6th Workshop on Numerical Methods in Applied
Science and Engineering (NMASE 07)
Vall de Núria, 10 a 12 de enero de 2007

SHOCK CAPTURING FOR DISCONTINUOUS GALERKIN METHODS

Eva Casoni y Antonio Huerta

Laboratori de Càlcul Numèric (LaCàN)
Departament de Matemàtica Aplicada III
Universitat Politècnica de Catalunya

e-mail:(eva.casoni,antonio.huerta)@upc.edu, web: http://www-lacan.upc.edu

Key words: Discontinuous Galerkin methods; Artificial diffusion; Non-linear conservation laws

Abstract. *This work is devoted to solve scalar hyperbolic conservation laws in the presence of strong shocks with discontinuous Galerkin methods (DGM). A standard approach is to use limiting strategies in order to avoid oscillations in the vicinity of the shock. Basically, these techniques reconstruct the solution with a lower order polynomial in those elements where discontinuities lie. These limiting procedures degrade the accuracy of the method and introduce an excessive amount of dissipation to the solution, in particular for high-order approximations. The aim of the present work is to use artificial diffusion instead of limiters to capture the shocks. We show preliminary results with the inviscid's Burgers equation and also with a convection-diffusion problem.*

1 INTRODUCTION

In the last years discontinuous Galerkin methods have been used in computational fluid dynamics with low order techniques. In the context of non-linear convection-dominated problems high-order discontinuous Galerkin methods may develop oscillations near discontinuities or shocks which lead to unbounded computational solutions.

There are two standard techniques to avoid unstable solutions and capture shocks with high order approximations: the use of limiting procedures [2, 4] and the introduction of artificial viscosity into the scheme [7]. In fact, the first technique is also a way to introduce some numerical dissipation without adding explicitly a high-order term to the original equations. The main problem in all the non-linear artificial viscosity techniques, is that such procedures affect the order of accuracy locally and, as a resulting, there is a reduction of the optimal high-order convergence rate.

The limiting techniques enforce a non-linear stability property (TVDM stability) by effects of a non-linear local projection operator. This operator, generally called the slope

limiter, is based on piecewise linear approximations. Because of this, the slope limiter reduces the interpolation degree in those regions where discontinuities are detected and reconstructs the solution with a lower order approximation. A special class of explicit Runge-Kutta time integration schemes are used in combination with slope limiters procedures and DG. The resulting scheme is TVD, but it requires small time steps. Up to now no implicit techniques are used combined with these techniques.

More recently Persson and Peraire in [7] have proposed a new technique based on introducing artificial viscosity in order to eliminate the high-frequencies in the solution, thus eliminating Gibbs-type oscillations. They also define a shock detection algorithm which is based on the decay of the expansion coefficients rather than in the residual of the solution like the most classical sensors do. It seems to be a reliable good indicator. The main difficulty of this method is the empirical choice of a constant which determines the amount of viscosity needed.

In these notes we propose a method to introduce the exact amount of artificial diffusion into the numerical scheme to capture the shocks. The amount of viscosity introduced (inside the elements and across their boundaries) tries to reproduce the effect of limiters in those regions where the solution must be reconstructed. The goal is to capture shocks with discontinuous Galerkin methods using polynomials of order $p \geq 1$ and maintaining high-order accuracy in regions where the solution is smooth and in the neighborhood of shocks. The introduction into the equation of the artificial viscosity term allows also to use implicit time integration schemes. We have also tested the introduction of the artificial viscosity by firstly detect the troubled regions using the discontinuity sensor of Persson and Peraire and we have compared these results with those ones obtained without the sensor.

2 SHOCK CAPTURING SCHEME

Consider the scalar hyperbolic conservation law subject to an initial condition:

$$u_t + (f(u))_x = 0, \qquad x \in \Omega,\ t > 0 \tag{1a}$$

$$u(x,0) = u^0(x), \qquad x \in \Omega \tag{1b}$$

and periodic boundary conditions.

The solution of this non-linear equation may develop discontinuities and formation of shocks even when the initial and boundary data are smooth. In order to prevent non-physical oscillations which lead to instabilities it is required to introduce some smoothing technique to ensure stability. In this work we are going to introduce the less possible amount of artificial viscosity to avoid oscillations and ensure stability of the solution only in those elements where discontinuities are present. The scheme is performed combining the ideas of the slope-limiters and the shock-capturing methods.

2.1 LDG Discretization

The added viscosity is introduced in (1a) by incorporating into the equation a dissipative term of the form

$$u_t + (f(u))_x - \varepsilon u_{xx} = 0, \quad x \in \Omega,\ t > 0 \tag{2}$$

There are several schemes to discretize high-order derivatives. In the present work we have chosen the Local Discontinuous Galerkin [3]. We require then to rewrite (2) in the mixed form by introducing the auxiliary variables σ

$$u_t + (f(u))_x - \sigma_x = 0, \qquad x \in \Omega,\ t > 0 \tag{3a}$$

$$\sigma - \varepsilon u_x = 0, \qquad x \in \Omega \tag{3b}$$

Remark 1. Notice that we have assumed constant viscosity.

Let $\{I_j\}_{j=1..J}$ with $I_j = (x_j, x_{j+1})$ be a partition of the interval Ω into J subintervals. Multiplying (3a) and (3b) by the test functions v and τ respectively, integrating over an element I_j and integrating by parts the flux terms, the following weak formulation of the problem is obtained

$$\int_{I_j} u_t v\ dx - \int_{I_j} f(u) v_x\ dx + \int_{I_j} \sigma v_x\ dx + \Big(\hat{f}_{j+1} v(x_{j+1}^-) - \hat{f}_j v(x_j^+)\Big) - \Big(\hat{\sigma}_{j+1} v(x_{j+1}^-) - \hat{\sigma}_j v(x_j^+)\Big) = 0 \tag{4a}$$

$$\int_{I_j} \sigma\tau\ dx + \int_{I_j} \varepsilon u \tau_x - \varepsilon\Big(\hat{u}_{j+1}\tau(x_{j+1}^-) - \hat{u}_j \tau(x_j^+)\Big) = 0 \tag{4b}$$

where we have introduced the typical numerical flux function $\hat{f}$ of hyperbolic DG methods, which can be taken as the Roe or Lax-Friedrichs flux [1]. There are also two other fluxes due to the discretization of LDG methods, $\hat{\sigma}$ and $\hat{u}$, which can be seen as approximations of the numerical traces of σ and u on the boundaries [3]. At interfaces we use the notation $x_j^{\pm}$, where

$$x_j^{\pm} = \lim_{\epsilon \longrightarrow 0} x_j \pm \epsilon \mathbf{n}_j \tag{5}$$

with $\mathbf{n}_j$ the exterior unit normal.

Note that the auxiliary equation (4b) allows us to update the value of ε at each time step. As we will show, ε is a non-linear coefficient which depends on the value of the solution u at time t^n: $\varepsilon = \varepsilon(u^n)$. The fluxes $\hat{\sigma}$ and $\hat{u}$ transport the shock information not only inside the elements but also across the element boundaries. Of course, they control the interfaces jumps too.

