
SSP _{3;3}	2.356	29.2%	$1.42 \cdot 10^{10}$
DGLDD _{3;7} [80]	1.412	48.8%	$8.53 \cdot 10^9$
PKD3S _{3;5} [81]	1.854	37.1%	$1.07 \cdot 10^{10}$
RDPKFSAL _{3;5} [83]	2.171	31.7%	$1.30 \cdot 10^{10}$
CKLLS _{3;4} [82]	2.350	29.3%	$1.45 \cdot 10^{10}$

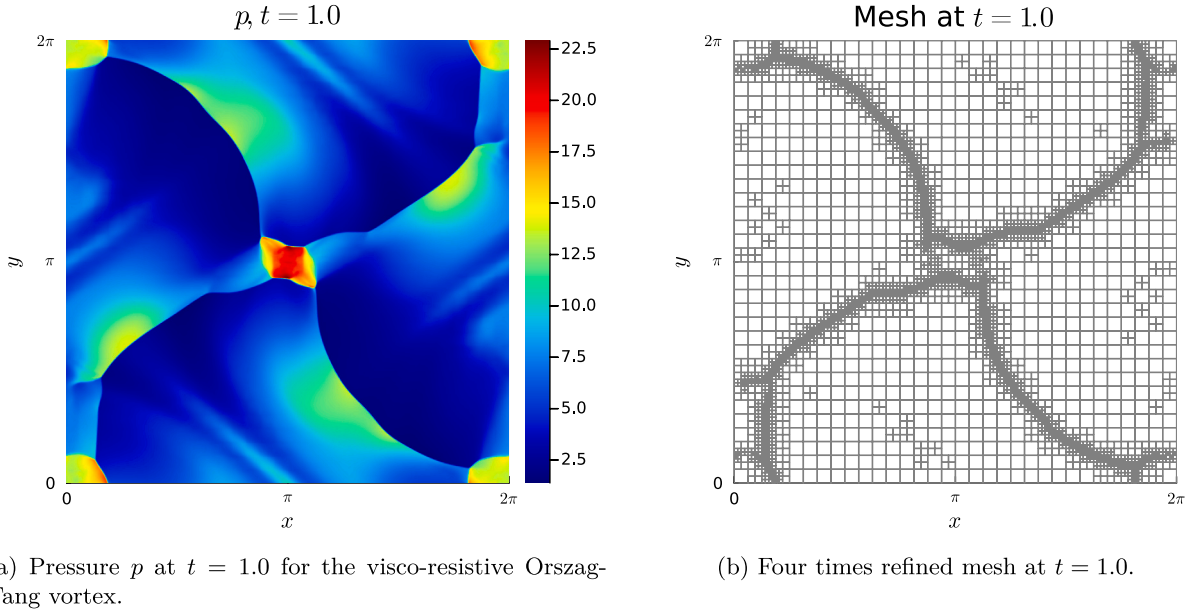


Fig. 8. Pressure p at $t = 1.0$ for the visco-resistive Orszag-Tang vortex (Fig. 8a) and corresponding adaptively refined mesh (Fig. 8b).

$$\mathbf{u}_{\text{primal}}(t_0 = 0, x, y) = \begin{pmatrix} \rho \\ v_x \\ v_y \\ v_z \\ p \\ B_x \\ B_y \\ B_z \end{pmatrix} = \begin{pmatrix} 1.0 \\ -2\sqrt{\pi} \sin(y) \\ 2\sqrt{\pi} \sin(x) \\ 0.0 \\ \frac{15}{4} + \frac{1}{4} \cos(4x) + \frac{4}{5} \cos(2x) \cos(y) - \cos(x) \cos(y) + \frac{1}{4} \cos(2y) \\ -\sin(2y) \\ \sin(2x) \\ 0.0 \end{pmatrix} \quad (7.2)$$

and the simulation is run until $t_f = 2.0$ as in [91]. Note that in the VRMHD equations employed in [91] the magnetic terms are nondimensionalized using viscous and resistive Lundquist numbers, which is different from [92] used in this work. The grid cells vary between $h = 2\pi/2^5$ and $h = 2\pi/2^9$, i.e., a mesh with five consecutive levels of refinement is employed.

To discretize the spatial derivatives a split-form DG method [70] is used. The surface-flux itself is split up into conserved terms which are discretized using the Rusanov/Local Lax-Friedrichs flux and a non-conserved part for which a specialized flux is employed [94,95]. Also for the volume flux/integral the conservative and non-conserved terms are treated differently. For this, we use the central flux for the conserved terms alongside the flux for the nonconservative Powell term [94]. As AMR indicator we employ the shock-capturing indicator developed in [72] based on density-pressure $\rho \cdot p$. This particular example benefits heavily from AMR since the configuration leads to initially sharp structures cf. Fig. 8 requiring accurate resolution which are then, due to the viscous effects, damped out over time with less need for high resolution.

For time discretization, we employ the three-level $p = 3, E = \{3, 4, 6\}$ optimized P-ERK family and compare it against the standalone $S = 6$ P-ERK scheme, the SSP_{3;3} method, DGLDD_{3;7}, and PKD3S_{3;5} with runtimes provided in Table 8. As the ideal GLM-MHD equations require a CFL-based timestep for the divergence cleaning mechanism, we also supply a CFL number to RDPKFSAL_{3;5} instead of an adaptive error-controlled timestep. As for the other cases, the P-ERK family is the fastest method among the considered ones. We observe that linear/absolute stability governs in this case the maximum admissible timestep as both the family as well as

Table 8
 Runtimes, optimality fraction (6.9), and number of scalar RHS evaluations (6.1) of different third-order integrators compared to the optimized $p = 3, S = \{3, 4, 6\}$ P-ERK integrator for the visco-resistive Orszag-Tang vortex.

Method	$\tau/\tau_{\text{P-ERK}_{3;\{3,4,6\}}}$	τ_{opt}/τ	$N_{\text{RHS}}^{\text{Actual}}$
P-ERK _{3;\{3,4,6\}}	1.0	58.2%	$1.44 \cdot 10^{10}$
P-ERK _{3;6}	1.248	46.6%	$2.07 \cdot 10^{10}$
SSP _{3;3}	2.005	29.0%	$3.28 \cdot 10^{10}$
DGLDD _{3;7} [80]	1.575	37.0%	$2.55 \cdot 10^{10}$
PKD3S _{3;5} [81]	1.717	33.9%	$2.72 \cdot 10^{10}$
RDPKFSAL _{3;5} [83]	1.649	35.3%	$2.72 \cdot 10^{10}$

the standalone optimized P-ERK schemes outperform the other schemes. Furthermore, the SSP_{3;3} scheme is the slowest Runge-Kutta method, which matches the expectations as this scheme is not optimized at all for a special class of semidiscretizations, in contrast to the other methods. As for the previous example, we obtain for a four times larger stable timestep for the P-ERK_{3;\{3,4,6\}} scheme compared to the SSP_{3;3} a speedup by a factor of 2.005 which is again the same ratio of speedup per timestep increase demonstrated in the examples in [2]. In terms of performance, there are $3.92 \cdot 10^9$ excess scalar RHS evaluations which lead to an overall 1.718 slower-than-optimal P-ERK scheme, i.e., for this case a significant speed-up would be possible if a full range of optimized methods without monotonicity violation could be employed. The update of the partition identifiers amounts to 1.0% of the total runtime, that is, the overhead due to the partitioning is again negligible. The memory requirements of the P-ERK_{3;\{3,4,6\}} scheme are on average 28% larger than for the standalone P-ERK_{3;6} scheme. As a sidenote, we note that despite having the same number of scalar RHS evaluations, the RDPKFSAL_{3;5} scheme is slightly faster than the PKD3S_{3;5} scheme which showcases the relevance of implementation of the timestepping algorithm itself.

7.1.3. 3D Taylor-Green vortex

The three-dimensional Taylor-Green vortex is a well-known reference problem showcasing emerging turbulence from a simple initial condition which is followed by the decay of the turbulent structures accompanied by kinetic energy dissipation [96,97]. The initial condition in primitive variables is given by

$$\mathbf{u}_{\text{prim}}(t_0 = 0, x, y, z) = \begin{pmatrix} \rho \\ v_x \\ v_y \\ v_z \\ p \end{pmatrix} = \begin{pmatrix} 1 \\ \sin(x) \cos(y) \cos(z) \\ -\cos(x) \sin(y) \cos(z) \\ 0 \\ p_0 + \frac{1}{16} \rho \left((\cos(2x) + \cos(2y)) (2 + \cos(2z)) \right) \end{pmatrix}, \tag{7.3}$$

where $p_0 = \frac{\rho}{M^2 \gamma}$ with Mach number $M = 0.1$ as in [98]. The Prandtl and Reynolds number are $\text{Pr} = 0.72$ and $\text{Re} = 1600$, and the compressible Navier-Stokes equations are simulated on $\Omega = [0, 2\pi]^3$ equipped with periodic boundaries until $t_f = 20.0$.

