



A p-Adaptive Discontinuous Galerkin Method with hp-Shock Capturing

Pascal Mossier¹ · Andrea Beck² · Claus-Dieter Munz¹

Received: 19 March 2021 / Revised: 19 October 2021 / Accepted: 11 January 2022 /
Published online: 18 February 2022
© The Author(s) 2022

Abstract

In this work, we present a novel hybrid Discontinuous Galerkin scheme with hp-adaptivity capabilities for the compressible Euler equations. In smooth regions, an efficient and accurate discretization is achieved via local p-adaptation. At strong discontinuities and shocks, a finite volume scheme on an h-refined element-local subgrid gives robustness. Thus, we obtain a hp-adaptive scheme that exploits both the high convergence rate and efficiency of a p-adaptive high order scheme as well as the stable and accurate shock capturing abilities of a low order finite volume scheme, but avoids the inherent resolution loss through h-refinement. A single a priori indicator, based on the modal decay of the local polynomial solution representation, is used to distinguish between discontinuous and smooth regions and control the p-refinement. Our method is implemented as an extension to the open source software FLEXI. Hence, the efficient implementation of the method for high performance computers was an important criterion during the development. The efficiency of our adaptive scheme is demonstrated for a variety of test cases, where results are compared against non adaptive simulations. Our findings suggest that the proposed adaptive method produces comparable or even better results with significantly less computational costs.

Keywords Discontinuous Galerkin · High order schemes · p-Adaptivity · Shock capturing · h-Refinement

1 Introduction

The stable and accurate simulation of transsonic and supersonic flow phenomena, that are encountered at high speed flight conditions, is a demanding task for the computational fluid

www.flexi-project.org, GNU GPL v3.0.

✉ Pascal Mossier
pascal.mossier@iag.uni-stuttgart.de

¹ Institute of Aerodynamics and Gas Dynamics, University of Stuttgart, Pfaffenwaldring 21, 70569 Stuttgart, Germany

² Laboratory of Fluid Dynamics and Technical Flows, Otto-von-Guericke-University Magdeburg, Magdeburg, Germany

dynamics community. The challenge, posed by the described flow conditions, is due to the presence of both regions with smooth and discontinuous solutions. Hence, a numerical approximation has to be able to capture shock waves, strong velocity gradients and shear layers, where fluid properties change sharply, as well as smooth laminar or turbulent flow. Due to their low approximation errors and favorable scale resolving capabilities, high order methods are currently objects of research and development. In particular, the discontinuous Galerkin (DG) method is well suited for practical applications since it features an element local polynomial solution representation that is only coupled by numerical fluxes. This makes it highly parallelizable and efficient for high-performance computing. The discretization error e of a polynomial approximation of degree $p - 1$ can be roughly estimated as $e \approx kh^p$, with the constant k depending on the solution regularity and the element size h . In regions of smooth flow, the high order discretization can profit from an exponential convergence rate with the exponent p . However, in regions with strong gradients or under-resolved regions, these schemes suffer from spurious oscillations, called the Gibb's instability, that causes stability issues and a loss of accuracy.

There exist different strategies to capture shocks and discontinuities. One approach is to smooth out shocks and strong gradients in the solution with artificial viscosity by locally applying an additional dissipation term, originally proposed by von Neumann and Richtmyer in [31]. Persson and Peraire [20] adapted this to high order discontinuous Galerkin methods to eliminate the high frequencies without widening the shock over a couple of cells. In a recent article, Zeifang et al. [33] used a smoothness indicator, based on artificial neural networks [2], to control the amount of artificial viscosity. Another approach is the application of limiting techniques such as those applied to the finite volume (FV) approach, adapted to the DG formulation. Qiu and Shu [22] combined the DG method with a weighted essentially non-oscillatory (WENO) limiting in [22] and more recently by Zhu et al. in [34]. A third approach is to change the discretization operator around a shock wave to a finite volume scheme and apply it on a refined sub-cell grid. This is done in the work of Dumbser et al. [7,8], where an ADER-DG scheme is limited by finite volume sub-cells with a WENO reconstruction for increased accuracy. A similar approach is used by Vilar [30] where a flux reconstruction scheme based on the the DG solution is used to increase the accuracy of the finite volume sub-cells. A variant is to decrease locally the order of accuracy while the loss of accuracy is remedied by local mesh refinement (h-refinement). This was applied in the work of Huerta et al. [11] and Persson et al. [21] by introducing piecewise constant ansatz functions and subdividing the DG element into sub-cells. A similar strategy, presented in [16], is called rp-refinement, where the mesh refinement is achieved by deforming the mesh and clustering DOFs around shocks, rather than refining the mesh through subdivision of elements. Both, hp- and rp-refinement have the common strategy of locally reducing the polynomial degree and increasing the mesh resolution to resolve shocks and discontinuities accurately. In this paper we employ a concept based on the work of Sonntag and Munz [26] that follows the same idea. A DG element containing shocks or strong gradients is subdivided into FV sub-cells that represent the solution with a stable second order FV-scheme. Low order FV schemes are well suited for shock capturing but suffer from a high discretization error. Therefore a finer sub-cell grid is used to compensate for the loss of accuracy. The combination of a high order DG method with a FV sub-cell scheme provides a stable numerical framework to resolve both smooth and discontinuous regions encountered in the transsonic and supersonic flow phenomena considered in this paper.

The goal of this paper is to extend the hybrid DG and FV sub-cell approach towards an hp-adaptive scheme to improve the accuracy and reduce the computational cost. Both p-adaptation and adaptive mesh refinement, called h-refinement, are common refinement

strategies and have been investigated in the past. In [9,12,23], h-refinement techniques were successfully applied to high order DG methods to solve the compressible Euler equations. P-adaptive DG methods, where adaptivity is achieved through a variable ansatz degree, have been investigated in [1,3,19]. Since Li and Jameson showed in [15], that h-refinement produces better results at flow discontinuities while p-refinement is more efficient in smooth regions we decide to use both techniques for our hybrid DG/FV scheme. In smooth regions, where the DG scheme is stable, adaptive p-refinement is applied by locally increasing and decreasing the polynomial degree. In the vicinity of shocks and discontinuities, the second order FV discretization is applied to a sub-grid of FV cells and allows for a better localization of discontinuities.

Thus, we propose to combine a FV sub-cell scheme with a p-adaptive DG scheme to obtain a novel, efficient hp-adaptive strategy that comes without the need for mesh adaptation during the run time. To allow for an increased local number of DOFs per element at shocks, we extended the efficient a priori FV sub-cell limiting of Sonntag and Munz [26] to allow for FV resolutions that can be chosen independently of the DG method. We use a sub-cell resolution of $2N + 1$ cells per direction that Dumbser et al. proposed in the context of an a posteriori FV sub-cell limiting [8]. The resulting scheme combines the advantages of high order schemes and p-adaptivity in smooth regions, with the robustness and accuracy of FV schemes with h-refinement for shock capturing. To switch between the two schemes, an indicator, based on the modal decay of the polynomial solution representation, proposed by Mavriplis [18], is used. Based on the modal decay rate, oscillations of the local solution polynomials within an element can be detected. Additionally, the modal decay rate inside an element provides an error estimate to chose the local polynomial degree of the DG elements.

This paper is organized as follows. In Sect. 2, the governing equations are introduced. In Sect. 3, we outline the DG discretization and the p-refinement technique, used in smooth regions. The FV sub-cell scheme, used for shock capturing, is introduced in Sect. 4. In Sect. 5 we briefly discuss the chosen indicator for FV sub-cell limiting and p-refinement. In Sect. 6, we propose an efficient implementation for the hybrid, adaptive scheme and in Sect. 7, the hp-adaptive method is assembled from the building blocks. Finally, in Sect. 8 we present and discuss numerical examples to validate the method and demonstrated its superior computational efficiency when compared to uniformly refined computations.

2 Governing Equations

In this work, we consider the Euler equations as the model for an inviscid compressible fluid flow in a computational domain Ω with the boundary $\Gamma = \partial\Omega$ on a finite time interval $(0, T]$. They can be written in the conservation form:

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla_{\mathbf{x}} \cdot \mathbf{F}(\mathbf{u}) = \mathbf{0} \quad \text{in } \Omega \times (0, T]. \quad (1)$$

Here, $[\rho, \rho \mathbf{v}, E]$ denotes the transposed vector of conserved variables \mathbf{u} with density ρ , velocity vector \mathbf{v} , and total energy E per unit volume. The vector \mathbf{F} is the convective flux and is written in components as $\mathbf{F} = [\rho \mathbf{v}, \rho \mathbf{v} \times \mathbf{v} + p \mathbf{I}, \mathbf{v}(E + p)]$. To compute the pressure p and to close the system, the equation of state of a perfect gas is assumed, given by

$$p(\rho, \rho \mathbf{v}, E) := (\gamma - 1) \left(E - \frac{1}{2} \rho \|\mathbf{v}\|_2^2 \right) \quad (2)$$

with the isentropic expansion coefficient $\gamma = 1.4$. The equation system is subjected to initial conditions, defined as $\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x})$, and appropriate boundary conditions, defined as $\mathbf{u}(\mathbf{x}_\Gamma, t) = \mathbf{u}_\Gamma(\mathbf{x}_\Gamma, t)$ for the spatial coordinate vector $\mathbf{x}_\Gamma \in \Gamma$. The boundary conditions are specified for the presented test cases in Sect. 8.

3 P-Adaptive Discontinuous Galerkin Discretization

In this section, a p-adaptive high order approximation for the Euler equations, based on the Discontinuous Galerkin Spectral Element Method (DGSEM), is outlined. The DGSEM is a computationally efficient DG variant that exhibits the basis functions in a tensor basis structure, described in detail in [13] and our version in [14].

3.1 Spatial Discretization on Curved Meshes

Equation (1) is approximated by the DGSEM for an arbitrary approximation order N on a computational domain $\Omega \in \mathbb{R}^3$ that is subdivided into $K \in \mathbb{N}$ non overlapping hexahedral elements Ω^e , so that $\Omega = \bigcup_{e=1}^K \Omega^e$ and $\bigcap_{e=1}^K \Omega^e = \emptyset$ are valid. We apply transformations from physical space $\mathbf{x} = (x_1, x_2, x_3)^T$ to reference space $\boldsymbol{\xi} = (\xi_1, \xi_2, \xi_3)^T$ and vice versa to map a physical element Ω^e to a reference element $E := [-1, 1]^3$ with the mappings

$$\mathbf{x}(\boldsymbol{\xi}) \quad \text{and} \quad \boldsymbol{\xi}(\mathbf{x}). \tag{3}$$

For the transformation of Eq. (1) to reference space, the inverse of the Jacobian \mathbf{J} of the mapping $\mathbf{x}(\boldsymbol{\xi})$ is required. With the covariant and contravariant basis vectors

$$\mathbf{a}_i = \frac{\partial \mathbf{x}}{\partial \xi_i}, \quad \mathbf{a}^i = \frac{\partial \boldsymbol{\xi}}{\partial x_i} \tag{4}$$

and the determinant J_{geo} of the Jacobian matrix \mathbf{J} , the inverse of \mathbf{J} is obtained as:

$$\mathbf{J}^{-1} = \frac{1}{J_{geo}} \begin{pmatrix} (\mathbf{a}_2 \times \mathbf{a}_3)^T \\ (\mathbf{a}_3 \times \mathbf{a}_1)^T \\ (\mathbf{a}_1 \times \mathbf{a}_2)^T \end{pmatrix} = \begin{pmatrix} \mathbf{a}^1 \\ \mathbf{a}^2 \\ \mathbf{a}^3 \end{pmatrix} \tag{5}$$

It follows, that the divergence of a Flux \mathbf{F} can be written in reference space as

$$\nabla_{\mathbf{x}} \cdot \mathbf{F} = \frac{1}{J_{geo}} \sum_{i=1}^3 \frac{\partial J_{geo} \mathbf{a}^i \cdot \mathbf{F}}{\partial \xi_i} = \frac{1}{J_{geo}} \nabla_{\boldsymbol{\xi}} \cdot \mathcal{F} \tag{6}$$

with \mathcal{F} denoting the flux in reference space, called contravariant flux.

$$\mathcal{F} = \begin{pmatrix} \mathcal{F}_1 \\ \mathcal{F}_2 \\ \mathcal{F}_3 \end{pmatrix} = \begin{pmatrix} J_{geo} \mathbf{a}^1 \cdot \mathbf{F} \\ J_{geo} \mathbf{a}^2 \cdot \mathbf{F} \\ J_{geo} \mathbf{a}^3 \cdot \mathbf{F} \end{pmatrix} \tag{7}$$

Finally, Eq. (1) in reference space is obtained as

$$J_{geo} \frac{\partial \mathbf{u}}{\partial t} + \nabla_{\boldsymbol{\xi}} \cdot \mathcal{F}(\mathbf{u}) = \mathbf{0}. \tag{8}$$

3.2 The Discontinuous Galerkin Spectral Element Method

Next, the DGSEM formulation is derived for the transformed equation system (8). First, the weak form is obtained by projection onto the test space spanned by $\phi \in \mathbb{P}$:

$$\int_E \left(J_{geo} \frac{\partial \mathbf{u}}{\partial t} \right) \phi d\Omega + \int_E (\nabla_{\xi} \cdot \mathcal{F}) \phi d\Omega = \mathbf{0} \tag{9}$$

Integration by parts yields

$$\int_E \left(J_{geo} \frac{\partial \mathbf{u}}{\partial t} \right) \phi d\Omega + \oint_{\partial E} (\mathcal{F} \cdot \mathbf{n}_{\xi}) \phi dS_{\xi} - \int_E (\mathcal{F} \cdot \nabla_{\xi}) \phi d\Omega = \mathbf{0} \tag{10}$$

with the unit normal vector \mathbf{n}_{ξ} and the surface of the reference element S_{ξ} . Since discontinuities are allowed across element interfaces, the term $\mathcal{F} \cdot \mathbf{n}_{\xi}$ is not unique at the interface and is approximated by a numerical flux $(\mathcal{F} \cdot \mathbf{n}_{\xi})^*$. The element local solution and the contravariant flux are both approximated as

$$\mathbf{u}(\xi, t) \approx \sum_{i,j,k=0}^N \hat{\mathbf{u}}_{ijk}(t) \psi_{ijk}(\xi), \tag{11}$$

$$\mathcal{F}(\xi, t) \approx \sum_{i,j,k=0}^N \mathcal{F}_{ijk}(\hat{\mathbf{u}}_{ijk}) \psi_{ijk}(\xi) \tag{12}$$

in the space spanned by tensor products of one-dimensional Lagrange polynomials of degree N :

$$\psi_{ijk}(\xi) = \ell_i(\xi^1) \ell_j(\xi^2) \ell_k(\xi^3) \tag{13}$$

and with the degrees of freedom (DOFs) $\hat{\mathbf{u}}_{ijk}$. Following Galerkin’s idea, we use the same polynomials for the basis functions ψ and the test functions ϕ . The node distribution for the one-dimensional Lagrange polynomials is defined by the Legendre–Gauss nodes of the reference element in every direction. If these points are also used for the numerical quadrature, we obtain a semi discrete DG operator with a tensor product structure and a reduced number of required operations per degree of freedom. Due to the tensor product structure, the DG operator can be split into one-dimensional operations that are applied for every direction of the reference element separately. The total number of degrees of freedom per variable for an element of order N in d space dimensions corresponds to $(N + 1)^d$. The node distribution is illustrated in Fig. 1 for the two-dimensional case and two elements with different local polynomial degrees. For a more detailed derivation of the DGSEM operator, which is used in the code framework, the reader is referred to [14].

