Adaptive application of characteristic projection for Central schemes

Gabriella Puppo[1]

Dip. di Matematica, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy *puppo@calvino.polito.it*

Abstract In this work, the application of central schemes to systems of conservation laws is studied. A comparison between the componentwise implementation of these schemes and the algorithm exploiting characteristic projection is carried out, studying the behaviour of the weights appearing in the reconstruction. Next, a more efficient adaptive strategy that employs characteristic projections only close to discontinuities is proposed.

1 Introduction

Central schemes based on staggered grids have proved to be effective and efficient tools for the integration of systems of conservation laws. Their main advantage is the simplicity of the resulting codes and their high speed. As a consequence, they have reached a wide range of applications, see for instance the review in [5].

The price paid for these advantages is weaker monotonicity preserving properties, with respect to the best upwind schemes, at least in some test cases for high order central schemes: compare for instance the tests in [1] and in [2]. An effective cure was recently proposed in [4]. It consists in applying central schemes projecting the equations along characteristic directions. This device however considerably slows down the code, and it also mars the simplicity of central schemes: while before it was only necessary to know the flux function of the system of equations, now it is necessary to know and compute the eigenstructure of the system.

In this work, we first investigate the behaviour of the reconstruction appearing in [4], and next we propose an adaptive strategy which permits to recover the speed of traditional central schemes, while enjoying the lack of oscillations obtained with the scheme in [4].

We apply these ideas to the recently proposed Central Runge Kutta scheme of order 4, [3]. The same techniques can be extended to other central schemes.

2 Description of the scheme

We consider the system of equations:

$$u_t + f_x(u) = 0, \tag{1}$$

with $u \in R^m$, $f : R^m \to R^m$ continuously differentiable. We suppose that the Jacobian of f, $A(u) = f'(u)$ has real eigenvalues and a complete set of eigenvectors.

Let Δt and h be the mesh widths in time and space, with $\lambda = \Delta t/h$. Let $I_j = [x_j - \frac{h}{2}, x_j + \frac{h}{2}]$ be the interval of width h centered around the grid point x_j. Let $\bar{u}_j^n$ be the cell average of the function $u(x,t)$ on the interval I_j at time $t^n = t_0 + n\Delta t$.

We briefly describe the 4-th order Central Runge Kutta scheme on which we will perform all numerical tests. For more details, see [3].

Given the cell averages $\{\bar{u}_j^n\}$ of the solution at time t^n, we reconstruct the function $u(x,t^n)$ with a suitable fourth order non oscillatory reconstruction operator R:

$$u^n(x) = R(x; \{\bar{u}^n\}) = \sum_j P_j^d(x)\chi_{I_j}(x), \tag{2}$$

where $P_j^d(x)$ are polynomials of degree $d = 2$ (for CRK4). The reconstruction is piecewise smooth, with jumps at the end points of the interval I_j.

Following the philosophy of Central Schemes, [5], we integrate the conservation law (1) on the *staggered* interval $I_{j+1/2} = [x_j, x_{j+1}]$. We obtain the exact equation:

$$\left.\frac{d\bar{u}}{dt}\right|_{j+1/2} = -\frac{1}{h}\left[f(u(x_{j+1},t)) - f(u(x_j,t))\right]. \tag{3}$$

As in all CRK schemes [3], this equation is first discretized in time with a Runge-Kutta scheme. The updated solution is given by:

$$\bar{u}_{j+1/2}^{n+1} = \bar{u}_{j+1/2}^n - \lambda \sum_{i=1}^{\nu} b_i K_{j+1/2}^{(i)}, \tag{4}$$

where ν is the number of steps of the RK scheme, b and a (appearing below) are the vector and the matrix defining the RK scheme, and:

$$K_{j+1/2}^{(i)} = f(u_{j+1}^{(i-1)}) - f(u_j^{(i-1)}) \qquad \text{with} \quad u_j^{(0)} = u^n(x_j). \tag{5}$$