In terms of spatial discretization, we employ solution polynomials of degree $k = 2$, HLLC surface flux [71], and an entropy-conservative volume flux [74,70]. We employ a simple AMR indicator based on the enstrophy

$$e := 0.5 \rho \boldsymbol{\omega} \cdot \boldsymbol{\omega}, \tag{7.4}$$

which refines cells once a specified threshold is exceeded. Doing so, we expect the mesh to be refined in regions where the vorticity $\boldsymbol{\omega} := \nabla \times \mathbf{v}$ is high, i.e., where turbulent structures are emerging. The initial condition is discretized using 16^3 cells which are allowed to be two times refined corresponding to a smallest cell size of $h = \frac{2\pi}{64}$. While the Taylor-Green vortex is certainly not a classic use case for AMR it allows us to examine the cost of the dynamic re-assignment of cells to partitions for a three-dimensional problem.

In terms of the temporal integrators, we employ the three-level $p = 3, E = \{3, 4, 6\}$ optimized P-ERK family and compare it against the standalone $S = 6$ P-ERK scheme, the SSP_{3;3} method, DGLDD_{3;7} [80], PKD3S_{3;5} [81], and RDPKFSAL_{3;5} [83] with runtimes provided in Table 9.

When comparing the runtimes of the different schemes in Table 9 we observe that the P-ERK_{3;\{3,4,6\}} scheme is again the fastest method among the considered ones, although the standalone $S = 6$ P-ERK scheme is only about 4% slower. This is not surprising, as for the majority of timesteps most cells are completely refined and thus only very little speed-up may be gained. This is also reflected by the speedup that could be gained from a hypothetical ideally scaling set of methods. The $5.44 \cdot 10^9$ excess scalar RHS evaluations lead only to an overall 1.034 slower-than-optimal P-ERK scheme. We included this example for instructive purposes since it showcases that also (at least for small) 3D problems the overhead of the re-computation of the partitioning datastructures does not exceed speedup gained. In particular, only 0.3% of the total runtime is spent on the construction of the flagging variables after mesh changes. In this case, the memory footprint of the P-ERK_{3;\{3,4,6\}} scheme is on average only 7% larger than the standalone P-ERK_{3;6} scheme.

Table 9
 Runtimes, optimality fraction (6.9), and number of scalar RHS evaluations (6.1) of different third-order integrators compared to the optimized $p = 3, S = \{3, 4, 6\}$ P-ERK integrator for the Taylor-Green vortex.

Method	$\bar{\tau}/\tau_{\text{P-ERK}_{3;\{3,4,6\}}}$	$\tau_{\text{opt}}/\bar{\tau}$	$N_{\text{RHS}}^{\text{Actual}}$
P-ERK _{3;\{3,4,6\}}	1.0	96.8%	$1.25 \cdot 10^{11}$
P-ERK _{3;6}	1.068	90.6%	$1.42 \cdot 10^{11}$
SSP _{3;3}	1.768	54.7%	$2.29 \cdot 10^{11}$
DGLDD _{3;7} [80]	1.250	77.4%	$1.62 \cdot 10^{11}$
PKD3S _{3;5} [81]	1.622	59.7%	$1.91 \cdot 10^{11}$
RDPKFSAL _{3;5} [83]	1.425	67.9%	$1.82 \cdot 10^{11}$

7.2. Hyperbolic equations

In addition to the isentropic vortex test case used for verification in Section 5 we present additional applications to inviscid, hyperbolic problems. In particular, this is the first time the P-ERK schemes are applied to inviscid problems, apart from the isentropic vortex used in convergence tests. For inviscid problems the issues concerning nonlinear stability discussed in Section 3.2 resurface and a simulation with three-level ($R = 3$) P-ERK schemes demands in general a reduction of the timestep compared to the optimized standalone scheme. Recall that for hyperbolic-parabolic problems the CFL number did not have to be reduced for simulations with P-ERK schemes compared to the standalone scheme. This clearly impairs the performance of the P-ERK schemes for inviscid problems and correspondingly, we do not see the same speedup as for the viscous problems.

7.2.1. Kelvin-Helmholtz instability

The inviscid Kelvin-Helmholtz instability is a suitable testcase for the AMR-based dynamic ODE partitioning developed in this work. The initial condition can be accurately represented with a quite coarse discretization while the emerging instability requires a significantly finer resolution. We use the setup from [66] where the initial condition is given by

$$\mathbf{u}_{\text{primal}}(t_0 = 0, x, y) = \begin{pmatrix} \rho \\ v_x \\ v_y \\ p \end{pmatrix} = \begin{pmatrix} 0.5 + 0.75 B(y) \\ 0.5(B(y) - 1) \\ 0.1 \sin(2\pi x) \\ 1 \end{pmatrix} \tag{7.5}$$

with $B(y) = \tanh(15y + 7.5) - \tanh(15y - 7.5)$ which corresponds to Mach number $\text{Ma} \leq 0.6$. The compressible Euler equations are discretized using the DGSEM with $k = 3$ local polynomials, HLLC surface-flux [71], subcell shock-capturing [72] to avoid spurious oscillations and entropy-conserving volume flux [74,73]. Initially, the domain $\Omega = [-1, 1]^2$ is discretized with $2^4 \times 2^4$ cells which are allowed to be refined five times, i.e., the finest cells may have edge length $h = 2^{-8}$. The indicator for AMR coincides in this case with the shock-capturing indicator [72] and is based on the density ρ . For time-integration, the $p = 3, E = \{4, 6, 11\}$ P-ERK family is employed where the admissible timestep doubles with increasing stage evaluations E . In contrast to the viscous flows from the previous section, the CFL number of the P-ERK_{3;\{4,6,11\}} scheme has to be reduced to 70% of the CFL number of the P-ERK_{3;11} scheme. This reduction is a consequence of the troubling nonlinear stability properties of the P-ERK schemes at scheme boundaries, cf. Section 3.2.

The computational mesh and solution at final time $t_f = 3.0$ are displayed in Fig. 9. As for the viscous problems, we compare the P-ERK family to the stand-alone $p = 3, S = 11$ scheme and the SSP_{3;3}, DGLDD_{3;7}, PKD3S_{3;5}, and RDPKFSAL_{3;5} schemes. The AMR-interval N_{AMR} (cf. (6.2)) is adjusted according to the timesteps undertaken by the different methods to have an approximately equal AMR load. The runtimes τ are averaged over ten runs on a shared-memory multicore thread-parallelized machine and presented in Table 10. The P-ERK_{3;\{4,6,11\}} is in this case the most efficient integrator with minimal runtime. Notably, the DGLDD_{3;7} method is only slightly slower and even faster than the optimized P-ERK_{3;11} scheme. This is attributed to favorable nonlinear stability properties (low-dispersion, low-dissipation), which are for this example (and for the inviscid problems in this section overall) more relevant than optimized absolute stability. Similarly, the SSP_{3;3} scheme performs also better than the PKD3S_{3;5} and RDPKFSAL_{3;5} schemes despite that these are optimized for DG methods and computational fluid dynamics, respectively. Note that this was not the case for the previous hyperbolic-parabolic examples, i.e., the SSP_{3;3} scheme did not outperform any of the optimized schemes for viscous problems.

The update of the identifier variables required for the partitioning amounts in this case to about 0.4% of the total runtime $\tau_{\text{P-ERK}}$. The memory footprint of the P-ERK_{3;\{4,6,11\}} scheme is on average 23% larger than the standalone P-ERK_{3;11} scheme. To assess the optimality, or, in other words, the room for further improvements of using, e.g., a P-ERK family with more members and higher stages, we compute the excess scalar RHS evaluations according to (6.2). For this example, there are $3.80 \cdot 10^8$ additional scalar RHS evaluations per conserved variable which increase the runtime of the employed P-ERK_{3;\{4,6,11\}} scheme to 1.035 of the optimal one, neglecting overhead due to additional levels. This value results from the fact that the vast majority of the cells is distributed among the finest three levels. In particular, at final time $t_f = 3.0$ the three non-optimal integrated levels form only 6% of the entire number of cells, cf. Fig. 9b. In other words, the majority of the cells are integrated optimally with results in a speed-up of 30% compared to the standalone P-ERK_{3;11} scheme.

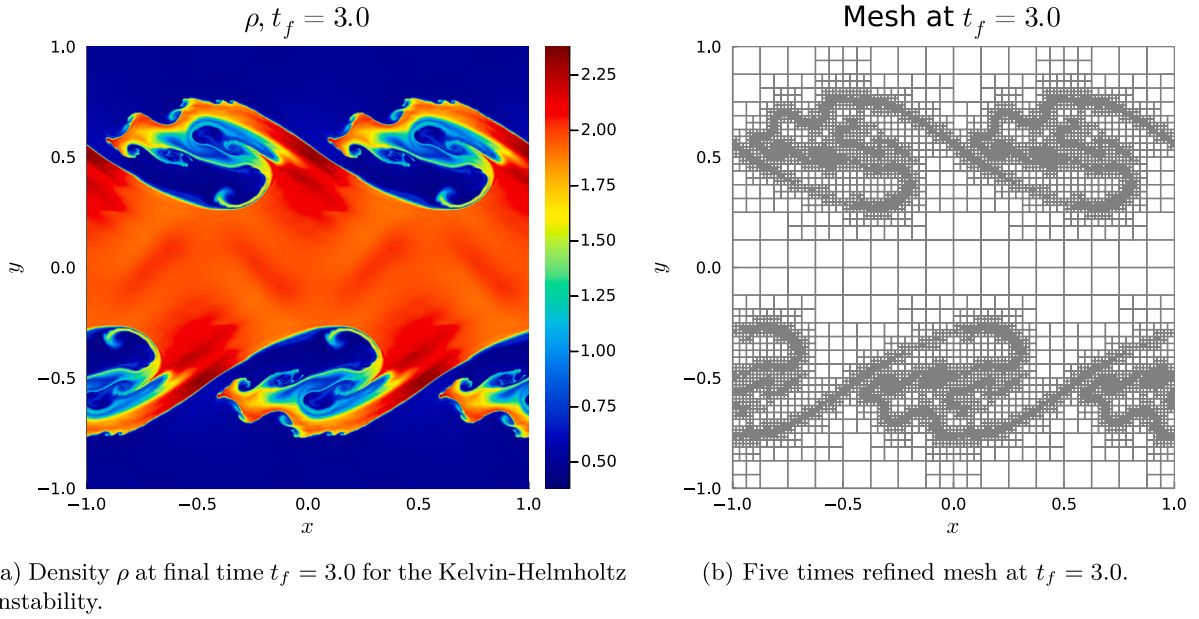


Fig. 9. Density ρ at final time $t_f = 3.0$ for the Kelvin-Helmholtz instability (7.5) (Fig. 9a) and corresponding adaptively refined mesh Fig. 9b.