3.3 Coupling of DG Elements with Variable Order

To construct a p-adaptive DGSEM scheme, interfaces between elements with varying polynomial degrees have to be considered. Since coupling is achieved through the numerical flux, the flux computation has to be adapted to allow for different degrees among adjacent elements. Furthermore, switching of the element local degree requires a transformation of the elements solution representation between different polynomial degrees. Without loss of generality, the transformation formula can be derived for the one-dimensional case, due to the tensor basis structure.

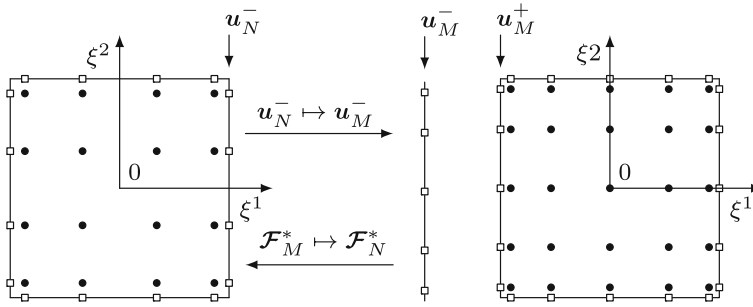


Fig. 1 Node distribution inside two-dimensional reference elements with degrees $N = 3$ (left) and $M = 4$ (right) indicated by dots and surface node distributions u_N^- and u_M^+ indicated by squares for the computation of the numerical flux \mathcal{F}_M^* between adjacent elements. For the flux computation, the surface node distribution u_N^- is transformed to the representation of the higher order element u_M^-

A transformation of the element local volume or surface solution from a degree N to a degree M with $N < M$ can be achieved by interpolating the solution polynomial at the new node positions of degree M . In the case of $N > M$, a conservative transformation is obtained by projecting the given polynomial of degree N to a polynomial of degree M . We achieve this projection through a modal cut-off filtering and subsequent interpolation. Therefore the solution representation u^N of degree N is first transformed to a modal Legendre basis representation u_{Leg}^N with the Vandermonde matrix V_{Leg}^N :

$$V_{Leg}^N u^N = u_{Leg}^N, \quad u^N, u_{Leg}^N \in \mathbb{R}^{N+1}. \tag{14}$$

Next, a modal cutoff is performed to reduce the number of modes to M :

$$u_{Leg}^M = \begin{pmatrix} u_{Leg}^N(0) \\ \vdots \\ u_{Leg}^N(M) \end{pmatrix}, \quad u_{Leg}^M \in \mathbb{R}^{M+1}. \tag{15}$$

The transformation is completed by transforming the modal solution back to the initial nodal Lagrange basis:

$$(V_{Leg}^M)^{-1} u_{Leg}^M = u^M, \quad u^M \in \mathbb{R}^{M+1}. \tag{16}$$

The transformation can be expressed as one matrix vector operation:

$$u^M = (V_{Leg}^M)^{-1} \begin{bmatrix} 1 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & 1 & \dots & 0 \end{bmatrix}^{M \times N} V_{Leg}^N u^N,$$

written in short as

$$u^M = V^{M2N} u^N. \tag{17}$$

The numerical flux between two elements that share a common side is evaluated using a polynomial solution representation of the surface data of both elements. Since the flux is computed for every node of the polynomial surface data, the number and position of the nodes has to match between neighboring sides. In case adjacent elements do not share a

common polynomial degree, depicted in Fig. 1, the node distribution of the element with the higher degree M is used. To that purpose, the surface data with the lower degree \mathbf{u}_N^+ is transformed to degree M of the neighboring side, using the aforementioned transformation rule. The numerical flux is then evaluated for all M nodes \mathbf{u}_M^+ and \mathbf{u}_M^- at the element side. The resulting flux \mathcal{F}_M^* can be applied to the element of degree M directly. For the element of degree N , the flux has to be transformed to a representation of degree N , using the introduced projection. In pseudo code, the flux computation reads as follows

$$\begin{aligned}
 & N \neq M, \quad N < M \quad (\text{Adjacent elements have different degrees}) \\
 & \mathbf{u}_N^+ \mapsto \mathbf{u}_M^+ \quad (\text{Transform side data of degree } N \text{ to degree } M) \\
 & \mathcal{F}_M^* := \text{Riemann solver}(\mathbf{u}_M^+, \mathbf{u}_M^-, \mathbf{n}) \quad (\text{Calculate numerical flux for } M \text{ nodes}) \\
 & \mathcal{F}_M^* \mapsto \mathcal{F}_N^* \quad (\text{Transform flux representation of degree } M \text{ to degree } N).
 \end{aligned}$$

Altogether, the formula, presented in this section, provide all basic ingredients for a p-adaptive discontinuous Galerkin discretization. Coupling of elements with different degrees through a numerical flux computation on a common polynomial representation was also addressed. For a more detailed derivation of the treatment of non conforming interfaces, so called mortars [17], the reader is referred to [6]. It should be noted, that the presented scheme does not impose any restrictions to the degree of neighboring elements, so that every arbitrary distribution of polynomial degrees among the elements, imposed by an indicator is possible.

In this work, we use an explicit 4 stage Runge–Kutta scheme for numerical integration in time to obtain a fully discrete scheme. The resulting time step restriction is obtained from the CFL condition by Courant, Friedrichs and Lewy [5] with an additional factor $\frac{1}{2N+1}$ that takes into account the polynomial degree N [4] and a scaling factor for the Runge–Kutta scheme $\alpha_{RK}(N) > 1$.

$$\Delta t = CFL \cdot \alpha_{RK}(N) \cdot \frac{\Delta x_{DG}}{(2N + 1) |\lambda^c|} \tag{18}$$

The remaining variables are the physical element size Δx_{DG} and the maximal signal velocity $|\lambda^c|$ given by the eigenvalues of the Euler equations.

4 An h-Adaptive Finite Volume Sub-cell Discretization

In Sect. 3, a p-adaptive DGSEM discretization for the compressible Euler equations was outlined. This adaptive high order scheme provides good stability and accuracy in smooth regions. However, in the presence of shocks or strong gradients within an element, the solution generates oscillations due to the Gibbs phenomenon. To circumvent this issue, we employ a shock capturing with finite volume sub-cells. Here, a d-dimensional DG element is subdivided into $(N_{FV})^d$ sub-cells to apply the time evolution by a stable second order FV scheme. A typical choice for the sub-cell resolution is $N_{FV} = 2N + 1$, since this is the highest resolution that does not impose a stricter time step restriction as the corresponding DG element of degree N . Thus we obtain a local reduction in approximation order, but combined with an increased spatial localization of strong gradients. In the following, we use a FV sub-cell shock capturing scheme on a predefined sub-cell mesh to achieve the favorable local mesh refinement properties of the hp- and rp-refinement strategies, without the need to perform remeshing or changing the number of elements in the mesh. For a detailed derivation of the FV sub-cell scheme and its implementation in FLEXI for $N_{FV} = N + 1$ the reader is referred

to the work of Sonntag and Munz in [26,27] and Kraiss et al. in [14]. In the following, we derive the spatial FV sub-cell discretization and its coupling to adjacent DG elements for arbitrary combinations of N and N_{FV} .

4.1 Spatial Discretization on Curved Meshes

We derive the spatial finite volume sub-cell discretization for the reference element E , following the mapping $\mathbf{x} \mapsto \boldsymbol{\xi}$ introduced in Sect. 3. The reference element of a local DG solution, in which shock capturing becomes necessary, is subdivided into N_{FV} equidistant pieces in every space dimension d . This results in a subdivision of the reference element and is then called FV element in this paper. Thus, the FV representation of such a troubled DG element contains $(N_{FV})^d$ sub-cells e . The metric terms J_{geo}^{FV} and $J_{geo}^{FV} \mathbf{a}^{i,FV}$, required for the mapping to reference space, are obtained as the integral mean of J_{geo} and $J_{geo} \mathbf{a}^i$ for each FV sub-cell e :

$$J_{geo}^{FV} = \frac{\int_e J_{geo}(\boldsymbol{\xi}) d\boldsymbol{\xi}}{\int_e d\boldsymbol{\xi}} \tag{19}$$

$$J_{geo}^{FV} \mathbf{a}^{i,FV} = \frac{\int_e J_{geo}(\boldsymbol{\xi}) \mathbf{a}^i(\boldsymbol{\xi}) d\boldsymbol{\xi}}{\int_e d\boldsymbol{\xi}} \quad \text{for } i = 1, 2, 3 \tag{20}$$

This is equivalent to multiplying with the transformation matrix \mathbf{V}^{DG2FV} defined in Sect. 4.3:

$$J_{geo}^{FV} = \mathbf{V}^{DG2FV} J_{geo}(\boldsymbol{\xi}) \quad , \quad J_{geo}^{FV} \mathbf{a}^{i,FV} = \mathbf{V}^{DG2FV} J_{geo}(\boldsymbol{\xi}) \mathbf{a}^i(\boldsymbol{\xi}) \tag{21}$$

With the definition of the FV metric terms, we obtain the contravariant flux.

$$\mathcal{F}^{FV} = \begin{pmatrix} \mathcal{F}_1^{FV} \\ \mathcal{F}_2^{FV} \\ \mathcal{F}_3^{FV} \end{pmatrix} = \begin{pmatrix} J_{geo}^{FV} \mathbf{a}^1 \cdot \mathbf{F} \\ J_{geo}^{FV} \mathbf{a}^2 \cdot \mathbf{F} \\ J_{geo}^{FV} \mathbf{a}^3 \cdot \mathbf{F} \end{pmatrix} \tag{22}$$

Consequently, Eq. (1) can be written in reference space for a sub-cell element e as:

$$J_{geo}^{FV} \frac{\partial \mathbf{u}^{FV}}{\partial t} + \nabla_{\boldsymbol{\xi}} \cdot \mathcal{F}^{FV}(\mathbf{u}^{FV}) = \mathbf{0}. \tag{23}$$

4.2 Finite Volume Discretization

Following the transformation to reference space, the FV sub-cell discretization for a reference sub-cell e is obtained from the weak form of (23) as

$$\int_e (J_{geo}^{FV} \partial_t \mathbf{u}^{FV}) \phi d\Omega_e + \oint_{\partial e} (\mathcal{F}^{FV} \cdot \mathbf{n}^{FV}) \phi dS_e = \mathbf{0}. \tag{24}$$

The term $\mathcal{F}^{FV} \cdot \mathbf{n}^{FV}$ represents the flux at the sub-cell boundaries and is approximated by a numerical flux \mathcal{F}^* :

$$(\mathcal{F}^{FV} \cdot \mathbf{n}^{FV}) \approx \mathcal{F}^*(\mathbf{u}_L^{FV}, \mathbf{u}_R^{FV}, \mathbf{n}^{FV}). \tag{25}$$

It depends on the normal vector \mathbf{n}^{FV} at the sub-cell interface and the sub-cell data right and left of the interface. A piecewise linear reconstruction with limiting is applied to get a second order total variation diminishing finite volume scheme on the sub-cell data. In all our simulations later we used simply the MinMod limiter [24].

4.3 Coupling of DG and FV Sub-cell Elements

Switching between DG and FV sub-cells as well as the flux computation at mixed DG/FV element interfaces requires a transformation between the polynomial solution representation of a DG element and a piecewise constant FV sub-cell representation. Therefore, we seek a transformation \mathbf{V}^{DG2FV} between a DG solution representation \mathbf{u}^{DG} of a degree N and a FV sub-cell solution representation \mathbf{u}^{FV} with N_{FV} sub-cells:

$$\mathbf{V}^{DG2FV} \mathbf{u}^{DG} = \mathbf{u}^{FV}. \tag{26}$$

A FV sub-cell representation \mathbf{u}^{FV} can be obtained from the polynomial DG solution representation \mathbf{u}^{DG} by piecewise projection of \mathbf{u}^{DG} on the N_{FV} sub-cells. With the piecewise constant FV sub-cell solution representation the projection can be written as the computation of the integral mean of \mathbf{u}^{DG} inside each sub-cell:

$$\int_E \mathbf{u} d\xi = \sum_{k=1}^{N_{FV}} \int_{-1+(k-1)w}^{-1+kw} \mathbf{u}^{DG} d\xi = \sum_{k=1}^{N_{FV}} \int_{-1+(k-1)w}^{-1+kw} \sum_{l=0}^N \hat{\mathbf{u}}_l^{DG} \psi_l(\xi) d\xi \stackrel{!}{=} \sum_{k=1}^{N_{FV}} \mathbf{w} \hat{\mathbf{u}}_k^{FV}. \tag{27}$$

For N_{FV} sub-cells, the width of a cell in reference space is given here as $w = \frac{2}{N_{FV}}$. Replacing the integral in Eq. (27) by a numerical quadrature with quadrature weights ω_m and quadrature nodes ξ_m^k inside the sub-interval k yields

$$\sum_{k=1}^{N_{FV}} \sum_{m=0}^N \sum_{l=0}^N \hat{\mathbf{u}}_l^{DG} \psi_l(\xi_m^k) \omega_m \stackrel{!}{=} \sum_{k=1}^{N_{FV}} \mathbf{w} \hat{\mathbf{u}}_k^{FV}. \tag{28}$$

For the numerical quadrature, the same nodes and weights are used as for the spatial DGSEM discretization. Equation (28) can be rewritten in matrix vector notation with the matrix \mathbf{V}^{DG2FV} defined as

$$\mathbf{V}^{DG2FV} := \frac{1}{2} \sum_{m=0}^N \psi_l(\xi_m^k) \omega_m \quad \forall l = 0, \dots, N, \quad \forall k = 1, \dots, N_{FV}. \tag{29}$$

To recover the polynomial DG representation \mathbf{u}^{DG} from piecewise constant sub-cell data \mathbf{u}^{FV} , an inverse transformation \mathbf{V}^{FV2DG} is required. For $N_{FV} + 1 > N$, this leads to an over determined system. This can be solved with a constrained least square approach as proposed by Dumbser and Zanotti in [8]. Integral conservation over the DG element serves herein as the constraint. The least square approach is equivalent to a discrete projection and is solved by finding the pseudo inverse of \mathbf{V}^{DG2FV} . Therefore both transformations fulfill the property

$$\mathbf{V}^{FV2DG} \circ \mathbf{V}^{DG2FV} = \mathbf{I}. \tag{30}$$

For two adjacent FV elements, the flux computation is straight forward and is performed on the piecewise constant sub-cell representation. If a mixed DG/FV interface is present, like in Fig. 2, we evaluate the flux on the sub-cell representation. Therefore, the solution on the surface of the DG element is transformed to a FV sub-cell representation via the matrix \mathbf{V}^{DG2FV} . Subsequently, the numerical flux is computed. Finally, the flux is transformed back to the DG representation using the inverse transformation matrix \mathbf{V}^{FV2DG} . As already discussed in Sect. 3.3, we advance our scheme in time with an explicit Runge–Kutta method and therefore need to fulfill a CFL time step condition. For a FV sub-cell with a physical

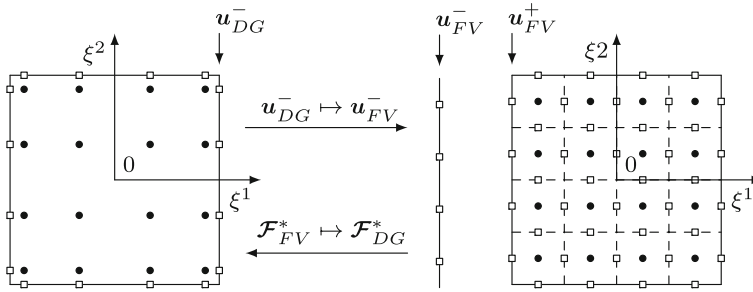


Fig. 2 DG reference element with degree $N = 3$ (left) and an adjacent DG element that is splitted into N_{FV}^2 FV sub-cells (right). DG volume nodes respectively FV sub-cell centers are indicated by dots. Surface nodes for the flux computation are indicated by squares. For the flux computation, the surface node distribution \mathbf{u}_{DG}^- is transformed to \mathbf{u}_{FV}^- to match the equidistant distribution of the FV sub-cells

element size Δx_{FV} and a piecewise constant solution representation of degree $N = 0$, the time step restriction (18) can be rewritten as

$$\Delta t = CFL \cdot \alpha_{RK}(0) \cdot \frac{\Delta x_{FV}}{|\lambda^c|} \tag{31}$$

As discussed in the work of Dumbser et al. [8], a comparison of both time step restrictions (18) and (31) yields that a FV sub-cell scheme does not suffer from a smaller time step than a corresponding DG discretization up to a sub-cell resolution of $N_{FV} = 2N + 1$. Therefore, an increased sub-cell resolution of $N_{FV} = 2N + 1$ compared to the resolution $N_{FV} = N + 1$ used in the open source FLEXI should not increase the total number of time steps of a simulation. This assumption is validated in Sect. 8, where the total number of time steps is compared for multiple test cases with different sub-cell resolutions.