The time integration is performed explicitly using a Runge-Kutta time scheme with a suitable small time step. We also point that there is no need to use the special RKDG-type schemes typical of the slope limiting procedures. In our numerical tests we have notice no difference.

2.2 Existing slope limiters methods

In this section we briefly review two slope limiters defined by Cockburn and Shu in [2] and the extension to high-order limiting procedure defined later by Biswas in [4].

2.2.1 The generalized slope limiter $\Lambda\Pi_h$ of Cockburn and Shu

The $\Lambda\Pi_h$-projection operator is constructed in such way that the following properties are satisfied:

- It preserves the conservation of mass element by element.
- It ensures that the gradient of the reconstructed solution is not bigger than the gradient of the not-reconstructed one.
- If the function u is smooth $\Lambda\Pi_h$ is the identity operator.

Consider an approximation of order p within each element. We express the solution in terms of the orthogonal Legendre polynomials (or) modal basis functions

$$u = \sum_{i=0}^{p} c_i P_i \tag{6}$$

where the degrees of freedom c_i are the so called modal coefficients. Let us point that for this set of basis functions the mean value of the solution in each element is reduced to c_0.

For a linear approximation ($p = 1$) the generalized slope limiter is defined as:

$$\Lambda\Pi_h(u) = c_0 P_0 + \text{minmod}(c_{1_{|I_j}}, c_{0_{|I_j}} - c_{0_{|I_{j-1}}}, c_{0_{|I_{j+1}}} - c_{0_{|I_j}}) \tag{7}$$

with the *minmod* function defined by

$$\text{minmod}\big(a_1, a_2, a_3\big) = \begin{cases} s\ \min_{1\le n\le 3}|a_n| & \text{if } s = sign(a_1) = sign(a_2) = sign(a_3) \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

Note that basically, the operator $\Lambda\Pi_h$ limits the variation of solution slopes, comparing each of the slopes with the neighboring elements and ensuring monotonicity of the means locally.

For high-order approximations the slope limiter reduces the approximate solution of arbitrary order p to a linear one:

$$\Lambda\Pi_h(u) = c_0 P_0 + \tilde{c_1} P_1 \tag{9}$$

setting $\tilde{c}_i = 0 \quad \text{for } i > 1$. We denote $\tilde{c}_i$ the reconstructed coefficients.

In order to ensure TVDM stability of the shceme this slope limiter procedure is combined with a special class of Runge-Kutta methods (for a detailed description of the method see [2]).

2.2.2 The extension to high-order limiting of Biswas and Flaherty

The method of Cockburn and Shu preserves monotonicity of solution averages, but do not modify the higher-order coefficients $c_2, c_3, ..., c_p$ to preserve monotonicity of the solution. The method defined by Biswas *et al.* in [4] tries to overcome this drawback. It uses limiting procedures to keep the ith moment monotone on neighboring elements, where the solution moments of a scalar problem are given by

$$\int_{-1}^{1} u P_i(\xi)\, d\xi = \frac{2}{2i+1} c_i \qquad i = 0, 1, ..., p \tag{10}$$

In fact, they use the slope limiter defined by Cockburn and Shu for linear approximations adaptively: the highest order coefficient is first limited, and the limitation procedure goes on until $\tilde{c}_i = c_i$ for some i (again, $\tilde{c}_i$ represents the reconstructed value after limitation).

The idea behind this method is to avoid destroying high-order accuracy where the solution is $\mathcal{C}^k$ with $k > 1$. By differentiating the numerical solution (6) it is possible to check that c_i is related (up to a constant) with the mean value of $\frac{\partial^i u}{\partial \xi^i}$ and c_{i+1} with its slope:

$$\frac{\partial^i u}{\partial \xi^i} = \prod_{m=1}^{i} (2m-1) c_i + \prod_{m=1}^{i+1} (2m-1) c_{i+1} \xi + \sum_{m=i+2}^{p} c_m \frac{\partial^i P_m}{\partial \xi^i} \tag{11}$$

Therefore, to keep the ith moment monote on neighboring elements we limit c_{i+1}, using the projection operator of Cockburn and Shu:

$$(2i+1)\tilde{c}_{i+1_{|I_j}} = \text{minmod}\Big((2i+1) c_{i+1_{|I_j}} \ , \ c_{i_{|I_j}} - c_{i_{|I_{j-1}}} \ , \ c_{i_{|I_{j+1}}} - c_{i_{|I_j}} \Big) \tag{12}$$

2.3 Artificial diffusion

Based on this high-order limiting algorithm we are going to define a constant viscosity coefficient for each element with the purpose of introducing the equivalent amount of

artificial viscosity to the limiting procedure. Therefore, we expect that we might be able to reproduce the dissipative effect of limiters instead of using them.

Consider the interval I_j. Our strategy consists on using the weak formulation to deduce the shock-capturing term. Multiplying (2) by a test function δv and integrating over the element I_j:

$$\int_{I_j} u_t \delta v \; dx - \int_{I_j} (f(u) - \varepsilon u_x)(\delta v)_x \; dx + \Big[F_{\mathbf{n}} \delta v \Big]_{x_j}^{x_{j+1}} = 0 \tag{13}$$

where we have introduce the Riemann solver $F_{\mathbf{n}} = (f(u) - \varepsilon u_x)\mathbf{n}$. We approximate it by $\hat{F}_{\mathbf{n}}(u)$, which is nothing but a numerical trace of the function $F_{\mathbf{n}}(u)$ on the boundaries. These operators are the so-called approximate Riemann solvers [6].

Remark 2. Taking the limit of $F_{\mathbf{n}}$ on the boundaries as ε goes to zero the approximate Riemann solver is reduced to $\hat{F}_{\mathbf{n}} = \hat{f}(u)\mathbf{n}$. (For a justification of this simplification see [1]).

Then, equation (13) can be rewritten in the following form,

$$\int_{I_j} u_t \delta v \; dx - \int_{I_j} f(u)(\delta v)_x \; dx + \Big[\hat{F}_{\mathbf{n}} \delta v \Big]_{x_j}^{x_{j+1}} + \int_{I_j} \varepsilon u_x (\delta v)_x \; dx = 0 \tag{14}$$

where we have obtained explicitly the shock-capturing term.

With the aim of introducing the effect of the slope limiters into the scheme, we split the method in two steps. A class of fractional-step methods operate at the level of the differential operators. The convection-diffusion operator splits into a sum of two components: one containing the convection operator and the other with the diffusion operator. Consider then equation (2) and split it in the following two steps:

$$\text{First step} \qquad \begin{cases} v_t + f_x(v) = 0, & \text{in } I_j \times [t^n, t^{n+1}[\\ v(t^n) = u^n, & \end{cases}$$

$$\text{Second step} \qquad \begin{cases} w_t - \varepsilon w_{xx} = 0, & \text{in } I_j \times [t^n, t^{n+1}[\\ w(t^n) = v^{n+1}, & \end{cases}$$

$$\text{Final update} \qquad u^{n+1} = w^{n+1}$$

Note that the first step is nothing but solving the hyperbolic conservation law (1a). The link between the slope limiters and the shock-capturing scheme can be easily established imposing that the effect of the limited function must be the same that the effect of adding some artificial diffusion to the solution of the hyperbolic problem (second step). That is:

$$\int_{I_j} \Lambda \Pi_h(v^{n+1})(\delta v) \; dx = \int_{I_j} v^{n+1}(\delta v) \; dx + \Delta t \Big[\int_{I_j} (\delta v)_x \varepsilon v_x^{n+1} \; dx \Big]^{n+1} \tag{15}$$

where, again, we have made the assumption that the viscosity goes to zero on the boundaries.

