J Sci Comput (2011) 47: 365–388
DOI 10.1007/s10915-010-9448-0

A Generic Stabilization Approach for Higher Order Discontinuous Galerkin Methods for Convection Dominated Problems

Andreas Dedner · Robert Klöfkorn

Received: 6 November 2009 / Revised: 19 July 2010 / Accepted: 1 December 2010 /
Published online: 21 December 2010

Abstract In this paper we present a stabilized Discontinuous Galerkin (DG) method for hyperbolic and convection dominated problems. The presented scheme can be used in several space dimension and with a wide range of grid types. The stabilization method preserves the locality of the DG method and therefore allows to apply the same parallelization techniques used for the underlying DG method. As an example problem we consider the Euler equations of gas dynamics for an ideal gas. We demonstrate the stability and accuracy of our method through the detailed study of several test cases in two space dimension on both unstructured and cartesian grids. We show that our stabilization approach preserves the advantages of the DG method in regions where stabilization is not necessary. Furthermore, we give an outlook to adaptive and parallel calculations in 3d.

Keywords Conservation laws · Higher order methods · Discontinuous Galerkin · Finite volume · Generic limiter

1 Introduction

A wide range of numerical methods has been developed for the approximation of nonlinear hyperbolic or convection dominated problems. Among these methods the RK-DG method [5] has become very popular over the last decade. The RK-DG method is stable when applied to linear hyperbolic systems; however for nonlinear problems spurious oscillations occur near strong shocks or steep gradients. Thus the RK-DG method requires some extra stabilization. In fact it is well known that only the first order scheme ($k = 0$) produces a monotonic structure in the shock region. Many approaches have been suggested

A. Dedner
Mathematics Institute, University of Warwick, Coventry CV4 7AL, UK
e-mail: A.S.Dedner@warwick.ac.uk

R. Klöfkorn (✉)
Applied Mathematics, University of Freiburg, Hermann-Herder-Strasse 10, 79104 Freiburg, Germany
e-mail: robertk@mathematik.uni-freiburg.de

to make this property available in higher order schemes, without introducing the amount of numerical viscosity, which is such a characteristic feature of first order schemes.

A first approach is motivated by higher order Finite Volume methods where slope limiters are applied in order to avoid oscillations. This is for example studied in [3, 5, 15–18, 21, 23, 25] and many others. Another approach is to add artificial diffusion to the problem in regions where shocks occur. Then the problem becomes sufficiently smooth and therefore stable. This is for example studied in [11, 22, 28]. A third approach has been suggested in [10] where a posteriori techniques are used to stabilize the DG method.

In this work we concentrate on the first approach which uses slope limiters to stabilize the RK-DG method. Although most of the components have been presented before, their combination and their extension to arbitrary grid structures and space dimensions is the major contribution of this work. Especially, the reconstruction process was so far not presented in this general setting and allows the combination of strong gradient with the DG approximation, while maintaining a small stencil. The presented RK-DG method thus contains all features needed nowadays for large scale computations such as locality of the method (only direct neighboring information are needed), works in several space dimensions such as 2d and 3d both on Cartesian grids and general unstructured (hybrid) grids with conforming or non-conforming local refinement.

As described for example in [18, 23], we combine the slope limiter with a shock detector. The shock detector should detect regions where a stabilization is necessary and the slope limiter is then used to stabilize the solution. The combination of shock detection and limiting should avoid over-excessive limiting in the extrema of the solution in regions where the solution is "smooth" enough, while keeping over- and undershoots in shock regions at a minimum. Especially slope limiting in regions of contact discontinuities (linear waves) should be treated with care since DG methods fare rather well here and too much numerical viscosity is in this case even more problematic than with compressive shock waves. Furthermore, due to the shock detector, the limiter is only applied to a small percentage of elements so that the computational cost of the limiter is not essential – the major cost is caused by the computation.

In contrast to the other approaches (see [3, 5, 15–18, 21, 23, 25]) the slope limiter presented here is independent of the space dimension and of the grid element geometry type used, i.e. the slope limiter can be used both with Cartesian and general unstructured grids in several space dimensions, including non-conforming or hybrid grids. From the literature mentioned above there is only one work, the work of Klieber and Rivière [16], dealing with non-conforming (triangular) meshes. The idea in [16] is to use an appropriate reconstruction on non-conforming intersections such that a conforming situation is emulated and then a slope limiter, developed for conforming meshes, is applied.

The rest of this paper is organized as follows. First, the general stabilization operator and its application to the RK-DG method is described. Second, the shock detection mechanism is explained followed by the description of the limiting procedure. The algorithm splits into three steps: first the shock detector is computed, then a set of admissible linear functions is computed, one of which is then chosen in the third step on cells where stabilization is required. For the shock detection, we use an approach based on [18], and for the third step the linear function with the largest gradient is selected. Other shock detectors could also be applied (see [24]) or the linear function with the largest angle could be used (see [4]). We have compared many combinations but will only report on those which for us proved the most efficient and accurate. Finally in Sect. 6, numerical results for the stabilized RK-DG method are presented that demonstrate the effectiveness and efficiency of the developed method. Preliminary numerical results were published in [7, 8].

2 The Runge-Kutta Discontinuous Galerkin Method

This sections describes a general approach for discretizing evolutions equations of the form

$$\partial_t \boldsymbol{u}(t,\cdot) = \mathcal{L}[\boldsymbol{u}(t,\cdot)](\cdot) \quad \text{in } ([0,T)\times\Omega)\subset(\mathbb{R}\times\mathbb{R}^d),\ d\in\{1,2,3\}. \tag{1}$$

Hereby, the spatial operator is defined by

$$\mathcal{L}[\boldsymbol{v}] = S(\boldsymbol{v}) - \nabla\cdot F(\boldsymbol{v}),$$

where $\boldsymbol{v}:\Omega\to\mathcal{U}\subseteq\mathbb{R}^r$ is in some suitable function space V. $\mathcal{U}$ denotes the set of states for a given problem. With the function $S(\boldsymbol{v})$ we denote a source term; $F(\boldsymbol{v})$ is the analytical flux function. By prescribing S and F a wide range of problems can be written in this from. A similar approach as the one described in the following sections can also be used for problems with including higher order derivatives if F is the dominating term, e.g., the Navier Stokes equations.

2.1 Higher Order Schemes for Systems of Evolutions Equations

In the following we focus on the spatial discretization, i.e., we construct a discrete operator $\mathcal{L}_\mathcal{G}$ which maps one finite-dimensional function space $V_\mathcal{G}$ onto another finite-dimensional function space $W_\mathcal{G}$.

For $\Omega\subset\mathbb{R}^d$ with $d\geq 1$ we choose $\Omega_\mathcal{G}\subseteq\Omega$ to be an polygonal approximation of the domain Ω which is partitioned by a tesselation $\mathcal{G}$ in the sense of the grid definition given in [9]. The spatial operator is defined by

$$\mathcal{L}[\boldsymbol{v}] = S(\boldsymbol{v}) - \nabla\cdot F(\boldsymbol{v}) \tag{2}$$

where the function $\boldsymbol{v}:\Omega_\mathcal{G}\to\mathcal{U}\subset\mathbb{R}^r,\ r\in\mathbb{N}$ is in V^r defined to be $V^r := \{\boldsymbol{v} : \Omega_\mathcal{G}\to\mathbb{R}^r \in L^2(\Omega_\mathcal{G}) \mid \boldsymbol{v}|_E\in H^1(E)\ \ \forall E\in\mathcal{G}\}$. For simplicity we choose $r=1$ in the following and use the abbreviation $V := V^1$. Note that $u\in V$ is a smooth function on the cells $E\in\mathcal{G}$ but might be discontinuous over the cell interfaces. Then a discretization of the operator $\mathcal{L}$ is constructed by multiplying equation (2) with a test function $v\in V$ and by integrating over the domain $\Omega_\mathcal{G}$. Thereby, the divergence term is integrated by parts over the cell E. Finally the discrete operator $\mathcal{L}_\mathcal{G}$ is defined by the (L^2-) projection of $\mathcal{L}$ onto a finite subset of V which we call $V_\mathcal{G}^k$. $V_\mathcal{G}^k$ is given by $V_\mathcal{G}^k := \{\varphi:\Omega_\mathcal{G}\to\mathbb{R}\in L^2(\Omega_\mathcal{G}) \mid \varphi|_E\in\mathbb{P}_k(E)\ \ \forall E\in\mathcal{G}\}\subset V$. We arrive at

$$\begin{aligned}\int_{\Omega_\mathcal{G}}\mathcal{L}_\mathcal{G}[u_\mathcal{G}]\varphi\,d\boldsymbol{x} = &\sum_{E\in\mathcal{G}}\int_E S(u_\mathcal{G})\varphi\,d\boldsymbol{x} + \sum_{E\in\mathcal{G}}\int_E F(u_\mathcal{G})\cdot\nabla\varphi\,d\boldsymbol{x}\\ &-\sum_{E\in\mathcal{G}}\int_{\partial E}\varphi\,G(u_\mathcal{G}^+,u_\mathcal{G}^-,\ldots)\cdot\mathbf{n}\,d\sigma \quad \forall\varphi\in V_\mathcal{G}^k.\end{aligned} \tag{3}$$

Hereby, $\mathbf{n}$ denotes the unit outer normal of the cell interface, $u_\mathcal{G}^+$ and $u_\mathcal{G}^-$ are the values of the function $u_\mathcal{G}$ on both sides of the cell interface. $G(u,v,\boldsymbol{x}):V\times V\times\mathbb{R}^d\to\mathbb{R}^d$ is called a numerical flux function. The Discontinuous Galerkin method is completely described by the physical parameters such as the functions S and F and by the numerical flux G. The physical parameters are determined by the equation whereas the choice of the numerical flux is the crucial part in the method. For a reasonable method the numerical flux G should

be consistent, Lipschitz continuous, and conservative. Many numerical fluxes for different kinds of equations can be found in standard textbooks such as [19].

To arrive at a fully discrete scheme a suitable time discretization method is applied; here we use the standard SSP Runge-Kutta methods as suggested in [13].

