Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 226 (2007) 879–896
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Limiters for high-order discontinuous Galerkin methods

Lilia Krivodonova *

Applied Mathematics Department, University of Waterloo, 200 University Ave West, Waterloo, ON, Canada N2L 3G1

Received 30 June 2006; received in revised form 1 May 2007; accepted 3 May 2007
Available online 25 May 2007

Abstract

We describe a limiter for the discontinuous Galerkin method that retains as high an order as possible, and does not automatically reduce to first order. The limiter is a generalization of the limiter introduced in [R. Biswas, K. Devine, J.E. Flaherty, Parallel adaptive finite element methods for conservation laws, Applied Numerical Mathematics 14 (1994) 255–284]. We present the one-dimensional case and extend it to two-dimensional problems on tensor-product meshes. Computational results for examples with both smooth and discontinuous solutions are shown.

Keywords: Limiters; High-resolution schemes; Discontinuous Galerkin methods; Euler equations

1. Introduction

Discontinuous Galerkin (DG) methods are becoming popular due to the ease of increasing the order of approximation while keeping the stencil local. They combine the ease of finite element approximations in handling complex geometry and adaptation with the shock-capturing abilities of finite volume schemes. One aspect of these methods, however, that is not yet satisfactory is limiting. When a DG solution is limited, most methods reduce the solution to first-order accuracy, and much of the advantage of high-order methods is lost.

Some form of nonlinear limiting seems necessary in high-order computations of discontinuous flows [9,22,10]. We propose a limiter for use with the DG schemes for hyperbolic conservation laws that can limit gradually, systematically reducing the order of accuracy depending on the behavior of the higher-order solution derivatives. It does not automatically reduce to first order. We then extend it to two-dimensional problems on tensor-product meshes. The limiter is a generalization of the *moment* limiter proposed by Biswas et al. [3].

The moment limiter itself is a generalization of the second-order accurate minmod limiter of van Leer [22] to higher orders of approximation. The minmod limiter reduces the slope in a cell if the solution in that cell exceeds the range of solution averages on neighboring cells. The moment limiter works in a similar way: it

* Tel.: +1 519 888 4567x38138.
E-mail address: lgk@math.uwaterloo.ca

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2007.05.011

limits the derivative of order i in a given cell using the derivatives of order $i-1$ in neighboring cells. As with van Leer type limiters, the strength of the moment limiter can be varied.

With DG methods, high-order limiting on general meshes remains an open question both from the theoretical and practical points of view. DG methods achieve formal order of accuracy $p+1$ by representing the solution as a polynomial of degree p in each computational cell. In the absence of high-order limiters, alternative techniques have been developed to control oscillations in approximations of order $p>1$. There are a number of discontinuity detection strategies where discontinuities are first detected and then a limiter (usually minmod) is applied only on the elements that are believed to contain a discontinuity. For an overview and comparison of such strategies, see [18]. Qiu et al. [19] proposed using a high-order WENO reconstruction instead of the minmod limiter in conjunction with a discontinuity detection strategy. Jaffre et al. [14] introduced the idea of adding artificial viscosity as a stabilization tool. The amount of viscosity is based on the size of the residual. An implementation of this approach can be found in [12]. More recently an artificial viscosity term based on h and p was used in [17], also in conjunction with a discontinuity detection strategy and sub-cell resolution. Modal filtering has been successfully applied to a nodal based DG, see for example [8]. Finally, Hoteit et al. [13] developed a limiter applicable to piecewise quadratic solutions on rectangular meshes. The limiter is vertex based: it requires the degree of freedom associated with a vertex to lie between the cell averages of all elements containing the vertex. A minimization problem needs to be solved on each cell. In contrast to these approaches, our limiter is closer in spirit to those used in finite volume schemes.

Our limiter is applied progressively, limiting first the high-order terms as needed (e.g. as the solution starts to steepen). The process continues until either a coefficient is found that does not need to be limited or all terms are limited. This has two beneficial effects. First, it achieves the highest possible accuracy when some limiting is necessary. Second, gradual introduction of the limiter seems to avoid artificial limiter-induced steepening – turning sine waves into square waves – that happens with some limiters.

One reason for the absence of high-order limiters might be the lack of analytical tools. The total variation diminishing (TVD) theory of Harten [10] has been very powerful in constructing second-order limiters in one space dimension. Harten looks at the total variation of the solution means

$$TV = \sum_j |\overline{U}_{j+1} - \overline{U}_j|, \tag{1}$$

which should be non-increasing with time. However, such schemes reduce to first-order near smooth extrema [11]. This leads to the conclusion that all TVD schemes are at most second-order accurate. However, there are a couple of interesting constructions for piecewise parabolic solutions of scalar problems in one dimension [20,16] that are TVD in a different sense. They measure the total variation of the entire function, consisting of the variation of the solution within mesh cells, and including the jumps between cells. It is interesting to note that with the minmod limiter, linear DG solutions in one dimension are TVD in means and are not TVD in this more general sense (see Example 4.1.2).

The limiter that we propose is not total variation diminishing in either sense. In our experiments, the solutions are total variation bounded, but we are unable to prove this analytically. The adaptive character of the limiter (we stop if a coefficient is found that does not need to be limited) makes it difficult to analyze. We should note that some commonly used schemes such as ENO/WENO [11,15] and PPM [6] are not provably TVD, but also seem to be nonlinearly stable.

The outline of the remainder of the paper is as follows. In Section 2 we present the moment limiter in the one-dimensional case, for both scalar equations and systems of equations. Section 3 extends the limiter to two space dimensions. Computational results on a variety of test cases in both one and two dimensions are presented in Section 4 with conclusions following in Section 5. We largely omit a description of local Runge–Kutta DG schemes. Classical papers of Cockburn and Shu are a good reference [5,4]; details of the specific implementation used by the author can be found in [7].

2. One-dimensional limiting

In this section, we present the moment limiter, which is a generalization of the minmod limiter. We will argue that the ith derivative of the numerical solution should not exceed forward and backward differences

of the $(i-1)$st derivative multiplied by a scaling factor. This will lead to a family of limiters, parametrized by a factor from the interval $[1/(2(2i-1)), 1]$. One end of the interval is too diffusive and the other end might not reduce the non-monotonicity enough. However, the solutions remained stable in our experiments for all factors from the interval. The amount of limiting depends on the order of the derivative with higher derivatives being allowed bigger variations relative to the neighboring values. We argue that it is safe to reduce the amount of limiting performed on the higher derivatives since they actually grow at discontinuities, while their contribution to the solution is small in smooth regions. We recommend the smallest amount of limiting, i.e. the right end of the interval, and use it in all numerical experiments presented in Section 4. We note that ENO schemes [11] also allow some oscillations to develop in higher derivatives.

We explain the idea behind the moment limiter in the simplest setting: a scalar equation

$$u_t + f(u)_x = 0 \tag{2}$$

on a uniform mesh in one space dimension. In the DG method, the solution is represented as a polynomial of degree p in each cell. Let ξ be a "computational coordinate" variable that ranges linearly from -1 to 1 on cell $[x_k, x_{k+1}]$. Within cell k we have

$$U_k = \sum_{i=0}^{p} c_i^k P_i(\xi), \tag{3}$$

where the P_i are the Legendre polynomials [1] normalized so that $P_i(1) = 1$, and c_i^k are the solution coefficients. The map between the physical and computational variables is given by

$$x = \frac{1-\xi}{2} x_k + \frac{1+\xi}{2} x_{k+1}. \tag{4}$$

We limit the solution (3) by limiting its coefficients. Starting with the highest coefficient (at the top level) $i = p$, we replace c_i^k with

$$\tilde{c}_i^k = \text{minmod}(c_i^k, D_i^{+k}, D_i^{-k}). \tag{5}$$

Here

$$\text{minmod}(a,b,c) = \begin{cases} \text{sgn}(a)\min(|a|,|b|,|c|) & \text{if } \text{sgn}(a) = \text{sgn}(b) = \text{sgn}(c) \\ 0 & \text{otherwise} \end{cases}. \tag{6}$$

The limiter is active if $\tilde{c}_i^k \neq c_i^k$. Formula (5) seeks to contain spurious growth in $c_i^k,\ i = 1, 2, \ldots, p$, by comparing them to and limiting them by suitable quantities D_i^{+k} and D_i^{-k}. Roughly speaking, c_i corresponds to the ith derivative of the solution, so it will be compared to the forward and backward differences of the $(i-1)$st derivative, which are alternative approximations to the ith derivative.

A Taylor series expansion of the solution reveals that in the absence of discontinuities, $c_i^k,\ i = 1, 2, \ldots, p$, is in fact an estimate of $\partial_x^i u$ on cell k up to a scaling factor:

$$c_i^k \approx C \Delta x^i \partial_x^i u(\zeta), \quad \zeta \in [x_k, x_{k+1}]. \tag{7}$$

Thus, limiting of c_i^k amounts to limiting the approximations of solution derivatives.