From this equation and assuming constant viscosity for each element, we can compute the reconstructed solution solving a convection-diffusion equation of the form (2) with the following artificial viscosity:

$$\varepsilon = \frac{\int_{I_j} \Big(\Lambda\Pi_h(v^{n+1}) - v^{n+1} \Big)(\delta v)\, dx}{\Delta t \Big[\int_{I_j} (\delta v)_x v_x^{n+1}\, dx \Big]^{n+1}} \tag{16}$$

2.3.1 Linear approximation

Let us start with the case of the linear approximation to describe the artificial diffusion method. Consider a linear approximation ($p = 1$ in the expansion (6))

$$u = c_0 P_0 + c_1 P_1 \tag{17}$$

The reconstructed solution of Biswas et al. would be

$$\tilde{u} = c_0 P_0 + \tilde{c}_1 P_1 \tag{18}$$

Inserting (17) and (18) with $\Lambda\Pi_h(v^{n+1}) = \tilde{u}$ and $v^{n+1} = u$ into (16) we obtain the amount of artificial diffusion to insert in (2) at each time step.

2.3.2 High-order approximation

For a high-order approximation of degree $p > 1$, the reconstructed solution is obtained after several projections: first, the highest order coefficient c_p is limited. The reconstructed solution in the adaptive process of Biswas et al. would be:

$$\tilde{u}_{(p)} = \sum_{i=0}^{p-1} c_i P_i + \tilde{c}_p P_p \tag{19}$$

As in the linear case, using expressions (15) or (16) with $\Lambda\Pi_h(v^{n+1}) = \tilde{u}_{(p)}$ and $v^{n+1} = u$ we impose the equivalence of the second step with the limited solution (19) and obtain the viscosity associated, denoted by ε_p.

Remark 3. Notice that $\varepsilon \neq 0$ in (16) only for $\delta v = P_p$. This is because we are modifying the degree of freedom c_p and therefore only viscosity associated with this coefficient is introduced. In fact, from (11) it can be easily checked that we are introducing viscosity for a linear approximation of $\frac{\partial^{p-1} u}{\partial \xi^{p-1}}$. Notice also that both viscosities are not the same.

If $\tilde{c}_p \neq c_p$ the next higher order coefficient, c_{p-1}, is limited. Now the not-limited solution v^{n+1} in the second step of the fractional method would be $\tilde{u}_{(p)}$ and the reconstructed one would be

$$\tilde{u}_{(p-1)} = \sum_{i=0}^{p-2} c_i P_i + \tilde{c}_{p-1} P_{p-1} + \tilde{c}_p P_p \tag{20}$$

Again, imposing (15) or (16) with $\Lambda\Pi_h(v^{n+1}) = \tilde{u}_{(p-1)}$, $v^{n+1} = \tilde{u}_{(p)}$ and for $\delta v = P_{p-1}$ we obtain the viscosity associated ε_{p-1}.

By following this procedure we can compute ε_i for $i = 1, ..., p$ associated to every projection of the solution. Roughly speaking, we state that for each coefficient limited there is a viscosity associated.

The main drawback of this adaptive process is that we can't compute a single equation (2) for every time step. The second step of the fractional-step method has to be solved for every reconstructed coefficient of the adaptive process. In order to improve this we have made the following simplification: having all the viscosities computed (which is not a hard task due to the orthogonality and hierarchy of the polynomials, see remark 4), we choose the maximum of all the viscosities. That is:

$$\varepsilon_e = \max\{\varepsilon_1, ..., \varepsilon_p\} \tag{21}$$

and introduce it in (2).

In fact, this choice corresponds to choose the last of the viscosities computed, which is the associated to the lowest-order degree of freedom:

$$\varepsilon_e = \max_{\{i|\tilde{c}_i \neq c_i\}} \varepsilon_i = \varepsilon_{\min\{i|\tilde{c}_i \neq c_i\}} \tag{22}$$

In our algorithm we use the LDG scheme described in section 2.1 instead of the fractional step method. The viscosity (16) is updated at each time step with the auxiliary equation (4b) and it is introduced in the conservation law with σ.

Remark 4. Using the orthogonality and hierarchy of Legendre polynomials and taking into account that there is no marching in time in (4b) we can compute ε from equation (16):

$$\varepsilon_i = \frac{h^2(\tilde{c}_i - c_i)}{2(2i+1)} \Big(\int_{-1}^{1} \frac{\partial u_{(i)}}{\partial \xi} \frac{\partial P_i}{\partial \xi} \, d\xi \Big)^{-1} \tag{23}$$

where h is the mesh size and $\tilde{c}_i$ is computed as in (12). Of course, if $\tilde{c}_i = c_i$ the viscosity associated is zero.

With our choice we are introducing less viscosity, but as we will show in the numerical tests, it is enough to maintain the solution stable and avoid oscillations. We are based upon the idea that the differentiation procedure tends to smooth the numerical solution. Therefore we hope that the slope of the ith derivative will be sharper than the slope of the following derivatives. In other words, we will need a bigger amount of viscosity for maintain monotone the first moment than for the higher-ones.

2.4 Shock detection

In [7] a smoothness indicator is defined with the aim of introducing only the diffusion where is needed. The sensor detects those elements in which discontinuites appear. It is defined for each element as

$$S_e = \frac{\int_{I_j} |u - \hat{u}|^2}{\int_{I_j} |u|^2} \tag{24}$$

where $\hat{u}$ represents the approximation of order $p-1$ and u an approximation of order p.

Assuming that the polynomial expansion has similar behavior to the Fourier expansion it is expected that $S_e \sim \frac{1}{p^4}$. The theoretical justification of these assumption is based on the following theorem:

Theorem *Given a function $f(x)$ and its periodic Fourier approximation*

$$SF(f) = \sum_{k=-\infty}^{k=\infty} g_k e^{ikx}$$

If $f(x) \in \mathcal{C}^m$ then $|g_k| \sim k^{-(m+1)}$

That means, if the solution is continuous, then it is expected that the coefficients c_i decay at least like $\frac{1}{p^2}$.

We can also avoid employing this sensor by using the *modified minmod* function. The modification consists on introducing some upper bound of the absolute value of the second order derivative of the solution at local extrema:

$$\text{minmod}\big(a_1, a_2, a_3\big) = \begin{cases} a_1 & \text{if } |a_1| < Mh^2 \\ \text{minmod}(a_1, a_2, a_3) & \text{otherwise} \end{cases} \tag{25}$$

where h is the size of the element and M is a nonnegative real number taken as

$$M = C \sup_{a \le y \le b} \{ |u^0_{xx}(y)| \ , \ u^0_x(y) = 0 \} \tag{26}$$

with $C = \frac{2}{3}$ (for the justification of this choice see [2]). The introduction of this constant provides a criterion to determine regions where the approximate solution must be limited and prevents the accuracy of the scheme near discontinuities. The choice $M = 0$ corresponds to the original *minmod* function, and so by increasing the value of M we are relaxing the shock detection criterion.