2.2 Example Problem—The Euler Equations

A compressible inviscid fluid is modeled by the Euler equations of gas dynamics:

$$\partial_t \boldsymbol{u} + \sum_{j=1}^{d} \partial_{x_j} \boldsymbol{f}_j(\boldsymbol{u}) = 0, \quad \text{in } ([0,T) \times \Omega_{\mathcal{G}} \subset \mathbb{R}^d),\ d \in \{1,2,3\} \tag{4}$$

where the vector of the conservative variables is $\boldsymbol{u} = (\rho, \rho\boldsymbol{v}, e)^T$, $\rho\boldsymbol{v} = (\rho v_1, \ldots, \rho v_d)^T$, and $e = \rho\mathcal{E}$, with $\boldsymbol{u}$ taking values in the set of states

$$\mathcal{U} := \left\{ (\rho, \rho\boldsymbol{v}, e) \;\middle|\; \rho > 0, \boldsymbol{v} \in \mathbb{R}^d, e - \frac{\rho}{2}|\boldsymbol{v}|^2 > 0 \right\}, \tag{5}$$

and the convective flux functions for $i = 1, ..., d$:

$$\boldsymbol{f}_i(\boldsymbol{u}) := \begin{pmatrix} \boldsymbol{u}_{i+1} \\ \boldsymbol{u}_{i+1}\boldsymbol{u}_2/\boldsymbol{u}_1 + \delta_{i,1}\, p(\boldsymbol{u}) \\ \vdots \\ \boldsymbol{u}_{i+1}\boldsymbol{u}_{d+1}/\boldsymbol{u}_1 + \delta_{i,d}\, p(\boldsymbol{u}) \\ (\boldsymbol{u}_{d+2} + P(\boldsymbol{u}))\, \boldsymbol{u}_{i+1}/\boldsymbol{u}_1 \end{pmatrix},$$

where $\delta_{i,d}$ is the Kronecker delta. The system is closed by the equation of state for an **ideal gas** where the pressure is given by

$$p(\boldsymbol{u}) = (\gamma - 1)\left[\boldsymbol{u}_{d+2} - \frac{\boldsymbol{u}_1}{2} \sum_{i=1}^{d} (\boldsymbol{u}_{i+1}/\boldsymbol{u}_1)^2 \right] \tag{6}$$

where γ is the adiabatic constant (see for example [19]).

Now, to rewrite the system in the form of (1) we simply have to define the operator $\mathcal{L}$. Since we have no source term, i.e. $\boldsymbol{S}(\boldsymbol{u}) = 0$, the operator $\mathcal{L}[\boldsymbol{u}] := -\nabla \cdot \boldsymbol{F}(\boldsymbol{u})$ only contains the conservative part with $\boldsymbol{F}(\boldsymbol{u}) := (\boldsymbol{f}_1, \ldots, \boldsymbol{f}_d)$. With this definition of $\mathcal{L}$ and an appropriate numerical flux such as the Local-Lax-Friedrichs flux function or the HLL flux function (cf. [19]) the spatial discretization is already obtained.

The boundary of the computational domain $\Omega_{\mathcal{G}}$ is partitioned into non-overlapping parts: $\partial\Omega_{in}$ (**Inflow**), $\partial\Omega_{out}$ (**Outflow**), and $\partial\Omega_{refl}$ (**Reflection**). These boundary conditions are realized such that a boundary value u_B is determined by a given boundary function or due to the boundary condition. u_B is then used to evaluate the numerical flux, i.e. $G = G(u_{\mathcal{G}}^+, u_B, \boldsymbol{x})$. For different boundary conditions the value $\boldsymbol{u}_B$ is defined to be: (**Inflow**) $u_B = g(\boldsymbol{x})$ where g is a given boundary function, (**Outflow**) $u_B = u_{\mathcal{G}}^+$, and (**Reflection**) u_B is obtained by toggling the sign of the normal component of the velocity in $u_{\mathcal{G}}^+$.

3 Stabilization

Let us recall the DG scheme presented in Sect. 2. For higher order schemes for advection dominated problems, a stabilization mechanism is required. A stabilized discrete operator is constructed by concatenation of the DG operator $\mathcal{L}_\mathcal{G}$ from (3) and a stabilization operator $\Pi_\mathcal{G}$, leading to a modified discrete spatial operator $\widetilde{\mathcal{L}_\mathcal{G}}[u_\mathcal{G}(t,\cdot)] := (\mathcal{L}_\mathcal{G} \circ \Pi_\mathcal{G})[u_\mathcal{G}(t,\cdot)]$. We call

$$u^*_\mathcal{G} := \Pi_\mathcal{G}[u_\mathcal{G}] \in V^*_\mathcal{G} \tag{7}$$

the stabilized approximate solution and $u^*_E = u^*_{\mathcal{G}|E}$ is the stabilized approximate solution restricted to an element. Before we describe the construction of $\Pi_\mathcal{G}$ we briefly describe some properties which a reasonable stabilization should satisfy:

- *Conservation property*:
  we require that $\bar{u}_E = \bar{u}^*_E, \forall\, E \in \mathcal{G}$, i.e. that $u_\mathcal{G}$ and $u^*_\mathcal{G}$ have the same average value on each element of the grid.
- *Physicality of* u^*_E:
  We require that the stabilized approximate solution satisfies $u^*_\mathcal{G}(\boldsymbol{x}) \in \mathcal{U}$.
- *Consistency for linear functions*:
  If the average values on E and its neighbors are given by the same linear function L_E then $u^*_E = L_E$ on E.
- *Identity in "smooth" regions*:
  In regions where the solution is "smooth" we claim $u^*_\mathcal{G} = u_\mathcal{G}$. This requires an indicator for the smoothness of the solution (here called shock detector, see Sect. 4.1).
- *Minimal stencil*:
  The stabilized DG method should have the same stencil as the original DG method, Thus u^*_E should only depend on the function $u_\mathcal{G}$ on E and its neighbors.
- *Maximum-minimum principle and monotonicity*:
  In regions where the solution is not smooth the function u^*_E should only take values between $\min_{E_e \in \mathcal{N}_E} u_E$ and $\max_{E_e \in \mathcal{N}_E} u_{E_e}$, where $\mathcal{N}_E$ denotes the set of neighbors of E. Also u^*_E should have the same monotonicity properties as the average values on E and its neighbors.

In general a stabilization operator can not fulfill all of these properties is a strong sense but should satisfy some approximation of these conditions. For example it suffices to enforce the physicality condition for all values needed for the definition of the operator $\mathcal{L}_\mathcal{G}$; in practice we use quadrature rules to approximate the element and boundary integrals in (3) and require that $u^*_\mathcal{G}$ is in the admissible set $\mathcal{U}$ for all quadrature points.

4 Stabilization Operator

In this section the stabilization operator is presented and in the following we describe the construction steps of this operator.

1. *Shock detection*:
   As stated in the previous section $u^*_E \equiv u_E$ should hold in regions where the solution is smooth. To detect these regions we use an indicator function $\mathcal{S}_E$ taking on the values 0 or 1. We require $\mathcal{S}_E = 0$ on cells in regions where the solution is smooth and $\mathcal{S}_E = 1$ on *troubled cells*, i.e., cells where the DG scheme might become unstable, e.g., near discontinuities or where u_E takes on unphysical values.

2. *Construction of admissible linear functions*:
   On cells where $\mathcal{S}_E \geq 1$ we construct a set of linear functions $\mathbb{L}_E$. To satisfies the minimum stencil properties, all the linear functions in $\mathbb{L}_E$ are constructed using the average values of $u_\mathcal{G}$ on E and on its neighbor. Furthermore, all functions in this set satisfy a version of the physicality, the minimum-maximum, and the monotonicity conditions.
3. *Definition of* u_E^*:
   finally u_E^* is chosen to be u_E on cells with $\mathcal{S}_E = 0$ and equal to one of the functions in the set of admissible linear functions (for example the steepest one, see (14)) in the case when $\mathcal{S}_E = 1$. Note that in regions where limiting is done the approximation order of the scheme is not higher than linear since the limited solution consists only of linear functions. In regions where no limiting is performed the scheme retains the original approximation order.

Note that if we take $\mathcal{S}_E = 1$ on all elements E we arrive at a second order Finite Volume scheme. Before we give more detail on our method, we describe how we will enforce the physicality condition:

Definition 4.1 (Non-physicality of the approximate solution) A local solution u_E of $u_\mathcal{G} \in V_\mathcal{G}^k$ on element E is called *non-physical* if the following condition is satisfied:

$$u_E \text{ is } \textit{non-physical} \iff \exists \boldsymbol{x} \in (Q_{\hat{E}}^{2k} \cup Q_{\partial\hat{E}}^{2k+1}) : u_E\big(F_E(\boldsymbol{x})\big) \notin \mathcal{U}, \tag{8}$$

where k is the polynomial order of the basis functions of $V_\mathcal{G}^k$ on E, $Q_{\hat{E}}^{2k}$ denotes an element quadrature which is exact of order $2k$, and $Q_{\partial\hat{E}}^{2k+1}$ a quadrature for the faces of the element which is exact of order $2k+1$. F_E is the reference mapping and $\mathcal{U}$ is the set of states (5) of the function $u_\mathcal{G}$. Notice that in the case that $\mathcal{U} = \mathbb{R}^r$ the solution u_E is never non-physical.

4.1 Detection of Troubled Cells—The Shock Detector

Using a scalar function ϕ depending on the left and right value of the discontinuous approximation $u_\mathcal{G}$ on cell boundaries, we define a discontinuity detector similar to the one given in [18]:

$$\bar{J}_E = \sum_{\substack{e \subset \partial E, \\ \boldsymbol{v}\cdot\mathbf{n}_e < 0}} \left(\frac{\int_e \phi(u_E, u_E)\, ds}{\alpha_d(k)\, h_E^{(k+1)/4}\, |e|} \right), \tag{9}$$

where e is a face of E and u_{E_e} is the value of $u_\mathcal{G}$ on the neighboring element E_e of E over e. The vector valued function $\boldsymbol{v}$ corresponds to some characteristic flow velocity and d is the element's dimension. In general the choice $\phi(u_E, u_{E_e}) = |u_E - u_{E_e}|$ could be used, but for the Euler equations the relative jump of pressure p from (6) leads to better results: $\phi(u_E, u_{E_e}) = 2\frac{p(u_E)-p(u_{E_e})}{p(u_E)+p(u_{E_e})}$. For the characteristic velocity $\boldsymbol{v}$ we use the velocity vector of the Euler equations.

According to [18] the denominator of (9) is in $O(h^d)$ near discontinuities and $O(h^{d+k+1})$ in smooth regions. Thus the quantity $\bar{J}_E$ is in $O(h^{-\beta})$ in regions of discontinuities and in $O(h^\alpha)$ otherwise, with $\alpha = \frac{3}{4}(k+1)$ and $\beta = \frac{1}{4}(k+1)$. Note that the authors of [18] suggested a normalization factor of $h^{(k+1)/2}$, but we obtained better results with a power of $(k+1)/4$. The local grid width h_E is defined as $h_E := \frac{|E|}{h_m^E}$ with $h_m^E := \min_{e \in \mathcal{I}_E} |e|$. Based on experimental results, the open parameter α was chosen to be $\alpha_d(k) := 0.016 \cdot d \cdot 5^k$.

Since $\bar{J}_E \to 0$ as either $h \to 0$ or $k \to \infty$ in smooth regions, and $\bar{J}_E \to \infty$ near a discontinuity,

$$\mathcal{S}_E = \begin{cases} 0 & \bar{J}_E \le 1, \\ 1 & \bar{J}_E > 1, \text{ or } u_E \text{ is unphysical} \end{cases} \tag{10}$$

can be used for shock detection in the stabilization scheme. Further indicators are studied in [24] but only a few are easily used on general grid structures.

4.2 Construction of Admissible Linear Functions

Once the detector (10) establishes that the current DG solution u_E on cell E should be limited, a set of admissible linear functions is constructed. This is achieved in two steps, where first a set linear functions is constructed using $u_{\mathcal{G}}$ on E and on the neighbors of E. Then a set of constraints is applied to each of these linear functions, to render them admissible in the sense described at the end of Sect. 3. Finally u_E^* is chosen to be equal to one of these admissible functions.