Table 10

Runtimes, optimality fraction (6.9), and number of scalar RHS evaluations (6.1) of different third-order integrators compared to the optimized $p = 3, S = \{4, 6, 11\}$ P-ERK integrator for the inviscid Kelvin-Helmholtz instability.

Method	$\tau/\tau_{\text{P-ERK}_{3;\{4,6,11\}}}$	τ_{opt}/τ	$N_{\text{RHS}}^{\text{Actual}}$
P-ERK _{3;\{4,6,11\}}	1.0	92.4%	$3.78 \cdot 10^9$
P-ERK _{3;11}	1.311	70.5%	$5.30 \cdot 10^9$
SSP _{3;3}	1.426	64.8%	$6.62 \cdot 10^9$
DGLDD _{3;7} [80]	1.169	79.1%	$5.19 \cdot 10^9$
PKD3S _{3;5} [81]	1.866	49.4%	$8.36 \cdot 10^9$
RDPKFSAL _{3;5} [83]	1.659	55.7%	$6.54 \cdot 10^9$

When comparing the number of performed scalar RHS evaluations $N_{\text{RHS}}^{\text{Actual}}$ to the runtime τ we observe not exactly a linear correspondence. This is due to two factors: First, the AMR contributes a fixed-cost to the overall measured runtime τ which does not decrease with the number of scalar RHS evaluations. Second, and more significantly, the employed shock-capturing mechanism [72] leads to different computational costs for different cells, as for some a plain DGSEM reconstruction is employed while for others a blending of DGSEM and subcell Finite Volume reconstruction is necessary. The influence of this can clearly be seen by comparing the runtimes of the standalone P-ERK_{3;11} scheme and the DGLDD_{3;7} scheme which have roughly the same number of scalar RHS evaluations. The DGLDD_{3;7} scheme is, however, over-proportionally faster than the P-ERK_{3;11} scheme, which is attributed to the lower computational cost of the employed shock-capturing mechanism. Similarly, the speedup of the P-ERK_{3;\{4,6,11\}} scheme compared to the P-ERK_{3;11} scheme is not exactly proportional to the number of scalar RHS evaluations, again due to increased computational cost of the employed shock-capturing mechanism. This can be seen as a price of the poor nonlinear stability properties of the P-ERK schemes which cause more cells to rely on a blend of DGSEM and subcell Finite Volume reconstruction.

7.2.2. Rayleigh-Taylor instability

Another classic example is the Rayleigh-Taylor instability, where a heavier fluid gets pushed into a lighter one, leading to complicated flow structures beneath the characteristic mushroom cap. The configuration follows [99] with reflective boundary conditions on the boundaries in x -direction and weakly enforced Dirichlet boundary conditions in y -direction. The domain $\Omega = [0, 0.25] \times [0, 1]$ is initially discretized with $N_x = 12, N_y = 4 \cdot N_x = 48$ third-order ($k = 3$) elements. The spatial discretization is analogous to the Kelvin-Helmholtz instability from Section 7.2.1 and the simulation is run up to $t_f = 1.95$ as in [99]. Following [99], we set the ratio of specific heats $\gamma = 5/3$ and the initial state to

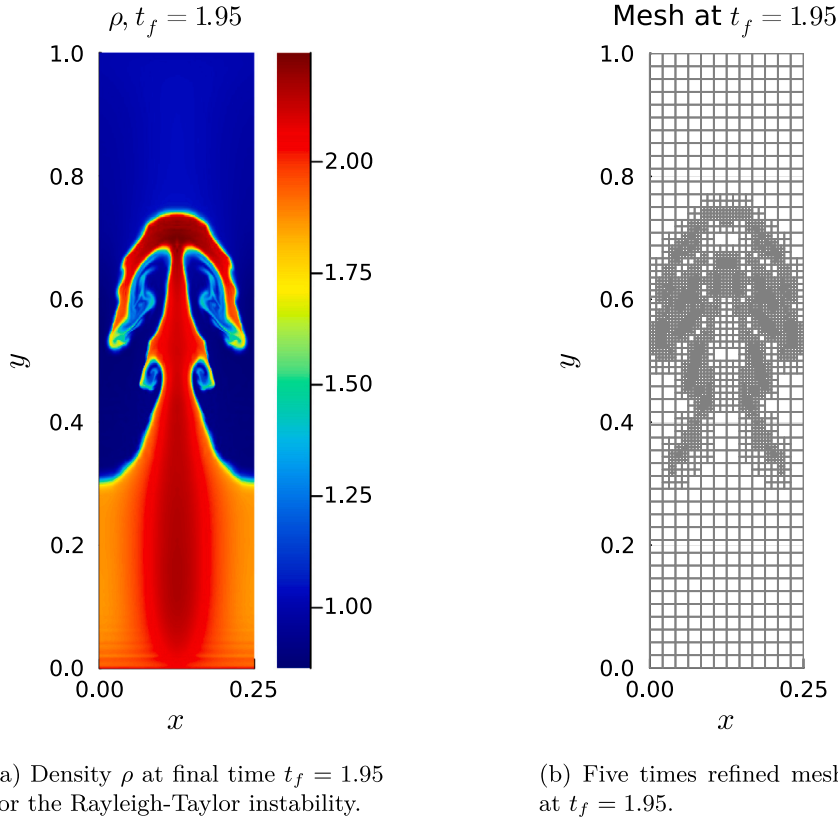


Fig. 10. Density ρ at final time $t_f = 1.95$ for the Rayleigh-Taylor instability (7.6) (Fig. 10a) and corresponding adaptively refined mesh Fig. 10b.

$$\mathbf{u}_{\text{prim}}(t_0 = 0, x, y) = \begin{pmatrix} \rho \\ v_x \\ v_y \\ p \end{pmatrix} = \begin{pmatrix} \begin{cases} 2.0 & y \leq 0.5 \\ 1.0 & y > 0.5 \end{cases} \\ 0.0 \\ ck \cos(8\pi x) \\ \begin{cases} 2.0 \cdot y + 1.0 & y \leq 0.5 \\ y + 1.5 & y > 0.5 \end{cases} \end{pmatrix} \quad (7.6)$$

where $c = \sqrt{\frac{\gamma p}{\rho}}$ denotes the speed of sound and $k = -0.025$ quantifies the strength of the initial perturbation.

Each cell is allowed to be refined at most six times, i.e., the theoretically maximum resolution is 768×3072 with $h = 1/3072$. For time integration we employ a $p = 3, E = \{3, 4, 6\}$ scheme where the maximum admissible timestep doubles for each family member. Thus, the finest three levels can be integrated optimally and only the remaining levels are not efficiently integrated. Compared to the standalone P-ERK_{3;6} scheme the CFL number of the P-ERK_{3;{3,4,6}} is reduced to 96% of the optimal value. We observe, however, that the P-ERK_{3;4} has a higher stage-normalized CFL number CFL/E , rendering it more efficient. Consequently, we employ P-ERK_{3;4} for the runtime comparisons in Table 11. The computation of the partitioning variables consumes about 4.4% of the total runtime. This increase compared to the other examples is due to the usage of the in principle unstructured `p4est` mesh which requires the computation of the minimum edge length for each cell. This is additional effort compared to the necessarily square/cubic cells forming the `TreeMesh`, where each cells' characteristic length is fully determined by the position of the associated node in the underlying tree datastructure. In terms of additional memory consumption, the P-ERK_{3;{3,4,6}} scheme requires on average 34% more memory than the standalone P-ERK_{3;4} scheme.

As for the Kelvin-Helmholtz instability, the P-ERK_{3;{3,4,6}} scheme is the fastest method among the considered ones, see Table 11 followed by the standalone optimized P-ERK_{3;4} scheme. Notably, the usually well-performing DGLDD_{3;7} [80] scheme is in this case relatively slow. As for the Kelvin-Helmholtz instability, we observe a discrepancy between the number of scalar RHS evaluations $N_{\text{RHS}}^{\text{Actual}}$ and the runtime τ . For instance, the P-ERK_{3;{3,4,6}} scheme has roughly half the number of scalar RHS evaluations of the PKD3S_{3;5} scheme. However, the P-ERK_{3;{3,4,6}} scheme is only about 1.4 times faster than the PKD3S_{3;5} scheme. This is attributed to more cells relying on the expensive subcell shock-capturing mechanism [72] for the P-ERK_{3;{3,4,6}} scheme than for the PKD3S_{3;5} scheme. Additionally, the AMR fixed-costs which are for the P-ERK_{3;{3,4,6}} scheme about 23% of the total runtime, do not decrease if $N_{\text{RHS}}^{\text{Actual}}$ is reduced.