Notice that by using the *modified minmod* function instead of the discontinuity sensor we aren't increasing the computational cost of the algorithm because the procedure to compute the added viscosity itself allows us to determine the troubled regions.

3 Numerical Results

In this section we provide numerical results for a non-linear conservation law and also for a linear convection-diffusion problem.

3.1 Example 1

Our first example is the inviscid's Burgers equation with smooth initial condition:

$$u_t + (\frac{u^2}{2})_x = 0, \quad x \in [0,1],\ t > 0 \tag{27a}$$

$$u(x,0) = \frac{1}{2} + sin(2\pi x), \quad x \in \Omega \tag{27b}$$

and periodic boundary conditions. Results are shown at time $T = 0.25$ and $T = 0.5$ where the shock is fully formed. We have made all the computations using a uniform mesh of 20 elements and interpolation degrees $p = 5, 10$. For the numerical integration a second order Padé scheme has been used.

If figures (1) and (2) we show the solution at time $T = 0.25$ with degrees of interpolation 5 and 10. We show the comparison between the slope limiters of Cockburn and Shu, Biswas et al. and our artificial viscosity method. Note that slope limiting procedures spreads out the shock over all the element and does not make special control of interelement jumps. For the case of an approximation $p = 10$ we haven't been able to compute the limiter of Biswas. In fact, we have observed that drawback for $p > 7$.

Next, in figure (3) we test the same problem solved with the artificial viscosity method but previously using the sensor (24). In blue we represent the solution with the discontinuity sensor and in red without it. We also show the amount of viscosity introduced in each element in both cases. In figure (4) the same test cases are shown at time $T = 0.5$. In all our computations we have taken $M = 40$.

The difference between both solutions is almost imperceptible. Despite the amount of artificial viscosity introduced without the discontinuity sensor is a bit bigger (specially when high-order interpolation polynomials are used) we believe that because of the small values of ε we can avoid using it and hence improve the computational cost of the algorithm. Therefore we conclude that the use of the discontinuity sensor is unnecessary because the minmod function acts as a sensor itself. Avoiding the discontinuity sensor makes the process more efficient but it implies the introduction of a problem-dependent constant M.

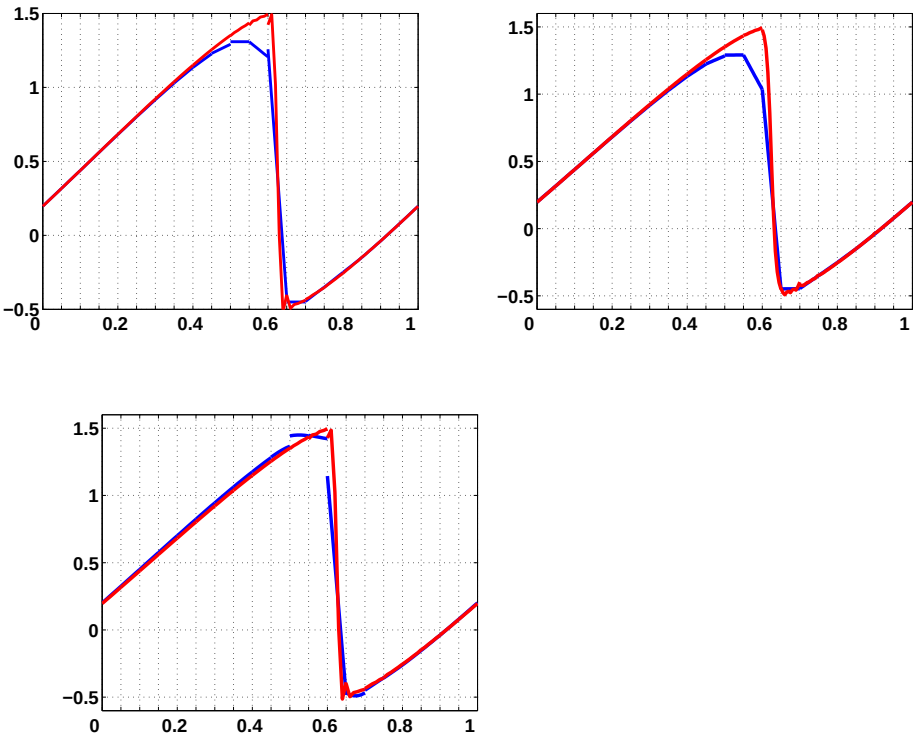

Figure 1: Approximation of degree $p = 5$ (left figures) and degree $p = 10$ (right figures) at time $T = 0.25$. On the top solution with Cockburn limiters (blue) versus artificial diffusion method (red). On the bottom the limiting moment of Biswas (blue) versus artificial diffusion method (red).

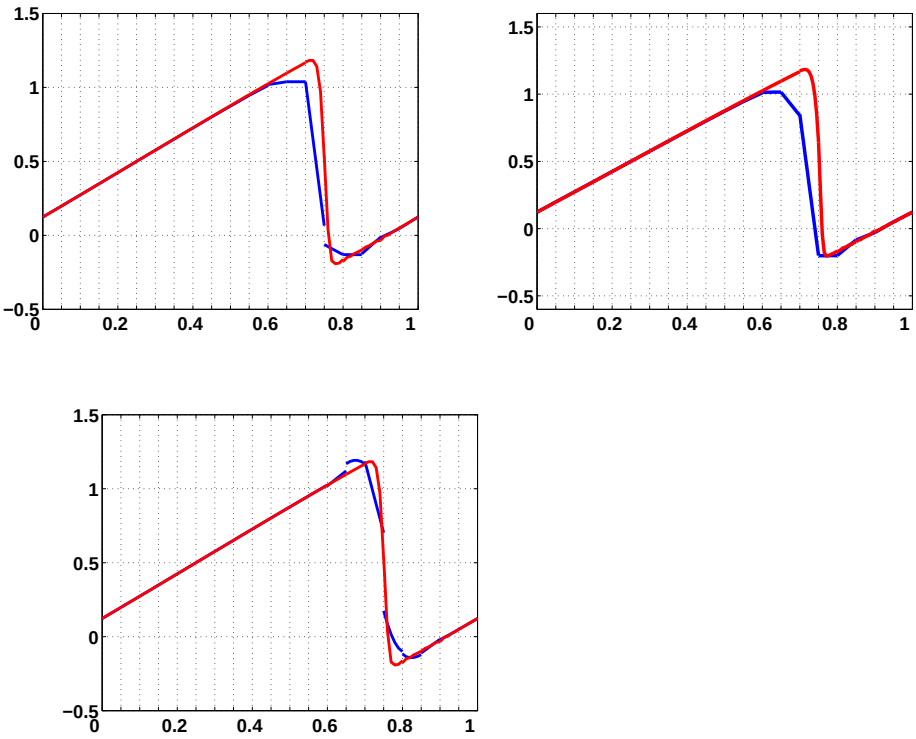

Figure 2: Approximation of degree $p = 5$ (left figures) and degree $p = 10$ (right figures) at time $T = 0.5$. On the top solution with Cockburn limiters (blue) versus artificial diffusion method (red). On the bottom the limiting moment of Biswas (blue) versus artificial diffusion method (red).