5 Modal Decay Indicator for Shock Detection and Error Estimation

An essential building block for the proposed p-adaptive DG/h-refined FV sub-cell scheme is a suitable smoothness indicator and an error estimator. In the current framework, we use an a priori indicator based on the decay rate of the modal polynomial solution representation inside an element, as proposed by Mavriplis in [18]. In the following, the indicator is derived for an element e of degree N . A sufficiently regular function $u(\xi)$ can be represented in terms of an infinite series with the polynomial basis functions $\zeta_i(\xi)$ and the coefficients \hat{u}_i :

$$u(\xi) = \sum_{i=0}^{\infty} \hat{u}_i \zeta_i(\xi) = \sum_{i=0}^N \hat{u}_i \zeta_i(\xi) + \sum_{i=N+1}^{\infty} \hat{u}_i \zeta_i(\xi). \tag{32}$$

If we approximate $u(\xi)$ with a finite series expansion up to a polynomial degree of N , we introduce a truncation error that corresponds to the rightmost term in Eq. (32). If a modal polynomial basis is used, the coefficients can be interpreted as the amplitude of the solution modes. For a sufficiently smooth solution, this amplitude decays exponentially. In case of a non smooth solution, a slower decay occurs. That way, the decay rate can be regarded as both a measure of the truncation error as well as the smoothness of the solution. To determine the modal decay, the nodal polynomial approximation \mathbf{u}_{nod} is first transformed to a modal

representation \mathbf{u}_{mod} via the Vandermonde matrix V_{Leg} :

$$\hat{\mathbf{u}}_{mod} = V_{Leg} \hat{\mathbf{u}}_{nod}. \tag{33}$$

The polynomial approximation in terms of the modal coefficients $\hat{\mathbf{u}}_{mod,ijk}$ and the product of the one-dimensional Legendre basis functions $\zeta(\boldsymbol{\xi})_{i,j,k}$ can be written as

$$\mathbf{u}(\boldsymbol{\xi}) \approx \sum_{i,j,k=0}^N \hat{\mathbf{u}}_{mod,ijk} \zeta_{ijk}(\boldsymbol{\xi}), \quad \zeta_{ijk}(\boldsymbol{\xi}) = \lambda_i(\xi^1) \lambda_j(\xi^2) \lambda_k(\xi^3). \tag{34}$$

We now determine the relative contribution w_m of the m -th mode in ξ^1 direction to the solution as follows

$$w_m = \left[\frac{\left(\sum_{j,k=0}^N \hat{\mathbf{u}}_{mod,ijk}^2 \right)^{i=m}}{\sum_{i,j,k=0}^N \hat{\mathbf{u}}_{mod,ijk}^2} \right]. \tag{35}$$

As a final step, the relative modal contributions w_m are fitted to the exponential function $w_m = ae^{-\sigma^m}$. This yields the modal decay rate σ^1 in the ξ^1 direction. The decay rate is evaluated for every direction and the final indicator is given as the minimum of the absolute values

$$ind = \min(|\sigma^1|, |\sigma^2|, |\sigma^3|) \tag{36}$$

Since the indicator is based on the assumption that the magnitude of the modes, representing a smooth solution, decays exponentially, the decay rate in smooth regions correlates to the approximation error. A large decay rate indicates that the error due to the truncation of the higher modes is small and the polynomial degree could be decreased. A small decay rate indicates a high approximation error due to the truncation of the higher modes and motivates to increase the local polynomial degree. In the presence of shocks or sharp gradients, the solution tends to develop oscillations that cause a large magnitude of the highest modes, which manifests as a very low decay rate. To distinguish between the described scenarios, thresholds are required that are discussed in Sect. 7.3. The indicator can also be evaluated for FV elements, following a transformation of the FV sub-cell solution representation to a DG polynomial.

6 Efficient Implementation of the Adaptive Scheme

The presented adaptive DG scheme with FV sub-cell shock capturing is implemented as an extension to the FLEXI framework. FLEXI's DGSEM implementation is based on static arrays, which store the polynomial solution representation for elements, faces and fluxes. An array based implementation allows for efficient tensor operations and is well suited for a non-adaptive scheme, with a constant global polynomial degree N and a FV sub-cell shock capturing with $N_{FV} = N + 1$ sub-cells, as suggested in [27]. Such a scheme profits from a constant number of DOFs for both DG and FV elements. Therefore both the polynomial DG solutions representation and the FV sub-cell representation can be stored in a common array. On the other hand, a p-adaptive DG scheme with an arbitrary FV sub-cell resolution, requires a data structure that allows for a variable number of DOFs per element. With elements switching between a DG or FV representation and changing their polynomial degree, the

storage requirement per element changes over time. A DG element with a local polynomial degree N requires $(N + 1)^d$ DOFs, a FV element with N_{FV} sub-cells requires N_{FV}^d DOFs. Data of changing size is usually stored in dynamic data structures like linked lists. However, to maintain FLEXI's static data structure, we chose an array based approach. For every polynomial degree of a predefined range $N \in [N_{min}, N_{max}]$, arrays are allocated to store the polynomial solution representation at the start of the computation. Following a domain decomposition, every proc is assigned a subset of $nElems$ elements with $nSides$ sides. For an equation system with $nVar$ components, volume data, like the element local solution, is stored in arrays of size $nVar \times nElems \times (N + 1)^d$. Surface data, like the solution at element faces and the fluxes are stored in arrays of size $nVar \times nSides \times (N + 1)^{d-1}$. The FV sub-cell solution representation is stored in a separate array of size $nVar \times nElems \times N_{FV}^d$ for volume data and $nVar \times nSides \times N_{FV}^{d-1}$ for surface data respectively. Therefore, solution arrays for every possible setup are allocated at the start of the computation for the whole computational domain. By providing storage space for all possible degrees and the FV discretization simultaneously for every element, the memory demand for the solution representation is increased. On the other hand, this trade off in memory efficiency facilitates a straightforward extension of FLEXI towards an adaptive extension with no need to reallocate memory during the run time.

Independent of the chosen data structure for the solution, a significant increase in memory is caused by the metric terms that have to be precomputed and stored at the start of the computation for the range of allowed degrees and the FV sub-cell discretization. Since the metric computation is not trivial in case of curved meshes, a metric computation during run time would have a severe influence on the performance. Therefore, we decide to precompute the metrics and store them for the whole mesh at the start of the computation. Thus, the increased memory demand due to the proposed static data structure for the solution data is largely shadowed by the memory requirement of the metric terms. In practical applications, an increased memory demand might lead to limitations. Since the original memory footprint of the DGSEM is quite small, this limitation is not too severe on the current architectures. In Sect. 8, the memory requirement of the adaptive scheme is compared for different setups. An advantage of the chosen data structure is the good single core performance, when compared to the non-adaptive open source FLEXI, which is analyzed in more detail in Sect. 8.3.

7 The hp-Adaptive DGSEM Scheme

For simplicity we abbreviate the scheme, which consists of a p-adaptive DG scheme in smooth parts of the solution and a shock-capturing by a h-refined FV scheme on sub-cells, as hp-adaptive. In this section, we explain how the presented building blocks are assembled and discuss the free parameter of the algorithm and how they may be chosen.

7.1 Setup of the Adaptive Spatial Approximation

For the adaptive spatial approximation, a range of possible polynomial degrees $N \in [N_{min}, N_{max}]$, the degree at the start of the computation N_{ini} and the FV sub-cell resolution N_{FV} are to be chosen for a simulation. Depending on the minimum and maximum allowed polynomial degrees N_{min} and N_{max} , arrays for the volume and surface solution are allocated. The volume and surface solution of the FV sub-cell discretization are stored in separate arrays, allocated for the resolution N_{FV} . This static data structure allows for an

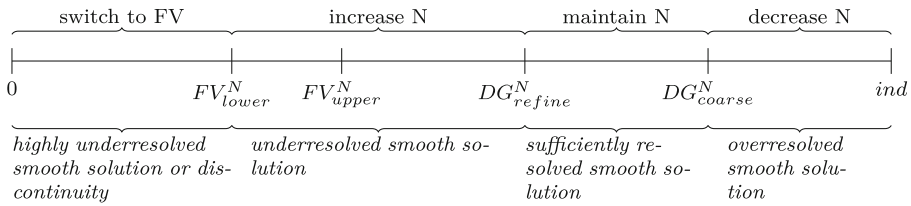


Fig. 3 For a DG element, four scenarios can be distinguished depending on the indicator value that corresponds to the modal decay rate: switch to FV for $ind < FV_{lower}^N$, increase N for $FV_{lower}^N < ind < DG_{refine}^N$, maintain N for $DG_{refine}^N < ind < DG_{coarse}^N$ and decrease N for $ind > DG_{coarse}^N$

efficient implementation without the need to reallocate memory during runtime. A drawback is the increased memory consumption compared to dynamic data structures, where only the memory needed for the element local polynomial degree is allocated. At the start of the simulation, precomputable building blocks like metric terms, node distributions, transformation matrices and interpolation matrices are precomputed and stored for polynomial degrees inside the allowed range. Thus, the overhead due to switching of the element local polynomial degree is reduced.

7.2 Interaction of Adaption and Time Stepping

Switching between the DG and FV sub-cell schemes and the adaption of the local polynomial degree is integrated in the explicit Runge–Kutta time integration. At the beginning of a time step, the indicator is computed for each element. The resulting indicator values are used to set the local polynomial degree for every DG element. After the p-adaption, the indicator has to be recomputed, since the computation of the indicator value depends on the local polynomial degree. Subsequently, the updated indicator value is used to switch elements between a DG and FV sub-cell discretization. DG elements containing shocks or discontinuities are detected and switched to a FV sub-cell representation. FV elements, where the indicator determines that the solution has sufficiently smoothed out are switched back to a DG representation. After each Runge–Kutta stage, the indicator is evaluated again to switch DG elements to FV elements to ensure a stable solution. The p-adaption is only performed at the beginning of a time step to reduce the overhead. Switching of FV elements to a DG representation is also omitted during the Runge–Kutta stages, since this could violate the time step restriction.

7.3 Parameter for the Shock and Refinement Indicator

The modal decay indicator has to fulfill two purposes. First, it has to determined which elements contain a smooth solution and which elements contain sharp gradients or shocks and flag them accordingly as DG or FV elements. Secondly, the polynomial degree is to be adjusted for each element based on an error estimate provided by the indicator in smooth regions. To choose the appropriate action depending on the indicator value, thresholds have to be defined. In the proposed framework, four thresholds exist for every allowed polynomial degree $N \in [N_{min}, N_{max}]$, FV_{lower}^N , FV_{upper}^N , DG_{refine}^N and DG_{coarse}^N . For a DG element of degree N , four scenarios can be distinguished as illustrated in the following Fig. 3:

- $ind > DG_{coarse}^N$: An indicator value above this threshold corresponds to a fast decay of the modes so that the highest modes have only a small contribution to the solution.

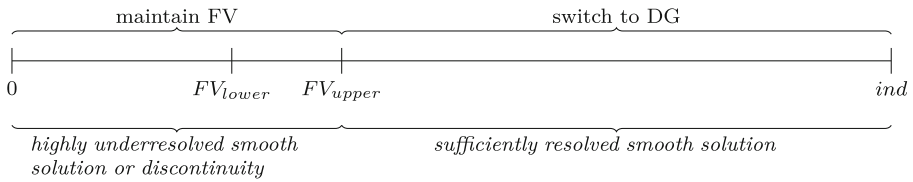


Fig. 4 For a FV element with an indicator value below FV_{upper}^N , the FV representation is maintained. An indicator value above FV_{upper}^N causes the element to switch back to a DG representation

In this case, given that the local polynomial degree is larger than the minimal allowed degree $N > N_{min}$, the local degree is reduced by one.

- $DG_{refine}^N < ind < DG_{coarse}^N$: For an indicator value between these thresholds the current polynomial degree is maintained.
- $FV_{lower}^N < ind < DG_{refine}^N$: An indicator value below DG_{refine}^N represents a large magnitude of the highest solution modes. This indicates an underresolved but smooth solution, since the indicator value is still above FV_{lower}^N . In case the element local degree is smaller than the maximum allowed degree $N < N_{max}$, the polynomial degree is increased by one.
- $ind < FV_{lower}^N$: In case of an oscillating solution polynomial, the highest modes have a large contribution to the solution and thus the modal decay rate is very small. An indicator value below the threshold FV_{lower}^N indicates an oscillating solution. This represents a non-smooth or severely underresolved underlying solution and therefore the DG element is switched to a FV sub-cell representation.

For elements already discretized with the FV sub-cell scheme, two actions are possible, depending on the indicator value as shown in Fig. 4:

- $ind < FV_{upper}^N$: An indicator value below this threshold indicates that the FV sub-cell solution should not be switched back to a DG representation since the corresponding solution polynomials exhibit oscillations.
- $ind > FV_{lower}^N$: An indicator value above this threshold indicates that the DG polynomial that corresponds to the current FV sub-cell solution is sufficiently smooth and the element can be switched back to a DG representation.

We use an upper and lower threshold with a margin for the DG/FV switch to avoid excessive switching that could occur for indicator values near the switching point. Since the indicator value depends on the current polynomial degree of the solution, thresholds for FV sub-cell limiting and p-refinement have to be defined for each degree. To reduce the number of free parameters, we assumed a linear dependency between the thresholds and the polynomial degree N . That way, only thresholds for the minimum and maximum allowed polynomial degrees N_{min} and N_{max} have to be defined. All remaining thresholds are then obtained by the following relation, here formulated for the FV thresholds:

$$FV_{lower}^N = FV_{lower}^{N_{min}} + N \frac{FV_{lower}^{N_{max}} - FV_{lower}^{N_{min}}}{N_{max} - N_{min}}. \tag{37}$$

A linear dependency is also assumed between the upper and lower thresholds. That way, the upper thresholds can be obtained by

$$FV_{upper}^N = FV_{lower}^{N_{min}} + (N + 1) \frac{FV_{lower}^{N_{max}} - FV_{lower}^{N_{min}}}{N_{max} - N_{min}}. \tag{38}$$

Through these practical relations, the number of free parameters is reduced to four: $FV_{lower}^{N_{min}}$, $FV_{lower}^{N_{max}}$, $DG_{refine}^{N_{min}}$ and $DG_{refine}^{N_{max}}$. These parameters were tuned empirically for the computations presented in Sect. 8.