4.2.1 Set of Possible Linear Functions

The most natural choice is the linear function given by the approximate solution $u_{\mathcal{G}}$ by ignoring higher order moments:

$$L_{DG}(\boldsymbol{x}) := \bar{u}_E + \nabla L_{DG} \cdot (\boldsymbol{x} - \mathbf{w}_E), \tag{11}$$

where $\int_E L_{DG}\,\phi = \int_E u_E\,\phi$, for all $\phi \in \mathbb{P}_1(E)$, where the average value of the solution u_E is given by $\bar{u}_E := \frac{1}{|E|}\int_E u_E(\boldsymbol{x})\,d\boldsymbol{x}$.

The set of possible linear functions can be enhanced using reconstruction techniques derived for higher order Finite Volume methods. The approach described in the following was first described for a Finite Volume method on conforming triangular meshes in the thesis [26]. Since we want to construct a limiter for arbitrary element geometry types and also non-conforming meshes these ideas are extended accordingly.

To define the reconstruction, we first recall the notations from [9]. For a given cell E we call E_i the neighbor of E over intersection e_i, the i-th intersection, and $i = 1, \ldots, \tilde{n}_E$, where $\tilde{n}_E$ is the number of all intersections of E with other elements or the boundary of $\Omega_{\mathcal{G}}$.

Now, for a given cell E we define

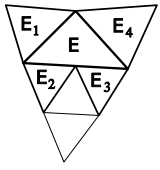
$$\widetilde{\mathcal{Y}}_E := \{1, \ldots, \tilde{n}_E\},$$
$$\mathcal{Y}_E := \{j := \{j_1, \ldots, j_d\} \subset \widetilde{\mathcal{Y}}_E \times \widetilde{\mathcal{Y}}_E \mid j_l \neq j_m \quad \text{for } l \neq m\ \forall l, m = 1, \ldots, d\}.$$

For example for a conforming triangular grid $\mathcal{G}$ we have

$$\widetilde{\mathcal{Y}}_E := \{1, 2, 3\} \quad \text{and} \quad \mathcal{Y}_E := \{\{1, 2\}, \{1, 3\}, \{2, 3\}\} \quad \forall E \in \mathcal{G}.$$

For a non-conforming situation (as shown in the right picture) these sets are

$$\widetilde{\mathcal{Y}}_E := \{1, \ldots, 4\}, \quad \text{and}$$
$$\mathcal{Y}_E := \{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}\}.$$



Using these two sets and the average values of the solution $u_{\mathcal{G}}$ on the neighbors of E as well as their barycenters ($\mathbf{w}_E$ denotes the barycenter of element E and so on) we proceed as follows: For all $j \in \mathcal{Y}_E$ we define linear functions L_j of the form

$$\begin{aligned} &\text{(i)} \quad L_j(\boldsymbol{x}) := \bar{u}_E + \nabla L_j \cdot (\boldsymbol{x} - \mathbf{w}_E), \quad \text{and} \\ &\text{(ii)} \quad L_j(\mathbf{w}_{E_i}) = \bar{u}_{E_i} \quad \forall\, i \in j, \end{aligned} \tag{12}$$

i.e. L_j is a linear function and L_j evaluated at the barycenters of the involved elements simply returns the average values of the solution on the corresponding element. The values of ∇L_j, $j \in \mathcal{Y}_E$, are calculated by solving $d \times d$ linear systems of the form

$$\begin{aligned} \nabla L_j \;&=\; \mathbf{A}_j^{-1}\, \mathbf{b}_j, \quad \forall\, j \in \mathcal{Y}_E, \\ \mathbf{A}_{jl} &:= (\mathbf{w}_{E_{j_l}} - \mathbf{w}_E), \quad \forall\, l = 1, \ldots, d, \\ \mathbf{b}_{jl} &:= \bar{u}_{E_{j_l}} - \bar{u}_E, \quad \forall\, l = 1, \ldots, d. \end{aligned}$$

Note that for Cartesian grids in particular but also on more general meshes the matrix $\mathbf{A}$ can become singular when the considered barycenters lie in a codimension 1 plain of the space. In this case we apply a least squares approach by successively adding other points $k \in \widetilde{\mathcal{Y}}_E$ such that $k \notin j$ and for the new $(d+1) \times d$ matrix $\tilde{\mathbf{A}}$ holds: $\det(\tilde{\mathbf{A}}^T \tilde{\mathbf{A}}) \neq 0$. The new right hand side is then given by $\tilde{\mathbf{b}} = \tilde{\mathbf{A}}^T \mathbf{b}$. For Cartesian grids this information has to be calculated only once since the matrix $\mathbf{A}$ only contains geometrical information.

Notice that with the given definition of the reconstructions the scheme retains its locality by only using information from direct neighboring cells and therefore fulfills the minimal stencil conditions. Furthermore, the method to obtain the linear functions described above is completely independent of the number of neighbors, the geometry type of the elements, and whether the intersections are conforming or not. Therefore, we state that this technique of calculating reconstructions works for arbitrary geometry types, conforming or non-conforming grids. In Sect. 6 this is demonstrated for some of the described situations, e.g., non-conforming meshes, different geometry types (triangular, tetrahedral, hexahedral, and Cartesian grids), for application in a parallel and adaptive environment.

4.2.2 Set of Admissible Linear Functions

In order to fulfill the monotonicity conditions suitable constraints are applied to each linear function in the set of possible linear functions. The constraint is that along all lines connecting the barycenter $\mathbf{w}_E$ of the element E with the barycenter $\mathbf{w}_{E_i}$ of a neighbor E_e each linear function stays between the average values $\bar{u}_E$ and $\bar{u}_{E_i}$ on E and E_e, respectively. This is achieved by reducing the gradient of the linear function by a factor $\tilde{m} \in [0, 1]$. Note that the constraint can always be satisfied by choosing $\tilde{m} = 0$. To define this factor we define for a set of linear functions ∇L_j, $j \in \mathcal{Y}_E$,

$$g_{j,i} := \nabla L_j \cdot (\mathbf{w}_{E_i} - \mathbf{w}_E) \quad \text{and} \quad d_{j,i} := \bar{u}_{E_i} - \bar{u}_E,\ \forall\, i \in \widetilde{\mathcal{Y}}_E.$$

Following the idea of [26] we now calculate a correction factor

$$m_{j,i} := \begin{cases} 0 & \text{if } g_{j,i}\, d_{j,i} < 0, \\ d_{j,i}/g_{j,i} & \text{if } g_{j,i}\, d_{j,i} > 0 \text{ and } |g_{j,i}| > |d_{j,i}|, \\ 1 & \text{otherwise.} \end{cases}$$

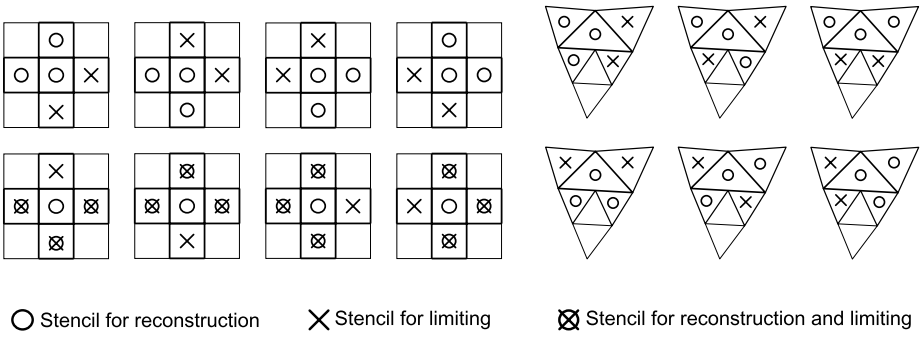


Fig. 1 The stencils (*circles*) used to define the set of linear functions are shown for a Cartesian and an unstructured non-conforming grid; the *lower row* in the Cartesian grid case shows all possible singular cases, where two neighbors and the center cell lie on a single line. In this case an additional point is added to define the linear function using a least squares approach. The *crosses* denote the neighbors which are used in the limiting process

In addition to [26], if $|g_{j,i}| \lll 1$ and $|d_{j,i}| \lll 1$ no limitation is applied to avoid rounding errors which in particular occur on Cartesian grids. This means that we set $m_{j,i} = 1$ in this case. In the numerical experiments 10^{-8} has been used as upper bound.

The limitation applied to the linear functions by means of the factor $m_{j,i}$, $j \in \mathcal{Y}_E$, $i \in \widetilde{\mathcal{Y}}_E$ has the following effect:

- If $d_{j,i} = 0$, i.e. $\bar{u}_E = \bar{u}_{E_i}$, or if $d_{j,i}$ and $g_{j,i}$ have different signs, any choice of $m_{j,i} > 0$ could cause an oscillation.
- If $d_{j,i}$ and $g_{j,i}$ have the same signs and $g_{j,i}$ is "steeper" than $d_{j,i}$, i.e. $|g_{j,i}| > |d_{j,i}|$, then $d_{j,i}/g_{j,i} \in (0, 1)$. In this case the gradient can be reduced such that the limited reconstruction L_j is still non-constant but *cannot* cause an oscillation.
- In the remaining cases no limitation is necessary.

Now we adjust the gradient of the functions L_j with the minimum of $m_{j,i}$,

$$\tilde{m}_j := \min_{i \in \widetilde{\mathcal{Y}}_E} m_{j,i}, \qquad \widetilde{\nabla L}_j := \tilde{m}_j \nabla L_j,$$

$$\tilde{L}_j(\boldsymbol{x}) := \bar{u}_E + \widetilde{\nabla L}_j \cdot (\boldsymbol{x} - \mathbf{w}_E).$$

Finally, we denote with $\mathbb{L}_E := \{\tilde{L}_j \mid j \in \mathcal{Y}_E \text{ and } \tilde{L}_j \text{ is physical}\}$ the set of *admissible linear reconstructions*.

Note that in the cases where no least squares approach was applied, the set $\widetilde{\mathcal{Y}}_E$ could for efficiency reasons also be replaced by the set $\widetilde{\mathcal{Y}}_E^j := (\widetilde{\mathcal{Y}}_E \setminus j)$, because for all $i \in j$ we have $m_{j,i} = 1$ which results from the construction of L_j in (12). The same approach can be also used for other possible linear functions, e.g., L_{DG}, which can then be added to $\mathbb{L}_E$.

Two examples with both the stencil used for each of the linear reconstructions together with the stencil used for the limiting process are shown in Fig. 1.

4.3 Choice of Linear Function u_E^* on Troubled Cells

All linear functions in $\mathbb{L}_E$ fulfill the conditions from Sect. 3 (at least in some weak sense) so that any one of them could be used to define u_E^*. We choose the reconstruction with the

steepest gradient from $\mathbb{L}_E$, i.e., we choose $l \in \mathcal{Y}_E$ such that

$$|\widetilde{\nabla L}_l| \geq |\widetilde{\nabla L}_j| \quad \forall\, j \in \mathcal{Y}_E. \tag{13}$$

In the case that there is more than one $l \in \mathcal{Y}_E$ such that (13) is fulfilled we select one of these by random choice. Now we obtain are limited discrete solution:

$$u_E^*(\boldsymbol{x}) := \bar{u}_E + \widetilde{\nabla L}_l \cdot (\boldsymbol{x} - \mathbf{w}_E). \tag{14}$$

5 Local Grid Adaptation

Computation time can be saved by applying local grid adaptivity which reduces complexity while keeping the accuracy of the numerical scheme. For the RK-DG method several a posteriori error estimate based approaches for scalar conservation laws can be found in the literature. One example is the a posteriori error estimate based adaptation strategy described in [10] which is based on ideas given in [20]. Another example are the adaptive simulations of two-phase flow in porous media presented in [16] based on an a posteriori error estimate from [12].