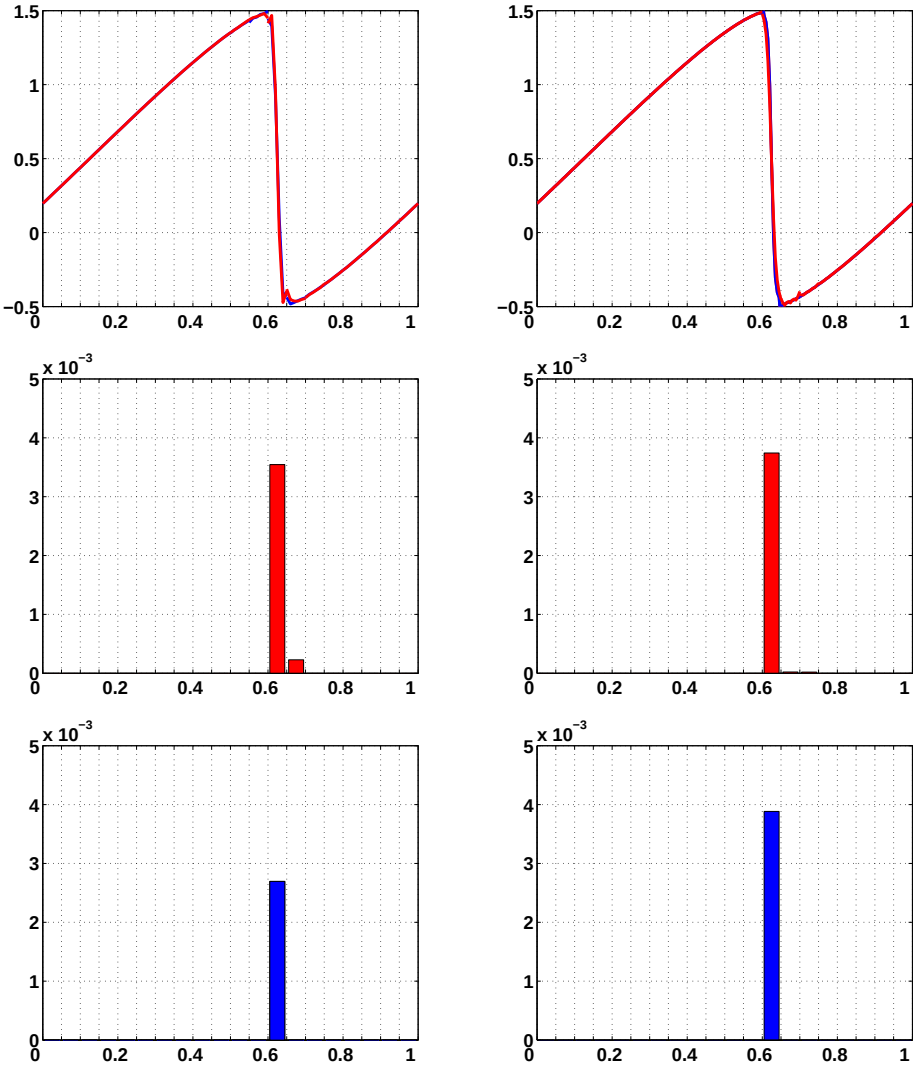

Figure 3: Approximation of degree $p = 5$ (left figures) and degree $p = 10$ (right figures) at time $T = 0.25$ with discontinuity sensor (blue) and without it (red). We also show the amount of artificial diffusion introduced for each one.

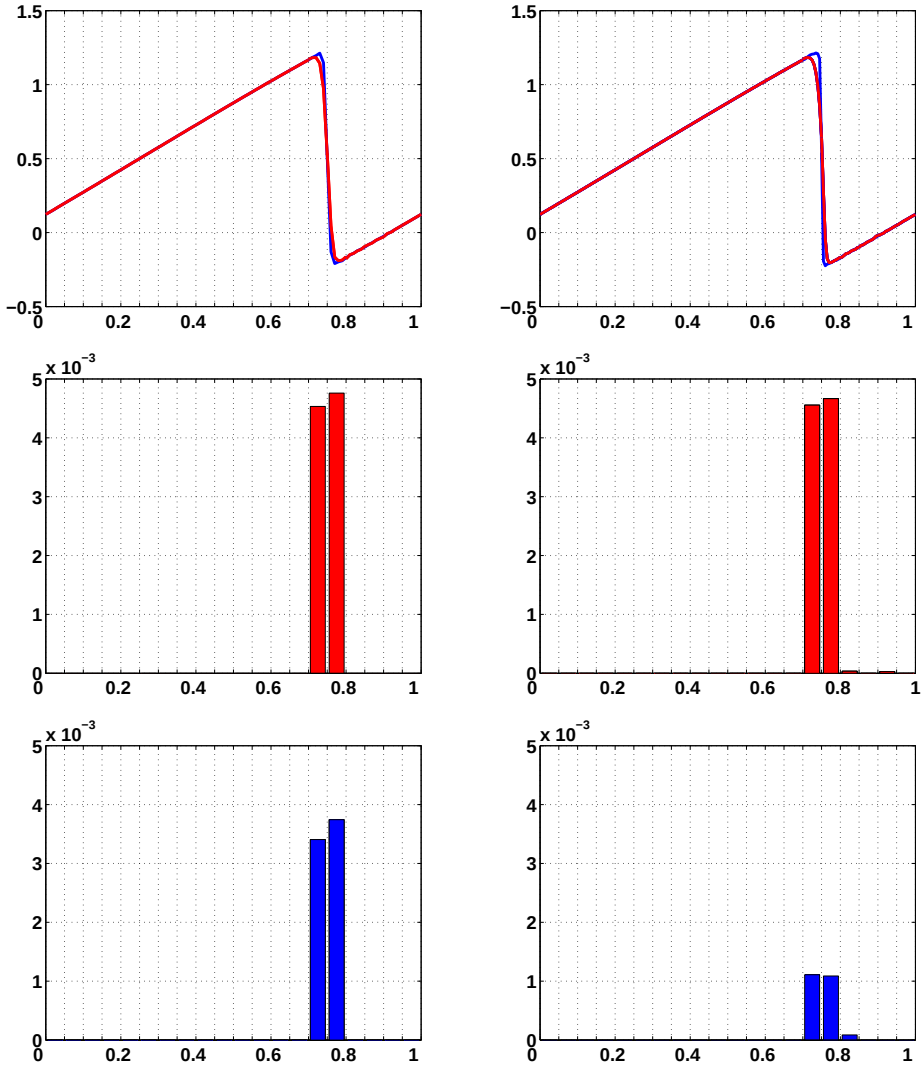

Figure 4: Approximation of degree $p = 5$ (left figures) and $p = 10$ (right figures) at time $T = 0.5$ with discontinuity sensor (blue) and without it (red). Under each approximation we show the amount of artificial diffusion introduced.