Our choice of the forward and backward differences in (5) can be motivated by the following close examination of the coefficients of the expansion and their relationship to derivatives. First, we compute the ith and $(i-1)$st derivatives of U_k with respect to x. Recall that the coefficient of the leading term $a_n x^n$ of the Legendre polynomial of order n with the chosen normalization is given by [1]

$$a_n = \frac{(2n)!}{2^n n! n!}. \tag{8}$$

Combining (8) with the mapping (4) gives the following expressions for the indicated derivatives

$$\frac{\partial^{i-1} U_k}{\partial x^{i-1}} = \left(\frac{2}{\Delta x}\right)^{i-1} (2i-3)!! c_{i-1}^k + \left(\frac{2}{\Delta x}\right)^{i-1} \frac{\partial^{i-1}}{\partial \xi^{i-1}} \sum_{m>i-1}^{p} c_m^k P_m(\xi) \tag{9}$$

and

$$\frac{\partial^i U_k}{\partial x^i} = \left(\frac{2}{\Delta x}\right)^i (2i-1)!! c_i^k + \left(\frac{2}{\Delta x}\right)^i \frac{\partial^i}{\partial \xi^i} \sum_{m>i}^{p} c_m^k P_m(\xi). \tag{10}$$

In (9), the terms corresponding to coefficients of order less than $i-1$ in x are identically zero, and those of order greater than $i-1$ are collected in the sum. Expression (10) has a similar form. Computing the forward difference of the $\partial_x^{i-1} U_k$ and rearranging the terms gives

$$\left(\frac{\partial^{i-1} U_{k+1}}{\partial x^{i-1}} - \frac{\partial^{i-1} U_k}{\partial x^{i-1}}\right) \Big/ \Delta x = \left(\frac{2}{\Delta x}\right)^i \frac{(2i-3)!!}{2} (c_{i-1}^{k+1} - c_{i-1}^k) + \left(\frac{2}{\Delta x}\right)^i \frac{\partial^{i-1}}{\partial \xi^{i-1}} \sum_{m>i-1}^{p} \frac{1}{2} (c_m^{k+1} - c_m^k) P_m(\xi). \tag{11}$$

Formulas (10) and (11) are approximations of the same quantity $\partial_x^i u$. Comparing the $(2/\Delta x)^i$ order terms in (10) and (11), and taking into account (7) reveals

$$c_i^k = \frac{c_{i-1}^{k+1} - c_{i-1}^k}{2(2i-1)} + \mathrm{O}(\Delta x^{i+1}). \tag{12}$$

A similar relation holds between c_i^k and the backward difference of c_{i-1}^k. In a special case when the exact solution of (2) belongs to the finite element space, i.e. is a polynomial of degree less than or equal to p, (12) is simplified and there are no higher order terms.

We could have based the limiter for c_i^k on (12), for example by setting $D_i^{+k} = (c_{i-1}^{k+1} - c_{i-1}^k)/2(2i-1)$ and $D_i^{-k} = (c_{i-1}^k - c_{i-1}^{k-1})/2(2i-1)$, but this choice is too diffusive. Instead, we try to find a limiter of the same general form

$$\tilde{c}_i^k = \mathrm{minmod}(c_i^k, \alpha_i(c_{i-1}^{k+1} - c_{i-1}^k), \alpha_i(c_{i-1}^k - c_{i-1}^{k-1})). \tag{13}$$

Note that α_i is a variable depending on the order of the coefficient. We propose the following range for α_i

$$\frac{1}{2(2i-1)} \leqslant \alpha_i \leqslant 1. \tag{14}$$

Choosing α_i outside of this region resulted in either loss of accuracy or numerical instability. In the experiments presented in Section 4, we used $\alpha_i = 1$, i.e. the mildest limiter possible.

We propose to use the limiter in the following way. Starting at the top level $i = p$, we replace c_i^k with

$$\tilde{c}_i^k = \mathrm{minmod}(c_i^k, c_{i-1}^{k+1} - c_{i-1}^k, c_{i-1}^k - c_{i-1}^{k-1}). \tag{15}$$

If $\tilde{c}_i^k = c_i^k$ we stop. Otherwise we limit c_{i-1}^k, continuing until $i = 1$ or stopping the first time $\tilde{c}_i^k = c_i^k$. Note that the limiter (15) is equivalent to allowing the leading term of the ith derivative to be $2(2i-1)$ times bigger than the forward and backward differences of the $(i-1)$st derivatives on the neighboring elements.

When $p = 1$, the limiter (15) becomes the familiar minmod limiter of [5] that compares the slope on k to twice the forward and backward differences of the solution averages.

Remark 1. It is important both to start limiting from the highest coefficient and to stop when the first coefficient that does not need to be limited is encountered. This adaptive action of the limiter can be seen in Example 4.1.1. Using the limiter as an indicator, i.e. either limiting all coefficients if at least one was limited, or not limiting at all if the limiter did not reach the lowest coefficient, does not work nearly as well. The first choice results in order reduction and the second in spurious oscillations in the solution. Limiting from the lowest coefficients up results in flattening of smooth extrema and reduction of accuracy. The moment limiter seems to identify and limit the exact amount of oscillations developed in the solution.

Remark 2. For nonlinear systems, the limiter must be applied to the characteristic variables. Applying the limiter to the conserved variables leads to spurious oscillations near discontinuities even with a linear spatial approximation [4]. The limiting is performed in the following way

$$(\widetilde{\mathbf{L}\mathbf{C}_i^k})_j = \mathrm{minmod}((\mathbf{L}\mathbf{C}_i^k)_j, (\mathbf{L}(\mathbf{C}_{i-1}^{k+1} - \mathbf{C}_{i-1}^k))_j, (\mathbf{L}(\mathbf{C}_{i-1}^k - \mathbf{C}_{i-1}^{k-1}))_j), \quad j = 1, 2, \ldots, N, \tag{16}$$

where $\mathbf{L}$ is a matrix of left eigenvectors of the Jacobian of (2) on cell k evaluated using the Roe average [4], $\mathbf{C}_i^k = (c_{i,1}^k, c_{i,2}^k, \ldots, c_{i,N}^k)$, and N is the number of equations in the system. The result is mapped back into the

conserved variables space by multiplication with the matrix of right eigenvectors. In this notation, $(\mathbf{LC}_i^k)_j$ is the ith coefficient of the expansion of the jth characteristic variable in the space of basis functions. Characteristic variables must be limited independently from one another: we stop limiting variable j if $(\widetilde{\mathbf{LC}}_i^k)_j = (\mathbf{LC}_i^k)_j$ for some i, but might have to continue to limit other characteristic variables. Presumably the same could be done for primitive variables, but we have not yet investigated this.

Remark 3. As a consequence of limiting of characteristic variables, some negative pressure values that are produced by the scheme might not be corrected by the limiter. Should this occur, we limit the coefficients corresponding to linear terms according to (5) and set higher order coefficients to zero. If this is not enough to eliminate negative pressure values, all coefficients except for c_0^k are set to zero.

3. Two-dimensional limiting

In this section we extend the moment limiter to tensor-product meshes in two space dimensions. This is less straightforward than one might think, because the x and y directions are coupled in the expansion of the solution. We use both directions to limit each coefficient, and we adjust the constant used in the limiting as well. The justification is outlined below.

We consider DG solutions of a two-dimensional scalar problem of the form (2) on a uniform rectangular grid with grid spacing Δx and Δy. We map each grid cell to a computational cell $[-1,1]\times[-1,1]$ with a map of the form (4) in both the x and y directions. The computational variables are now ξ and η. We construct a tensor-product basis using the Legendre polynomials. The basis functions are normalized so that $\|P_iP_j\|_{L2}=1$ on the computational cell. The solution on cell k,m in terms of this basis is given by

$$U_{k,m}=\sum_{i=0,j=0}^{p}\frac{\sqrt{(2i+1)(2j+1)}}{2}c_{i,j}^{k,m}P_i(\xi)P_j(\eta). \tag{17}$$

As in one space dimension, we seek to limit solution derivatives. First, we differentiate $U_{k,m}$ i times in x and j times in y when i,j are not simultaneously equal to zero to get

$$\frac{\partial^{i+j}U_{k,m}}{\partial x^i\partial y^j}=\left(\frac{2}{\Delta x}\right)^i\left(\frac{2}{\Delta y}\right)^j\left[\frac{\sqrt{(2i+1)(2j+1)}}{2}(2i-1)!!(2j-1)!!c_{i,j}^{k,m}+\frac{\partial^{i+j}}{\partial\xi^i\,\partial\eta^j}\sum_{u>i,v>j}^{p}c_{u,v}^{k,m}P_u(\xi)P_v(\eta)\right]. \tag{18}$$

We limit $c_{i,j}^{k,m}$ using forward and backward differences of lower derivatives of $U_{k,m}$. Two derivatives can serve this purpose: $\partial_x^i\partial_y^{j-1}U$ and $\partial_x^{i-1}\partial_y^jU$. After some algebra similar to the one-dimensional case, we find

$$\frac{\partial^{i+j-1}U_{k,m}}{\partial x^i\,\partial y^{j-1}}=\left(\frac{2}{\Delta x}\right)^i\left(\frac{2}{\Delta y}\right)^{j-1}\left[\frac{\sqrt{(2i+1)(2j-1)}}{2}(2i-1)!!(2j-3)!!c_{i,j-1}^{k,m}+\frac{\partial^{i+j-1}}{\partial\xi^i\partial\eta^{j-1}}\sum_{u>i,v\geqslant j}^{p}c_{u,v}^{k,m}P_u(\xi)P_v(\eta)\right],\quad j>0, \tag{19}$$

and

$$\frac{\partial^{i+j-1}U_{k,m}}{\partial x^{i-1}\,\partial y^j}=\left(\frac{2}{\Delta x}\right)^{i-1}\left(\frac{2}{\Delta y}\right)^j\left[\frac{\sqrt{(2i-1)(2j+1)}}{2}(2i-3)!!(2j-1)!!c_{i-1,j}^{k,m}+\frac{\partial^{i+j-1}}{\partial\xi^i\,\partial\eta^{j-1}}\sum_{u\geqslant i,v>j}^{p}c_{u,v}^{k,m}P_u(\xi)P_v(\eta)\right],\quad i>0. \tag{20}$$