Table 11
 Runtimes, optimality fraction (6.9), and number of scalar RHS evaluations (6.1) of different third-order integrators compared to the optimized $p = 3, S = \{3, 4, 6\}$ P-ERK integrator for the inviscid Rayleigh-Taylor instability.

Method	$\bar{\tau}/\tau_{\text{P-ERK}_{3;\{3,4,6\}}}$	$\tau_{\text{opt}}/\bar{\tau}$	$N_{\text{RHS}}^{\text{Actual}}$
P-ERK _{3;\{3,4,6\}}	1.0	82.5%	$1.25 \cdot 10^{10}$
P-ERK _{3;4}	1.126	73.3%	$1.84 \cdot 10^{10}$
SSP _{3;3}	1.205	68.5%	$2.04 \cdot 10^{10}$
DGLDD _{3;7} [80]	1.751	47.1%	$3.32 \cdot 10^{10}$
PKD3S _{3;5} [81]	1.402	58.9%	$2.40 \cdot 10^{10}$
RDPKFSAL _{3;5} [83]	1.292	63.9%	$1.44 \cdot 10^{10}$

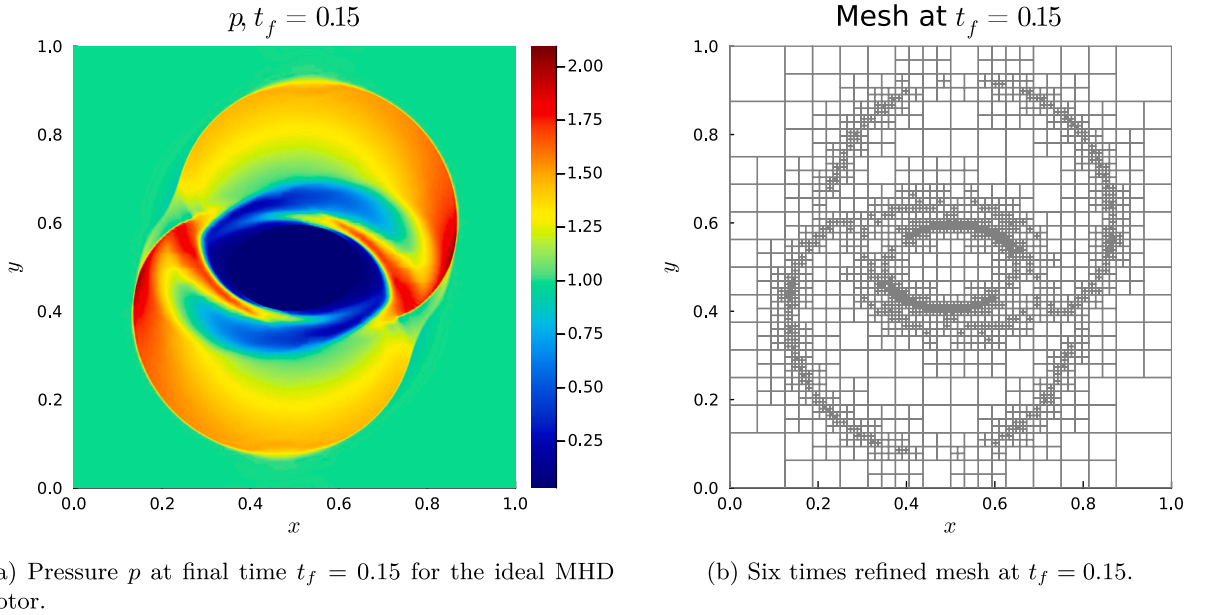


Fig. 11. Pressure p at final time $t_f = 0.15$ for the MHD rotor [100] (Fig. 11a) and corresponding adaptively refined mesh Fig. 11b.

7.2.3. MHD rotor

In order to go beyond the Euler equations of gas dynamics we consider the ideal MHD equations with divergence cleaning through a GLM approach [92,93]. A classic testcase thereof is the MHD rotor [100] which we also study here. The somewhat lengthy initial condition is provided in Appendix B.

In terms of spatial discretization, we employ $k = 4$ local polynomials with the fluxes being discretized analogous to the inviscid fluxes from the visco-resistive Orszag-Tang vortex studied in Section 7.1.2. Again, for AMR indication the shock-capturing mechanism from [72] based on density-pressure $\rho \cdot p$ is employed. The pressure distribution at final time $t_f = 0.15$ is displayed in Fig. 11 together with the corresponding mesh. For this example the $p = 3, E = \{4, 6, 10\}$ scheme is a suitable choice for the P-ERK family where the maximum admissible timestep doubles for each family member. Similar to the Rayleigh-Taylor instability, the CFL number of the P-ERK_{3;\{4,6,10\}} scheme has not to be significantly reduced compared to the standalone P-ERK_{3;10} scheme. We compare the P-ERK family to the best performing stand-alone P-ERK_{3;6} optimized method, SSP_{3;3}, DGLDD_{3;7}, PKD3S_{3;5}, and RDPKFSAL_{3;5} (with CFL-determined timestep). The results are tabulated in Table 12. As for the previous examples, the P-ERK family is the fastest method among the considered ones with the standalone $S = 6$ P-ERK scheme being about 10% slower. The runtimes for the PKD3S_{3;5} and RDPKFSAL_{3;5} scheme are almost identical which is not surprising since the scheme RDPKFSAL_{3;5} developed in [83] is based on the PKD3S_{3;5} scheme [81]. For this case, a fully optimal P-ERK scheme could save $7.71 \cdot 10^7$ scalar RHS calls per field. Correspondingly, the here employed three-level P-ERK scheme is 1.135 times slower than a hypothetical optimal P-ERK family, again neglecting potential overhead due to the usage of additional levels. The dynamic updating of the partitioning indicator variables amounts in this case to 0.8% of the total runtime. The memory footprint of the P-ERK_{3;\{4,6,10\}} scheme is on average 11% larger than the standalone P-ERK_{3;6} scheme. As for the other two inviscid examples, the decrease in runtime τ of the P-ERK schemes is not proportional to the number of scalar RHS evaluations $N_{\text{RHS}}^{\text{Actual}}$. The reason behind this is that for the P-ERK_{3;\{4,6,10\}} scheme a larger number of cells require the subcell stabilizing mechanism [72] compared to the standalone schemes. This is due to the poor nonlinear stability properties of the P-ERK schemes and thus an increased computational cost per RHS evaluation.

Table 12
Runtimes, optimality fraction (6.9), and number of scalar RHS evaluations (6.1) of different third-order integrators compared to the optimized $p = 3, S = \{4, 6, 10\}$ P-ERK integrator for the ideal MHD rotor.

Method	$\bar{\tau}/\tau_{\text{P-ERK}_{3,\{4,6,10\}}}$	$\tau_{\text{opt}}/\bar{\tau}$	$N_{\text{RHS}}^{\text{Actual}}$
P-ERK _{3,\{4,6,10\}}	1.0	88.1%	$2.35 \cdot 10^9$
P-ERK _{3,6}	1.097	80.6%	$3.23 \cdot 10^9$
SSP _{3,3}	1.492	57.9%	$4.43 \cdot 10^9$
DGLDD _{3,7} [80]	1.148	76.7%	$3.51 \cdot 10^9$
PKD3S _{3,5} [81]	1.195	73.7%	$3.56 \cdot 10^9$
RDPKFSAL _{3,5} [83]	1.192	73.9%	$3.56 \cdot 10^9$

8. Conclusions

In this work, we applied the Paired-Explicit Runge-Kutta (P-ERK) schemes by Vermeire et al. [1,2] to convection-dominated, compressible flows with adaptive mesh refinement. For a range of hyperbolic and hyperbolic-parabolic examples we demonstrate speed-up compared to state-of-the-art optimized Runge-Kutta methods. In particular, the P-ERK outperform for every considered case the non-multirate schemes by factors up to 2.4 for viscous problems and 1.8 for inviscid problems.

Additional data structures are required for the partitioning, which can for all considered examples be set up in less of 5% of the total runtime. The memory footprint of partitioning variables leads to 7 – 34% additional memory consumption compared to the standalone schemes. For the shown applications a public reproducibility repository is provided [27].

In general, the P-ERK schemes perform better for PDEs involving viscous fluxes compared to purely inviscid problems. This is due to the fact that the P-ERK schemes are in general not monotonicity-preserving and thus require the stabilizing shock-capturing strategy [72] on a larger number of cells than the standalone schemes. We investigated the mechanism behind this behavior by conducting a thorough nonlinear stability analysis of the fully discrete, partitioned system. In particular, we show that the usage of the P-ERK schemes leads in general to a loss of the monotonicity properties of the spatial discretization. More precise, the monotonicity properties are violated at the intersection of different schemes, i.e., where the partitioning is performed. This restricts the P-ERK schemes to configurations with limited difference in stage evaluations among partitions for an efficient time integration.

Future work will focus on the construction of fourth-order consistent P-ERK schemes and the application to unstructured meshes with adaptive mesh refinement.