8 Numerical Results and Discussion

In this section, we apply our hp-adaptive hybrid DG/FV sub-cell scheme to a succession of 1D, 2D, and 3D test cases to show its applicability and efficiency. Our investigation is focused on supersonic and transsonic flows that exhibit discontinuous flow features, like shocks as well as smooth areas and regions with intricate flow features, like shear layers with vortical structures. Computations are performed with three different setups that can be distinguished as follows:

- **Flexi-static:** A non-adaptive computation with a polynomial degree N and a FV sub-cell resolution of $N_{FV} = N + 1$, leading to a constant number of DOFs per element. This corresponds to the baseline FLEXI code without the improvements discussed herein.
- **Flexi-p-adaptive:** A p-adaptive computation with variable polynomial degree $N \in [N_{min}, N_{max}]$ and a FV sub-cell resolution of $N_{FV} = N_{Max} + 1$.
- **Flexi-hp-adaptive:** A p-adaptive computation with variable polynomial degree $N \in [N_{min}, N_{max}]$ and a refined FV sub-cell resolution of $N_{FV} = 2N_{Max} + 1$.

A central aspect of this section is to compare and discuss the quality of the results, the employed number of DOFs and the computation time for the listed setups. Additionally, free-stream preservation and a convergence test are performed to validate the coupling of DG cells with variable polynomial degree and the single core performance is analyzed. All computations were performed on a single core of a Xeon Gold 6140 processor if not stated otherwise. For all multidimensional examples, the visualisation is done with Paraview.

8.1 Free-Stream Preservation

To validate the proposed hp-adaptive hybrid DG/FV scheme, free-stream preservation is demonstrated in this subsection. This can be shown by imposing an initial solution $\mathbf{u} = (1, 1, 1, 1)^T$ on a computational domain Ω with periodic boundaries. The computational domain Ω is defined as a cube, that is distorted by the mapping

$$\mathbf{x} \mapsto \mathbf{x} + 0.1 \cdot \sin(x_1) \cdot \sin(x_2) \cdot \sin(x_3) \quad (39)$$

and the curved geometry of the elements are approximated with a polynomial degree of $N_{geo} = 2$. To prove free-stream preservation for the key features of the hybrid scheme, the domain Ω is divided along the x-axis in two parts. In one half, the solution is discretized with a p-adaptive DG scheme, where elements with a degree of $N = 2$ and $N = 7$ are placed in a checkerboard like pattern. The second half is discretized with a FV sub-cell scheme with a sub-cell resolution of $N_{FV} = 2N_{max} + 1 = 15$. The resulting domain is visualized in Fig. 5. The computation is run until the final time $t = 0.5$ is reached, which amounts to 165 time steps. Subsequently, the L_2 and L_∞ error norms are evaluated and listed in Table 1. Since the error norms lie in the order of machine precision, we can conclude that our proposed scheme satisfies free-stream preservation on curved meshes.

Fig. 5 Distorted cubic domain used for the free-stream preservation test. The left half is discretized with DG elements of degree $N = 2$ and $N = 7$ and in the left half, a FV sub-cell discretization is used with $N_{FV} = 15$ sub-cells per direction

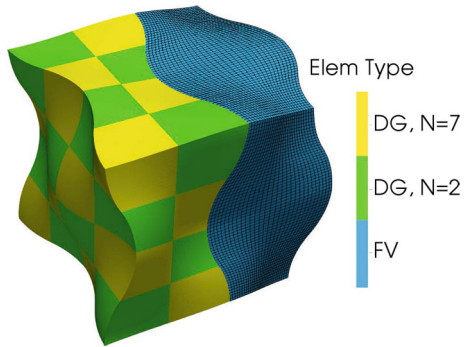


Table 1 L_2 and L_∞ error of a free-stream computation on a curved mesh with a hybrid DG/FV discretization and a variable polynomial degree $N = [2, 7]$ after 165 time steps

	u_1	u_2	u_3	u_4	u_5
L_2	$7.79e^{-16}$	$1.34e^{-15}$	$1.09e^{-15}$	$1.08e^{-15}$	$3.37e^{-15}$
L_∞	$9.10e^{-14}$	$1.17e^{-13}$	$6.35e^{-14}$	$9.16e^{-14}$	$2.89e^{-13}$

8.2 Experimental Order of Convergence

In this subsection, we validate our p-adaptive DGSEM implementation by investigating the error convergence experimentally. For the non-adaptive case, the order of convergence of the applied DGSEM and FV sub-cell scheme was already studied in [27]. Therefore, we restrict our analysis to the p-adaptive case. The convergence measurement is performed with a simple test case of a periodic diagonal density sine wave, which is advected in the direction $(1, 1, 1)^T$ through a Cartesian box $\Omega = [-1, 1]^3$. Since the problem is perfectly smooth, no FV sub-cells are necessary and we can compare the results against the expected order of convergence $p = N + 1$, given by the truncation error $e \approx kh^p$. All tests are performed with two different element local polynomial degrees that are distributed inside the cube in a checkerboard like pattern. The higher of the two employed degrees is chosen to be $N_{Max} = 6$, the lower one varied between $N_{Min} = [2, 5]$. To avoid overwhelming of the spatial discretization error by the error of the time integration, a small CFL number of 0.05 is used. The resulting L^2 and L^∞ errors of the density and the order of L^2 and L^∞ convergence are listed in Table 2. As expected, the theoretical order of convergence of the lowest present polynomial degree is matched perfectly.

8.3 Single Core Performance

The motivation for a hp-adaptive approach is to reduce the overall number of DOFs and thus the cost of a given computation. Therefore, it is imperative to keep the overhead, caused by the more complex algorithm and data structure, to a minimum. Since our adaptive code is based on the non-adaptive fast open source code FLEXI, the goal is to retain its original performance as closely as possible. The variable number of DOFs per element causes a significant load imbalance, in case of parallel computations. The implementation of an effective load balancing is a challenging task, and is work in progress. Therefore, we restrict ourselves

Table 2 Errors and convergence rates of the density for a diagonally advected sinus for a p-adaptive DG method

Poly. deg.	Cells	L^2 error	L^2 order	L^∞ error	L^∞ order	Theor. order
N=2	4	6.41e-03		2.04e-02		3
	8	7.43e-04	3.11	8.24e-03	1.31	
	16	9.46e-05	2.97	1.19e-03	2.79	
	32	1.18e-05	3.00	1.49e-04	2.99	
	64	1.48e-06	3.00	1.84e-05	3.02	
N=3	4	5.34e-04		4.23e-03		4
	8	3.51e-05	3.93	6.22e-04	2.76	
	16	2.25e-06	3.96	4.51e-05	3.78	
	32	1.42e-07	3.98	2.88e-06	3.97	
	64	8.95e-09	3.99	1.83e-07	3.98	
N=4	4	4.35e-05		4.04e-04		5
	8	1.21e-06	5.17	1.74e-05	4.54	
	16	3.84e-08	4.97	5.59e-07	4.96	
	32	1.21e-09	4.99	1.74e-08	5.01	
	64	3.81e-11	4.99	5.31e-10	5.03	
N=5	4	3.27e-06		3.23e-05		6
	8	5.29e-08	5.95	5.83e-07	5.79	
	16	8.28e-10	6.00	9.73e-09	5.90	
	32	1.29e-11	5.99	1.54e-10	5.98	
	64	2.06e-13	5.98	2.67e-12	5.96	

Polynomial degrees are tested in pairs for the following combinations ($N = [2, 5], N = 6$)

in this paper to analyzing single core performance. For all measurements, the code was compiled with the *GNU compiler* version 8.3.0 and the optimization flag O3. To quantify the performance of the adaptive code compared to the static open source code, we measure the performance index (PID) of both codes for different setups. The PID is defined as the mean compute time required to update one DOF for one time step, including the Runge–Kutta stages, and computed as follows:

$$PID = \frac{\text{wall-clock-time} \cdot \text{cores}}{\text{DOF} \cdot \text{time steps}}$$

We perform the PID measurements on a curved, three-dimensional periodic computational domain Ω that is defined as a distorted cube with the mapping (39) from Sect. 8.1. As an initial condition, a sine shaped density distribution is imposed. The density distribution is then convected diagonally along the direction $(1, 1, 1)^T$. PID measurements are performed for three different setups, illustrated in Fig. 6. In a first step, we measure the PID of the adaptive (hp-FLEXI) and non-adaptive (os-FLEXI) code for the *same* constant polynomial degrees in a range between $N = 2, 7$. We want to emphasize that even for the test with constant polynomial degree, the hp-FLEXI is initialized with a data structure that allows for computations up to a polynomial degree $N_{Max} = 7$. From the measured results, visualized in Fig. 7, we conclude that the PID of the adaptive code is only slightly larger than the PID of the original, non-adaptive implementation. Particularly for degrees above 4, almost the same performance is observed. The largest difference is present for degree $N = 2$, where

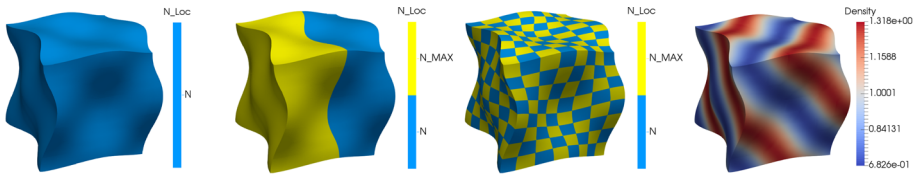


Fig. 6 Curved periodic domain with an initial density distribution given as a diagonal sine shaped wave (right). Three different discretizations are used for the PID measurement: a uniform polynomial degree N (left), one half with degree N and the other half with N_{Max} (middle) and a checkerboard like distribution of the polynomial degrees N and N_{Max}

the adaptive code is about 15% slower. In case of a varying local polynomial degree among adjacent elements, transformation operations, described in Sect. 3.3, are necessary to compute the fluxes. This introduces additional operations per DOF and an increased PID has to be expected. To measure the influence of the transformations, computations were performed with two different distributions of polynomial degrees. First the domain is split into two halves and for a second test a checkerboard like pattern was used. In both cases, computations were performed with two different polynomial degrees. The larger one was fixed to $N = 7$ and the smaller varied between $N = [2, 7]$. The influence of the transformations is clearly visible in Fig. 7. For a half-half distribution, an almost constant PID can be observed that is at most 15% slower than the PID of the open source code. In contrast, the checkerboard distribution causes a significant increase of the PID up to 80%. The data point at $N = 7$ matches the open source code since here all elements are of degree 7. We can conclude from the measured data that the overhead of the adaptive code is minor, as long as the number of interfaces between cells of different ansatz degree is small. We note that for this comparison, the checkerboard distribution constitutes the worst case scenario, as each element has neighbors with a different operator, while the half and half distribution is the least severe case. In practise, we thus expect the true performance to lie between these two extremes. It is also clear that this single metric is not sufficient to judge the overall performance of the hp-adaptive version, as we will show later on. In the following, we employ our framework to a range of test cases of increasing complexity and compare it against the os-Flexi in terms of accuracy and efficiency.

8.4 Sod's Shock Tube Problem

The initial conditions of the well known shock tube problem of Sod consist of two constant states, separated by a discontinuity in the middle of the domain. The initial conditions are $\rho = 1, u = 0, p = 1$ on the left side and $\rho = 0.125, u = 0, p = 0.1$ on the right side. The one-dimensional computational domain $\Omega = [0, 1]$ is discretized with 10 elements. At the boundaries, Dirichlet boundary conditions are applied. For both shock detection and p-refinement, the modal decay indicator from Sect. 5 is applied to the density and tuned with thresholds set to $[DG_{refine}^{Nmin}, DG_{refine}^{Nmax}] = [3.5, 5.0]$ and $[FV_{lower}^{Nmin}, FV_{lower}^{Nmax}] = [2.4, 3.3]$. As a Riemann solver, the approximate Roe solver with an entropy fix is used [10]. Two computations with different setups are performed and compared against the exact solution, which is e.g. given in [29]. First a global polynomial degree $N = 5$ and a FV sub-cell resolution of $N_{FV} = N + 1 = 6$ are applied. A second, p-adaptive computation is performed with a polynomial degree $N \in [2, 5]$ and a FV sub-cell resolution of $N_{FV} = 2N + 1 = 11$. The resulting density distributions are visualized in Figs. 8 and 9. Both computations show good agreement with the exact solution and only apply FV sub-cells at the shock, which is

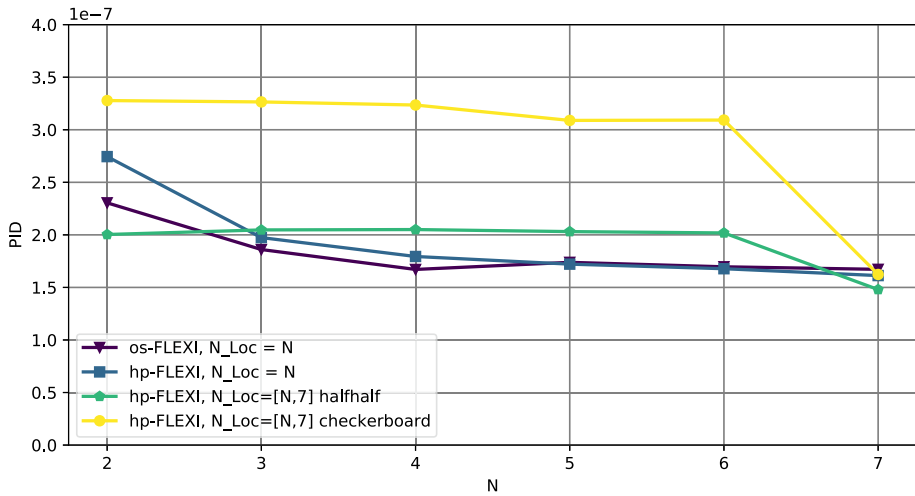


Fig. 7 Performance index for computations with polynomial degrees between $N = 2$ and $N = 7$ with the open source FLEXI and our p-adaptive FLEXI extension. Different distributions of the element local degree are applied for p-adaptive computations

successfully detected. The p-adaptive computation resolves the shock more sharply, due to the higher sub-cell resolution, which can be observed in Fig. 8. A slightly increased polynomial degree is applied at the rarefaction wave and the maximal degree is only applied at the contact discontinuity. This is clearly a desired feature, and shows the usefulness of our indicator strategy discussed in Sect. 7.3. A good agreement between the p-adaptive and globally refined computations can be observed, apart from the shock, which is better resolved in case of the adaptive computation. With an average of 151.6 DOFs per element, the adaptive computation required about 25% less DOFs per element when compared to the globally refined computation with 216.0 DOFs per element. Therefore the adaptive computation achieved a better result with less DOFs.