The term $\bar{u}_{j+1/2}^n$ in (4) is computed directly using the reconstruction (2). To evaluate the intermediate states, $u_j^{(i)}$, we exploit the fact that the reconstruction $u^n(x)$ is smooth except for jumps at the cell walls, $x_{j\pm1/2}$. Thus, $u(x_j,t)$ remains smooth for $t \in [t^n, t^n + \Delta t]$, if Δt is small enough. As in all central schemes based on staggered grids, we can therefore evaluate the intermediate states $u_j^{(i)}$ integrating the conservation law (1) in its differential form, namely: $u_t = -f_x(u)$. Thus:

$$u_j^{(i)} = u_j^{(0)} + \Delta t \sum_{l=1}^{i} a_{i,l}\hat{K}_j^{(l)}, \qquad \hat{K}_j^{(l)} = -\left.\frac{\partial f(u^{l-1})}{\partial x}\right|_j \tag{6}$$

for $i = 1, \cdots, \nu - 1$. Here, the standard fourth order Runge-Kutta scheme is used; with our notation: $b = (1/6, 1/3, 1/3, 1/6)$, while a is a diagonal matrix, with $a(1,1) = a(2,2) = 1/2$ and $a(3,3) = 1$.

We now consider space discretization, which plays a particularly important role in the extension to systems of equations.

At the beginning of each time step, we need a reconstruction algorithm to compute $\bar{u}^n_{j+1/2}$ and $u^{(0)}_j$. Next, we need a reconstruction scheme which computes $f_x(u^{(i)})$, starting from the point values $f(u^{(i)})$, for $i = 0, \cdots, \nu - 1$. We use the CWENO reconstruction [2], outlined below.

The reconstruction from cell averages at the beginning of each time step is a non-linear convex combination of three parabolas. On the interval I_j:

$$u^n(x)|_{I_j} = R_j(x) = \omega_j^{-1} P_{j-1}(x) + \omega_j^0 P_j(x) + \omega_j^{+1} P_{j+1}(x), \tag{7}$$

where the ω_j^k are the non linear weights, which satisfy $\sum_k \omega_j^k = 1$. The coefficients of the parabola P_{j+k} are computed interpolating the data $\bar{u}^n_{j+k+l}$, $l = -1, 0, 1$ in the sense of cell averages.

The weights ω_j^k are determined in order to maximize accuracy in smooth regions and to prevent the onset of spurious oscillations. As in [1], we define the weights with the following formulae:

$$\omega_j^k = \frac{\alpha_j^k}{\sum_{l=-1}^{1} \alpha_j^l} \qquad \alpha_j^k = \frac{C^k}{\left(\epsilon + \mathrm{IS}_j^k\right)^2}. \tag{8}$$

In the present case, $C^{-1} = C^1 = 3/16$, while $C^0 = 5/8$. The parameter ϵ prevents a vanishing denominator. For the CRK4 scheme, we always have $\epsilon = 10^{-6}$.

Finally, $\mathrm{IS}_j^k, k = -1, 0, 1$ are called *smoothness indicators*. They measure the regularity of the polynomial P_{j+k} on the interval I_j. Following [1], we take:

$$\mathrm{IS}_j^k = \sum_{l=1}^{2} \int_{x_{j-1/2}}^{x_{j+1/2}} h^{2l-1} \left(\frac{d^l P_{k+j}}{dx^l} \right)^2 dx, \qquad k = -1, 0, 1. \tag{9}$$

Thus $\omega_j^k = C^k + O(h^2)$ on smooth regions, while $\omega_j^k = O(h^4)$ if the data in the stencil of the polynomial P_{j+k} contain a jump discontinuity.

Finally, for flux derivatives, the polynomials P_{j+k} are determined interpolating the data $f(u(x_{j+k+l}))$ in the sense of point values. The reconstruction in this case must yield accurate estimates of the derivatives of f. The structure of the non linear weights does not change, but we must use a different set of constants, namely: $C^{-1} = C^1 = 1/6$, while $C^0 = 1/3$, see [2].

3 Systems of equations

We consider two extensions of central schemes to systems of equations. We test the resulting algorithms on two classical problems from 1D gas dynamics,

namely: Lax' Riemann Problem and Woodward and Colella blast wave problem, which describes the interaction of two Riemann problems, characterized by strong shocks. These test problems have been widely used in the literature. The initial and boundary conditions and the details of the implementation for central WENO schemes can be found in [2].