Code availability & reproducibility

We provide a reproducibility repository [27] containing the source code and scripts and data to reproduce the results from Section 5 and Section 7.

CRediT authorship contribution statement

Daniel Doehring: Writing – original draft, Software, Methodology, Investigation, Formal analysis. **Michael Schlottke-Lakemper:** Writing – review & editing, Software, Funding acquisition, Conceptualization. **Gregor J. Gassner:** Writing – review & editing, Funding acquisition, Conceptualization. **Manuel Torrilhon:** Writing – review & editing, Supervision, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Daniel Doehring reports financial support was provided by German Research Foundation. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

We provide a reproducibility repository on Github that contains data and code. All other data will be provided as supplementary material.

Acknowledgements

Funding by German Research Foundation (DFG) under Research Unit FOR5409: “Structure-Preserving Numerical Methods for Bulk- and Interface-Coupling of Heterogeneous Models (SNUbic)” (grant #463312734). The authors thank Lambert Theisen for fruitful discussion concerning the iterative spectrum computation.

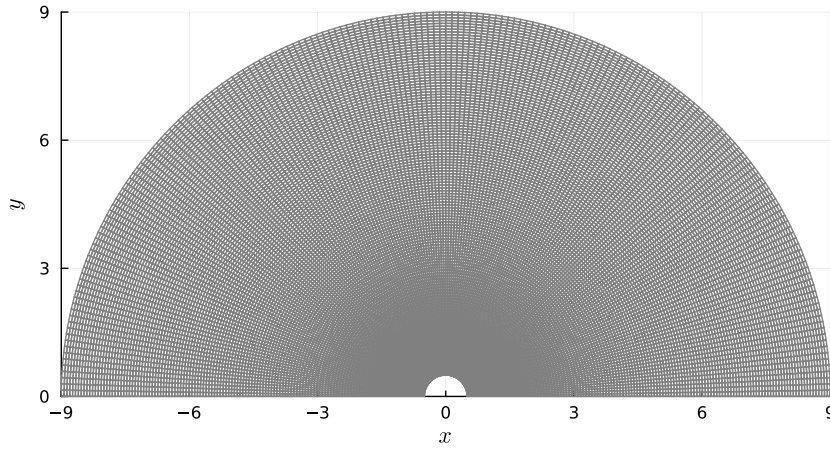


Fig. A.12. High-resolution, structured mesh for demonstration of working P-ERK schemes on smooth meshes.

Appendix A. P-ERK schemes on a smoothly refined, high-resolution mesh

In order to give further evidence for the nonlinear stability analysis conducted in Section 3.2, we present a testcase where the P-ERK schemes are applied to a smooth, high-resolution mesh. The refinement factor α of neighboring cells is for this example close to 1, i.e., the discontinuities in the mesh are substantially less pronounced than for the quad/octree-based AMR meshes. To this end, we construct a structured curved mesh by defining the mapping of the edges of the cartesian square $(\xi, \eta) \in [-1, 1]^2$ onto curved edges $e_i, i = 1, \dots, 4$ of the computational domain. In particular, we employ the mapping

$$e_1(\xi, \eta) = \begin{pmatrix} r_0 + 0.5(r_1 - r_0)(\xi + 1) \\ 0 \end{pmatrix} \quad (\text{A.1a})$$

$$e_2(\xi, \eta) = \begin{pmatrix} r_1 \cos(0.5\pi(\eta + 1)) \\ r_1 \sin(0.5\pi(\eta + 1)) \end{pmatrix} \quad (\text{A.1b})$$

$$e_3(\xi, \eta) = \begin{pmatrix} -r_0 - 0.5(r_1 - r_0)(\xi + 1) \\ 0 \end{pmatrix} \quad (\text{A.1c})$$

$$e_4(\xi, \eta) = \begin{pmatrix} r_0 \cos(0.5\pi(\eta + 1)) \\ r_0 \sin(0.5\pi(\eta + 1)) \end{pmatrix} \quad (\text{A.1d})$$

where the edges e_i of the $[-1, 1]^2$ cube are labeled counter-clockwise starting from the bottom $(\xi, \eta = 0)$ edge. This mapping corresponds to a half-circle with radius r_1 from which a half-circle with radius r_0 is subtracted. The mesh is discretized using $N_\xi = 192 \times N_\eta = 128$ cells and is displayed in Fig. A.12.

For this testcase we simulate the inviscid Euler equations of gas dynamics with $\gamma = 1.4$ with initial condition

$$\mathbf{u}_{\text{prim}}(t_0 = 0, x, y) = \begin{pmatrix} \rho \\ v_x \\ v_y \\ p \end{pmatrix} = \begin{pmatrix} 1.0 \\ 0.0 \\ 0.0 \\ 1.0 + 2 \frac{(x - 0.6r_1)^2 + y^2}{0.45^2} \end{pmatrix} \quad (\text{A.2})$$

which corresponds to a pressure perturbation with magnitude $p_{\text{max}} = 2$ located at the right-bottom of the half-circle, cf. Fig. A.13a.

In terms of spatial discretization we employ the DGSEM with $k = 2$ local polynomials, Harten-Lax-Van Leer (HLL) surface-flux and an entropy-conservative volume flux [74,70] with reflective boundaries. In particular, no techniques like shock capturing or positivity-preserving limiters are employed to suppress spurious oscillations.

For time-integration, second-order accurate stability polynomials with degrees $S = \{4, 6, 8, \dots, 16\}$ are optimized, based on which we construct a range of P-ERK schemes. The gridcells are sorted into R (number of P-ERK family members) bins based on their minimum edge length which govern the associated Runge-Kutta method. We test all combinations of P-ERK schemes with $E^{(1)} = 4$ minimum stage evaluations, i.e., from the standalone $S = 4$ scheme up to the $S = 16, E = \{4, 6, 8, \dots, 16\}$ family. For all combinations, the maximum admissible timestep is increased according to highest number of stage evaluations $E^{(R)}$. For this setup, across different P-ERK schemes, the stage-evaluation normalized CFL-number for stable simulation is essentially constant across P-ERK families, as shown in Table A.13. For the sake of completeness we also provide the speedup compared to the Shu-Osher $p = 3, S = 3$ scheme, the reference method from [1,2] in Table A.13.

For illustration, the final pressure distribution p at $t_f = 5.0$ is displayed in Fig. A.13 alongside the initial perturbation according to (A.2).

Table A.13

Normalized CFL numbers and speedup across different P-ERK schemes for the structured mesh testcase. Here, $E^{(1)} = 4$ is the base scheme utilized in all cases.

$E^{(R)}$	4	6	8	10	12	14	16
$\text{CFL} \cdot \frac{E^{(1)}}{E^{(R)}}$	0.95	0.97	0.97	0.94	0.94	0.94	0.93
$\frac{\tau_{\text{P-ERK}_{2,4,\dots,E^{(R)}}}}{\tau_{\text{SSP}_{3,3}}}$	1.26	1.64	1.90	2.06	2.22	2.37	2.45

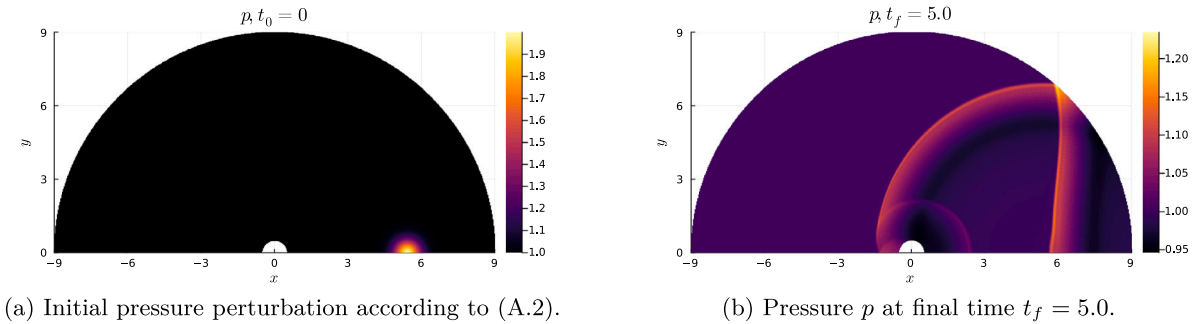


Fig. A.13. Initial pressure perturbation according to (A.2) (Fig. A.13a) and pressure p at final time $t_f = 5.0$ (Fig. A.13b).

We repeated this testcase for a significantly coarser mesh with $N_x = 48 \times N_y = 32$ cells and obtained qualitatively similar results. In particular, the normalized CFL number is again of similar value, with the largest value $\text{CFL} = 1.04$ for the P-ERK scheme with $E^{(R)} = 10$ and smallest value for $E^{(R)} = 16$ with $\text{CFL} = 0.95$.

This example demonstrates that the P-ERK schemes are a suitable multirate scheme, even for inviscid problems without stabilization. In particular, due to the smooth meshes, the discontinuity in the mesh is effectively eliminated and does not coincide with the discontinuity in the time integration method. This is the main difference to quad/octree-based AMR meshes where the discontinuity in the mesh coincides with the discontinuity in the time integration method. In this case, the mesh refinement factor is always $\alpha = 2$ and the difference in stage evaluations scales quadratically with the number of levels, leading to a discontinuity in methods for the finest levels.