In this work, for the RK-DG method a heuristic adaptation indicator is applied which is based on the ideas of the shock detector from (9). In comparison to the shock detector, the adaptation indicator is calculated over the whole boundary of an element E,

$$\mathcal{A}_E := \sum_{e \subset \partial E} \left(\frac{\int_e \chi(u_E, u_{E_e})\, ds}{\alpha_d(k)\, h_E^{(k+1)/4}\, |e|} \right), \tag{15}$$

where χ is the function describing the jump of $u_{\mathcal{G}}$ on the cell interface e. This could be the same function as for the shock detector, but it could also be different, for example, for the Euler equations we choose $\chi(u_E, u_{E_e}) = 2\frac{\rho_E - \rho_{E_e}}{\rho_E + \rho_{E_e}}$, i.e. the jump of the density instead of the pressure. We also ensure that a maximal refinement level m is not exceeded. We now mark an element for refinement if either $\mathcal{S}_E = 1$ (see (9) and (10)) or $\mathcal{A}_E > \theta_r$ (see (15)); an element not mark for refinement can be coarsened if $\mathcal{A}_E < \theta_c,\ \theta_c \in (0, \theta_r)$.

6 Numerical Results

In the following we demonstrate the effectiveness of our stabilized scheme. If not noted otherwise we use the technique described in the previous sections using the reconstructed linear functions $\mathbb{L}$ to compute the set of admissible functions—this scheme will be denoted by $\mathcal{DG} + \mathcal{R}$. We use well known test cases for the Euler equations of gas dynamics (4) for a perfect gas law with adiabatic constant $\gamma = 1.4$ (see Sect. 2.2 for details).

We use an implementation based on the DUNE-FEM library [9], a generic framework for the development of grid based numerical schemes based on the DUNE interface [1, 2]. Methods based on higher order DG discretizations on arbitrary grid structures, including non-conform or conform local adaptivity and parallelization with load balancing, are easy to implement within the DUNE-FEM context.

For the first set of test cases exact solutions are known or can be computed thus allowing us to compute the approximation error using the L^1-norm and the experimental order of convergence for different polynomial degrees.

The second set of test cases are well known problems taken from the literature for which we do not have exact solutions, but for which the quality of our approach can be measured through the comparison with other simulation results found in the literature.

6.1 Test Cases with Exact Solution

We start the numerical study of our $\mathcal{DG}+\mathcal{R}$ scheme with a set of standard Riemann problems. For these problems "exact" solutions can calculated using Chorin's method, for example found in [19, Sect. 4.2]. We demonstrate the effectiveness of the $\mathcal{DG}+\mathcal{R}$ scheme by studying the L^1-error under grid refinement and the resulting experimental order of convergence (EOC).

Numerical experiments in [19] show that one can at least expect $EOC \approx \frac{1}{2}$ for first order schemes. For schemes of higher order we expect an $EOC > \frac{1}{2}$.

Although the exact solution of the Riemann problems remains one dimensional throughout the whole time interval, we compute our results on the domain $[0,1]\times[0,0.25]\subset\mathbb{R}^2$. Thus we can study both the error and the convergence rate EOC of the scheme as well as how good it retains the one dimensional structure of the solution. This second property is challenging for any scheme on unstructured grids. Therefore, we compute results on the unstructured grid shown in Fig. 2 (details of how to reproduce this grid are given in Appendix); we refine this grid by quartering all elements in each step. To demonstrate the performance of our scheme on structured grids we will also show results for a Cartesian grid starting with $\Delta x=\Delta y=0.25$ and reducing the spacing such that the grid width is bisected for each step.

The first test case is a Riemann problem with a solution consisting of a rarefaction, a contact, and a shock wave. The second test consists of a right and a left moving rarefaction without a contact discontinuity; this problem can lead to difficulties caused by negative densities. For both problems we start the simulation with discontinuous initial data so that we cannot expect a rate of convergence above one. To study the performance of the scheme in the case of continuous solutions we use the problem with the two rarefaction waves but initialize the simulation not with the piecewise constant Riemann data but with the exact solution to the problem at a time $t>0$; thus the solution is continuous over the whole time interval. We will first describe the setting for all four test cases and give some interpretation of the results later. The initial data is defined using the primitive variables $V=(\rho,v_x,v_y,p)$.

(*TC1*) *Test Case*: *Withman problem*

Initial data:

$$V_0(x,y)=(\rho_0,v_x,v_y,p_0)(x,y)=\begin{cases}(1,0,0,1) & x<\frac{1}{2}\\ (\frac{1}{10},0,0,\frac{1}{100}) & \text{otherwise}\end{cases}$$

Boundary cond.:	$\partial\Omega_{out}=\partial\Omega_{\mathcal{G}}$ and $\partial\Omega_{in}=\partial\Omega_{refl}=\emptyset$
Final time:	$T=0.15$
Solution:	left moving rarefaction, right moving contact and shock wave (see also Fig. 6b)
L^1-error & *EOC*:	see Fig. 3 for results with $\mathcal{DG}+\mathcal{R}$

The results for the first Riemann problem which includes both a contact and a shock demonstrates that with respect to the absolute error neither the polynomial degree or the grid structure has a strong influence on our scheme. This shows the ability of the scheme to retain the one dimensional structure of the solution even on unstructured grids, which is also confirmed by the solution plot shown in Fig. 6. For all polynomial degrees the order of convergence is clearly above 0.8.

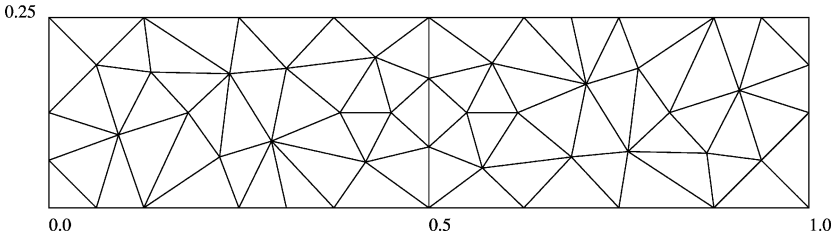


Fig. 2 Unstructured triangular grid used for the test cases with exact solution (see TC1–TC4). The grid was produced by the program `triangle` (see Appendix for the input data and link). Note that the line $\{0.5\} \times [0, 0.25]$ is represented by element boundaries to make sure that the initial data projection can be done without introducing additional errors

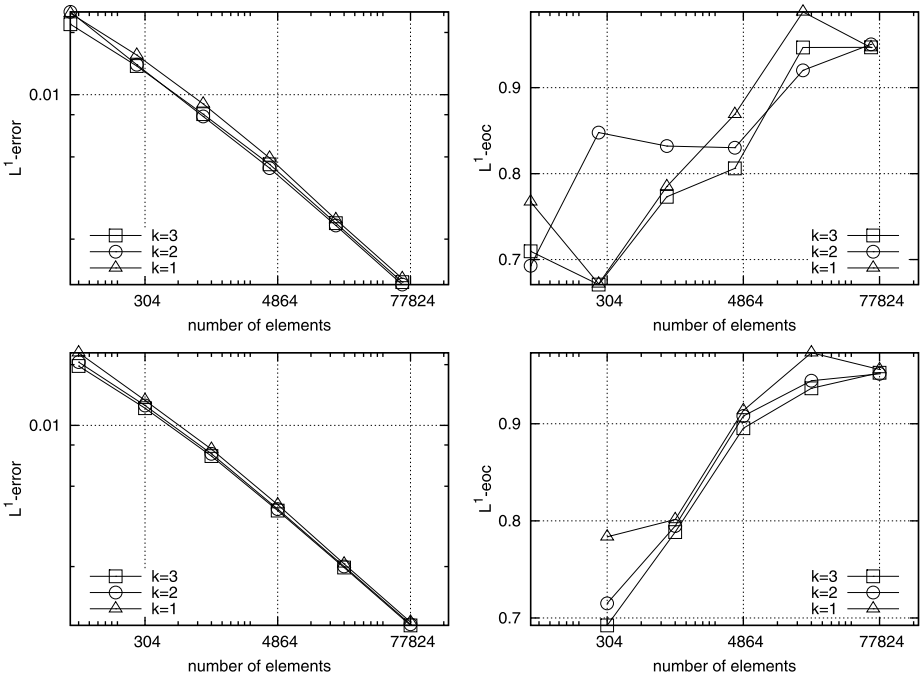


Fig. 3 Results for TC1 on Cartesian grids (*top row*) and unstructured triangular grids (*bottom row*). We compare the L^1-error and the *EOC* for polynomial degree $k = 1, 2, 3$

(*TC2*) *Test Case*: *Two rarefaction problem*
Initial data:

$$V_0(x, y) = (\rho_0, v_x, v_y, p_0)(x, y) = \begin{cases} (1, -1, 0, 1) & x < \frac{1}{2} \\ (1, 1, 0, 1) & \text{otherwise} \end{cases}$$

Final time:	$T = 0.15$
Boundary cond.:	$\partial\Omega_{out} = \partial\Omega_{\mathcal{G}}$ and $\partial\Omega_{in} = \partial\Omega_{refl} = \emptyset$
Solution:	left and right moving rarefaction, no contact discontinuity (see also Fig. 6c)
L^1-error & *EOC*:	see Fig. 4 for results with $\mathcal{DG} + \mathcal{R}$

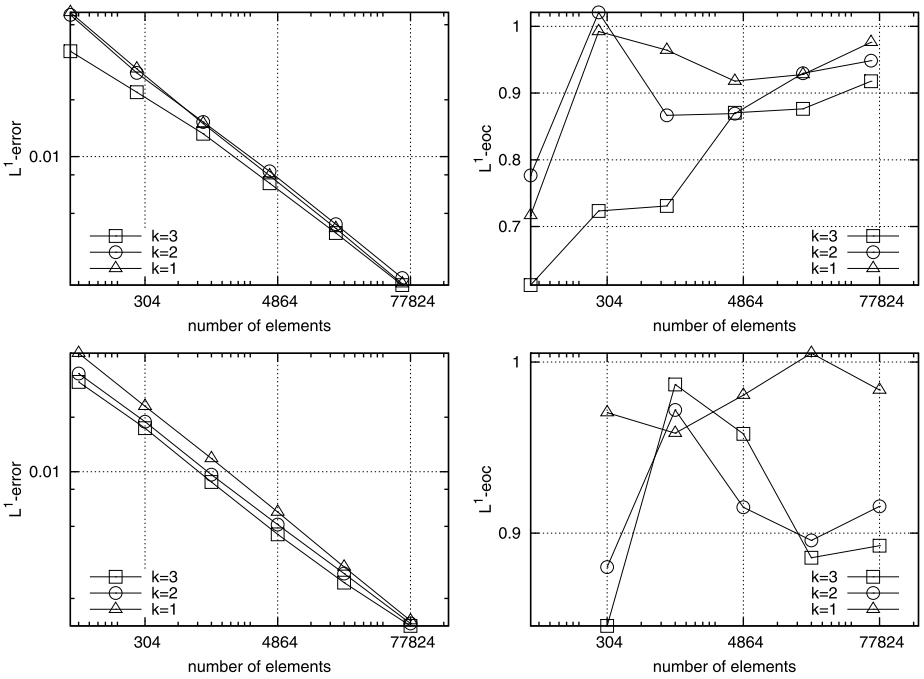


Fig. 4 Results for TC2 on Cartesian grids (*top row*) and unstructured triangular grids (*bottom row*). We compare the L^1-error and the *EOC* for polynomial degree $k = 1, 2, 3$

The remarks made so far are also basically true for the two rarefaction wave problems although the order of convergence is slightly lower in this case. The advantage of the higher order scheme in the case of continuous initial data is clearly visible in Fig. 5. The order of convergence is above 1.6 for all polynomial orders and both on the structured and the unstructured grid. Although the *EOC* is not higher for the higher order polynomials the error on a fixed grid is clearly smaller for $k = 3$ then both for $k = 2$ or $k = 1$. This shows the advantage of the scheme in smooth regions of the solution. A higher order of convergence is not be expected since the solution is not continuously differentiable. The quality of the solution can be seen in Fig. 6.