We now describe the two algorithms we are compairing. The simplest approach to the solution of systems of equations is to apply the scheme component by component, following the scalar recipe. It was found in [2] that the results improve if all components of the solution share the same smoothness indicator. At the beginning of each time step a Global Smoothness indicator is computed, namely:

$$IS_k^j = \frac{1}{m}\sum_{r=1}^{m}\frac{1}{||\bar{u}_r||_2}\left(\sum_{l=1}^{2}\int_{I_j} h^{2l-1}\left(\frac{d^l P_{j+k,r}}{dx^l}\right)^2 dx\right), \tag{10}$$

for $k = -1, 0, 1$. Here r denotes the r-th component of the solution and of the vector-valued interpolating polynomial. Compairing with (9), we see that the Global Smoothness Indicator is just a weighted average of the Smoothness Indicators given by each component.

These quantities are then used to compute the non linear weights throughout the time step. This will be called *Componentwise Implementation (CW)* of Central schemes.

In the *CW* approach, very little information on the physics of the system of equations is required. It is enough to provide the flux function f and an estimate of the characteristic speeds of propagation, to satisfy the stability condition, which, for the CRK4 scheme, is $\lambda \leq 12/25 * 1/\max|\mu(u)|$, where μ are the eigenvalues of the Jacobian of f, see [3].

The solution computed in this fashion is fast and simple to program. However small wiggles may appear. These spurious oscillations decrease in amplitude as the grid is refined. See for instance the left part of Fig. 1.

A second approach was suggested in [4]. It consists in applying to the central framework a technique that is widely employed in schemes based on flux splitting. Specifically, the computation of the weights in the WENO reconstruction is performed component by component but using characteristic directions, instead of conserved variables.

Let $Q(u)$ be the non-singular matrix whose columns are the right eigenvectors of $A(u)$. At each grid point j, compute $Q(\bar{u}_j^n)$. Let $\{\bar{u}_{j+l}^n\}$ be the data in the stencil of the jth cell. For CRK4, the stencil of the reconstruction consists of 5 points; thus $l = -2, \cdots, 2$. Map each cell average belonging to the stencil of the jth cell along the characteristic directions, i.e. compute the vectors:

$$\bar{v}_{j+l} = Q^{-1}(\bar{u}_j^n)\bar{u}_{j+l}^n \qquad l = -2, \cdots, 2.$$

Construct the conservative interpolant Π_v^j for the data $\{\bar{v}_{j+l}\}$, using the CWENO recipe described above *componentwise*. Then define the interpolant for the conserved variables u through the equation:

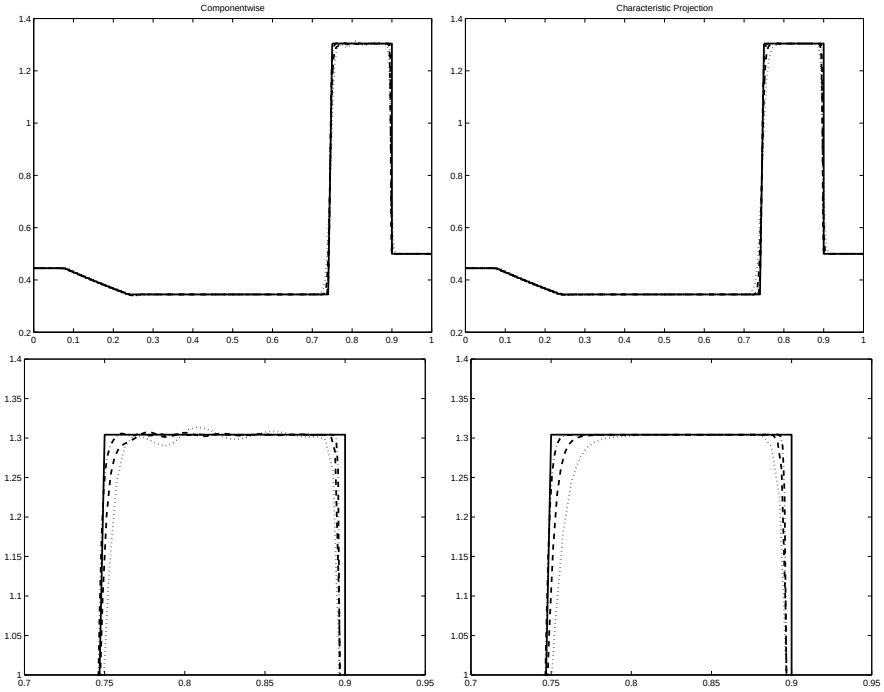

Fig. 1. Lax' Riemann problem. Componentwise (left) and Characteristic Projection (right) implementation of CRK4 scheme, for various grid sizes. **Bottom:** Detail of the density peak. $N = 200$ (dots), $N = 400$ (dash), $N = 800$ (dash-dot).