3.2 Example 2

We have also test the method with a linear convection-diffusion problem:

$$u_t + u_x - \nu u_{xx} = 1, \qquad x \in [0,1],\ t > 0 \tag{28a}$$
$$u(0,t) = u(1,t) = 0, \qquad t > 0 \tag{28b}$$
$$u(x,0) = 0, \qquad x \in \Omega \tag{28c}$$

with exact solution at steady state:

$$u(x) = x - \frac{1 - e^{\frac{x}{\nu}}}{1 - e^{\frac{1}{\nu}}} \tag{29}$$

Despite the problem is linear, a shock is well formed at $x = 1$ when high Peclet numbers are used. In order to check that our algorithm do not introduce extra artificial diffusion when the solution is smooth, we have tested it with a low Peclet number ($Pe = 0.5$). The results are shown in figure (5), and no artificial diffusion has been needed. Note that the exact solution at stationary state is well described by the approximate one.

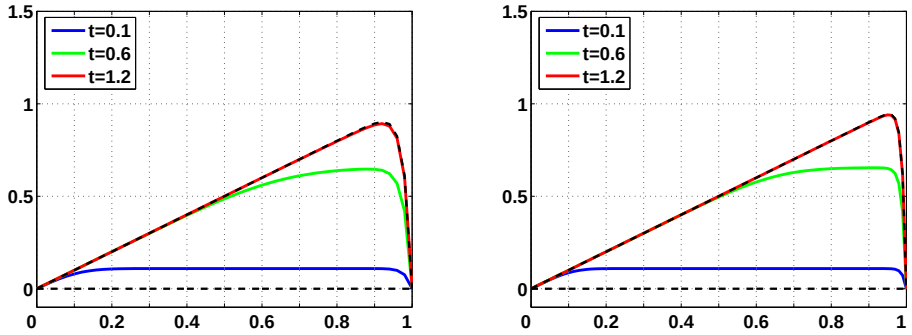

Figure 5: Approximation with $p = 5, 10$ for Peclet$= 0.5$. No artificial diffusion is needed.

Increasing the Peclet number the problem is convection-dominated and a front is developed when arriving at stationary state. We show results at times $t = 0.1,\ 0.6,\ 1.1$ and compare them with the stationary solution (figure 6). Again we show the amount of artificial diffusion introduced in each element and point out the order of the limiting procedure, that is, the subscript i corresponding to the viscosity added (22). Subscript equal to 0 means that no artificial diffusion has been added. Note that some amount of artificial diffusion is added to the last element.

In all our computations we have used $M = 5$ for the constant of the modified minmod function (25).

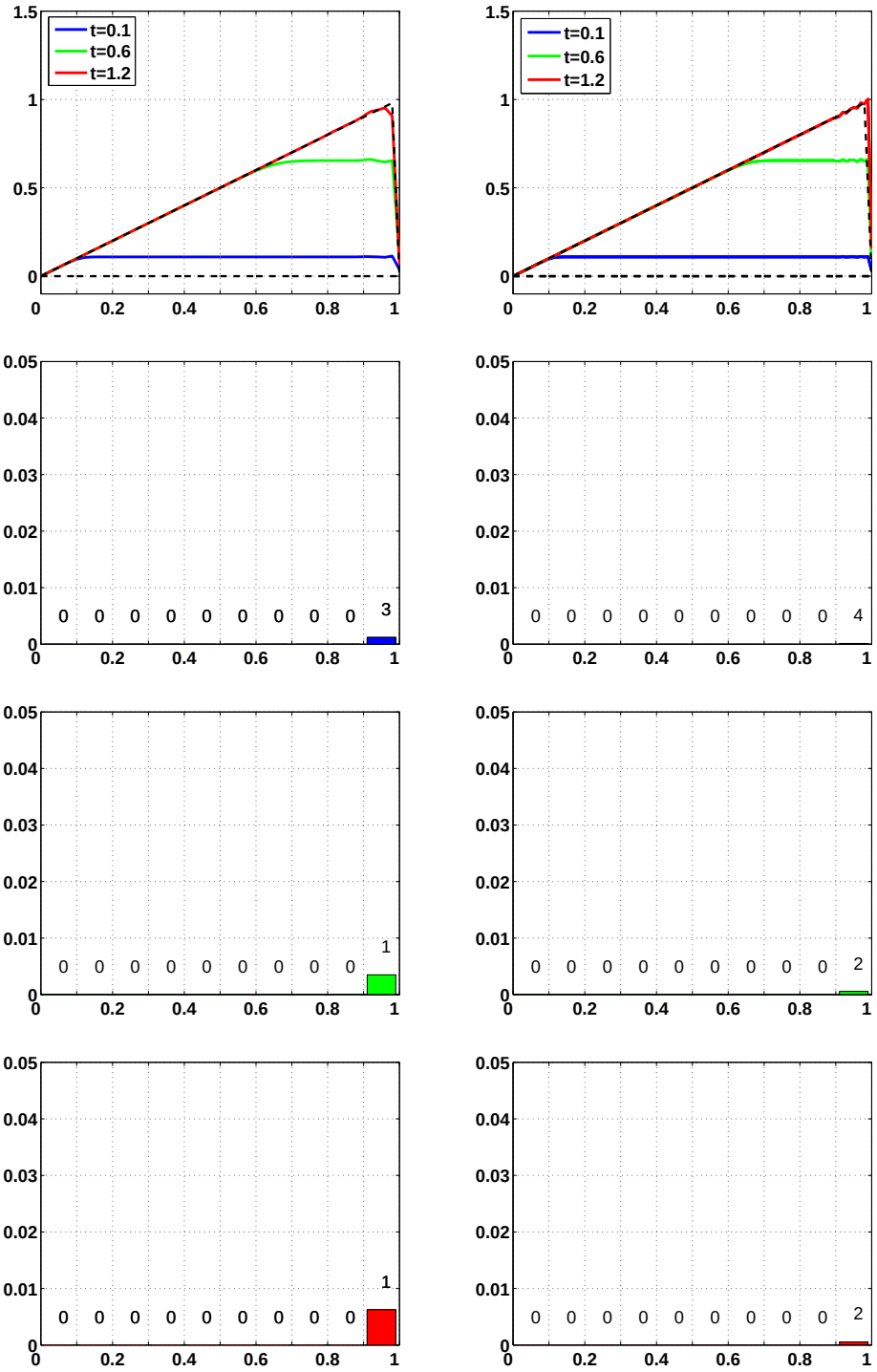

Figure 6: Peclet= 5. Right figures show the solution of an approximation with $p = 5$ and the viscosity added at each time with $M = 5$. Left figures show the same tests with $p = 10$.

One of the advantages of DG methods is that they are well-suited for p-adaptivity. The goal is to achieve good results using only a few elements by increasing the degree of interpolation. With that purpose in the following test we have set a fixed number of degrees of freedom and we have compared results using coarsest meshes by increasing the order of the approximation.

We have test example 2 for Peclet 5 setting 10 degrees of freedom. That choice corresponds to take 10 linear elements for $p = 1$, 2 elements for $p = 5$ and only one element for $p = 1$.

In figures 8 and 9 we can see the influence of M in the modified minmod function (26). It seems to be that M has influence not only with the amount of artificial diffusion to include but also with the order of accuracy in the element. Notice that for $M = 0$ we are reducing the order too much. Further investigations are on the track to improve the choice of this constant.