(*TC3*) *Test Case*: *Two rarefaction problem (smooth case)*	
Initial data:	the exact solution to TC2 is used at time $t = 0.05$
	the setting is otherwise as in TC2
Solution:	see Fig. 6d
L^1-error & *EOC*:	see Fig. 5 for results with $\mathcal{DG} + \mathcal{R}$

We show the density distribution as a function of the x-coordinate together with a logarithmic view of the shock indicator $\bar{J}_E$ (9) (using the right horizontal axes). The time evolution of this indicator is shown in Fig. 6a The final setting is truly two dimensional. An initial density profile and constant velocity (v_x, v_y) and pressure are chosen which results in a solution of the form $u(x, y, t) = u_0(x - v_x t, y - v_y t)$. We tested our scheme with a smooth density profile which leads to optimal convergence rates of $k + 1$. More interesting for our type of application is the case where the density profile includes a discontinuity. It is

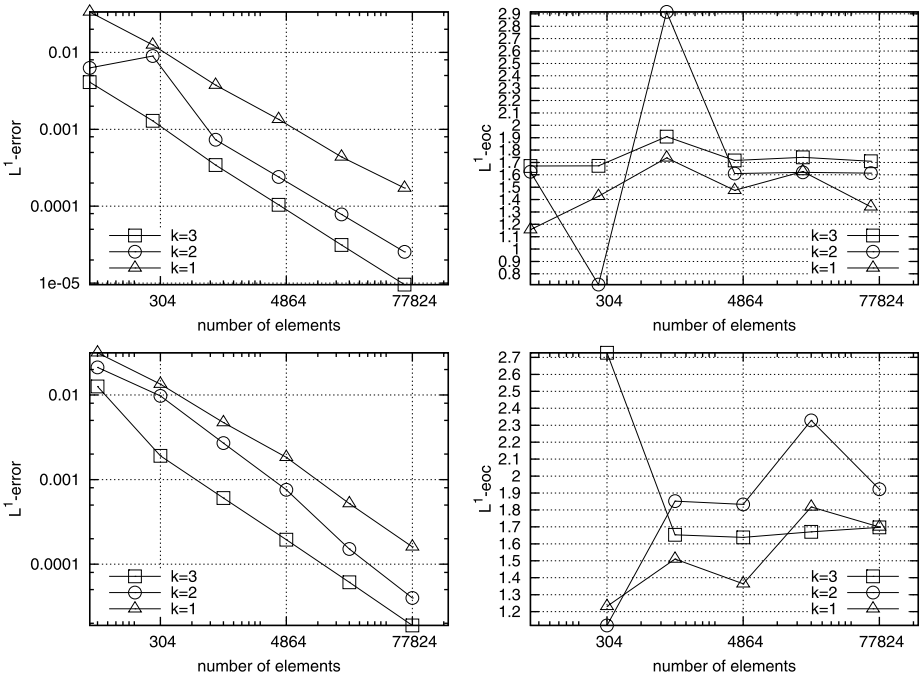


Fig. 5 Results for TC3 on Cartesian grids (*top row*) and unstructured triangular grids (*bottom row*). We compare the L^1-error and the *EOC* for polynomial degree $k = 1, 2, 3$

well known that the $\mathcal{DG}$ scheme resolves this type of contact discontinuity especially well and our test demonstrates to what degree this good resolution is retained by our $\mathcal{DG} + \mathcal{R}$ scheme.

(*TC4*) *Test Case*: *Discontinuous advection problem*
Initial data:

$$V_0(x, y) = (\rho_0, v_x, v_y, p_0)(x, y) = \begin{cases} (\rho_0, \frac{9}{2}, \frac{1}{2}, \frac{2}{5}) & x - \pi y > 0 \\ (\rho_0 + 1, \frac{9}{2}, \frac{1}{2}, \frac{2}{5}) & \text{otherwise} \end{cases}$$

with $\rho_0(x, y) = \frac{3}{5} + \frac{1}{2}|\sin(\pi x)||\sin(2\pi x)|$

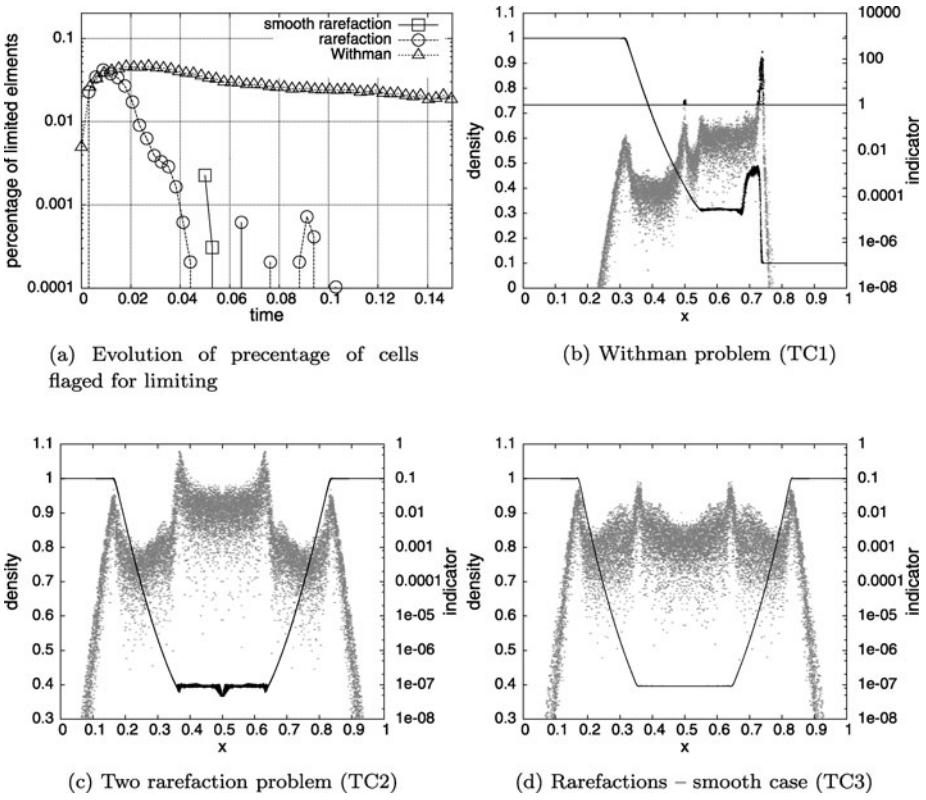
Final time: $T = 0.15$

Solution: $V(x, y, t) = V_0(x - \frac{9}{2}t, y - \frac{1}{2}t)$
where we extend V_0 periodically to the whole of $\mathbb{R}^2$.

Boundary cond.: $\partial\Omega_{in} = \partial\Omega_{\mathcal{G}}$ and $\partial\Omega_{out} = \partial\Omega_{refl} = \emptyset$,
as boundary data the exact solution is used

L^1-error & *EOC*: see Fig. 7 for results with $\mathcal{DG} + \mathcal{R}$

In order to study the optimality of the results we compare the error and convergence rates of the approximation with the projection of the initial data onto the triangular grid (Fig. 8). Note that the grid is chosen such that the initial data can be projected to the grid with producing an error, so that we only show this comparison for test cases 3 and 4. Although in the case of the discontinuous advection problem (TC4) almost no elements are limited (only a few elements in the first time steps) the convergence is smaller than the rate of 1 shown by the projection error. However, as has already been shown the convergence rate increases



(a) Evolution of precentage of cells flaged for limiting

(b) Withman problem (TC1)

(c) Two rarefaction problem (TC2)

(d) Rarefactions – smooth case (TC3)

Fig. 6 Scatter plots of the density obtained from 2d calculations with the $\mathcal{DG}+\mathcal{R}$ scheme. The values of the shock detector from (9) are also shown using the unstructured grid on level 4 with quadratic polynomials. The *solid line* indicates values of the shock detector equal to 1, which is the threshold used for applying the limiter. The evolution of the percentage of limited elements is also plotted. Note the starting time for the smooth rarefaction wave test case (TC3) is $t = 0.05$ so that the corresponding curve does not start at $t = 0$

with higher order of approximation. In the case of the smooth rarefaction problem the global convergence rate is very close to the convergence rate of the projection error.

We conclude this section with a closer look at the choice for the set of possible linear functions. Calculations for TC3 on the triangular grid are shown in Fig. 9 comparing two different strategies for choosing the linear functions: the $\mathcal{DG}$ scheme where only the linear part of the DG approximation is limited, and the $\mathcal{DG}+\mathcal{R}$ scheme where in addition linear reconstructions are calculated. Since the solution remains smooth, the $\mathcal{DG}$ scheme without limitation is stable and can be viewed as a reference solution. On coarser grids, using the $\mathcal{DG}+\mathcal{R}$ approach increases the accuracy of the scheme by an order in the grid resolution, compared to the cheaper approach, the $\mathcal{DG}$ scheme, where only the DG solution is limited; on finer grids no limiting is applied during the whole computation and the errors are the same for all three computations.

6.2 Test Cases Without an Exact Solution

In this section we consider well-known test cases for which we can compare our method with results found in the literature.