Computing the forward differences of $\partial_x^{i-1}\,\partial_y^jU$ and $\partial_x^i\,\partial_y^{j-1}U$ in the x and y directions, respectively, and comparing them to the leading term of (18) results in

$$c_{i,j}^{k,m}=\frac{c_{i-1,j}^{k+1,m}-c_{i-1,j}^{k,m}}{2\sqrt{4i^2-1}}+\mathrm{O}(\Delta x^{i+1}\Delta y^{i+1}) \tag{21a}$$

and

$$c_{i,j}^{k,m} = \frac{c_{i,j-1}^{k,m+1} - c_{i,j-1}^{k,m}}{2\sqrt{4j^2-1}} + \mathrm{O}(\Delta x^{i+1}\Delta y^{j+1}). \tag{21b}$$

As in one space dimension, (21a) and (21b) are exact equalities if the solution actually belongs to the space spanned by the basis. We use both to limit $c_{i,j}^{k,m}$ but with different coefficients

$$\tilde{c}_{i,j}^{k,m} = \mathrm{minmod}\left(c_{i,j}^{k,m}, \alpha_j(c_{i,j-1}^{k,m+1} - c_{i,j-1}^{k,m}), \alpha_j(c_{i,j-1}^{k,m} - c_{i,j-1}^{k,m-1}), \alpha_i(c_{i-1,j}^{k+1,m} - c_{i-1,j}^{k,m}), \alpha_i(c_{i-1,j}^{k,m} - c_{i-1,j}^{k-1,m})\right). \tag{22}$$

For $c_{0,j}^{k,m}$ and $c_{i,0}^{k,m}$, $i, j = 1, 2, \ldots, p$, the limiter (22) has only three terms. The coefficients α_i and α_j are defined by using the same reasoning as the one-dimensional case. We do not allow $c_{i,j}^{k,m}$ to exceed $2(2i-1)$ times the lower derivative in the x direction and $2(2j-1)$ times the lower derivative in the y direction. This and (21) result in the following choice of α

$$\frac{1}{2\sqrt{4n^2-1}} \leqslant \alpha_n \leqslant \sqrt{\frac{2n-1}{2n+1}}. \tag{23}$$

Note that the difference in the factors between the one- and two-dimensional cases are due to the difference in normalization of the basis functions.

The least diffusive choice of α, the one corresponding to the right end of the interval (23), is used in all computations presented in Section 4. As in the one-dimensional case, the limiter is applied adaptively starting with the highest coefficient $c_{p,p}$ and continuing to lower ones if necessary. The limiting is performed by applying (22) to consecutively lower terms of (17) until a coefficient $c_{i,i}$ or a pair of coefficients $c_{i,j}$ and $c_{j,i}$ are found that are not changed by the limiter. For coefficients $c_{i,j}$, $i \neq j$, we require a symmetric pair $c_{i,j}$ and $c_{j,i}$ not be changed by the limiter in order to assume that no further limiting is necessary. More specifically, the order of limiting is the following. First, any coefficient with a subscript p is limited, and within those, they are limited from highest to lowest. For example, the highest coefficient $c_{p,p}$ is examined. If it does not need limiting, the procedure stops. If it does, the pair of coefficients $c_{p,p-1}$ and $c_{p-1,p}$ is examined. If they do not need limiting, the procedure stops, otherwise the pair $c_{p,p-2}$ and $c_{p-2,p}$ is examined. When all coefficients with a p are finished, those with at least a $p-1$ subscript that have not yet been examined go next. The following describes the ordering of elements for limiting from the highest to the lowest:

$$\begin{aligned}
&c_{p,p}, c_{p,p-1} \text{ and } c_{p-1,p}, c_{p,p-2} \text{ and } c_{p-2,p}, \ldots, c_{p,0} \text{ and } c_{0,p},\\
&c_{p-1,p-1}, c_{p-1,p-2} \text{ and } c_{p-2,p-1}, c_{p-1,p-3} \text{ and } c_{p-3,p-1}, \ldots, c_{p-1,0} \text{ and } c_{0,p-1},\\
&\ldots\\
&c_{1,1}, c_{1,0} \text{ and } c_{0,1}.
\end{aligned} \tag{24}$$

Detailed formulas illustrating the use of limiter (22) for a quadratic approximation are given in Appendix A.

4. Numerical examples

This section shows the behavior of the limiter on a variety of test problems. Examples with both smooth solutions and discontinuous solutions, and in one and two space dimensions, are presented. The implementation of the limiter is straightforward.

4.1. One-dimensional examples

4.1.1. Linear advection

We consider the linear initial value problem

$$\begin{aligned}
&u_t + u_x = 0, \quad -1 \leqslant x < 1, \ t > 0,\\
&u(x,0) = u_0(x),\\
&u(x,-1) = u(x,1),
\end{aligned} \tag{25}$$

with

$$u_0(x) = \sin \pi x. \tag{26}$$

The exact solution of this problem is smooth and does not require limiting. To measure the loss of accuracy inflicted by the moment limiter, we solve the problem with and without the limiter and compare the results. We present errors ϵ_1 in the L^1 norm at $t = 2$ after one period on uniform meshes having 16, 32, 64, 128, and 256 elements in Table 1 (no limiter) and Table 2 (moment limiter) for p ranging from 1 to 4. Even on this smooth example, the limiter decreases the accuracy quite a bit. This is expected, but it is no worse than a corresponding finite volume scheme, see for example [2]. Also note that the convergence rate is preserved. (In fact, since the coarsest grid does so poorly the rate appears higher than expected, but would settle back down). Next, we advect the initial profile further in time to $t = 100$ using the minmod and moment limiters and plot the solutions in Fig. 1 for $p = 4$ on a 32 element mesh with 11 points per element. There is no visual change in the shape of the solution obtained with the moment limiter while the minmod limiter transforms the sinusoidal wave into a square pulse. The L^1 errors are 3.85e−07 and 2.49e−01, respectively. This indicates that the moment limiter might be especially beneficial for problems requiring long-time simulations.

Next, we solve (25) with the initial conditions [15]

$$u_0(x) = \begin{cases} \frac{1}{6}(G(x,\beta,z-\delta) + G(x,\beta,z+\delta) + 4G(x,\beta,z)), & -0.8 \leqslant x \leqslant -0.6, \\ 1, & -0.4 \leqslant x \leqslant -0.2, \\ 1 - |10(x-0.1)|, & 0 \leqslant x \leqslant 0.2, \\ \frac{1}{6}(F(x,\alpha,a-\delta) + F(x,\alpha,a+\delta) + 4F(x,\alpha,z)), & 0.4 \leqslant x \leqslant 0.6, \\ 0 & \text{otherwise}, \end{cases} \tag{27a}$$

$$G(x,\beta,z) = \mathrm{e}^{-\beta(x-z)^2}, \tag{27b}$$

$$F(x,\alpha,a) = \sqrt{\max(1-\alpha^2(x-a)^2, 0)}, \tag{27c}$$

where $a = 0.5$, $z = -0.7$, $\delta = 0.005$, $\alpha = 10$, and $\beta = \frac{\log 2}{36\delta^2}$. The initial profile consists of a combination of Gaussians, a square pulse, a sharp triangle, and a combination of half-ellipses. We solve the problem on a 200 element uniform mesh with $p = 1, 2, 3, 4$ using the moment limiter. The solutions are plotted in Fig. 2 with 11 points per element. We observe that the solution extrema are better resolved with higher p. The rectangular pulse is captured better as well. The action of the limiter on the solution is plotted beneath each plot. In this figure, a dot corresponds to the highest coefficient not limited (one dot per element). For example, a dot at "three" means that the third and lower coefficients were not limited on the element; only coefficients higher than 3 would be limited. We observe that the limiter acts on the solution adaptively and does minimal limiting in smooth regions. Almost all coefficients are limited in the regions of near constant state since higher coefficients are on the order of machine precision there.

Next, we plot the exact and numerical derivatives of the solution computed with $p = 3$ on the 200 element mesh in Fig. 3. The numerical derivatives were computed and plotted locally on each element by differentiating (3). At discontinuities, the "exact" derivatives in Fig. 3 were plotted using one-sided differentiation. There is a good agreement between the exact and computed derivatives for the Gaussian. In approximating the pulse

Table 1
L^1 errors ϵ_1 and convergence rates r for the sine wave

N	$p=1$		$p=2$		$p=3$		$p=4$	
	ϵ_1	r	ϵ_1	r	ϵ_1	r	ϵ_1	r
16	2.18e−02	–	3.15e−04	–	6.67e−06	–	1.43e−07	–
32	5.42e−03	2.01	3.91e−05	3.01	4.12e−07	4.02	4.46e−09	5.01
64	1.35e−03	2.01	4.89e−06	3.00	2.56e−08	4.01	1.39e−10	5.00
128	3.37e−04	2.00	6.11e−07	3.00	1.59e−09	4.00	4.35e−12	5.00
256	8.45e−05	2.00	7.63e−08	3.00	9.99e−11	4.00	1.36e−13	4.99

No limiter.