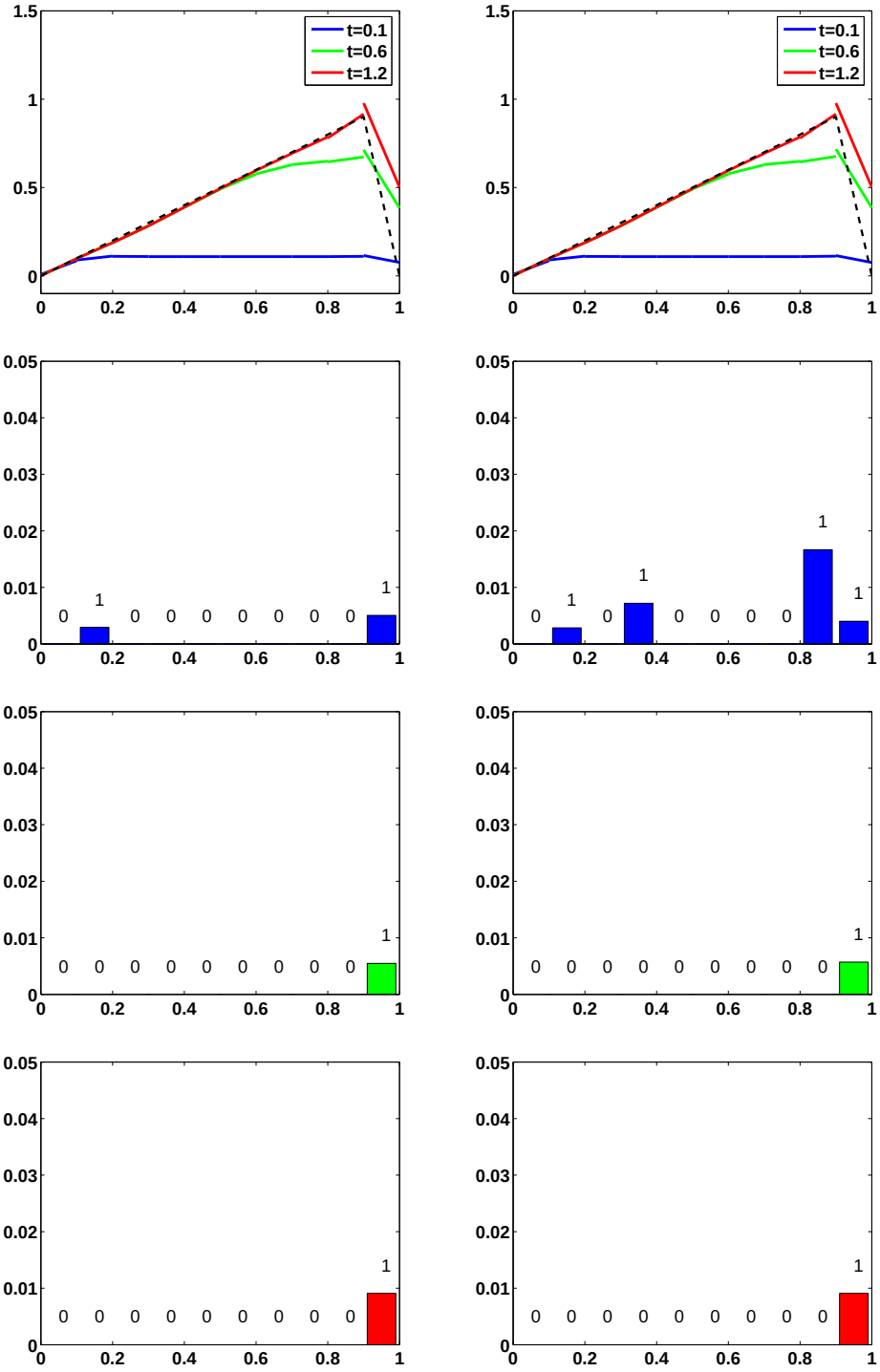

Figure 7: Approximation of degree $p = 1$ at times $t = 0.1,\ 0.6,\ 1.1$ for Peclet equal to 5. On left figures $M = 1$ and on right figures $M = 0$.

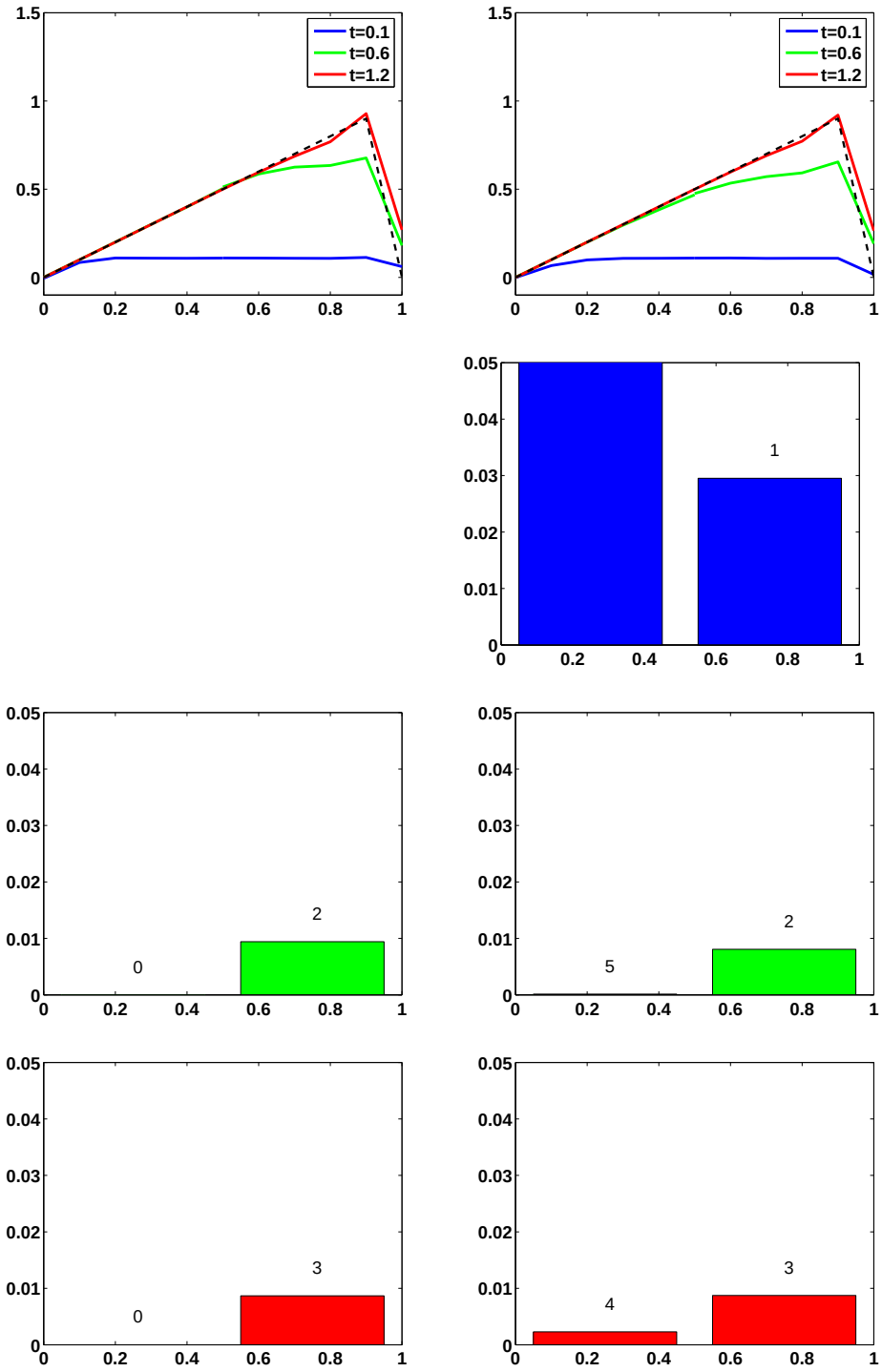

Figure 8: Approximation of degree $p = 5$ at times $t = 0.1,\ 0.6,\ 1.1$ for Peclet equal to 5. On left figures $M = 1$ and on right figures $M = 0$.