Appendix B. Ideal MHD rotor: initial condition

For the simulation of the ideal MHD rotor we employ the initial condition from [100]. The radius r_0 of the spinning cylinder is set to $r_0 = 0.1$ while the medium is initially at rest for $r \geq r_1 = 0.115$ with radius $r(x, y) = \sqrt{(x - 0.5)^2 + (y - 0.5)^2}$. The magnetic field is set uniformly to $\mathbf{B}(t_0 = 0, x, y) = \left(\frac{5}{2\sqrt{\pi}} \quad 0 \quad 0 \right)$ as well as pressure $p(0, x, y) = 1$ and $v_z(0, x, y) = 0$. Inside the spinning cylinder, i.e., $r \leq r_0$ we set the density to $\rho(0, x, y) = 10$ and the velocities to $v_x(0, x, y) = -20(y - 0.5)$, $v_y(0, x, y) = 20(x - 0.5)$. In the transition from r_0 to r_1 we set the density to $\rho(0, x, y) = 1 + 9f(r(x, y))$ and the velocities to $v_x(0, x, y) = -20(y - 0.5)f(r(x, y))$, $v_y(0, x, y) = 20(x - 0.5)f(r(x, y))$ with $f(r(x, y)) := \frac{r_1 - r(x, y)}{r_1}$. Outside of r_1 , i.e., $r > r_1$ the medium is at rest $v_x(t_0, x, y) = 0$, $v_y(t_0, x, y) = 0$ and density is set to $\rho(0, x, y) = 1$.

References

- [1] B.C. Vermeire, Paired explicit Runge-Kutta schemes for stiff systems of equations, J. Comput. Phys. 393 (2019) 465–483, <https://doi.org/10.1016/j.jcp.2019.05.014>.
- [2] S.H. Nasab, B.C. Vermeire, Third-order paired explicit Runge-Kutta schemes for stiff systems of equations, J. Comput. Phys. 468 (2022) 111470, <https://doi.org/10.1016/j.jcp.2022.111470>.
- [3] E. Godlewski, P.-A. Raviart, Numerical Approximation of Hyperbolic Systems of Conservation Laws, 2 ed., Applied Mathematical Sciences, vol. 118, Springer Science & Business Media, 2013.
- [4] R.J. LeVeque, Finite Volume Methods for Hyperbolic Problems, Cambridge Texts in Applied Mathematics, vol. 31, Cambridge University Press, 2002.
- [5] M.J. Berger, J. Olinger, Adaptive mesh refinement for hyperbolic partial differential equations, J. Comput. Phys. 53 (1984) 484–512, [https://doi.org/10.1016/0021-9991\(84\)90073-1](https://doi.org/10.1016/0021-9991(84)90073-1).
- [6] B. Cockburn, C.-W. Shu, Runge–Kutta discontinuous Galerkin methods for convection-dominated problems, J. Sci. Comput. 16 (2001) 173–261, <https://doi.org/10.1023/A:1012873910884>.
- [7] R. Courant, K. Friedrichs, H. Lewy, On the partial difference equations of mathematical physics, IBM J. Res. Dev. 11 (1967) 215–234, <https://doi.org/10.1147/rd.112.0215>.
- [8] H. Bijl, M.H. Carpenter, V.N. Vatsa, C.A. Kennedy, Implicit time integration schemes for the unsteady compressible Navier–Stokes equations: laminar flow, J. Comput. Phys. 179 (2002) 313–329, <https://doi.org/10.1006/jcph.2002.7059>.
- [9] C. Yang, X.-C. Cai, A scalable fully implicit compressible Euler solver for mesoscale nonhydrostatic simulation of atmospheric flows, SIAM J. Sci. Comput. 36 (2014) S23–S47, <https://doi.org/10.1137/130919167>.
- [10] U.M. Ascher, S.J. Ruuth, B.T. Wetton, Implicit-explicit methods for time-dependent partial differential equations, SIAM J. Numer. Anal. 32 (1995) 797–823, <https://doi.org/10.1137/0732037>.

- [11] P. Rentrop, Partitioned Runge-Kutta methods with stiffness detection and stepsize control, *Numer. Math.* 47 (1985) 545–564, <https://doi.org/10.1007/BF01389456>.
- [12] J. Bruder, K. Strehmel, R. Weiner, Partitioned adaptive Runge-Kutta methods for the solution of nonstiff and stiff systems, *Numer. Math.* 52 (1988) 621–638, <https://doi.org/10.1007/BF01395815>.
- [13] M. Günther, A. Kvaerno, P. Rentrop, Multirate partitioned Runge-Kutta methods, *BIT Numer. Math.* 41 (2001) 504–514, <https://doi.org/10.1023/A:1021967112503>.
- [14] E.M. Constantinescu, A. Sandu, Multirate timestepping methods for hyperbolic conservation laws, *J. Sci. Comput.* 33 (2007) 239–278, <https://doi.org/10.1007/s10915-007-9151-y>.
- [15] M.J. Grote, M. Mehlin, T. Mitkova, Runge–Kutta-based explicit local time-stepping methods for wave propagation, *SIAM J. Sci. Comput.* 37 (2015) A747–A775, <https://doi.org/10.1137/140958293>.
- [16] L. Krivodonova, An efficient local time-stepping scheme for solution of nonlinear conservation laws, *J. Comput. Phys.* 229 (2010) 8537–8551, <https://doi.org/10.1016/j.jcp.2010.07.037>.
- [17] M. Schlegel, O. Knoth, M. Arnold, R. Wolke, Multirate Runge–Kutta schemes for advection equations, *J. Comput. Appl. Math.* 226 (2009) 345–357, <https://doi.org/10.1016/j.cam.2008.08.009>.
- [18] M. Günther, A. Sandu, Multirate generalized additive Runge-Kutta methods, *Numer. Math.* 133 (2016) 497–524, <https://doi.org/10.1007/s00211-015-0756-z>.
- [19] G.J. Cooper, A. Sayfy, Additive Runge-Kutta methods for stiff ordinary differential equations, *Math. Comput.* 40 (1983) 207–218, <https://doi.org/10.1090/S0025-5718-1983-0679441-1>.
- [20] P. Jenny, Time adaptive conservative finite volume method, *J. Comput. Phys.* 403 (2020) 109067, <https://doi.org/10.1016/j.jcp.2019.109067>.
- [21] E. Hairer, Order conditions for numerical methods for partitioned ordinary differential equations, *Numer. Math.* 36 (1981) 431–445, <https://doi.org/10.1007/BF01395956>.
- [22] W. Hundsdorfer, D.I. Ketcheson, I. Savostianov, Error analysis of explicit partitioned Runge-Kutta schemes for conservation laws, *J. Sci. Comput.* 63 (2015) 633–653, <https://doi.org/10.1007/s10915-014-9906-1>.
- [23] I. Higuera, Strong stability for additive Runge–Kutta methods, *SIAM J. Numer. Anal.* 44 (2006) 1735–1758, <https://doi.org/10.1137/040612968>.
- [24] I. Higuera, Characterizing strong stability preserving additive Runge-Kutta methods, *J. Sci. Comput.* 39 (2009) 115–128, <https://doi.org/10.1007/s10915-008-9252-2>.
- [25] W. Hundsdorfer, A. Mozartova, V. Savcenko, Analysis of explicit multirate and partitioned Runge-Kutta schemes for conservation laws, Technical Report, Centrum voor Wiskunde en Informatica, 2007, [https://research.tue.nl/en/publications/analysis-of-explicit-multirate-and-partitioned-\(R\)unge-\(K\)utta-scheme](https://research.tue.nl/en/publications/analysis-of-explicit-multirate-and-partitioned-(R)unge-(K)utta-scheme), eWI report. MAS-E; vol. 0715.
- [26] D.I. Ketcheson, C.B. MacDonald, S.J. Ruuth, Spatially partitioned embedded Runge–Kutta methods, *SIAM J. Numer. Anal.* 51 (2013) 2887–2910, <https://doi.org/10.1137/130906258>.
- [27] D. Doehring, M. Schlottke-Lakemper, G.J. Gassner, M. Torrilhon, Reproducibility repository for “Multirate time-integration based on dynamic ode partitioning through adaptively refined meshes for compressible fluid dynamics”, <https://doi.org/10.5281/zenodo.10792779>, 2024, <https://github.com/trixi-framework/paper-2024-amr-paired-rk>.
- [28] E. Hairer, G. Wanner, S.P. Nørsett, *Solving Ordinary Differential Equations I: Nonstiff Problems*, 2 ed., Springer Series in Computational Mathematics, Springer Berlin, Heidelberg, 1993.
- [29] E. Hofer, A partially implicit method for large stiff systems of odes with only few equations introducing small time-constants, *SIAM J. Numer. Anal.* 13 (1976) 645–663, <https://doi.org/10.1137/0713054>, arXiv:10.1137/0713054.
- [30] E. Griepentrog, *Gemischte Runge-Kutta-verfahren für steife systeme*, Seminarbereich Sect. Math. (1978) 19–29.
- [31] L. Jay, Symplectic partitioned Runge-Kutta methods for constrained Hamiltonian systems, *SIAM J. Numer. Anal.* 33 (1996) 368–387, <https://doi.org/10.1137/0733019>, arXiv:10.1137/0733019.
- [32] L. Abia, J. Sanz-Serna, Partitioned Runge-Kutta methods for separable Hamiltonian problems, *Math. Comput.* 60 (1993) 617–634, <https://doi.org/10.1090/S0025-5718-1993-1181328-1>.
- [33] C.A. Kennedy, M.H. Carpenter, Additive Runge–Kutta schemes for convection–diffusion–reaction equations, *Appl. Numer. Math.* 44 (2003) 139–181, [https://doi.org/10.1016/S0168-9274\(02\)00138-1](https://doi.org/10.1016/S0168-9274(02)00138-1), <https://www.sciencedirect.com/science/article/pii/S0168927402001381>.
- [34] A. Sandu, M. Günther, A generalized-structure approach to additive Runge–Kutta methods, *SIAM J. Numer. Anal.* 53 (2015) 17–42, <https://doi.org/10.1137/130943224>.
- [35] P. Albrecht, A new theoretical approach to Runge–Kutta methods, *SIAM J. Numer. Anal.* 24 (1987) 391–406, <https://doi.org/10.1137/0724030>.
- [36] Z. Jackiewicz, R. Vermiglio, Order conditions for partitioned Runge-Kutta methods, *Appl. Math.* 45 (2000) 301–316, <http://dml.cz/dmlcz/134441>.
- [37] G. Wanner, E. Hairer, *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*, 2 ed., Springer Series in Computational Mathematics, vol. 375, Springer Berlin, Heidelberg, 1996.
- [38] R.I. McLachlan, Y. Sun, P. Tse, Linear stability of partitioned Runge–Kutta methods, *SIAM J. Numer. Anal.* 49 (2011) 232–263, <https://doi.org/10.1137/100787234>.
- [39] J.P. Hespanha, *Linear Systems Theory*, 2 ed., Princeton University Press, 2018, <https://press.princeton.edu/books/hardcover/9780691179575/linear-systems-theory>.
- [40] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, *J. Comput. Phys.* 77 (1988) 439–471, [https://doi.org/10.1016/0021-9991\(88\)90177-5](https://doi.org/10.1016/0021-9991(88)90177-5).
- [41] S. Gottlieb, D. Ketcheson, C.-W. Shu, *Strong Stability Preserving Runge-Kutta and Multistep Time Discretizations*, World Scientific, 2011.
- [42] J.F.B.M. Kraaijevanger, Contractivity of Runge-Kutta methods, *BIT Numer. Math.* 31 (1991) 482–528, <https://doi.org/10.1007/BF01933264>.
- [43] E.J. Kubatko, B.A. Yeager, D.I. Ketcheson, Optimal strong-stability-preserving Runge–Kutta time discretizations for discontinuous Galerkin methods, *J. Sci. Comput.* 60 (2014) 313–344, <https://doi.org/10.1007/s10915-013-9796-7>.
- [44] N. Krais, A. Beck, T. Bolemann, H. Frank, D. Flad, G. Gassner, F. Hindenlang, M. Hoffmann, T. Kuhn, M. Sonntag, et al., FLEXI: a high order discontinuous Galerkin framework for hyperbolic–parabolic conservation laws, *Comput. Math. Appl.* 81 (2021) 186–219, <https://doi.org/10.1016/j.camwa.2020.05.004>.
- [45] T. Warburton, T. Hagstrom, Taming the CFL number for discontinuous Galerkin methods on structured meshes, *SIAM J. Numer. Anal.* 46 (2008) 3151–3180, <https://doi.org/10.1137/060672601>.
- [46] A. Ralston, Runge-Kutta methods with minimum error bounds, *Math. Comput.* 16 (1962) 431–437, <https://doi.org/10.1090/S0025-5718-1962-0150954-0>.
- [47] D. Ketcheson, A. Ahmadi, Optimal stability polynomials for numerical integration of initial value problems, *Commun. Appl. Math. Comput. Sci.* 7 (2013) 247–271, <https://doi.org/10.2140/camcos.2012.7.247>.
- [48] R. Jettsch, O. Nevanlinna, Largest disk of stability of explicit Runge-Kutta methods, *BIT Numer. Math.* 18 (1978) 500–502, <https://doi.org/10.1007/BF01932030>.
- [49] B. Owren, K. Seip, Some stability results for explicit Runge-Kutta methods, *BIT Numer. Math.* 30 (1990) 700–706, <https://doi.org/10.1007/BF01933217>.
- [50] A. Abdulle, Explicit stabilized Runge-Kutta methods, Technical Report, Mathematics Institute of Computational Science and Engineering, School of Basic Sciences - Section of Mathematics EPFL Lausanne 2011, https://www.epfl.ch/labs/mathicse/wp-content/uploads/2018/10/27.2011_AA.pdf.
- [51] P. Van der Houwen, The development of Runge-Kutta methods for partial differential equations, *Appl. Numer. Math.* 20 (1996) 261–272, [https://doi.org/10.1016/0168-9274\(95\)00109-3](https://doi.org/10.1016/0168-9274(95)00109-3).