Table 2
L^1 errors ϵ_1 and convergence rates r for the sine wave

N	$p=1$		$p=2$		$p=3$		$p=4$	
	ϵ_1	r	ϵ_1	r	ϵ_1	r	ϵ_1	r
16	1.22e−01	–	1.95e−03	–	6.71e−05	–	1.92e−06	–
32	2.65e−02	2.21	1.81e−04	3.43	2.95e−06	4.51	4.38e−08	5.45
64	5.64e−03	2.23	1.78e−05	3.35	1.35e−07	4.45	9.89e−10	5.47
128	1.17e−03	2.27	1.74e−06	3.35	6.13e−09	4.46	2.20e−11	5.48
256	2.45e−04	2.25	1.79e−07	3.28	2.92e−10	4.39	5.02e−13	5.45

Moment limiter.

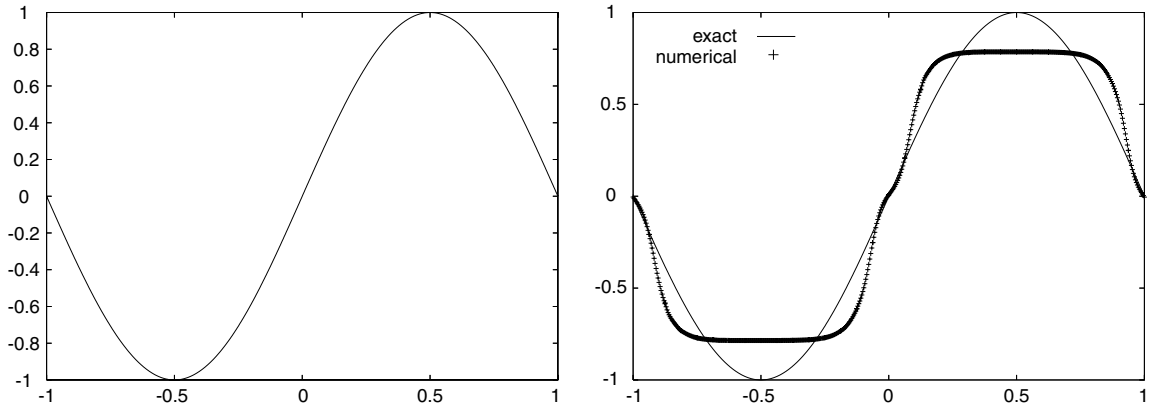


Fig. 1. Linear advection (25) and (26) with $p=4$ on a 32 element uniform mesh; $t=100$. Left: the exact and numerical solutions are indistinguishable with the moment limiter. Right: solid line – exact solution, plus signs – numerical solution.

(the second shape from the left), we see two delta-function-like shapes, then four spikes and then six. The limiter does not cut the accuracy to first order, rather the numerical derivatives are a good approximation for the delta function. We also note that the derivatives are not approximated monotonically. This is actually the case even for linear approximation, as can be seen in Fig. 4.

4.1.2. Burgers equation

In this example we study how the moment limiter affects the total variation of the solution and its coefficients when solving the Burgers equation

$$\begin{aligned} &u_t + (u^2/2)_x = 0, \quad -1 \leqslant x < 1, \ t > 0, \\ &u(x,0) = \frac{1+\sin \pi x}{2}, \\ &u(-1,t) = u(1,t). \end{aligned} \tag{28}$$

The solution of this problem forms a shock at $t = 2/\pi$ that moves to the right. The evolution of the shock in time is shown in Fig. 5, right. We solve the problem on a 32 element uniform mesh with $p=1$ and $p=2$ and plot the total variation in means (TVM) and the real total variation (TV) of the computed solution in Fig. 6. The real total variation is computed as a sum of the solution variation within each cell plus the solution jumps between cells. The total variation in means is given by (1). The exact total variation remains constant until $t=1$ and monotonically decreases after that (Fig. 5, right). The minmod limiter ($p=1$) results in a diminishing TVM as expected. This can be seen in Fig. 5, left, as a flattened local maximum. The moment limiter ($p=2$) retains a near constant TVM until the shock forms. However, the TV with both $p=1$ and $p=2$ starts noticeably growing as the solution steepens into a shock. The growth in the TV is roughly 15% and 30% for $p=1$ and $p=2$, respectively. The oscillatory structure of the TV plot is a result of the solution profile moving

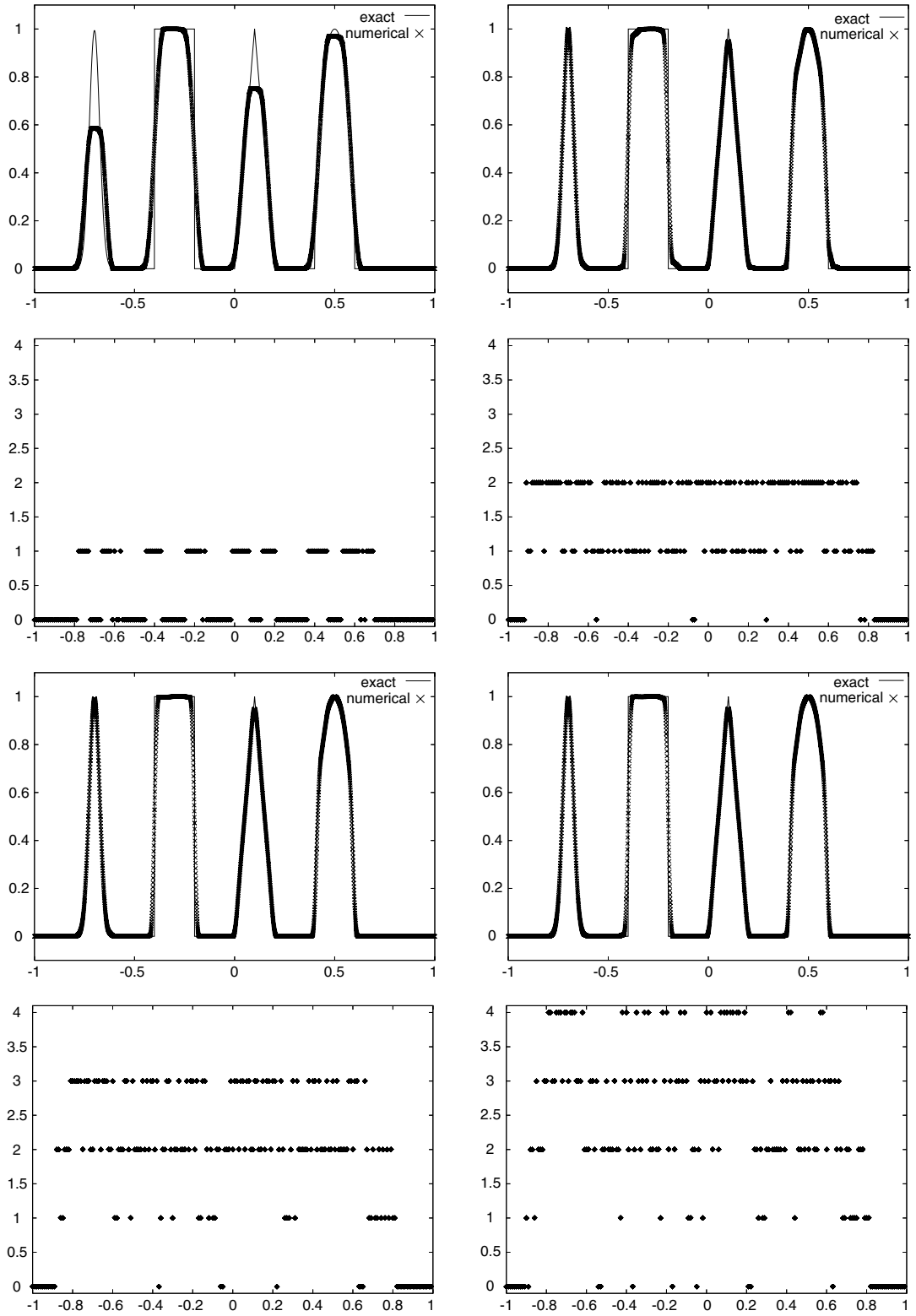


Fig. 2. Solutions of linear advection problem (25), (27) at $t = 8.0$, $N = 200$, $p = 1, 2, 3, 4$ from left to right and from top to bottom. Solid line – exact solution, crosses – numerical solution. The moment limiter's action is shown beneath each plot. A dot corresponds to the highest coefficient not limited.