8.5 Shu-Osher Density Fluctuations Shock Wave Interaction Problem

With the density fluctuation shock wave interaction problem proposed by Shu and Osher in [25], we investigate the shock capturing and p-refinement abilities of the adaptive code for a more demanding one-dimensional test case. The setup features a domain $\Omega \in [-5, 5]$, discretized with 100 elements that contains a shock at $x = -4$. States left and right of the shock are given as initial conditions

$$(\rho, u, p) = \begin{cases} (3.857143, 2.629369, 10.33333) & x \leq -4 \\ (1.0 + 0.2 \cdot \sin(5x), 0.0, 1.0) & x \geq -4 \end{cases}$$

and the boundary values are defined through Dirichlet boundary conditions. The resulting flow phenomena can be described as a Mach 3 shock front, traveling into a density field with a sinusoidal distribution. As a reference solution we performed a second order FV computation on a very fine mesh. The results for a final time of $t = 1.8$ are visualized in Fig. 10. In the wake of the shock, high frequency density fluctuations are present, as well as weak secondary shocks further downstream. We compare a computation with a uniformly

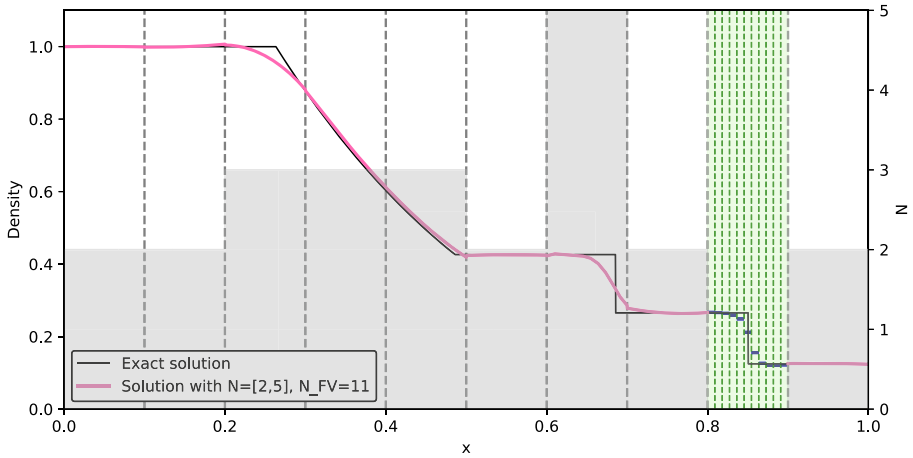


Fig. 8 Density distribution of the Sod shock tube problem at $t=0.2$, computed on a mesh with 10 grid cells with a variable polynomial degree $N \in [2, 5]$ and a FV sub-cell resolution of $N_{FV} = 11$. When compared to the non-adaptive computation with $N + 1$ sub-cells, the adaptive computation with an increased sub-cell resolution of $2N + 1$ provides a less dissipative solution at the shock

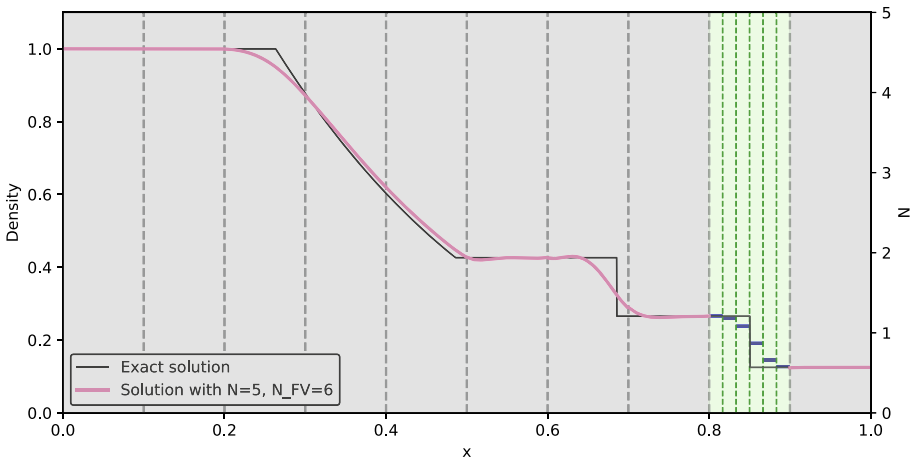


Fig. 9 Density distribution of the Sod shock tube problem at $t=0.2$, computed on a mesh with 10 grid cells with a constant polynomial degree $N = 5$ and a FV sub-cell resolution of $N_{FV} = 6$

high polynomial degree $N = 5$ and a FV sub-cell resolution of $N_{FV} = N + 1 = 6$ with a p-refined computation with $N \in [2, 5]$ and a refined FV sub-cell resolution of $N_{FV} = 2N + 1 = 11$. The approximate Roe solver is used as a Riemann solver for all setups. Again, the modal decay indicator from Sect. 5 is applied on the density for both shock capturing and p-refinement. To tune the indicator, we define the thresholds $[DG_{refine}^{N_{min}}, DG_{refine}^{N_{max}}] = [4.5, 6.0]$ and $[FV_{lower}^{N_{min}}, FV_{lower}^{N_{max}}] = [3.6, 4.5]$. The shock indicator behaves equivalently for both computations and successfully detects the primary shock, as well as the weak shocks that form downstream. A perfect behavior of the p-refinement can be observed, since only the high frequency waves and the first of the weak shocks are computed with the maximal polynomial degree. In regions with constant density or with low frequency sinus waves the minimal

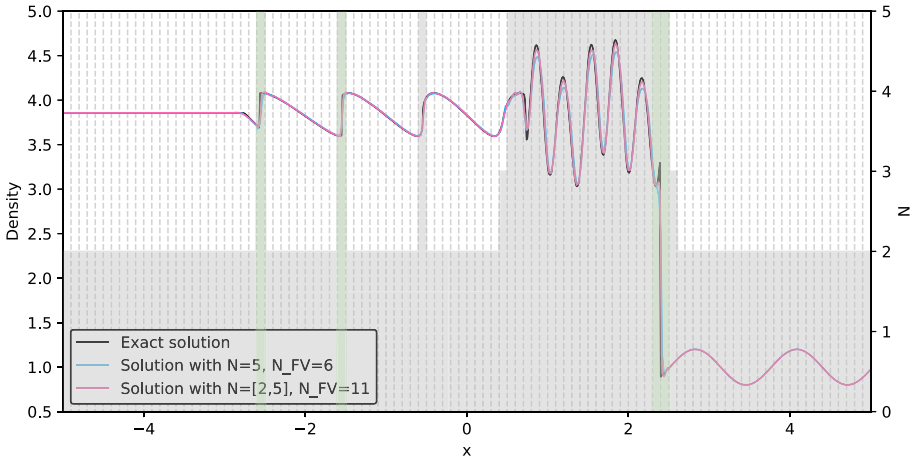


Fig. 10 Shu-Osher problem at $t = 1.8$, computed on a mesh of 100 grid cells. A p-adaptive computation with a refined sub-cell mesh is compared to a non-adaptive computation with a constant number of DOFs per element

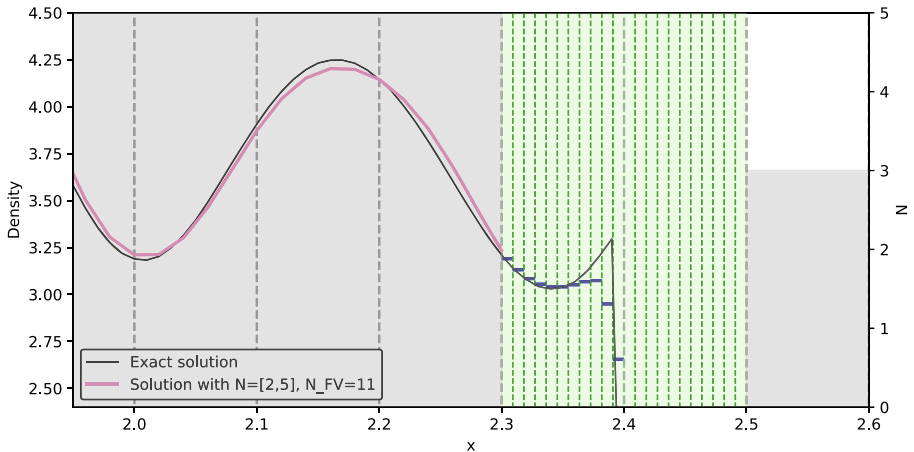


Fig. 11 Close up view of the high frequency density fluctuation and the primary shock of the Shu-Osher problem at $t = 1.8$. A close agreement of the adaptive computation with the reference solution can be observed due to the fine FV sub-cell resolution of $N_{FV} = 2N + 1$

polynomial degree is chosen. As a result of the well distributed element local polynomial degree of the adaptive computation, a good agreement with the uniform computation is achieved. Through the high FV sub-cell resolution, the primary and secondary shocks are resolved more sharply and with less dissipation. Therefore, the adaptive computation matches the exact solution better, especially at the high frequency waves immediately after the shock, visible in Fig. 11. The more accurate result, provided by the adaptive computation, was achieved with a mean of 70.8 DOFs per element, whereas the non-adaptive computation used 216 DOFs per element.

8.6 Double Mach Reflection

The double Mach reflection (DMR) problem was proposed by Woodward and Colella in [32] and has become a standard benchmark. It can be interpreted as a horizontally traveling Mach 10 shock wave that encounters a ramp with an inclination of 30° . By the Rankine–Hugoniot conditions the initial conditions are obtained as

$$(\rho, v_1, v_2, v_3, p) = \begin{cases} (8.0, 8.25 \cdot \cos(30^\circ), -8.25 \cdot \sin(30^\circ), 0.0, 116.5) & x \leq x_0 + \sqrt{\frac{1}{3}}y \\ (1.4, 0.0, 0.0, 0.0, 1.0) & x \geq x_0 + \sqrt{\frac{1}{3}}y \end{cases}$$

with $x_0 = \frac{1}{6}$ and the heat capacity ratio $\gamma = 1.4$. The computational domain $\Omega = [0, 4] \times [0, 1]$ is discretized with $240 \times 60 \times 1$ elements corresponding to a characteristic mesh size of $h = \frac{1}{60}$. In x-direction, inflow and outflow conditions are imposed. In y-direction, the bottom boundary is modelled as a reflecting wall and the upper boundary is given by the exact solution of an oblique shock traveling at Mach 10. Since we compute the 2D problem with a 3D code, the z-direction is discretized with one element layer and periodic boundary conditions are imposed. The problem is advanced in time until the final time $t = 0.28$ is reached. As a Riemann solver, we used a HLLC solver and perform shock capturing with the modal decay indicator from Sect. 5 that is applied on the pressure. The same indicator is also used to determine the polynomial degree for p-adaptive computations. The indicator is tuned with thresholds defined as $[DG_{refine}^{N_{min}}, DG_{refine}^{N_{max}}] = [5.6, 6.5]$ and $[FV_{lower}^{N_{min}}, FV_{lower}^{N_{max}}] = [3.2, 4.1]$. Three computations with the numerical setups, listed in Table 3, are performed. First the non-adaptive setup *Flexi-static* is applied, to obtain a reference solution. It is compared against results, obtained by the p-adaptive setups *Flexi-p-adaptive* and *Flexi-hp-adaptive*. The comparison between *Flexi-p-adaptive* and *Flexi-hp-adaptive* demonstrates the effect of an increased FV sub-cell resolution on the solution. In Fig. 12, the density distribution with equidistant contour lines is visualized for the three cases at the final time. The underlying DG and FV sub-cell mesh is included in the plots, to illustrate the coarse DG mesh and the different FV sub-cell grids. To highlight shocks, strong gradients and shear layers, schlieren images are provided in Fig. 13. The distribution of FV elements and the local polynomial degree can be seen in Fig. 14. Our code successfully detects the shock fronts and resolves them with FV sub-cells. Equal behavior of the indicator can be observed for all three cases. As expected, shocks are resolved more sharply for a higher FV sub-cell resolution, best visible in Fig. 13. To resolve the vortical structures at the primary slip line, a high resolution is necessary. This is successfully detected by the modal decay indicator, and the local polynomial degree is increased along the primary slip line. In a narrow band around the shocks, p-refinement is performed as well. For the remaining domain, a local polynomial degree of $N = 2$ is used. This is in line with our indicator strategy in Sect. 7.3. Since the areas where a high polynomial degree is necessary are detected well, the results of the p-adaptive computation matches the non-adaptive, globally refined simulation well. The vortical structures at the slip line are very similar and only slightly better resolved for the non-adaptive case. As already discussed, the third computation resolves shocks more sharply, due to the higher sub-cell resolution. This results in a minor deviation of the position of shocks and triple points. Vortical structures on the primary slip stream, near the primary triple point are resolved even better for this setup, when compared to the non-adaptive result. We note that an FV/DG interface can trigger acoustic waves. This is a result of the different resolution capabilities and numerical transfer functions of the DG and FV operator and is most pronounced for the cases with $N_{FV} = N + 1$, as the difference

Table 3 Comparison of wall time, average number of DOFs per element, total number of time steps and memory requirement between the non-adaptive and p-adaptive computations

	Flexi-static	Flexi-p-adaptive	Flexi-hp-adaptive
Resolution	$N = 5, N_{FV} = 6$	$N \in [2, 5], N_{FV} = 6$	$N \in [2, 5], N_{FV} = 11$
DOFs per element	216	35	54
Time steps	4695	4726	4765
Wall time [s]	15,578	3856	6569
Memory [GB]	4.9	5.9	19.0

in resolution between FV and DG is largest. The thus generated disturbances are damped by the reduction in polynomial degree away from the shock in the p-adaptive FLEXI version. In regions with the maximal polynomial degree, acoustics are damped far less, which is a result of the low dissipation error of the high order DG scheme. This is best observed “inside” the DMR region in Fig. 13. Increasing the FV resolution to $N_{FV} = 2N + 1$ almost completely avoids the generation of these artefacts, since the jump of the resolution between a DG and FV element is decreased. Therefore the third setup, provides the solution with the least noise. The primary advantage of the adaptive computations is the reduced computational effort. Table 3 lists the mean number of DOFs per element, the computation time, number of time steps and the memory requirement for the different setups. The computation *Flexi-p-adaptive* achieved a speedup of a factor 4 and the computation *Flexi-hp-adaptive* with an increased FV sub-cell resolution still achieved a speedup of almost a factor 2.5. We note that although the reduction in DOF alone might suggest a larger reduction in computing time, the loss in operator efficiency, investigated in Sect. 8.3, balances this to some degree. However, in all practical applications investigated by us, a significant overall gain in wall time remained. When comparing the memory footprint of the different setups, the following can be observed: the *Flexi-p-adaptive* scheme requires about 20% more memory than the *Flexi-static* computation and the *Flexi-hp-adaptive* setup increases the memory requirement by a factor of 4. A similar behavior can be observed for all presented setups. As discussed in Sect. 6, the main reason for the increased memory requirement lies with the storage of metric terms and the solution for all allowed polynomial degrees and the FV sub-cell discretization at the same time.

8.7 Forward Facing Step

With the forward facing step (FFS) we consider a second test problem by Woodward and Colella [32] that models a Mach 3 wind tunnel with a step. The computational domain $\Omega = [0, 3] \times [0, 1] \setminus [0.6, 3.0] \times [0, 0.2]$ is initialized with free-stream conditions defined as

$$(\rho, v_1, v_2, v_3, p) = (1.4, 3.0, 0.0, 0.0, 1.0)$$

and the heat capacity ratio $\gamma = 1.4$. Inflow and outflow conditions are imposed in x-direction and reflective walls are used to describe the channel walls in y-direction. Since the two-dimensional test case is again computed with a three-dimensional setup, periodic boundary conditions are applied in z-direction. We discretize the domain with $180 \times 60 \times 1 - 144 \times 12 \times 1$ elements, corresponding to a characteristic length $h = \frac{1}{60}$. As an approximate Riemann solver, the HLLE is used.