- [52] J.G. Verwer, B.P. Sommeijer, W. Hundsdorfer, RKC time-stepping for advection–diffusion–reaction problems, *J. Comput. Phys.* 201 (2004) 61–79, <https://doi.org/10.1016/j.jcp.2004.05.002>.
- [53] M. Torrilhon, R. Jetsch, Essentially optimal explicit Runge-Kutta methods with application to hyperbolic–parabolic equations, *Numer. Math.* 106 (2007) 303–334, <https://doi.org/10.1007/s00211-006-0059-5>.
- [54] L. Ferracina, M. Spijker, An extension and analysis of the Shu-Osher representation of Runge-Kutta methods, *Math. Comput.* 74 (2005) 201–219, <https://doi.org/10.1090/S0025-5718-04-01664-3>.
- [55] A. Harten, High resolution schemes for hyperbolic conservation laws, *J. Comput. Phys.* 135 (1997) 260–278, <https://doi.org/10.1006/jcph.1997.5713>, <https://www.sciencedirect.com/science/article/pii/S0021999197957132>.
- [56] S. Gottlieb, C.-W. Shu, Total variation diminishing Runge-Kutta schemes, *Math. Comput.* 67 (1998) 73–85, <https://doi.org/10.1090/S0025-5718-98-00913-2>.
- [57] X. Zhang, C.-W. Shu, Maximum-principle-satisfying and positivity-preserving high-order schemes for conservation laws: survey and new developments, *Proc. R. Soc. A, Math. Phys. Eng. Sci.* 467 (2011) 2752–2776, <https://doi.org/10.1098/rspa.2011.0153>, <https://royalsocietypublishing.org/doi/abs/10.1098/rspa.2011.0153>.
- [58] Z.J. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H.T. Huynh, et al., High-order CFD methods: current status and perspective, *Int. J. Numer. Methods Fluids* 72 (2013) 811–845, <https://doi.org/10.1002/fld.3767>.
- [59] B. Sande, A.E. Veldman, Constraint-consistent Runge–Kutta methods for one-dimensional incompressible multiphase flow, *J. Comput. Phys.* 384 (2019) 170–199, <https://doi.org/10.1016/j.jcp.2019.02.001>.
- [60] D.I. Ketcheson, L. Lóczy, M. Parsani, Internal error propagation in explicit Runge-Kutta methods, *SIAM J. Numer. Anal.* 52 (2014) 2227–2249, <https://doi.org/10.1137/130936245>.
- [61] H. Ranocha, M. Schlottke-Lakemper, A.R. Winters, E. Faulhaber, J. Chan, G. Gassner, Adaptive numerical simulations with Trixi.jl: a case study of Julia for scientific computing, *Proc. JuliaCon Conf.* 1 (2022) 77, <https://doi.org/10.21105/jcon.00077>, arXiv:2108.06476.
- [62] M. Schlottke-Lakemper, A.R. Winters, H. Ranocha, G. Gassner, A purely hyperbolic discontinuous Galerkin approach for self-gravitating gas dynamics, *J. Comput. Phys.* 442 (2021) 110467, <https://doi.org/10.1016/j.jcp.2021.110467>, arXiv:2008.10593.
- [63] M. Schlottke-Lakemper, G. Gassner, H. Ranocha, A.R. Winters, J. Chan, Trixi.jl: adaptive high-order numerical simulations of hyperbolic PDEs in Julia, <https://doi.org/10.5281/zenodo.3996439>, 2021, <https://github.com/trixi-framework/Trixi.jl>.
- [64] D. Doehring, G.J. Gassner, M. Torrilhon, Many-stage optimal stabilized Runge-Kutta methods for hyperbolic partial differential equations, *J. Sci. Comput.* 99 (1) (2024) 1–40, <https://doi.org/10.1007/s10915-024-02478-5>.
- [65] R.B. Lehoucq, D.C. Sorensen, C. Yang, *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*, SIAM, 1998.
- [66] A.M. Rueda-Ramírez, G.J. Gassner, A subcell finite volume positivity-preserving limiter for DGSEM discretizations of the Euler equations, <https://doi.org/10.48550/arXiv.2102.06017>, 2021, preprint, arXiv:2102.06017.
- [67] K. Black, A conservative spectral element method for the approximation of compressible fluid flow, *Kybernetika* 35 (1999) 133–146, <https://www.kybernetika.cz/content/1999/1/133/paper.pdf>.
- [68] D.A. Kopriva, *Implementing Spectral Methods for Partial Differential Equations: Algorithms for Scientists and Engineers*, Scientific Computation (SCIENTCOMP), 1 ed., Springer Science & Business Media, 2009.
- [69] G.J. Gassner, A skew-symmetric discontinuous Galerkin spectral element discretization and its relation to SBP-SAT finite difference methods, *SIAM J. Sci. Comput.* 35 (2013) A1233–A1253, <https://doi.org/10.1137/120890144>.
- [70] G.J. Gassner, A.R. Winters, D.A. Kopriva, Split form nodal discontinuous Galerkin schemes with summation-by-parts property for the compressible Euler equations, *J. Comput. Phys.* 327 (2016) 39–66, <https://doi.org/10.1016/j.jcp.2016.09.013>.
- [71] B. Einfeldt, On Godunov-type methods for gas dynamics, *SIAM J. Numer. Anal.* 25 (1988) 294–318, <https://doi.org/10.1137/0725021>.
- [72] S. Hennemann, A.M. Rueda-Ramírez, F.J. Hindenlang, G.J. Gassner, A provably entropy stable subcell shock capturing approach for high order split form DG for the compressible Euler equations, *J. Comput. Phys.* 426 (2021) 109935, <https://doi.org/10.1016/j.jcp.2020.109935>.
- [73] T.C. Fisher, M.H. Carpenter, High-order entropy stable finite difference schemes for nonlinear conservation laws: finite domains, *J. Comput. Phys.* 252 (2013) 518–557, <https://doi.org/10.1016/j.jcp.2013.06.014>.
- [74] H. Ranocha, Entropy conserving and kinetic energy preserving numerical methods for the Euler equations using summation-by-parts operators, in: *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2018*, vol. 134, 2020, pp. 525–535, <https://library.oapen.org/handle/20.500.12657/41274>.
- [75] E.F. Toro, M. Spruce, W. Speares, Restoration of the contact surface in the HLL-Riemann solver, *Shock Waves* 4 (1994) 25–34, <https://doi.org/10.1007/BF01414629>.
- [76] W.H. Hundsdorfer, J.G. Verwer, *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*, 1 ed., Springer Series in Computational Mathematics, vol. 33, Springer, 2010.
- [77] C. Burstedde, L.C. Wilcox, O. Ghattas, p4est: scalable algorithms for parallel adaptive mesh refinement on forests of octrees, *SIAM J. Sci. Comput.* 33 (2011) 1103–1133, <https://doi.org/10.1137/100791634>.
- [78] D.A. Kopriva, A conservative staggered-grid Chebyshev multidomain method for compressible flows. II. A semi-structured method, *J. Comput. Phys.* 128 (1996) 475–488, <https://doi.org/10.1006/jcph.1996.0225>.
- [79] C. Rackauckas, Q. Nie, *DifferentialEquations.jl – a performant and feature-rich ecosystem for solving differential equations in julia*, *J. Open Res. Softw.* 5 (2017), <https://doi.org/10.5334/jors.151>, <https://app.dimensions.ai/details/publication/pub.1085583166>, <http://openresearchsoftware.metajnl.com/articles/10.5334/jors.151/galley/245/download/>, exported from <https://app.dimensions.ai> on 2019/05/05.
- [80] T. Toulorge, W. Desmet, Optimal Runge–Kutta schemes for discontinuous Galerkin space discretizations applied to wave propagation problems, *J. Comput. Phys.* 231 (2012) 2067–2091, <https://doi.org/10.1016/j.jcp.2011.11.024>.
- [81] M. Parsani, D.I. Ketcheson, W. Deconinck, Optimized explicit Runge–Kutta schemes for the spectral difference method applied to wave propagation problems, *SIAM J. Sci. Comput.* 35 (2013) A957–A986, <https://doi.org/10.1137/120885899>.
- [82] C.A. Kennedy, M.H. Carpenter, R.M. Lewis, Low-storage, explicit Runge–Kutta schemes for the compressible Navier–Stokes equations, *Appl. Numer. Math.* 35 (2000) 177–219, [https://doi.org/10.1016/S0168-9274\(99\)00141-5](https://doi.org/10.1016/S0168-9274(99)00141-5).
- [83] H. Ranocha, L. Dalcin, M. Parsani, D.I. Ketcheson, Optimized Runge-Kutta methods with automatic step size control for compressible computational fluid dynamics, *Commun. Appl. Math. Comput.* 4 (2022) 1191–1228, <https://doi.org/10.1007/s42967-021-00159-w>.
- [84] F. Bassi, S. Rebay, A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations, *J. Comput. Phys.* 131 (1997) 267–279, <https://doi.org/10.1006/jcph.1996.5572>.
- [85] G.J. Gassner, A.R. Winters, F.J. Hindenlang, D.A. Kopriva, The BR1 scheme is stable for the compressible Navier–Stokes equations, *J. Sci. Comput.* 77 (2018) 154–200, <https://doi.org/10.1007/s10915-018-0702-1>.
- [86] D.L. Brown, M.L. Minion, Performance of under-resolved two-dimensional incompressible flow simulations, *J. Comput. Phys.* 122 (1995) 165–183, <https://doi.org/10.1006/jcph.1995.1205>.
- [87] H. Ranocha, Comparison of some entropy conservative numerical fluxes for the Euler equations, *J. Sci. Comput.* 76 (2018) 216–242, <https://doi.org/10.1007/s10915-017-0618-1>.
- [88] R. Löhner, An adaptive finite element scheme for transient problems in CFD, *Comput. Methods Appl. Mech. Eng.* 61 (1987) 323–338, [https://doi.org/10.1016/0045-7825\(87\)90098-3](https://doi.org/10.1016/0045-7825(87)90098-3).