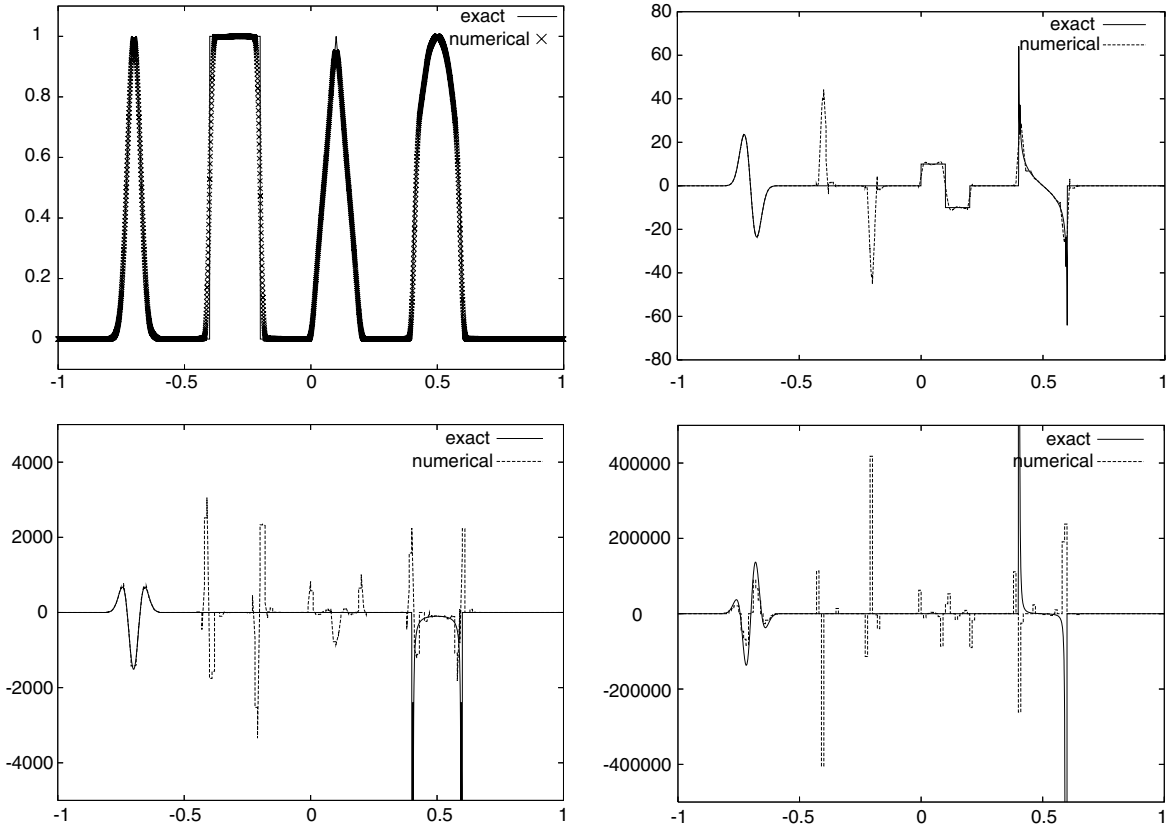


Fig. 3. Solution and first, second, and third derivatives of (25), (27) (from left to right and from top to bottom), $N = 200$, $p = 3$.

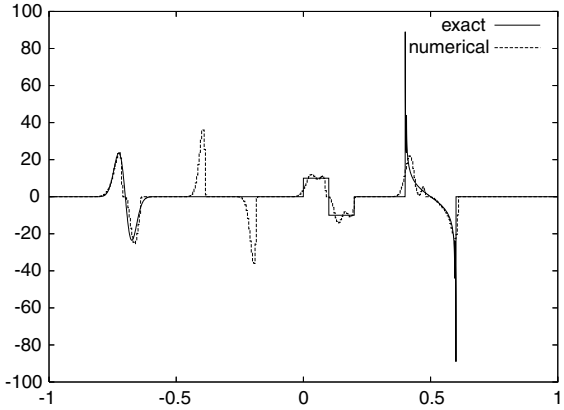


Fig. 4. First derivative of the solution of (25), (27), $N = 400$, $p = 1$.

through the grid, because TV depends on the positions of solution peaks relative to the mesh. The period of the oscillations is $a/\Delta x$, where a is the wave speed. Eleven peaks after $t = 2/\pi$ in Fig. 6 correspond to 11 grid cells through which the shock moved. A similar phenomenon is observed with the minmod limiter. The minmod enforces the TVD property by suppressing the upward movement in each period, which results in the staircasing pattern in Fig. 6, left.

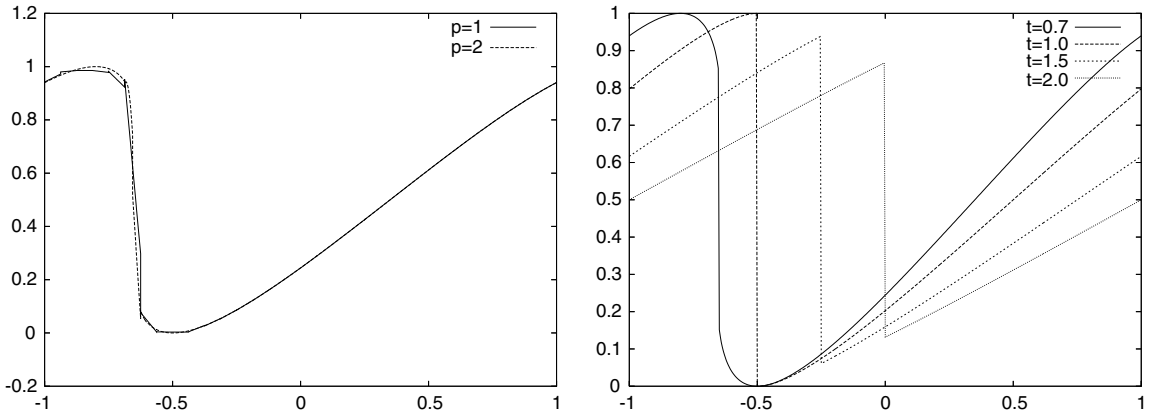


Fig. 5. left: Solutions of the Burgers equation with $p = 1$ (solid line) and $p = 2$ (dashed line) on a 32 element mesh with the moment limiter, $t = 0.7$. right: The exact solution of the Burgers equation at $t = 0.7, 1.0, 1.5, 2.0$.

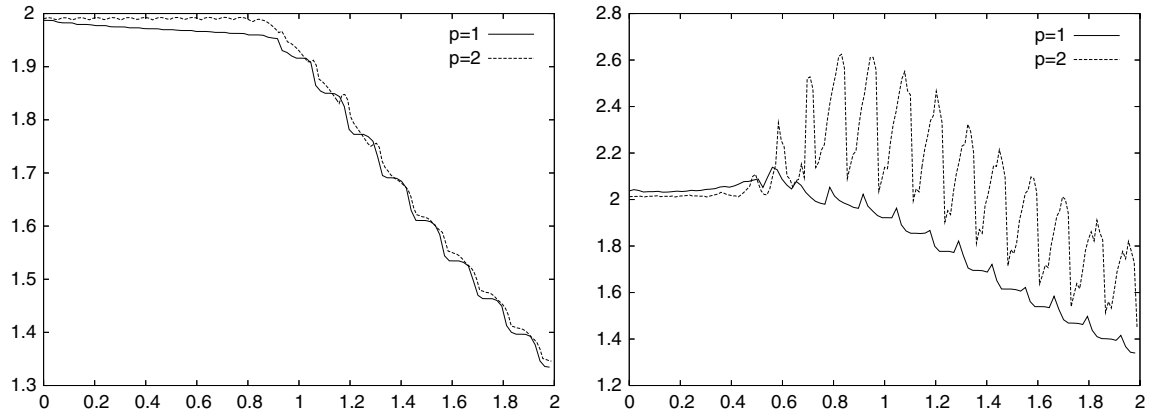


Fig. 6. Total variation (right) and total variation in means (left) as a function of time for the Burgers equation, $p = 1$ (solid line) and $p = 2$ (dashed line).

Next, we solve (28) on the same mesh with $p = 3$, compute the following quantities:

$$TV_{i-1} = \sum_j |c_i^j|, \quad i = 1, 2, 3, \tag{29}$$

at every timestep, and plot them in Fig. 7 using a log scale. Recall that the c_i are approximations to the solution derivatives times a scaling factor involving Δx^i. We notice that the scales remain clearly separated even after the shock has formed. TV_0 is an approximation to the total variation, which decays with time. By analogy, TV_1 and TV_2 can be viewed as approximations of the total variation of the first and second derivatives. They start to grow earlier and grow faster than TV_0, with the third coefficients growing faster than the second ones. This confirms our hypothesis that oscillations first appear in higher derivatives and then propagate into lower ones.

4.1.3. Blast waves

We consider the Euler equations $\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x = \mathbf{0}$ with

$$\mathbf{u} = (\rho, \rho q, E)^{\mathrm{T}}, \quad \mathbf{f}(\mathbf{u}) = q\mathbf{u} + (0, P, qP)^{\mathrm{T}}, \tag{30a}$$

and an equation of state

$$P = (\gamma - 1)\left(E - \frac{1}{2}\rho q^2\right). \tag{30b}$$

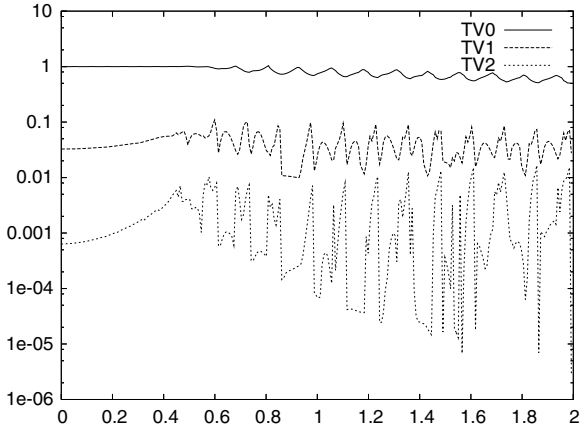


Fig. 7. TV_i, $i = 0, 1, 2$, defined by (29) for the Burgers equation as a function of time, $p = 3$, logscale.