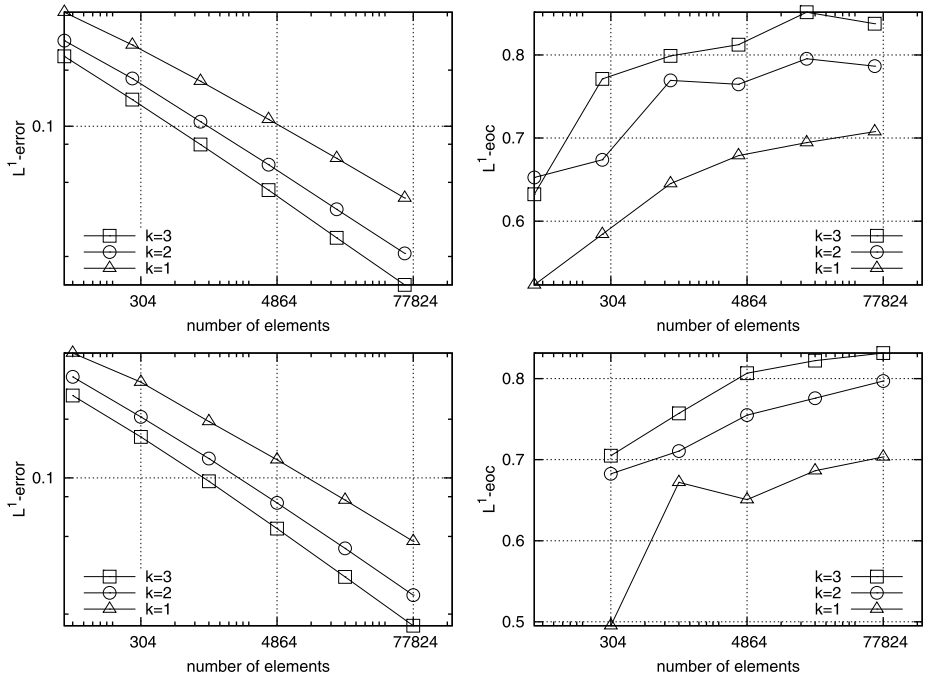
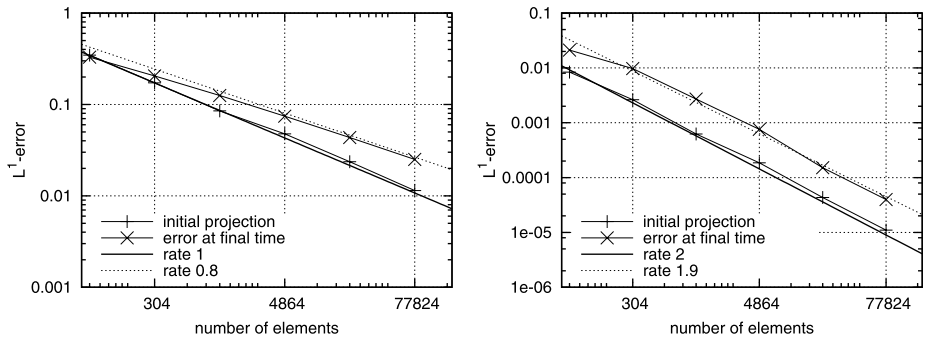


Fig. 7 Results for the $\mathcal{DG} + \mathcal{R}$ for the discontinuous advection problem (TC4) on Cartesian grids (*top row*) and unstructured triangular grids (*bottom row*). We compare the L^1-error and the *EOC* for polynomial degree $k = 1, 2, 3$



(a) Discontinuous advection (TC4) (b) Two rarefac. smooth (TC3)

Fig. 8 A comparison between the projection of the initial data onto the grid with the numerical solution at the final time is shown for the triangular grid are shown

6.2.1 The Forward Facing Step Problem

Over the last decades the *Forward Facing Step* has become a well known benchmark problem for the Euler equations for a compressible gas, see for example [5, 27] and many others.

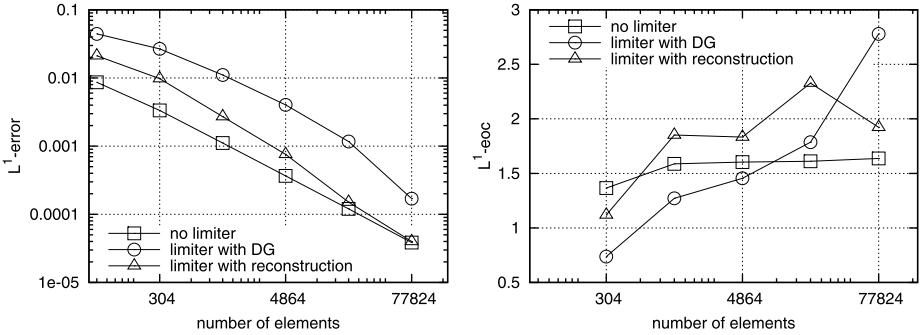


Fig. 9 We compare different versions of stabilization: limiting only the linear part of the DG solution full linear reconstruction (our $\mathcal{DG}+\mathcal{R}$ scheme), and the scheme without any stabilization. The results have been obtained using triangular grids

(*TC5*) *Test Case*: *Forward Facing Step*	
Comp. domain:	$\Omega_{\mathcal{G}} := ([0,3]\times[0,1]) \setminus ([0.6,3]\times[0,0.2]) \subset \mathbb{R}^2$
Initial data:	$V_0(x,y) = (\rho_0, v_x, v_y, p_0)(x,y) = (1.4, 3, 0, 1)$
Final time:	$T = 4$
Boundary cond.:	$\partial\Omega_{in} = (\{0\}\times[0,1])$,
	$\partial\Omega_{out} = (\{3\}\times[0.2,1])$, $\partial\Omega_{refl} = \partial\Omega \setminus (\Omega_{in} \cup \Omega_{out})$
Initial grid:	see Appendix

The results for the calculation on Cartesian grids are presented in Fig. 10. We see that the shock front is well captured by the scheme. Also, the resolution of the Kelvin-Helmholtz instability is very well developed behind the Mach stem at the top boundary. Here, one can see the advantage of the higher polynomial ansatz space. While for both, $k = 1$ and $k = 2$ the shock front is well captured, the resolution of the Kelvin-Helmholtz instability is much better with the third order ($k = 2$) method. By comparing these results with the results presented in [5] we see that the full vortex structure of the Kelvin-Helmholtz instability with our method is obtained with a grid width $\triangle x = \triangle y = 1/160$. With the method presented in [5] and [6] a grid width of $\triangle x = \triangle y = 1/320$ is necessary to see the full vortex structure. In [5] with $k = 1$ the vortex structure has not been resolved on a grid with $\triangle x = \triangle y = 1/160$. Using the $\mathcal{DG}+\mathcal{R}$ method it seems that we obtain the same resolution while saving one global refinement level.

In Fig. 11a results for the Forward Facing Step problem (TC5) on a Cartesian grid obtained with a first order FV ($k = 0$ for DG) method and in Fig. 11b for a second order FV method (taking the reconstruction from $\mathcal{DG}+\mathcal{R}$ on all cells) are shown on a Cartesian grid with spacing $1/160$. Compared with the DG solutions presented in Fig. 10 one can see that with the first order FV (DG) method the resolution of the shock as well as for the Kelvin-Helmholtz instability is very poor. With the second order FV method the resolution in the shock regions is about the same as for the second order DG ($k = 1$) method. However, the Kelvin-Helmholtz instability is not resolved at all compared to the higher order $\mathcal{DG}+\mathcal{R}$ method. In Fig. 11c and 11d results for the second order FV method on finer grids with spacing $1/320$ and $1/640$ are presented. One can see that the resolution in the shock regions becomes very good. But even on the finer grids the second order FV method is not able to resolve the structure of the Kelvin-Helmholtz instability.

The results for the calculation on a triangular grid can be found in Fig. 12. The initial triangular grid has been chosen such that the number of elements is approximately twice

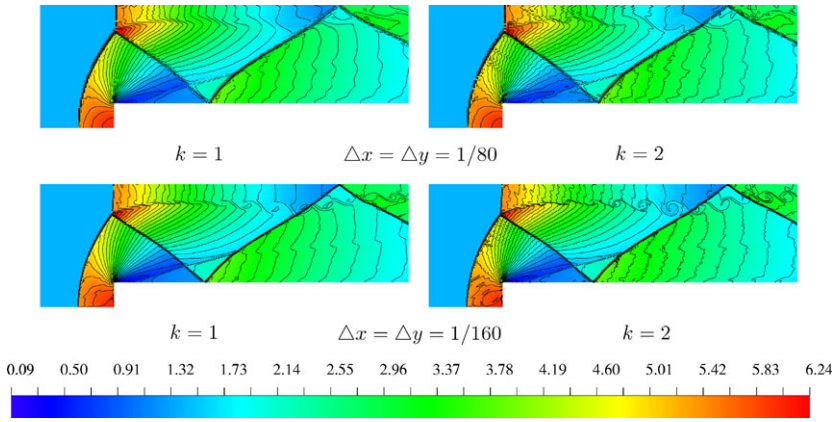


Fig. 10 Results for the $\mathcal{DG} + \mathcal{R}$ scheme for the Forward Facing Step problem (TC5) calculated on a Cartesian grid. From *top* to *bottom* the grid width is 1/80, 1/160. On the *left hand side* of the picture we see the results for $k = 1$ and on the *right hand side* the results for $k = 2$

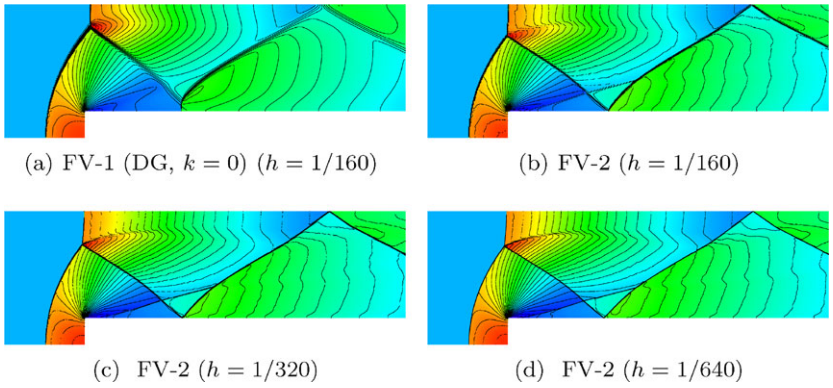


Fig. 11 Results for the Forward Facing Step problem (TC5) calculated with a first order FV (FV-1) method (**a**) and with a second order FV (FV-2) method (**b**)–(**d**) on a Cartesian grids with $h = \triangle x = \triangle y$

the number of cells in the Cartesian grid. Thus, the grid width should be more or less comparable. The overall quality for both grid structures seems similar, with the results on the Cartesian grid being slightly better. Also, we see again that our results compare favorably with those presented in [5]. In Fig. 12 we see the results for the DG method with $k = 2$, where we used our reconstruction strategy and only applied the limiting to the linear part of the DG solution ($\mathcal{DG}$ scheme). While the resolution of the Kelvin-Helmholtz instability is more or less the same (here the shock indicator hardly flags any cells for limiting) we see that the resolution of the shock is superior when the full linear reconstruction ($\mathcal{DG} + \mathcal{R}$) is applied. This is in agreement with the results presented in Fig. 8.

6.3 Test Cases for Adaptive and 3-Dimensional Computations

In the following, adaptive simulations using the adaptation indicator described in Sect. 5 are presented. We start with the 2d Forward Facing Step (TC5) problem, studied in the previous section. We then conclude with results for two well known 3d problems.