- [89] B. Fryxell, K. Olson, P. Ricker, F. Timmes, M. Zingale, D. Lamb, P. MacNeice, R. Rosner, J. Truran, H. Tufo, FLASH: an adaptive mesh hydrodynamics code for modeling astrophysical thermonuclear flashes, *Astrophys. J. Suppl. Ser.* 131 (2000) 273, <https://doi.org/10.1086/317361>.
- [90] S.A. Orszag, C.-M. Tang, Small-scale structure of two-dimensional magnetohydrodynamic turbulence, *J. Fluid Mech.* 90 (1979) 129–143, <https://doi.org/10.1017/S002211207900210X>.
- [91] T. Warburton, G.E. Karniadakis, A discontinuous Galerkin method for the viscous mhd equations, *J. Comput. Phys.* 152 (1999) 608–641, <https://doi.org/10.1006/jcph.1999.6248>.
- [92] D. Derigs, A.R. Winters, G.J. Gassner, S. Walch, M. Bohm, Ideal GLM-MHD: about the entropy consistent nine-wave magnetic field divergence diminishing ideal magnetohydrodynamics equations, *J. Comput. Phys.* 364 (2018) 420–467, <https://doi.org/10.1016/j.jcp.2018.03.002>.
- [93] C.-D. Munz, P. Omnes, R. Schneider, E. Sonnendrücker, U. Voss, Divergence correction techniques for Maxwell solvers based on a hyperbolic model, *J. Comput. Phys.* 161 (2000) 484–511, <https://doi.org/10.1006/jcph.2000.6507>.
- [94] M. Bohm, A.R. Winters, G.J. Gassner, D. Derigs, F. Hindenlang, J. Saur, An entropy stable nodal discontinuous Galerkin method for the resistive MHD equations. Part I: Theory and numerical verification, *J. Comput. Phys.* 422 (2020) 108076, <https://doi.org/10.1016/j.jcp.2018.06.027>.
- [95] K.G. Powell, P.L. Roe, T.J. Linde, T.I. Gombosi, D.L. De Zeeuw, A solution-adaptive upwind scheme for ideal magnetohydrodynamics, *J. Comput. Phys.* 154 (1999) 284–309, <https://doi.org/10.1006/jcph.1999.6299>.
- [96] J. DeBonis, Solutions of the Taylor-Green vortex problem using high-resolution explicit finite difference methods, in: 51st AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, 2013, pp. 1–20, <https://arc.aiaa.org/doi/abs/10.2514/6.2013-382>, <https://arc.aiaa.org/doi/pdf/10.2514/6.2013-382>.
- [97] J.R. Bull, A. Jameson, Simulation of the compressible Taylor Green vortex using high-order flux reconstruction schemes, in: 7th AIAA Theoretical Fluid Mechanics Conference, 2014, p. 3210, <https://arc.aiaa.org/doi/abs/10.2514/6.2014-3210>, <https://arc.aiaa.org/doi/pdf/10.2514/6.2014-3210>.
- [98] C.T. Jacobs, S.P. Jammy, N.D. Sandham, Opensbli: a framework for the automated derivation and parallel execution of finite difference solvers on a range of computer architectures, *J. Comput. Sci.* 18 (2017) 12–23, <https://doi.org/10.1016/j.jocs.2016.11.001>.
- [99] J. Shi, Y.-T. Zhang, C.-W. Shu, Resolution of high order WENO schemes for complicated flow structures, *J. Comput. Phys.* 186 (2003) 690–696, [https://doi.org/10.1016/S0021-9991\(03\)00094-9](https://doi.org/10.1016/S0021-9991(03)00094-9).
- [100] D. Derigs, G.J. Gassner, S. Walch, A.R. Winters, Entropy stable finite volume approximations for ideal magnetohydrodynamics, *Jahresber. Dtsch. Math.-Ver.* 120 (2018) 153–219, <https://doi.org/10.1365/s13291-018-0178-9>.