Here, ρ is density, q velocity, P pressure, and E energy. We solve the problem subject to the initial conditions

$$(\rho, q, P)(x, 0) = \begin{cases} (1, 0, 1000), & 0 \leqslant x < 0.1, \\ (1, 0, 0.01), & 0.1 \leqslant x < 0.9, \\ (1, 0, 100), & 0.9 \leqslant x \leqslant 1, \end{cases} \tag{31}$$

on the interval $[0, 1]$. Reflecting boundary conditions are imposed at the end points. The problem models interaction of two blast waves and was studied extensively in [23]. We solve the problem on 200 and 400 element meshes with linear and quadratic approximation. The results at $t = 0.38$ are reported in Fig. 8. The numerical solutions are compared with a solution obtained with $p = 1$ on a 3000 element mesh that is referred to as "exact" in Fig. 8. Although the structure of the solution is quite complex, there are no smooth regions where the advantages of a high-order scheme could be made apparent. The purpose of this example, thus, is to show that a high-order DG with the moment limiter results in accurate, stable, overshoot-free solutions for this difficult problem involving multiple interactions of strong shocks and rarefactions. Even for this case we observe that the quadratic approximation is more accurate, and the shocks and contact regions are better resolved on both meshes.

4.1.4. Shock–entropy interaction

Consider the Euler equations (30) subject to the initial data [21]

$$(\rho, q, P)(x, 0) = \begin{cases} (3.857143, -0.920279, 10.33333), & x \leqslant 0, \\ (1 + 0.2\sin(5x), -3.549648, 1.00000), & 0 < x < 10, \\ (1.0000, -3.549648, 1.00000), & x \geqslant 10. \end{cases} \tag{32}$$

This example involves the interaction of a stationary shock at $x = 0$ with a leftward-moving flow having a sinusoidal density variation. As the density perturbation passes through the shock, it produces oscillations developing into shocks of smaller amplitude. A poor limiting strategy would damp the oscillations and, thus, the aim is to avoid this. In Fig. 9 we present solutions at $t = 2.0$ computed on a uniform mesh with elements of size 0.04 with $p = 1$ and $p = 2$. The high-frequencies behind the shock are better resolved by the quadratic approximation. This indicates that the DG might be competitive for problems involving interactions of shocks and fine structures.

4.2. Two-dimensional examples

4.2.1. Linear advection

We consider the linear advection problem on a square $\Omega = (-1, 1) \times (-1, 1)$

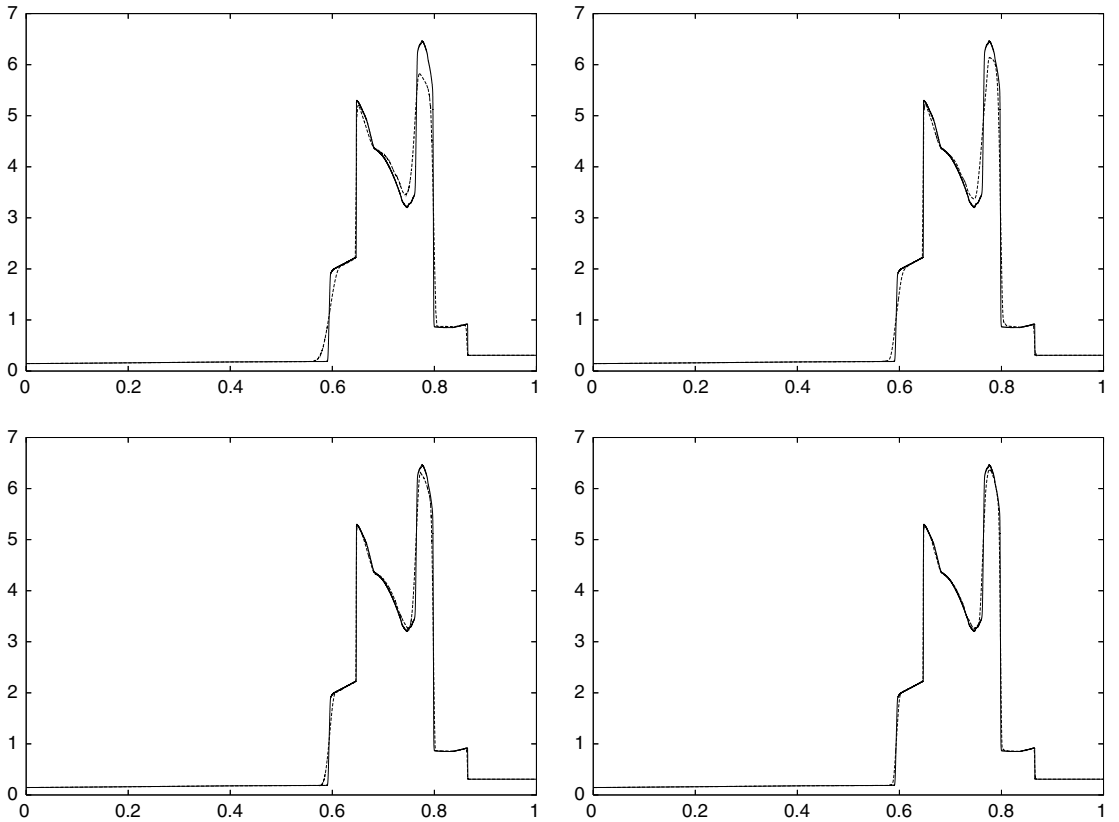
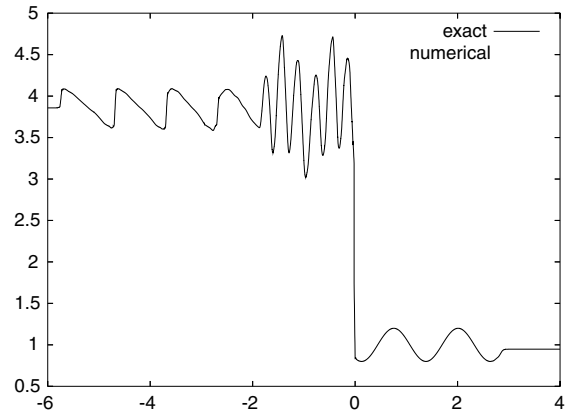


Fig. 8. Interaction of two blast waves (30), (31) at $t = 0.38$, computed with $p = 1$ (left) and $p = 2$ (right) on $N = 200$ and $N = 400$ (top and bottom). Solid line – exact solution, dashed line – numerical solution.



numerical

625

$$\begin{aligned} &u_t + \mathbf{a} \cdot \nabla u = 0, \quad t > 0, \\ &u(x, y, 0) = u_0(x, y). \end{aligned} \tag{33}$$

First, we solve (33) with sinusoidal initial data

$$u_0(x, y, 0) = \sin \pi(x - 2y), \tag{34}$$

Table 3
L^1 errors ϵ_1 and convergence rates r for the sinusoidal wave

N	Limiter		No Limiter	
	$p=1$			
	ϵ_1	r	ϵ_1	r
16×16	1.677712e−01	–	1.179655e−01	–
32×32	2.510205e−02	2.74	1.946301e−02	2.60
64×64	5.148220e−03	2.29	4.697913e−03	2.05
	$p=2$			
16×16	4.691721e−03	–	4.6849e−03	–
32×32	5.7393e−04	3.03	5.7393e−04	3.03
64×64	7.1462e−05	3.01	7.1462e−05	3.01

and $\mathbf{a}=(3,1)$. Periodic boundary conditions are imposed. The initial conditions and the velocity $\mathbf{a}$ are chosen so that the isolines of the solution and the direction of the flow do not coincide with the orientation of the mesh. We solve this problem on a sequence of Cartesian meshes: 16×16, 32×32, and 64×64 with $p=1,2,3,4$. The exact errors in the L^1 norm for limited and unlimited solutions are reported in Table 3. For $p>1$ the errors in limited and unlimited solutions are virtually the same for all but the coarsest mesh and to save space we do not report them here.

Next, we consider a profile consisting of a cone and a square pulse rotating clock-wise around the origin with velocity $\mathbf{a}=(2\pi y,-2\pi x)$. The initial conditions are given by

$$u_0(x,y,0)=\begin{cases}\cos^2(2\pi r), & r <= 0.25,\\ 1, & 0.1\leqslant x\leqslant 0.6\ \&\&\ -0.25\leqslant y\leqslant 0.25,\\ 0, & \text{otherwise},\end{cases} \tag{35}$$

where $r=(x+0.5)^2+y^2$. We set $u=0$ on the boundaries. The solutions obtained on an 80 by 80 element mesh are shown in Fig. 10 with 11 isolines. The solutions are reasonably symmetric with no visible overshoots. The cross-section of the solutions along the line $y=0$ is plotted in Fig. 11. The peak of the cone is still equal to one for the quadratic approximation but is only 0.82 for the linear approximation.

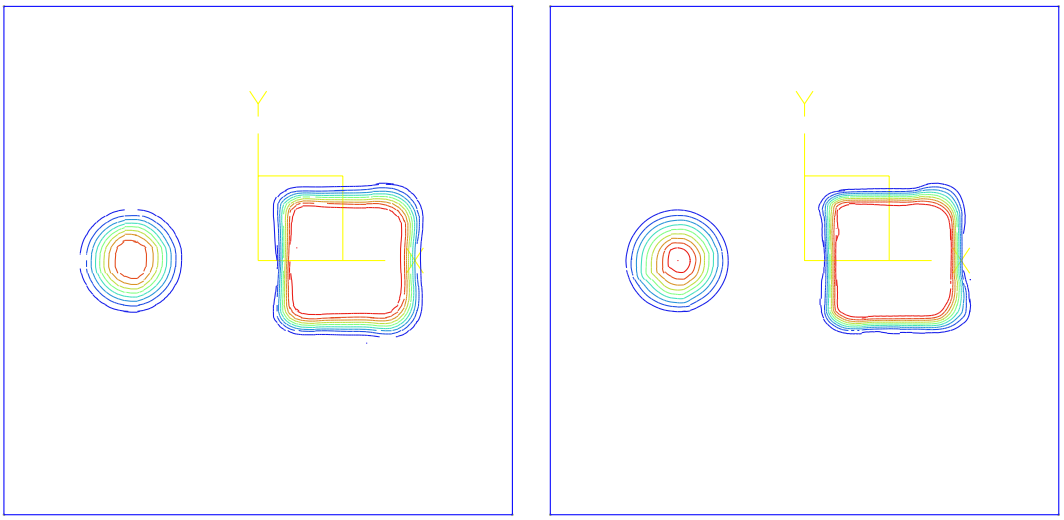


Fig. 10. Rotating cone and square pulse, $t=1$, $p=1$ (left) and $p=2$ (right).