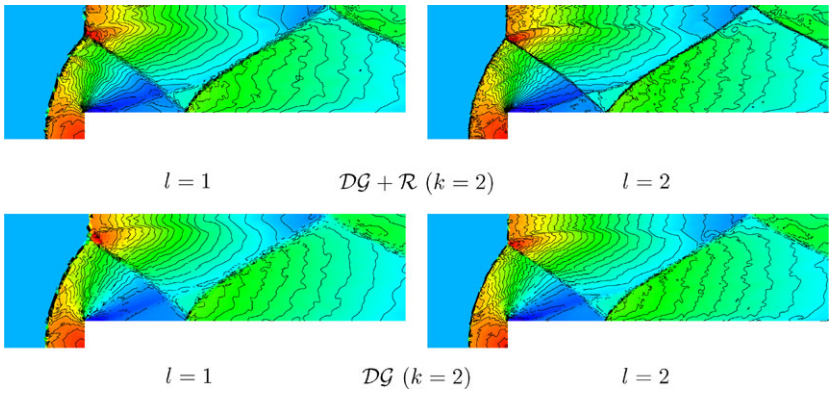


Fig. 12 Results for the $\mathcal{DG} + \mathcal{R}$ scheme for the Forward Facing Step problem (TC5) calculated on a triangular grid. From *top* to *bottom* the refinement level is 1, 2, 3. On the *left hand side* of the picture we see the results for $k = 1$ and on the *right hand side* the results for $k = 2$

6.3.1 The Forward Facing Step Problem

We consider the setting for the Forward Facing Step test case TC5 using the following parameters for the adaptation strategy.

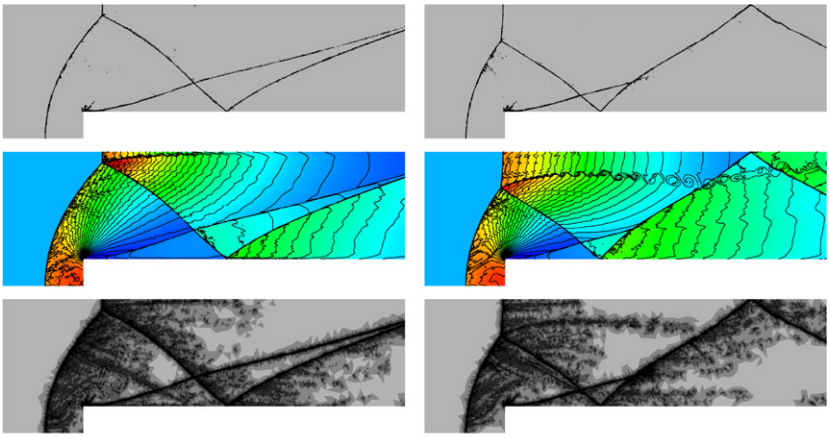
(*TC6*) *Test Case*: *Forward Facing Step (adaptive)*	
Refinement tolerance:	$\theta_r = 0.25$ (see also Sect. 5)
Coarsening factor:	$\theta_c = 0.05$ (see also Sect. 5)
Maximal refinement level:	$m = 4$

The results of the computation are presented in Fig. 13. We plot the values of the shock detector $\bar{J}_E$ (9), the density ρ, and the refinement level of each cell for the simulation times $t = 2$ and $t = 4$. One can see that the cells where $\bar{J}_E > 1$ (indicated by black regions) are only located in the region of the strong shock—the region where the method actually needs extra stabilization—regions where the density isolines are very close to each other. Cells with unphysical values almost never appear. From measurements inside the simulation program we know that in each time step there are less than 1‰ of cells which contain unphysical values in the approximate solution and thus would cause the whole simulation to break down. This means that although the unphysicality check is needed it does not dominate the numerical scheme.

The Kelvin-Helmholtz instability is resolved very nicely and in this region the grid is also refined, due to the fact that for the adaptation indicator $\mathcal{A}_E$ (15) the jump of the density has been considered. The fine cells of the grid (those with refinement level 4) are located around regions where jumps in the density occur.

6.3.2 The 3d Forward Facing Step Problem

For the 3-dimensional Euler equations (4), $d = 3$, we present a 3d version of the test case TC5 with a truly 3-dimensional computational domain.



(a) $t = 2$ (b) $t = 4$

Fig. 13 Results for the $\mathcal{DG} + \mathcal{R}$ scheme for the adaptive computation of test case TC6. Each selection shows from *top* to *bottom*: values of the shock detector $\bar{J}_E$ (9) *dark regions* indicate that limiting is applied, the density ρ, and the refinement level (dark means refinement level 4) of each cell

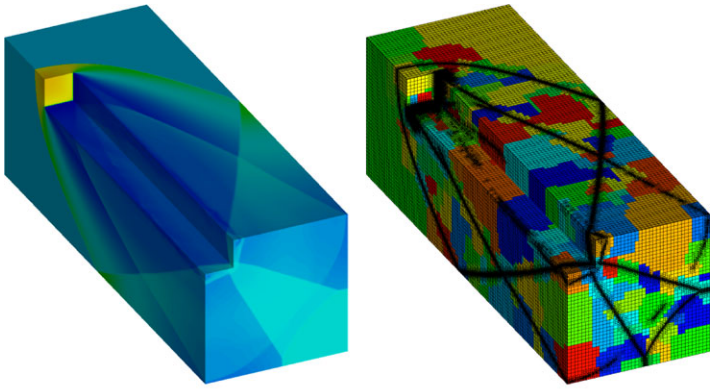
(*TC7*) *Test Case*: *3d Forward Facing Step (adaptive)*	
Comp. domain:	$\Omega_{\mathcal{G}} := ([0,3] \times [0,1] \times [0,1]) \setminus ([0.6,3] \times [0,0.2]) \times [0,0.2])$.
Initial data:	$V_0(x,y,z) = (\rho_0, v_x, v_y, v_z, p_0)(x,y,z) = (1.4, 3, 0, 0, 1)$
Final time:	$T = 2$
Boundary cond.:	$\partial\Omega_{in} = (\{0\} \times [0,1] \times [0,1])$
	$\partial\Omega_{out} = (\{3\} \times [0.2,1] \times [0.2,1])$
	$\partial\Omega_{refl} = \partial\Omega \setminus (\Omega_{in} \cup \Omega_{out})$
Initial grid:	$\triangle x = \triangle y = \triangle z = 1/40$
Refinement tol.:	$\theta_r = 1$ (see also Sect. 5)
Coarsening fac.:	$\theta_c = 0.2$ (see also Sect. 5)
Max. ref. level.:	$m = 3$

In Fig. 14 the density distribution including the adaptively refined grid can be found. One can also see the underlying grid partitioning. The simulation has been done on the parallel supercomputer XC4000 of the SCC Karlsruhe using 512 processors. For this simulation quadratic basis functions ($k = 2$) have been used. The final grid contains about 4.5 million grid cells which leads to about $2.25 \cdot 10^8$ unknowns. Grid adaptation is performed in each time step. If the local grid adaptation leads to an unbalance of work load then a dynamic load balancing is performed such that the work load is balanced again.

In Fig. 15a and 15b the parallel performance of the code is presented. In Fig. 15a the run times for one time step as well as for the ODE solver alone are plotted for the runs on 128, 256, and 512 processors. In Fig. 15b the efficiency based no these run times is shown. One can see that the overall efficiency is above 0.93 and the efficiency of the serial part (ODE solver) of the code is above 0.96 which is very close to the optimal value of 1. This indicates that the parallelization of $\mathcal{DG} + \mathcal{R}$ method is very efficient.

6.3.3 The Shock Bubble Problem

As a last test we consider the so called *Shock Bubble* problem which can for example be found in [14]. In addition, on the homepage of the software package CLAWPACK (see



(a) ρ (from below) (b) grid and partitioning (from below)

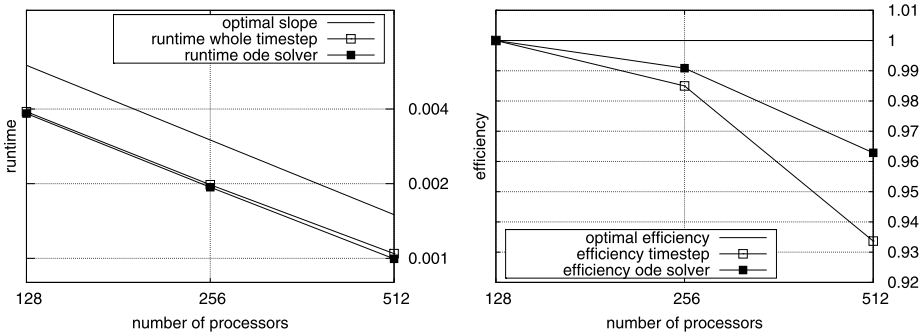
Fig. 14 Density distribution obtained with the $\mathcal{DG}+\mathcal{R}$ scheme, adapted grid and partitioning of the grid at time $t=2$ for test TC7. The calculation used ALUCUBEGRID (a hexahedral grid from DUNE) and 512 processors. Quadratic basis functions ($k=2$) have been used. The initial grid contains 185856 hexahedrons and the final grid contains about 4.5 million hexahedrons

http://www.amath.washington.edu/~claw/) results for this test problem can be found. The setup is the following.

In Fig. 16 the density and the adapted grid at time $t=0.3$ for Test Case TC8 is shown. Again, we can see that the grid is refined around the shock wave that has already passed the bubble.

(*TC8*) *Test Case*: *3d Shock Bubble*	
Computational domain:	$\Omega_{\mathcal{G}} := ([0,1.2]\times[0,0.5]\times[0,0.5]) \subset \mathbb{R}^3$.
Initial data:	$V_0(x,y,z) = (\rho_0, v_x, v_y, v_z, p_0)(x,y,z) =$
	$\begin{cases} (2.81818, 1.60644, 0, 0, 5) & x < \frac{1}{5} \\ (\frac{1}{10}, 0, 0, 0, 1) & \sqrt{((x,y,z)-(\frac{1}{2},\frac{1}{2},0))^2} < \frac{1}{5} \\ (1,0,0,0,1) & \text{otherwise} \end{cases}$
Final time:	$T = 0.3$
Boundary conditions:	$\partial\Omega_{in} = (\{0\}\times[0,0.5]\times[0,0.5])$
	$\partial\Omega_{out} = (\{1.2\}\times[0,0.5]\times[0,0.5])$
	$\partial\Omega_{refl} = \partial\Omega \setminus (\Omega_{in} \cup \Omega_{out})$
Refinement tolerance:	$\theta_r = 1$ (see also Sect. 5)
Coarsening factor:	$\theta_c = 0.2$ (see also Sect. 5)
Maximal refinement level:	$m = 3$
Initial grid:	see Appendix

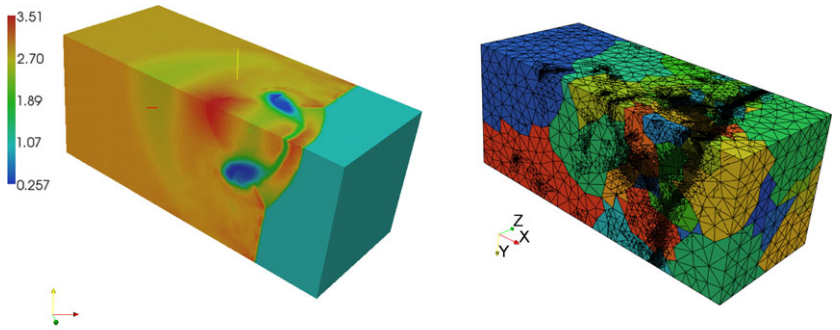
The results obtained for this test are in good agreement with the results shown on the CLAWPACK homepage (see above). We can also see that the shock and bubble structures are quite well retained on this unstructured tetrahedral grid. This also confirms the functionality of the $\mathcal{DG}+\mathcal{R}$ scheme for tetrahedral grids including non-conform refinement.