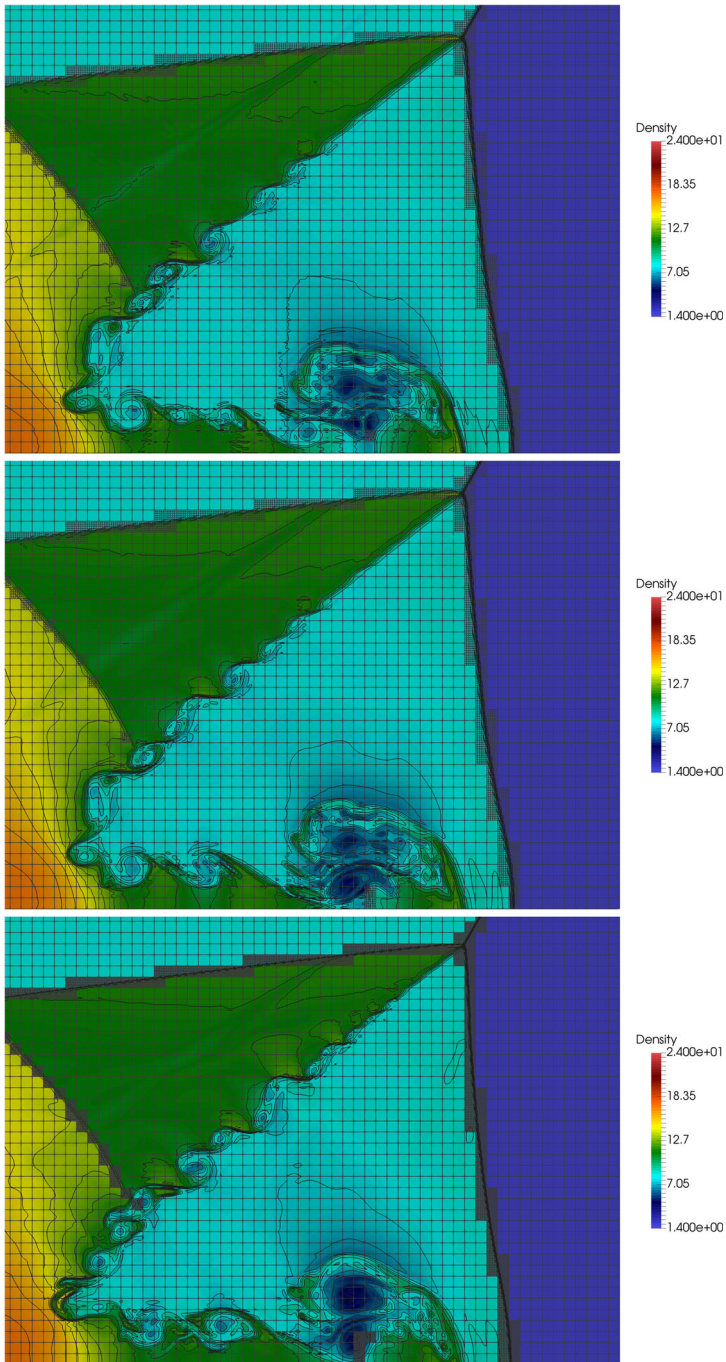


Fig. 12 Results for the double Mach reflection problem computed with a DG method and limited by FV sub-cells on $240 \times 60 \times 1$ elements. The density with equidistant contour lines is displayed for the three different computational settings *Flexi-static* (top), *Flexi-p-adaptive* (middle) and *Flexi-hp-adaptive* (bottom)

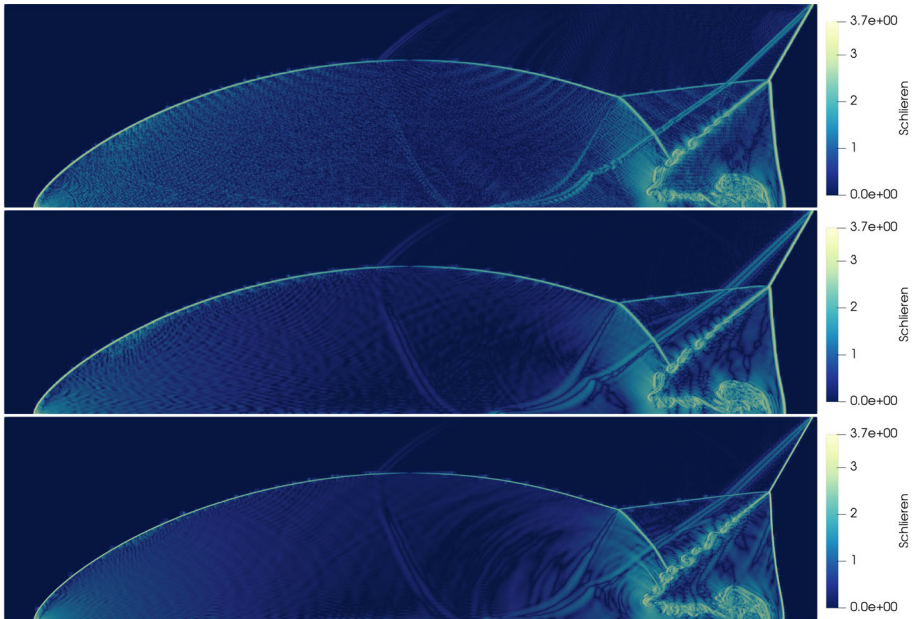


Fig. 13 Resulting schlieren images, computed as $\log_{10}(|\nabla\rho| + 1)$, for the double Mach reflection problem and three different computational setting *Flexi-static* (top), *Flexi-p-adaptive* (middle) and *Flexi-hp-adaptive* (bottom)

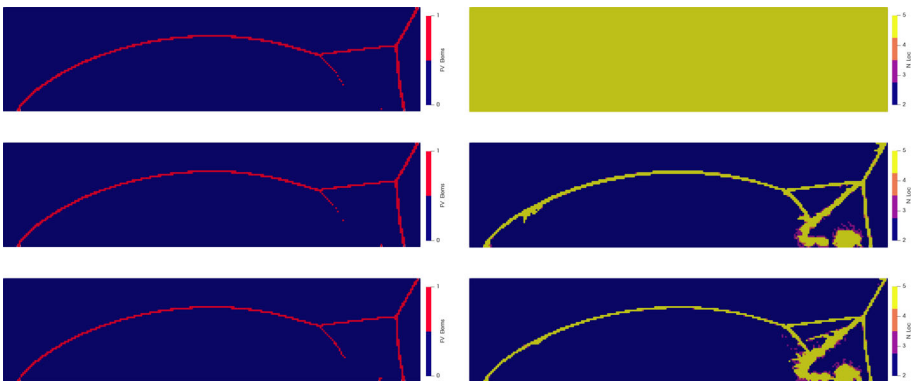


Fig. 14 FV sub-cell distribution (left) and element local polynomial degree (right) used for the DMR computation with the setups *Flexi-static* (top), *Flexi-p-adaptive* (middle) and *Flexi-hp-adaptive* (bottom)

Shock capturing, as well as p-refinement is controlled by the modal decay indicator from Sect. 5, operating on the density. The corresponding thresholds are given as $[DG_{refine}^{N_{min}}, DG_{refine}^{N_{max}}] = [4.5, 6.0]$ and $[FV_{lower}^{N_{min}}, FV_{lower}^{N_{max}}] = [3.0, 4.2]$. To demonstrate and assess the performance of our adaptive code, compared to the non-adaptive version, we use the setups from Sect. 8.6 again. In Table 4, the polynomial degrees and FV sub-cell resolutions for the three cases are given. The density distributions with equidistant contour lines are visualized in Fig. 15. Shocks, sharp gradients and shear layers are highlighted in the schlieren images in Fig. 16. FV sub-cells and the distribution of the local polynomial degree

Table 4 Comparison of wall time, average number of DOFs per element, total number of time steps and memory requirement between the non-adaptive and p-adaptive computations

	Flexi-static	Flexi-p-adaptive	Flexi-hp-adaptive
Resolution	$N = 5, N_{FV} = 6$	$N \in [2, 5], N_{FV} = 6$	$N \in [2, 5], N_{FV} = 11$
DOFs per element	216	44	82
Time steps	19,969	19,136	19,993
Wall time [s]	65,782	19,774	40,217
Memory [GB]	4.8	5.8	18.7

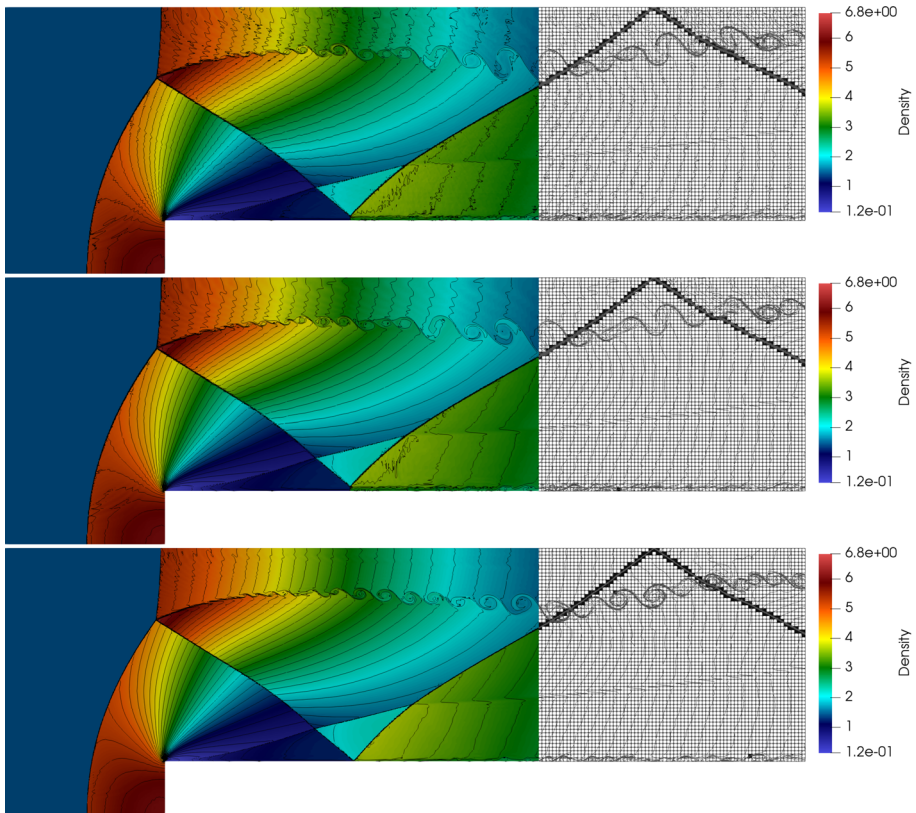


Fig. 15 Forward facing step problem computed with a DG method and limited by FV sub-cells. The density with equidistant contour lines is given for the three computational setups *Flexi-static* (top), *Flexi-p-adaptive* (middle) and *Flexi-hp-adaptive* (bottom). In the right part of the plots, the DG mesh and the FV sub-cell grids are visualized

are depicted in Fig. 17. The shock fronts, interacting with the channel boundaries are successfully detected and resolved with FV sub-cells, as can be seen in Fig. 17. Only very few FV elements are placed at the Kelvin-Helmholtz instabilities, that develop along the shear layer after the primary triple point. This demonstrates a very good behavior of the modal decay indicator for shock capturing. The schlieren images in Fig. 16 showcase the influence of the increased FV sub-cell resolution in the third computational setup. Here, shocks are resolved

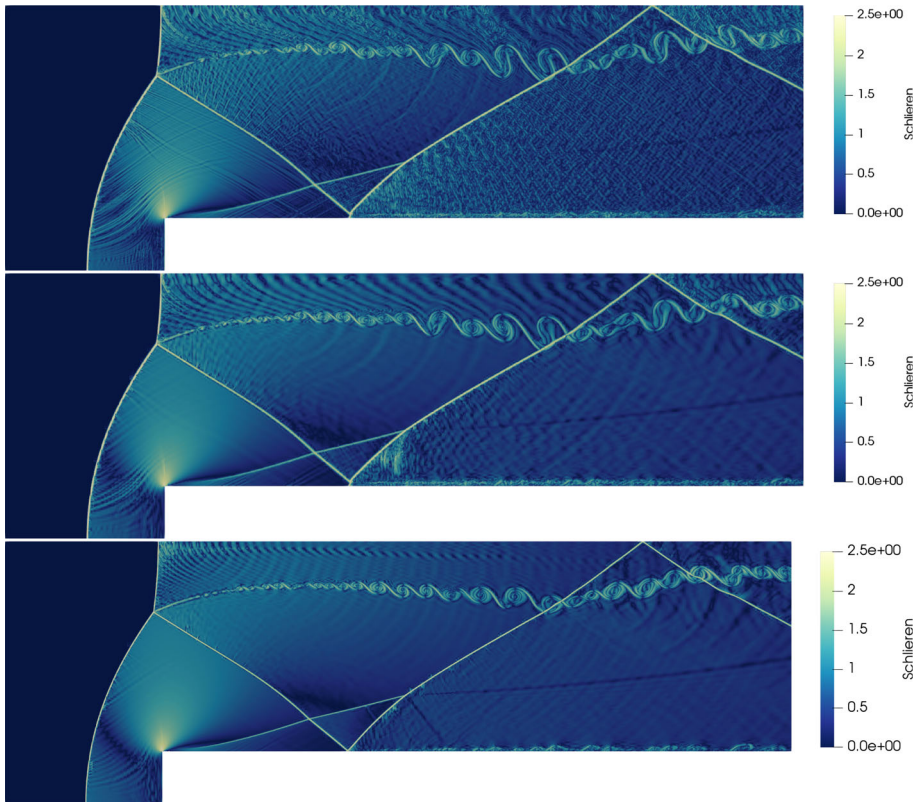


Fig. 16 Resulting schlieren images, computed as $\log_{10}(|\nabla\rho| + 1)$, for the forward facing step problem and the three computational setups *Flexi-static* (top), *Flexi-p-adaptive* (middle) and *Flexi-hp-adaptive* (bottom)

much sharper and the interface between FV and DG elements produced less artefacts and less noise. For the p-refinement, the indicator has to detect areas where a high resolution is required. The resulting distribution of element local polynomial degrees shows that a high order is applied at the sharp edge of the step, in the direct vicinity of shocks, in the shear layer near the channel floor and especially at the Kelvin-Helmholtz instabilities of the upper shear layer. The density distribution at the final computation time and the corresponding schlieren images show, that the adaptive setup *Flexi-hp-adaptive* reproduces the solution of the non-adaptive *Flexi-static* setup very well. Results obtained with *Flexi-p-adaptive* are only slightly more dissipative and the vortical structures at the Kelvin-Helmholtz instability are very similar. Like already observed in the DMR test case, a finer FV sub-cell grid resolves shocks better, which in turn results in a minor deviation of the positioning of shock fronts and triple points. Therefore, a slightly different pattern of vortices is generated at the Kelvin-Helmholtz instability. Structures near the primary triple point appear even richer than in the non-adaptive, globally refined case. Acoustic waves, that are generated by the vortical structures, are present at the upper shear layer and the shear layer near the channel bottom and are mostly ignored by the p-refinement indicator and damped. This can be seen best at the density contour lines and in the schlieren images. While providing comparable and in parts even better results, the adaptive computations required in average far less DOFs per

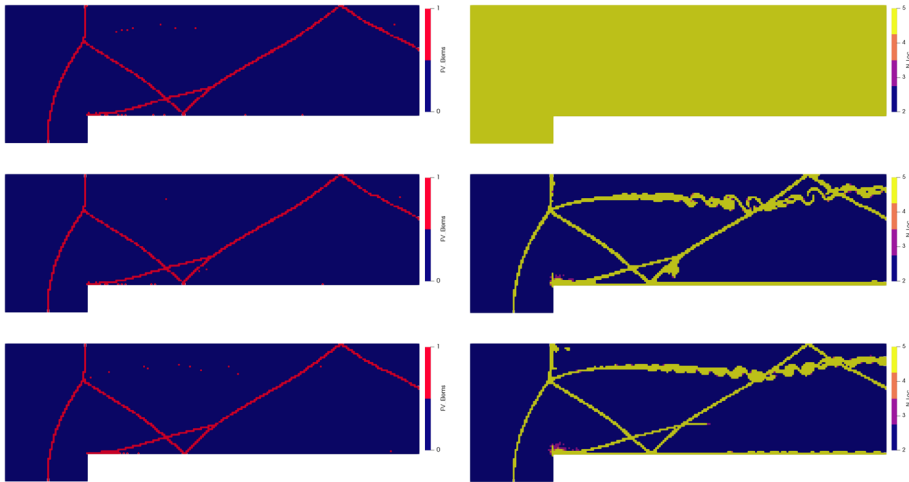


Fig. 17 FV sub-cell distribution (left) and element local polynomial degree (right) used for the forward facing step computation with the setups *Flexi-static* (top), *Flexi-p-adaptive* (middle) and *Flexi-hp-adaptive* (bottom)

element and therefore less computation time. From Table 4, we can deduce a speed up of almost a factor 3 for the setup *Flexi-p-adaptive* and still a factor of about 1.6 for the setup *Flexi-hp-adaptive*, where an increased FV sub-cell resolution was applied additionally.