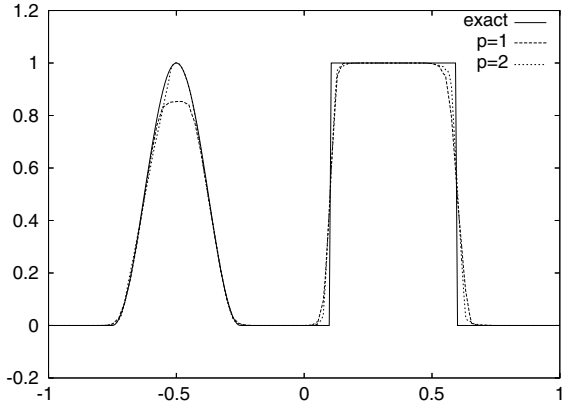


Fig. 11. Cross section along $y = 0$ of rotating cone and pulse at $t = 1$.

4.2.2. Double mach reflection

We consider the reflection of a Mach 10 planar shock by a wedge having a half-angle of 30°. The detailed set-up and analysis of the problem can be found in [23]. The problem is solved on a uniform mesh with spacings $\Delta x = \Delta y = 1/240$ on the $[0, 4] \times [0, 1]$ domain. Initially, the shock is located at $x = 1/6$ and forms a 60° angle with x axis. Reflecting boundary conditions are imposed on the lower boundary starting at $x = 1/6$; values corresponding to the exact motion of the Mach 10 shock are used on the rest of the boundaries. The density contours with $p = 1$ and $p = 2$ are plotted in Figs. 12 and 13. The fine structures of the solution are better resolved with quadratic approximation. In particular, the instability of the contact is better captured with

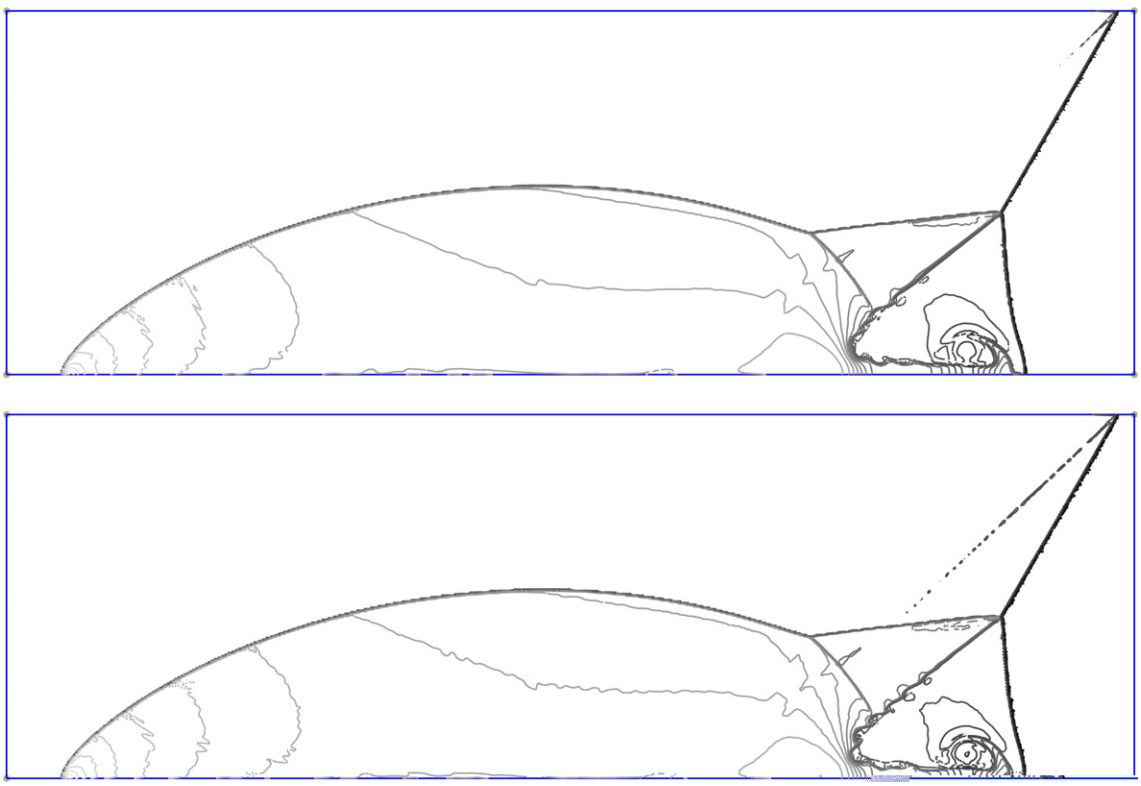


Fig. 12. Double Mach reflection on 240 by 960 mesh. Density, 30 isolines, $p = 1$ (top) and $p = 2$ (bottom).

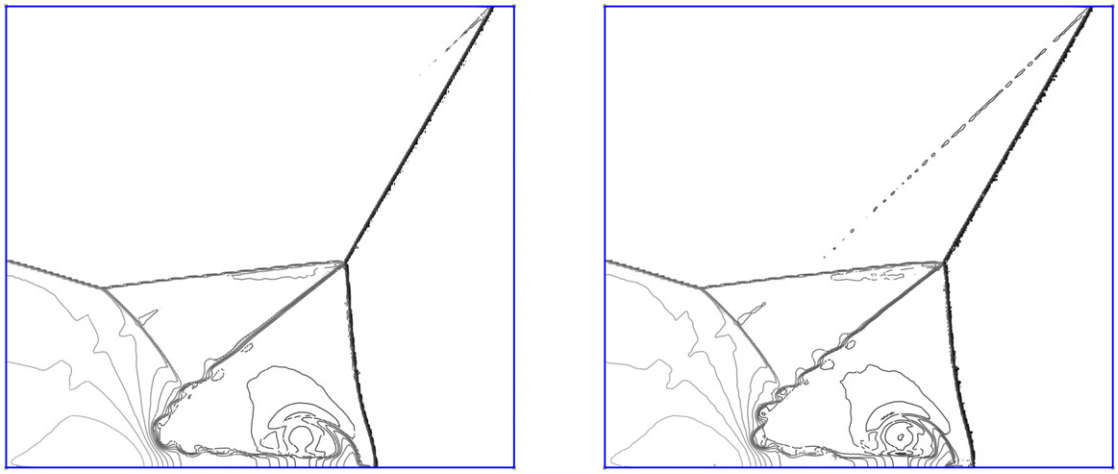


Fig. 13. Double Mach reflection on 240 by 960 mesh. Density (zoom), 30 isolines, $p = 1$ (left) and $p = 2$ (right).

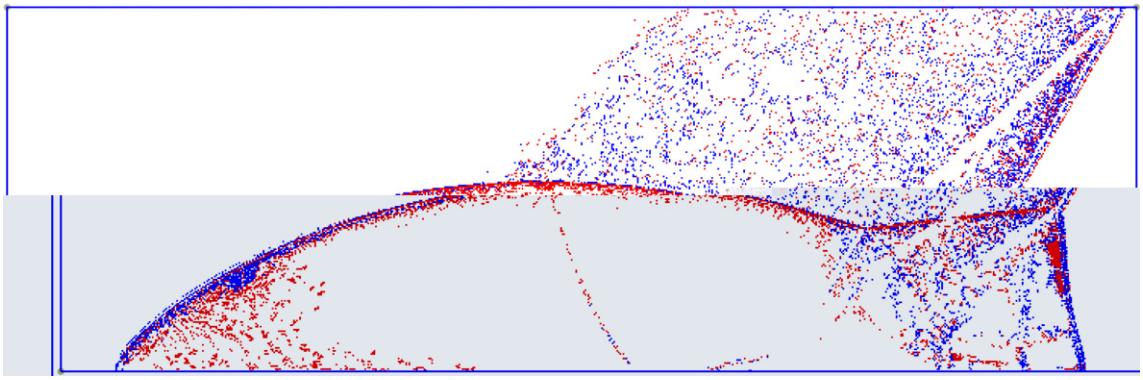


Fig. 14. Double Mach reflection on 240 by 960 mesh, $p = 2$, moment limiter. The elements where all coefficients were limited are marked.

$p = 2$. The action of the limiter after the final time step was performed is shown in Fig. 14. The elements where all coefficients were limited are marked. The full action of the limiter is mainly restricted to the shock regions. The limiter is also activated in the region behind the incident shock where the solution has near constant values.

5. Conclusions

We have described the moment limiter for discontinuous Galerkin methods that can limit a high-order approximation without reducing it to first order. The limiter was developed in one space dimension, and extended to two dimensions for tensor-product meshes. The most important next step is to try to extend the limiter to unstructured triangular meshes. It would also be interesting to test these ideas on discontinuous Galerkin methods with different basis functions, and derive the appropriate scalings for those cases.