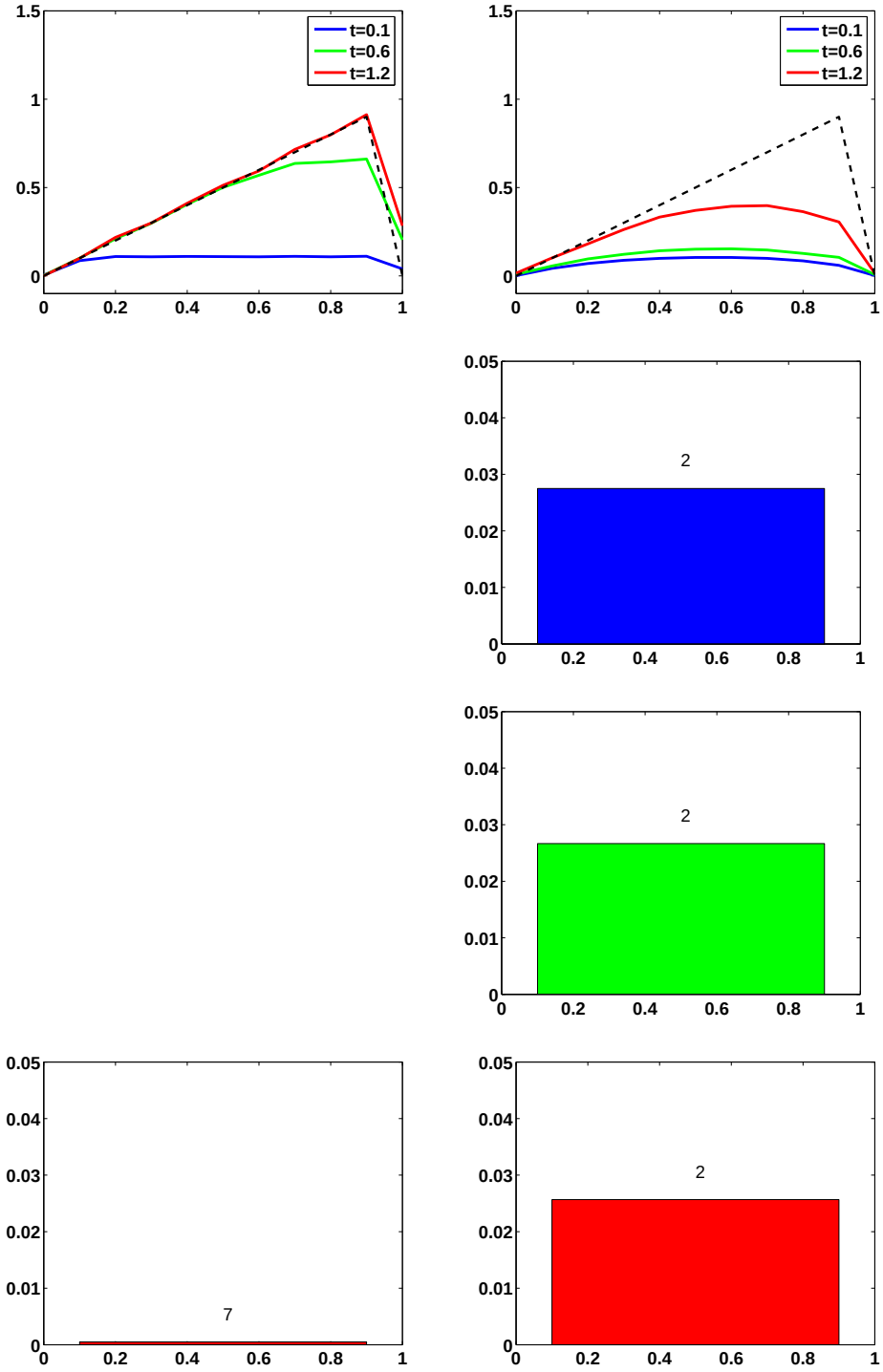

Figure 9: Approximation of degree $p = 10$ at times $t = 0.1,\ 0.6,\ 1.1$ for Peclet equal to 5. On left figures $M = 1$ and on right figures $M = 0$.

4 CONCLUSIONS

We have presented a technique to solve scalar conservation laws in the presence of shocks and discontinuities with high-order DG methods. It is based on defining a viscosity coefficient for each element of the discretization. Our approach takes into account all the expansion terms of the approximate solution and it is able to detect regions where the solution is non-smooth. The method obtained is capable to capture shocks maintaining high resolution when they are contained in one element but also when they cross between elements without the use of a discontinuity sensor. It can also be used in combination with an explicit or an implicit time integration scheme. Future work is devoted to introduce a non-constant viscosity per element and to extend this procedure in multidimensional problems.

REFERENCES

[1] B.Cockburn, Devising discontinuous Galerkin methods for non-linear hyperbolic conservation-laws, *J. Comp. Appl. Math.*, **128**, 187-204, (2001)

[2] B.Cockburn and C.-W. Shu, TVB Runge-Kutta local projection discontinuous galerkin methods for scalar conservation laws II: General Framework, *Mathematics of Computation*, **52**, 411-435 (1989)

[3] B.Cockburn and C.-W. Shu, The local discontinuous Galerkin method for time-dependent convection-diffusion systems, *SIAM J. Numer. Anal.* **35, No. 6**, 2440-2463, (1998)

[4] R.Biswas, K.Devine and J.Flaherty, Parallel adaptive finite element methods for conservation laws, *Applied Numerical Mathematics*, **14**, 255-284, (1994)

[5] J.Donea, A.Huerta, Finite Element Methods for Flow Problems, Wiley (2003)

[6] R.J. LeVeque, Numerical Methods for Conservation Laws, Birkhäuser (1992)

[7] P.-O. Persson and J.Peraire, Sub-Cell Shock Capturing for Discontinuous Galerkin methods, *Manuscript* (2005)

[8] C.-W. Shu and S.Osher, Efficient Implementation of Essentially Non-oscillatory Shock-Capturing Schemes, *J. Comp. Phys.*, 439-471 **77**, (1988)

Con el apoyo de *Universitat Politècnica de Catalunya* y *E.T.S. d'Enginyers de Camins, Canals i Ports de Barcelona*