8.8 Airfoil at Transsonic Conditions

In this subsection, we simulate the flow around an airfoil at transsonic conditions to validate our framework against a more realistic setup. The airfoil is placed in the computational domain Ω , that is discretized with 8156 hexahedral elements. The simulation is initialized with Mach 0.73 freestream conditions, defined as

$$(\rho, v_1, v_2, v_3, p) = (1.0, 0.998, 0.061, 0.0, 1.340)$$

and the heat capacity ratio $\gamma = 1.4$. The airfoils surface is modeled as a reflecting wall boundary and inflow and outflow conditions are imposed. Again, we perform a 3D computation for a 2D problem and apply periodic boundary conditions in the z -direction. Shock capturing and p-refinement is performed with the modal decay indicator applied on the pressure with the thresholds $[DG_{refine}^{N_{min}}, DG_{refine}^{N_{max}}] = [5.5, 6.05]$ and $[FV_{lower}^{N_{min}}, FV_{lower}^{N_{max}}] = [3.0, 4.0]$. As a Riemann solver, the approximate Roe solver is used. We compute the test case with the three different numerical setups *Flexi-static*, *Flexi-p-adaptive* and *Flexi-hp-adaptive* and compare the results. The exact setups are provided in Table 5. Results at the final time $t = 10$ are provided in Fig. 18. Both the density distribution with equidistant contour lines and corresponding schlieren images are on display. A region with supersonic flow conditions and a shock are visible above the airfoil. Vortex shedding can be observed at the trailing edge and in the wake. The underlying distribution of FV sub-cells and the element local polynomial degree is visualized in Fig. 19. For all three setups, FV sub-cells are positioned exclusively at the shock and directly at the trailing edge. The influence of the increased FV sub-cell resolution in the case of the third computational setup is highlighted in the schlieren image. The shock is resolved significantly sharper and vortical structures in the wake are richer and

Table 5 Comparison of wall time, average number of DOFs per element, total number of time steps and memory requirement between the non-adaptive and p-adaptive computations

	Flexi-static	Flexi-p-adaptive	Flexi-hp-adaptive
Resolution	$N = 4, N_{FV} = 5$	$N \in [2, 4], N_{FV} = 5$	$N \in [2, 4], N_{FV} = 9$
DOFs per element	125	30.5	32.5
Time steps	176,737	61,589	61,410
Wall time [s]	10,160	24,692	27,149
Memory [GB]	20.2	25.5	66.7

less damped. We want to empathize that only about 3 elements are limited with FV sub-cells in the wake. Nonetheless, the refined sub-cell resolution has a substantial impact on the quality of the computation. As visible in Fig. 19, the p-refinement indicator successfully detects the nose area, the vicinity of the shock and the airfoils wake as regions, where a high resolution is required. Here, the maximum allowed polynomial degree is applied. The rest of the computational domain is computed with the lowest allowed polynomial degree. When compared to the globally refined computation, the p-adaptive computations show good agreement and recover intricate features like the vortex sheet in the wake of the airfoil very well. The results of the adaptive computation with a refined sub-cell grid even surpass the globally refined computation when considering the vortical structures in the wake. This is especially significant, since only an average of 32.5 DOFs per element were required in this setup. With only 30.5 DOFs per element, the p-adaptive computation without increased sub-cell resolution was even cheaper, while reproducing the non-adaptive result very well. Since the globally refined computation was too expensive to run on a single core, it was performed on 32 cores in parallel*. This makes a direct comparison of the computation time difficult. On a single core, the adaptive computations required about 3 times more time than the globally refined computation on 32 cores. Since good parallel scaling of the FLEXI was demonstrated in [27], we can conclude from these measurements, that the adaptive computations were significantly cheaper and would outperform the non-adaptive code on a single core by a large factor. A contributing factor to the decreased wall time of the adaptive computations for this setup is the number of time steps, listed in Table 5. The adaptive computations required about three times less time steps than the globally refined computation. This effect occurs, when elements with the most restrictive time step requirement are not discretized with the highest possible polynomial degree. This results in an increased global time step, as observed for both p-adaptive airfoil computations.

8.9 A Three-Dimensional Explosion Problem

Finally, to validate our adaptive scheme for three-dimensional computations, we consider a spherically symmetric problem by extending the one-dimensional shock tube problem from Sect. 8.4 to a three-dimensional spherical problem. The resulting setup on the computational domain $\Omega = [-1, 1]^3$ is defined by the initial conditions

$$(\rho, u, p) = \begin{cases} (1, 0, 0, 0, 1) & r \leq 0.5, \\ (0.125, 0, 0, 0, 0.1) & r \geq 0.5, \end{cases}$$

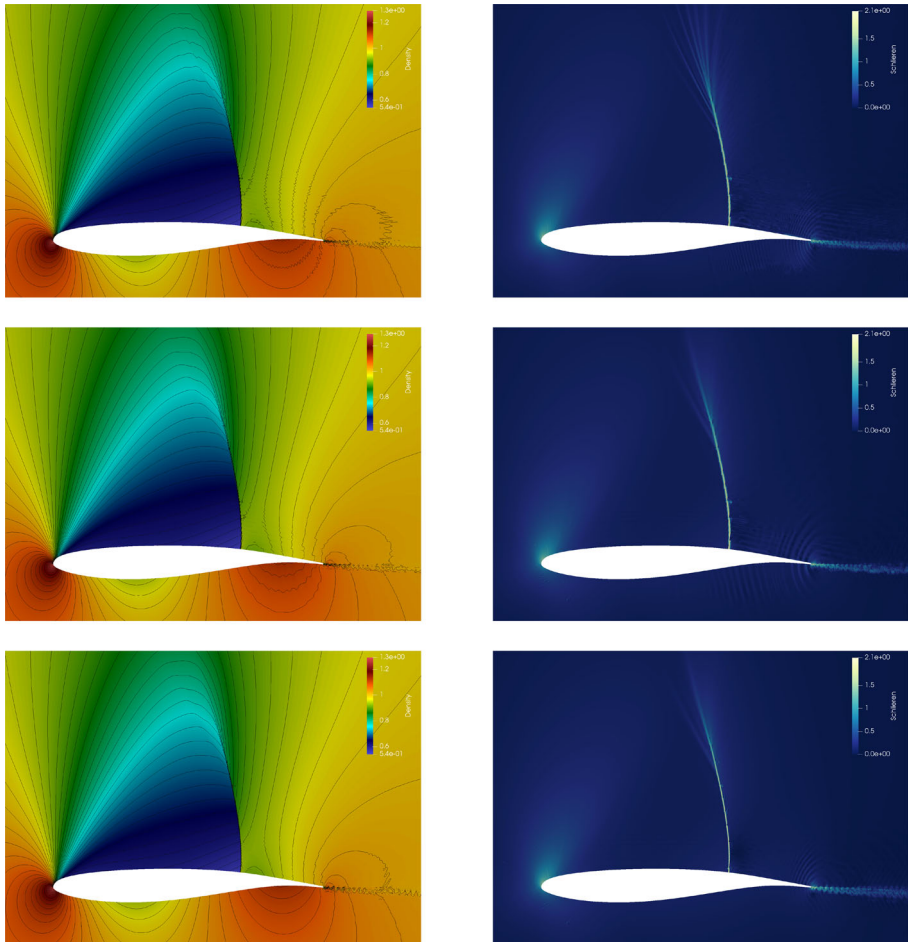


Fig. 18 Density distribution with equidistant contour lines (left) and schlieren images (right), computed as $\log_{10}(|\nabla\rho| + 1)$, for an airfoil at transonic conditions. We compare the three computational setup *Flexi-static* (top), *Flexi-p-adaptive* (middle) and *Flexi-hp-adaptive* (bottom)

where r denotes the radial coordinate $r = \sqrt{x^2 + y^2 + z^2}$. The computational domain is still Cartesian and discretized by $50 \times 50 \times 50$ elements with Dirichlet boundary conditions. As an approximate Riemann solver, we use here for this focusing wave problem the robust local Lax-Friedrichs solver. The modal decay indicator is evaluated for the density and thresholds are set as $[DG_{refine}^{Nmin}, DG_{refine}^{Nmax}] = [4.5, 6.0]$ and $[FV_{lower}^{Nmin}, FV_{lower}^{Nmax}] = [2.5, 4.0]$. The described setup is of particular interest since it features the propagation of waves that are not aligned to the grid. Due to the problem being spherically symmetric, an equivalent reference solution can be computed in one dimension using spherical coordinates and a geometric source term [28]. Three computations with the numerical setups *Flexi-static*, *Flexi-adaptive* and *Flexi-hp-adaptive* are performed and compared against the reference solution. Table 6 contains a detailed definition of the setups. To evaluate the schemes ability to propagate waves diagonally through the Cartesian grid cells, Fig. 20 shows the density distribution over a line from the origin to the corner (1, 1, 1) at the final time $t = 0.2$. It is to be noted that in this direction,

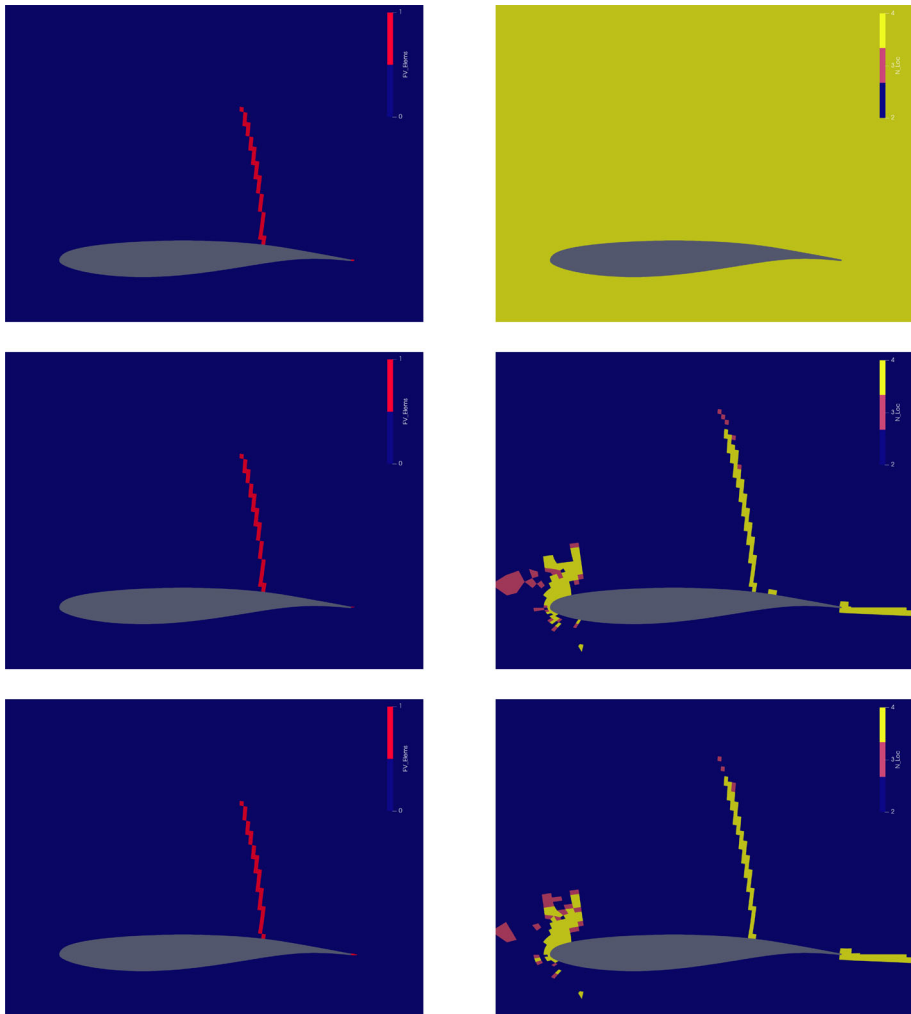
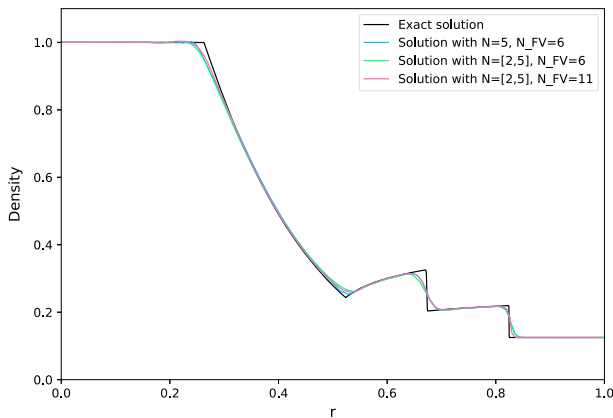


Fig. 19 FV sub-cell distribution (left) and element local polynomial degree (right) for the setups *Flexi-static* (top), *Flexi-p-adaptive* (middle) and *Flexi-hp-adaptive* (bottom)

the interval of interest $[0, 1]$ is only discretized by 15 elements. The density is visualized for the three computational setups. To demonstrate the spherical symmetry of the result, Fig. 21 compares the density distribution in x -direction and the distribution diagonal to the mesh direction from the origin to the corner $(1, 1, 1)$. For the hp -adaptive setup with $N_{FV} = 11$ sub-cells, a very close agreement can be observed. In Fig. 22, the contour of the density from the $z = 0$ plane is plotted together with the FV sub-cell distribution and the distribution of the local polynomial degree for setup *Flexi-hp-adaptive*. The shock is successfully detected and limited with FV sub-cells. In contrast to the one-dimensional Sod shock tube computation of Sect. 8.4, the contact discontinuity is also limited with FV sub-cells. This is necessary due to the significantly coarser mesh of the three-dimensional setup. A high polynomial degree is applied in the vicinity of the shock and the contact discontinuity as well as at the rarefaction fan. The density distributions in Fig. 20 show that the adaptive setups match the non-adaptive,

Table 6 Comparison of wall time, average number of DOFs per element, total number of time steps and memory requirement between the non-adaptive and p-adaptive computations

	Flexi-static	Flexi-p-adaptive	Flexi-hp-adaptive
Resolution	$N = 5, N_{FV} = 6$	$N \in [2, 5], N_{FV} = 6$	$N \in [2, 5], N_{FV} = 11$
DOFs per element	216	56	141
Time steps	235	241	248
Wall time [s]	10,839	3557	8601
Memory [GB]	39.8	48.6	157.3

**Fig. 20** Density distribution of the 3D explosion problem along a line from the origin to the corner (1, 1, 1) at the final time $t = 0.2$. The reference solution is compared against a non-adaptive and two adaptive setups

globally refined solution quite well. A slightly better resolution of the shock can be observed in the case of the refined FV sub-cell resolution. In Table 6, the average number of DOFs per element and the computation time is compared for the three setups. For the test case at hand, computations were performed on a single core of an AMD EPYC 7302 processor. With an average of 56 DOFs per element, the adaptive computation *Flexi-p-adaptive* was three times faster than the non-adaptive computation *Flexi-static*. Since an average of almost 8% of all elements are limited with FV sub-cells, the number of DOFs and therefore the computation time increased significantly for the setup *Flexi-hp-adaptive*, where an increased sub-cell resolution is applied. Therefore, only a speedup of 25% was achieved. Since both adaptive computations produce very similar results, the setup *Flexi-p-adaptive* is the more efficient one for this test case. We can conclude that our adaptive framework is able to provide comparable results for a three-dimensional test case while reducing the computational effort significantly.