(a) Run time for one complete timestep and the run time of one timestep taken by the ODE solver are shown.

(b) Efficiency of the parallel code.

Fig. 15 Run times for a parallel computation (**a**) and efficiency of the parallel code (**b**) for the Euler equations using the third order $\mathcal{DG}+\mathcal{R}$ scheme using hexahedral elements (ALUCUBEGRID)



(a) ρ at final time $t = 0.3$

(b) grid and partitioning at $t = 0.3$

Fig. 16 Density distribution (**a**) at time $t = 0.3$ obtained with the $\mathcal{DG}+\mathcal{R}$ scheme, adapted grid and partitioning of the grid (**b**) at $t = 0.3$ for test TC8. The calculation used ALUSIMPLEXGRID (a tetrahedral grid from DUNE) and 64 processors. Quadratic basis functions ($k = 2$) have been used. The initial grid contains 14999 tetrahedrons and the final grid contains about $6 \cdot 10^5$ tetrahedrons

7 Conclusions

In this paper we have demonstrated for various numerical experiments that the newly developed $\mathcal{DG}+\mathcal{R}$ method is applicable in 2d and 3d (on Cartesian as well as on general unstructured grids) for hyperbolic problems. Also, parallel and adaptive computations on both unstructured and structured non-conforming grids (including dynamic load balancing) were considered. An efficiency study shows that this method can be used for efficient high-resolution simulations on parallel supercomputers.

In the presented method it is easily possible to use either strong gradients on troubled cells or to limit the linear part of the DG approximation. While in our test cases the former led to better results, the possibility to switch between the two different approaches is a feature of the presented approach and could be of advantage in other settings. In the future we will investigate the possible extension of the method to less restrictive limiting, keeping

higher order moments of the DG solution where possible. This could, for example, increase the resolution of the method for problems like the shock interaction with entropy waves (see [18, Example 2] and [25, Example 3.7]) where our results coincide with the results obtained in [18] and [25]. This would make it possible to keep more of the subscale resolution which is characteristic of the DG method.

Acknowledgements The first author was supported by the Landesstiftung Baden-Württemberg. The second author was supported by the German Bundesministerium für Forschung und Bildung (BMBF) under contract number 03SF0310C.

Appendix: Initial Triangular Grids

The initial triangular grids for the Riemann problems (TC1–TC4) and the Forward Facing Step (TC5) problem have been created with the Delaunay triangulator `triangle` (see http://www.cs.cmu.edu/~quake/triangle.html, version 1.6). The command line call for `triangle` to create the initial triangular grid for TC1–TC4 is: `triangle -ej -q30 -a0.005 rp.node` and for TC5: `triangle -ej -Ap -q30 -a0.00195 ffs.poly`. For the Shock Bubble (TC8) problem the 3d Delaunay triangulator `tetgen` has been used (see http://tetgen.berlios.de/). The command line call for `tetgen` to create the **SB** initial tetrahedral grid (needs the files `sb.node`, `sb.ele`, `sb.face`): `tetgen -rq25a3.8e-05 sb`. The files needed by `triangle`/`tetgen` are the following:

`rp.node`

```
8 2 0 0
0   0 0
1   1 0
2   1 0.25
3   0 0.25
4   0.5 0
5   0.5 0.25
6   0.5 0.17
7   0.5 0.08
0 1
0
0
```

`ffs.poly`

```
6  2  0  0
0  0     0
1  0.6   0
2  0.6   0.2
3  3     0.2
4  3     1
5  0     1
6 1
0 0 1 3
1 5 0 1
2 1 2 4
3 2 3 4
4 3 4 2
5 4 5 4
0
0
```

`sb.node`

```
8  3  0  0
0  0  0  0
1  0  0.5  0
2  0  0.5  0.5
3  0  0  0.5
4  1.2  0  0
5  1.2  0.5  0
6  1.2  0.5  0.5
7  1.2  0  0.5
```

`sb.ele`

```
6  4 0
0  0 1 2 4
1  1 2 4 5
2  2 4 5 6
3  2 4 6 7
4  4 2 0 7
5  2 0 7 3
```

`sb.face`

```
12  0
0  0  1  2
1  0  4  1
2  1  5  2
3  4  5  1
4  4  6  5
5  5  6  2
6  4  7  6
7  6  7  2
8  0  7  4
9  2  3  0
10 0  3  7
11 7  3  2
```

References

1. Bastian, P., Blatt, M., Dedner, A., Engwer, C., Klöfkorn, R., Kornhuber, R., Ohlberger, M., Sander, O.: A generic grid interface for parallel and adaptive scientific computing. II: Implementation and tests in DUNE. Computing **82**(2–3), 121–138 (2008)
2. Bastian, P., Blatt, M., Dedner, A., Engwer, C., Klöfkorn, R., Ohlberger, M., Sander, O.: A generic grid interface for parallel and adaptive scientific computing. I: Abstract framework. Computing **82**(2–3), 103–119 (2008)
3. Burbeau, A., Sagaut, P., Bruneau, Ch.-H.: A problem-independent limiter for high-order Runge-Kutta discontinuous Galerkin methods. J. Comput. Phys. **169**(1), 111–150 (2001)
4. Christov, I., Popov, B.: New non-oscillatory central schemes on unstructured triangulations for hyperbolic systems of conservation laws. J. Comput. Phys. **227**(11), 5736–5757 (2008)

5. Cockburn, B., Shu, C.-W.: TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws V: Multidimensional systems. J. Comput. Phys. **141**, 199–224 (1998)
6. Cockburn, B., Shu, C.-W.: Runge-Kutta discontinuous Galerkin methods for convection-dominated problems. J. Sci. Comput. **16**(3), 173–261 (2001)
7. Dedner, A., Klöfkorn, R.: A generic stabilization approach for higher order discontinuous Galerkin methods for convection dominated problems. Preprint No. 8, Universität Freiburg (2008)
8. Dedner, A., Klöfkorn, R.: Stabilization for discontinuous Galerkin methods applied to systems of conservation laws. In: Proc. of the 12th International Conference on Hyperbolic Problems: Theory, Numerics, Applications, Maryland (2008)
9. Dedner, A., Klöfkorn, R., Nolte, M., Ohlberger, M.: A generic interface for parallel and adaptive scientific computing: Abstraction principles and the DUNE-FEM module. Preprint No. 3, Mathematisches Institut, Universität Freiburg (2009)
10. Dedner, A., Makridakis, C., Ohlberger, M.: Error control for a class of Runge-Kutta discontinuous Galerkin methods for nonlinear conservation laws. SIAM J. Numer. Anal. **45**(2), 514–538 (2007)
11. Dolejší, V., Feistauer, M., Schwab, C.: On some aspects of the discontinuous Galerkin finite element method for conservation laws. Math. Comput. Simul. **61**(3–6), 333–346 (2003)
12. Ern, A., Proft, J.: A posteriori discontinuous Galerkin error estimates for transient convection-diffusion equations. Appl. Math. Lett. **18**(7), 833–841 (2005)
13. Gottlieb, S., Shu, C.-W., Tadmor, E.: Strong stability-preserving high-order time discretization methods. SIAM Rev. **43**(1), 89–112 (2001)
14. Hagen, T.R., Lie, K.-A., Natvig, J.R.: Solving the Euler equations on graphics processing units. In: Alexandrov V.N., et al. (eds.) Proceedings, Part IV, Computational Science—ICCS 2006. 6th International Conference, Reading, UK, 28–31 May, 2006. Lecture Notes in Computer Science, vol. 3994, pp. 220–227. Springer, Berlin (2006)
15. Hoteit, H., Ackerer, Ph., Mosé, R., Erhel, J., Philippe, B.: New two-dimensional slope limiters for discontinuous Galerkin methods on arbitrary meshes. Int. J. Numer. Methods Eng. **61**(14), 2566–2593 (2004)
16. Klieber, W., Rivière, B.: Adaptive simulations of two-phase flow by discontinuous Galerkin methods. Comput. Methods Appl. Mech. Eng. **196**(1–3), 404–419 (2006)
17. Krivodonova, L.: Limiters for high-order discontinuous Galerkin methods. J. Comput. Phys. **226**(1), 879–896 (2007)
18. Krivodonova, L., Xin, J., Remacle, J.-F., Chevaugeon, N., Flaherty, J.E.: Shock detection and limiting with discontinuous Galerkin methods for hyperbolic conservation laws. Appl. Numer. Math. **48**(3–4), 323–338 (2004)
19. Kröner, D.: Numerical Schemes for Conservation Laws. Wiley, Teubner, New York, Leipzig (1997)
20. Kröner, D., Ohlberger, M.: A posteriori error estimates for upwind finite volume schemes for nonlinear conservation laws in multi dimensions. Math. Comput. **69**, 25–39 (2000)
21. Luo, H., Baum, J.D., Löhner, R.: A Hermite WENO-based limiter for discontinuous Galerkin method on unstructured grids. J. Comput. Phys. **225**(1), 686–713 (2007)
22. Persson, P.-O., Peraire, J.: Sub-cell shock capturing for discontinuous Galerkin methods. In: 44th AIAA Aerospace Sciences Meeting, AIAA-2006-0112, Reno, Nevada, Department of Aeronautics & Astronautics. Massachusetts Institute of Technology (2006)
23. Qiu, J., Shu, C.-W.: A comparison of troubled-cell indicators for Runge-Kutta discontinuous Galerkin methods using weighted essentially nonoscillatory limiters. SIAM J. Sci. Comput. **27**(3), 995–1013 (2005)
24. Qiu, J., Shu, C.-W.: A comparison of troubled-cell indicators for Runge-Kutta discontinuous Galerkin methods using weighted essentially nonoscillatory limiters. SIAM J. Sci. Comput. **27**(3), 995–1013 (2005)
25. Qiu, J., Shu, C.-W.: Runge-Kutta discontinuous Galerkin method using WENO limiters. SIAM J. Sci. Comput. **26**(3), 907–929 (2005)
26. Wesenberg, M.: Efficient finite—volume schemes for magnetohydrodynamics simulations in solar physics. Ph.D. thesis, Universität Freiburg (2003)
27. Woodward, P., Colella, P.: The numerical simulation of two-dimensional fluid flow with strong shocks. J. Comput. Phys. **54**, 115–173 (1984)
28. Xin, Jianguo, Flaherty, Joseph E.: Viscous stabilization of discontinuous Galerkin solutions of hyperbolic conservation laws. Appl. Numer. Math. **56**(3–4), 444–458 (2006)