Acknowledgments

The author is grateful to Marsha Berger and Jonathan Goodman for encouragement and fruitful discussions. The author was supported in part by the Department of Energy Grants DE-FG02-00ER25053 and DE-FC02-01ER25472.

Appendix A

For completeness, we include the formulas for limiting a quadratic approximation using the Legendre tensor-product basis functions. The second order basis is given by

$$
\begin{aligned}
&c_{0,0} = \frac{1}{2}, \quad c_{1,0} = \frac{\sqrt{3}\xi}{2}, \quad c_{0,1} = \frac{\sqrt{3}\eta}{2}, \quad c_{1,1} = \frac{3\xi\eta}{2},\\
&c_{2,0} = \frac{\sqrt{5}(3\xi^2-1)}{2}, \quad c_{0,2} = \frac{\sqrt{5}(3\eta^2-1)}{2}, \quad c_{2,1} = \frac{\sqrt{15}(3\xi^2-1)\eta}{2},\\
&c_{1,2} = \frac{\sqrt{15}(3\eta^2-1)\xi}{2}, \quad c_{2,2} = \frac{5(3\xi^2-1)(3\eta^2-1)}{2}.
\end{aligned}
\tag{36}
$$

We limit the solution on mesh cell (k,m) using the following formulas that correspond to formula (22) in the paper,

$$
\begin{aligned}
\tilde{c}^{k,m}_{2,2} &= \text{minmod}\left(c^{k,m}_{2,2}, \frac{c^{k,m+1}_{2,1}-c^{k,m}_{2,1}}{\sqrt{15}}, \frac{c^{k,m}_{2,1}-c^{k,m-1}_{2,1}}{\sqrt{15}}, \frac{c^{k+1,m}_{1,2}-c^{k,m}_{1,2}}{\sqrt{15}}, \frac{c^{k,m}_{1,2}-c^{k-1,m}_{1,2}}{\sqrt{15}}\right)\\
\tilde{c}^{k,m}_{1,2} &= \text{minmod}\left(c^{k,m}_{1,2}, \frac{c^{k,m+1}_{1,1}-c^{k,m}_{1,1}}{\sqrt{15}}, \frac{c^{k,m}_{1,1}-c^{k,m-1}_{1,1}}{\sqrt{15}}, \frac{c^{k+1,m}_{0,2}-c^{k,m}_{0,2}}{\sqrt{3}}, \frac{c^{k,m}_{0,2}-c^{k-1,m}_{0,2}}{\sqrt{3}}\right)\\
\tilde{c}^{k,m}_{2,1} &= \text{minmod}\left(c^{k,m}_{2,1}, \frac{c^{k,m+1}_{2,0}-c^{k,m}_{2,0}}{\sqrt{3}}, \frac{c^{k,m}_{2,0}-c^{k,m-1}_{2,0}}{\sqrt{3}}, \frac{c^{k+1,m}_{1,1}-c^{k,m}_{1,1}}{\sqrt{15}}, \frac{c^{k,m}_{1,1}-c^{k-1,m}_{1,1}}{\sqrt{15}}\right)\\
\tilde{c}^{k,m}_{0,2} &= \text{minmod}\left(c^{k,m}_{0,2}, \frac{c^{k,m+1}_{0,1}-c^{k,m}_{0,1}}{\sqrt{15}}, \frac{c^{k,m}_{0,1}-c^{k,m-1}_{0,1}}{\sqrt{15}}\right)\\
\tilde{c}^{k,m}_{2,0} &= \text{minmod}\left(c^{k,m}_{2,0}, \frac{c^{k+1,m}_{1,0}-c^{k,m}_{1,0}}{\sqrt{15}}, \frac{c^{k,m}_{1,0}-c^{k-1,m}_{1,0}}{\sqrt{15}}\right)\\
\tilde{c}^{k,m}_{1,1} &= \text{minmod}\left(c^{k,m}_{1,1}, \frac{c^{k,m+1}_{1,0}-c^{k,m}_{1,0}}{\sqrt{3}}, \frac{c^{k,m}_{1,0}-c^{k,m-1}_{1,0}}{\sqrt{3}}, \frac{c^{k+1,m}_{0,1}-c^{k,m}_{0,1}}{\sqrt{3}}, \frac{c^{k,m}_{0,1}-c^{k-1,m}_{0,1}}{\sqrt{3}}\right)\\
\tilde{c}^{k,m}_{0,1} &= \text{minmod}\left(c^{k,m}_{0,1}, \frac{c^{k,m+1}_{0,0}-c^{k,m}_{0,0}}{\sqrt{3}}, \frac{c^{k,m}_{0,0}-c^{k,m-1}_{0,0}}{\sqrt{3}}\right)\\
\tilde{c}^{k,m}_{1,0} &= \text{minmod}\left(c^{k,m}_{1,0}, \frac{c^{k+1,m}_{0,0}-c^{k,m}_{0,0}}{\sqrt{3}}, \frac{c^{k,m}_{0,0}-c^{k-1,m}_{0,0}}{\sqrt{3}}\right).
\end{aligned}
\tag{37}
$$

Limiting is performed in the following manner:

- limit $\tilde{c}^{k,m}_{2,2}$, if not changed, stop.
- limit $\tilde{c}^{k,m}_{1,2}$ and $\tilde{c}^{k,m}_{2,1}$, if both not changed, stop.
- limit $\tilde{c}^{k,m}_{0,2}$ and $\tilde{c}^{k,m}_{2,0}$, if both not changed, stop.
- limit $\tilde{c}^{k,m}_{1,1}$, if not changed, stop.
- limit $\tilde{c}^{k,m}_{0,1}$ and $\tilde{c}^{k,m}_{1,0}$.

References

[1] M. Abramowitz, I.A. Stegun (Eds.), Handbook of Mathematical Functions, Dover, New York, 1965.

[2] M. Aftosmis, D. Gaitonde, T. S. Tavares. On the accuracy, stability and monotonicity of various reconstruction algorithms for unstructured meshes, AIAA-94-0415, January, 1994.

[3] R. Biswas, K. Devine, J.E. Flaherty, Parallel adaptive finite element methods for conservation laws, Applied Numerical Mathematics 14 (1994) 255–284.

[4] B. Cockburn, S.Y. Lin, C.-W. Shu, TVB Runge–Kutta local projection discontinuous Galerkin methods for scalar conservation laws III: one dimensional systems, Journal of Computational Physics 84 (1989) 90–113.
[5] B. Cockburn, C.-W. Shu, TVB Runge–Kutta local projection discontinuous Galerkin methods for scalar conservation laws II: general framework, Mathematics of Computation 52 (1989) 411–435.
[6] P. Colella, P. Woodward, The piecewise parabolic method (PPM) for gas-dynamical simulations, Journal of Computational Physics 54 (1984) 174–201.
[7] J.E. Flaherty, L. Krivodonova, J.-F. Remacle, M.S. Shephard, Some aspects of dicontinuous Galerkin methods for hyperbolic conservation laws, Finite Elements in Analysis and Design 38 (2002) 889–908.
[8] F.X. Giraldo, J.S. Hesthaven, T. Warburton, Nodal high-order discontinuous Galerkin methods for the spherical shallow water equations, Journal of Computational Physics 181 (2002) 499–525.
[9] S.K. Godunov, A finite difference method for computing discontinuous solutions of fluid dynamics problems, Matematichekii Sbornik 47 (1959) 271–306 (in Russsian).
[10] A. Harten, High resolution schemes for hyperbolic conservation laws, Journal of Computational Physics 49 (1983) 357–393.
[11] A. Harten, B. Engquist, S. Osher, S. Chakravarthy, Uniformly high-order accurate essentially nonoscillatory schemes III, Journal of Computational Physics 71 (1987) 231–303.
[12] R. Hartmann, P. Houston, Adaptive discontinuous Galerkin methods for the compressible Euler equations, Journal of Computational Physics 183 (2002) 508–532.
[13] H. Hoteit, P. Ackerer, R. Mose, J. Erhel, B. Philippe, New two-dimensional slope limiters for discontinuous Galerkin methods on arbitrary meshes, Technical Report INRIA Report 4491, INRIA, 2002.
[14] J. Jaffre, C. Johnson, A. Szepessy, Convergence of the discontinuous Galerkin finite element method for hyperbolic conservation laws, Mathematical Models and Methods in Applied Sciences 5 (1995) 367–386.
[15] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, Journal of Computational Physics 126 (1996) 202–228.
[16] X.-D. Liu, S. Osher, Nonoscillatory high order accurate self-similar maximum principle satisfying shock capturing schemes I, SINUM 33 (1996) 760–779.
[17] P.-O. Persson, J. Peraire, Sub-cell shock capturing for discontinuous Galerkin methods, AIAA-2006-112, January, 2006.
[18] J. Qiu, C.-W. Shu, A comparison of troubled cell indicators for Runge–Kutta discontinuous Galerkin methods using WENO limiters, SIAM Journal on Scientific Computing 27 (2005) 995–1013.
[19] J. Qiu, C.-W. Shu, Runge–Kutta discontinuous Galerkin methods using WENO limiters, SIAM Journal on Scientific Computing 26 (2005) 907–929.
[20] R. Sanders, A third-order accurate variation nonexpansive difference scheme for single nonlinear conservation laws, Mathematics of Computation 51 (1988) 535–558.
[21] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes II, Journal of Computational Physics 83 (1989) 32–78.
[22] B. van Leer, Towards the ultimate conservation difference scheme V, Journal of Computational Physics 32 (1979) 1–136.
[23] P. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, Journal of Computational Physics 54 (1984) 115–173.