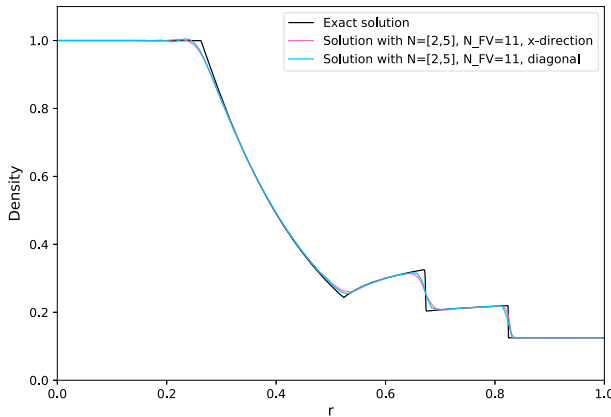


Fig. 21 Density distribution of the 3D explosion problem along a line from the origin to the corner (1, 1, 1) and along the x-axis at the final time $t = 0.2$. The reference solution is compared against the hp adaptive setup

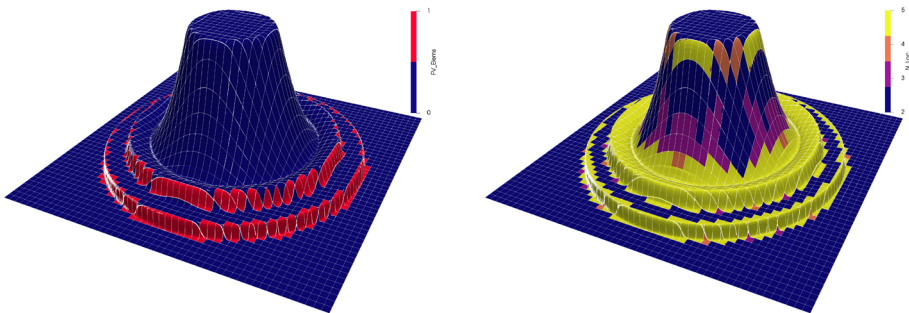


Fig. 22 Contour of the density from the $z = 0$ plane of the 3D explosion problem at the final time $t = 0.2$, computed with an adaptive setup with $N = [2, 5]$ and a FV sub-cell resolution of $N_{FV} = 2N + 1 = 11$. On the left, FV elements are highlighted in red and on the right, the distribution of the local polynomial degree is visualized

9 Conclusion

In this paper, we combined a p-adaptive discontinuous Galerkin scheme with a finite volume shock capturing strategy with variable h-refinement. This hybrid approach profits from the high convergence rate of p-adaptive high order methods in smooth areas as well as from the stable and accurate shock capturing abilities by a second order total variation diminishing finite volume scheme. In the troubled cells with oscillations, the finite volume scheme acts on a h-refined sub-cell grid. The refinement causes a better localisation of the strong gradients within the troubled DG grid cell. Hence, the advantages of the discontinuous Galerkin scheme and the finite volume scheme are successfully combined. An indicator based on the modal decay of the solution polynomials managed the switching between both schemes and controlled the adaptation of the local polynomial degree during run time. To obtain an implementation with a minimal overhead, we introduced an efficient, static, array based data structure that recovers the single core performance of non-adaptive reference computations. With a variety of numerical test cases, we compared non-adaptive setups with different adap-

tive setups and achieved significant speedups, while producing comparable or even more accurate results.

In the future, we will extend the adaptive framework to more complex setups like a shock boundary interaction for a transsonic airfoil. This requires multiple extensions of the code and the numerical scheme. To increase the overall robustness, a split DG formulation is available as a de-aliasing approach to deal with under resolved flow features. This application was omitted in this paper, to highlight the stabilizing effect of the FV sub-cell shock capturing. Furthermore, we need to adapt our indicator strategy to distinguish between under resolved turbulence and shocks. A possible strategy could be to apply the indicator on the pressure and on the density simultaneously to identify the shock wave. Finally, to achieve efficient scaling of our code on massively parallel systems for large scale simulations, a load balancing is required, since the different resolutions and operators introduce an imbalance. Within this context we will also consider the efficiency of a higher accuracy in the finite volume sub-cells by reconstruction, as proposed in [30] or in [8].

Funding Open Access funding enabled and organized by Projekt DEAL. Research reported in this publication was supported by the German Research Foundation (DFG) through the project GRK 2160/1 “Droplet Interaction Technologies” under the project number 270852890 and through Germany’s Excellence Strategy EXC 2075 under the project number 390740016.

Availability of Data and Materials All data generated or analysed during this study are included in this published article.

Code Availability The open source code FLEXI, on which all extensions are based, is available at www.flexi-project.org under the GNU GPL v3.0 license.

Declarations

Conflict of interest The corresponding author states on behalf of all authors, that there is no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Abbà, A., Recanati, A., Tugnoli, M., Bonaventura, L.: Dynamical p-adaptivity for LES of compressible flows in a high order DG framework. *J. Comput. Phys.* **420**, 109720 (2020). <https://doi.org/10.1016/j.jcp.2020.109720>
2. Beck, A.D., Zeifang, J., Schwarz, A., Flad, D.G.: A neural network based shock detection and localization approach for discontinuous Galerkin methods. *J. Comput. Phys.* **423**, 109824 (2020). <https://doi.org/10.1016/j.jcp.2020.109824>
3. Chalmers, N., Agbaglah, G., Chrust, M., Mavriplis, C.: A parallel hp-adaptive high order discontinuous Galerkin method for the incompressible Navier–Stokes equations. *J. Comput. Phys.* **X 2**, 100023 (2019). <https://doi.org/10.1016/j.jcp.2019.100023>
4. Cockburn, B., Shu, C.W.: Runge–Kutta discontinuous Galerkin methods for convection-dominated problems. *J. Sci. Comput.* **16**(3), 173–261 (2001). <https://doi.org/10.1023/A:1012873910884>
5. Courant, R., Friedrichs, K., Lewy, H.: Über die partiellen differenzengleichungen der mathematischen physik. *Math. Ann.* **100**, 32–74 (1928)

6. Duerrwaechter, J., Kurz, M., Kopper, P., Kempf, D., Munz, C.D., Beck, A.: An efficient sliding mesh interface method for high-order discontinuous Galerkin schemes. *Comput. Fluids* **217**, 104825 (2020). <https://doi.org/10.1016/j.compfluid.2020.104825>
7. Dumbser, M., Loubère, R.: A simple robust and accurate a posteriori sub-cell finite volume limiter for the discontinuous Galerkin method on unstructured meshes. *J. Comput. Phys.* **319**, 163–199 (2016). <https://doi.org/10.1016/j.jcp.2016.05.002>
8. Dumbser, M., Zanotti, O., Loubère, R., Diot, S.: A posteriori subcell limiting of the discontinuous Galerkin finite element method for hyperbolic conservation laws. *J. Comput. Phys.* **278**, 47–75 (2014). <https://doi.org/10.1016/j.jcp.2014.08.009>
9. Giri, P., Qiu, J.: A high order Runge–Kutta discontinuous Galerkin method with a sub-cell limiter on adaptive unstructured grids for two-dimensional compressible inviscid flows: RKDG with a sub-cell limiter on adaptive unstructured grids. *Int. J. Numer. Methods Fluids* **91**, 367–394 (2019). <https://doi.org/10.1002/fld.4757>
10. Harten, A., Hyman, J.M.: Self adjusting grid methods for one-dimensional hyperbolic conservation laws. *J. Comput. Phys.* **50**(2), 235–269 (1983). [https://doi.org/10.1016/0021-9991\(83\)90066-9](https://doi.org/10.1016/0021-9991(83)90066-9)
11. Huerta, A., Casoni, E., Peraire, J.: A simple shock-capturing technique for high-order discontinuous Galerkin methods. *Int. J. Numer. Methods Fluids* **69**, 1614–1632 (2012). <https://doi.org/10.1002/fld.2654>
12. Kopera, M.A., Giraldo, F.X.: Analysis of adaptive mesh refinement for IMEX discontinuous Galerkin solutions of the compressible Euler equations with application to atmospheric simulations. *J. Comput. Phys.* **275**, 92–117 (2014). <https://doi.org/10.1016/j.jcp.2014.06.026>
13. Kopriva, D.A.: *Implementing Spectral Methods for Partial Differential Equations: Algorithms for Scientists and Engineers*. Springer Publishing Company, Incorporated, Cham (2009)
14. Kraiss, N., Beck, A., Bolemann, T., Frank, H., Flad, D., Gassner, G., Hindenlang, F., Hoffmann, M., Kuhn, T., Sonntag, M., Munz, C.D.: Flexi: a high order discontinuous Galerkin framework for hyperbolic–parabolic conservation laws. *Comput. Math. Appl.* **81**, 186–219 (2020). <https://doi.org/10.1016/j.camwa.2020.05.004>
15. Li, Y., Premasathan, S., Jameson, A.: Comparison of h- and p-adaptations for spectral difference methods. In: *AIAA-201010 of 11 American Institute of Aeronautics and Astronautics*, pp. 10–1007 (2010)
16. Marcon, J., Castiglioni, G., Moxey, D., Sherwin, S., Peiro, J.: rp-adaptation for compressible flows. *Int. J. Numer. Methods Eng.* **121**, 5405–5425 (2020). <https://doi.org/10.1002/nme.6529>
17. Mavriplis, C.: Nonconforming discretizations and a posteriori error estimators for adaptive spectral element techniques. Dissertation, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology. <http://hdl.handle.net/1721.1/14526> (1989)
18. Mavriplis, C.: A posteriori error estimators for adaptive spectral element techniques. In: Wesseling, P. (ed.) *Proceedings of the Eighth GAMM-Conference on Numerical Methods in Fluid Mechanics*, pp. 333–342. Vieweg+Teubner Verlag, Wiesbaden (1990)
19. Moxey, D., Cantwell, C.D., Mengaldo, G., Serson, D., Ekelschot, D., Peiró, J., Sherwin, S.J., Kirby, R.M.: Towards p-adaptive spectral/hp element methods for modelling industrial flows. In: Bittencourt, M.L., Dumont, N.A., Hesthaven, J.S. (eds.) *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2016*, pp. 63–79. Springer International Publishing, Cham (2017)
20. Persson, P.O.: Shock capturing for high-order discontinuous Galerkin simulation of transient flow problems (2013). <https://doi.org/10.2514/6.2013-3061>
21. Persson, P.O., Stamm, B.: A discontinuous Galerkin method for shock capturing using a mixed high-order and sub-grid low-order approximation space. *J. Comput. Phys.* **449**, 110765 (2019)
22. Qiu, J., Shu, C.W.: Hermite WENO schemes and their application as limiters for Runge–Kutta discontinuous Galerkin method: one-dimensional case. *J. Comput. Phys.* **193**(1), 115–135 (2004). <https://doi.org/10.1016/j.jcp.2003.07.026>
23. Remacle, J.F., Flaherty, J., Shephard, M.: An adaptive discontinuous Galerkin technique with an orthogonal basis applied to compressible flow problems. *SIAM Rev.* **45**, 53–72 (2003). <https://doi.org/10.1137/S00361445023830>
24. Roe, P.: Characteristic-based schemes for the Euler equations. *Annu. Rev. Fluid Mech.* **18**(1), 337–365 (1986). <https://doi.org/10.1146/annurev.fl.18.010186.002005>
25. Shu, C.W., Osher, S.: Efficient implementation of essentially non-oscillatory shock-capturing schemes. II. *J. Comput. Phys.* **83**(1), 32–78 (1989). [https://doi.org/10.1016/0021-9991\(89\)90222-2](https://doi.org/10.1016/0021-9991(89)90222-2)
26. Sonntag, M., Munz, C.D.: Shock capturing for discontinuous Galerkin methods using finite volume subcells. In: Fuhrmann, J., Ohlberger, M., Rohde, C. (eds.) *Finite Volumes for Complex Applications VII*, vol. 78, pp. 945–953. Springer, Berlin (2014). https://doi.org/10.1007/978-3-319-05591-6_96

27. Sonntag, M., Munz, C.D.: Efficient parallelization of a shock capturing for discontinuous Galerkin methods using finite volume sub-cells. *J. Sci. Comput.* **70**, 1262–1289 (2017). <https://doi.org/10.1007/s10915-016-0287-5>
28. Toro, E.: *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*. Springer, Berlin (2009). <https://books.google.de/books?id=SqEjX0um8o0C>
29. Toro, E.F., Clarke, J.F.: *Numerical Methods for Wave Propagation*, 1st edn. Springer Publishing Company, Incorporated, Cham (2011)
30. Vilar, F.: A posteriori correction of high-order discontinuous Galerkin scheme through subcell Finite Volume formulation and flux reconstruction. *J. Comput. Phys.* **387**, 245–279 (2019). <https://doi.org/10.1016/j.jcp.2018.10.050>
31. VonNeumann, J., Richtmyer, R.D.: A method for the numerical calculation of hydrodynamic shocks. *J. Appl. Phys.* **21**, 232–237 (1950)
32. Woodward, P., Colella, P.: The numerical simulation of two-dimensional fluid flow with strong shocks. *J. Comput. Phys.* **54**(1), 115–173 (1984). [https://doi.org/10.1016/0021-9991\(84\)90142-6](https://doi.org/10.1016/0021-9991(84)90142-6)
33. Zeifang, J., Beck, A.: A data-driven high order sub-cell artificial viscosity for the discontinuous Galerkin spectral element method. *J. Comput. Phys.* **441**, 110475 (2021). <https://doi.org/10.13140/RG.2.2.26934.32322/1>
34. Zhu, J., Zhong, X., Shu, C.W., Qiu, J.: Runge–Kutta discontinuous Galerkin method using a new type of WENO limiters on unstructured meshes. *J. Comput. Phys.* **248**, 200–220 (2013). <https://doi.org/10.1016/j.jcp.2013.04.012>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.