$$\Pi_u^j(x - x_j) = Q(\bar{u}_j^n)\Pi_v^j(x - x_j).$$

Finally, use this function to evaluate the two staggered half-cell averages:

$$\bar{u}_{j,-}^n = \frac{1}{h}\int_{x_j - \frac{h}{2}}^{x_j} \Pi_u^j(x - x_j)\, dx \qquad \bar{u}_{j,+}^n = \frac{1}{h}\int_{x_j}^{x_j + \frac{h}{2}} \Pi_u^j(x - x_j)\, dx,$$

with $\bar{u}_{j+1/2}^n = \bar{u}_{j,+}^n + \bar{u}_{j+1,-}^n$ and the point value $u(x_j, t^n) = \Pi_u^j(0)$. Notice that because the mapping between conservative and characteristic variables is linear, then a conservative reconstruction in v will provide a conservative reconstruction in u.

According to the results in [4] it is necessary to perform this computation only at the beginning of each time step, to evaluate $\bar{u}_{j+1/2}^n$. Our results confirm this finding. When we use characteristic projection, the derivatives of the fluxes are still computed with the non linear weights obtained with the Global Smoothness indicator. This algorithm will be called *Characteristic Projection (CP)* scheme.

The results obtained with this recipe can be seen on the right of Fig. 1. The spurious oscillations in the density peak of the solution of Lax' Riemann Problem have disappeared.

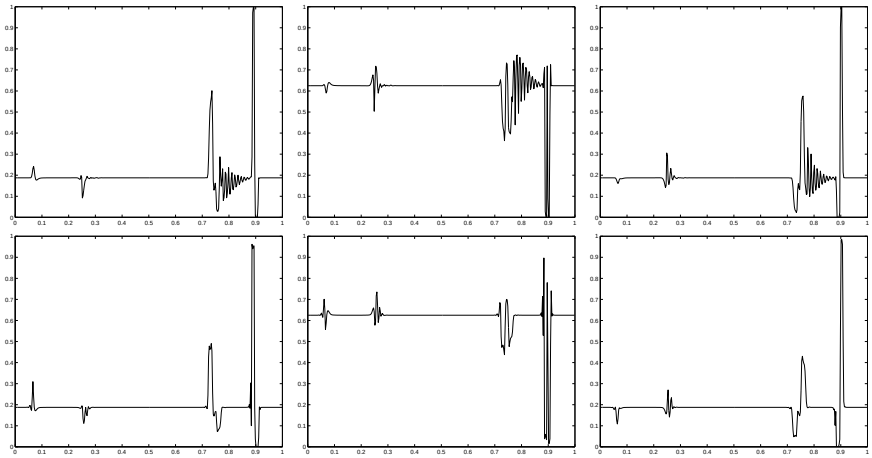

Fig. 2. Global Weights for $N = 400$. From left to right: Left, Central and Right weight. Solution computed with *CW* (top) and *CP* (bottom) algorithms.

The improvement is due to a better separation of the effects of the contact and the shock wave on the weights. Fig. 2 shows a comparison of the non linear global weights which result from the solution to Lax' problem computed with the *CW* (top) and the *CP* (bottom) algorithms. The weights shown in the figure are computed using (10), and they are needed to evaluate the flux derivatives, in both strategies. If we compare with the corresponding solution given in Fig. 1, we see that the fluctuations in all weights clearly signal the location of the shock, the contact, and the head and tail of the rarefaction wave. However in the *CW* implementation, between the shock and the contact the weights have a very irregular behaviour. This effect is absent in the results obtained with the *CP* implementation. This is probably the origin of the wiggles appearing on the left of Fig. (1).

Further insight in the behaviour of the weights can be obtained plotting the weights computed with the *CP* algorithm along each characteristic field, see Fig. 3. Here correctly the contact wave appears only in the second characteristic field, belonging to the eigenvalue $\mu = v$ of the Jacobian $A(u)$, where v is the gas velocity. The head and tail of the rarefaction appear only in very small wiggles in the weights computed for the first characteristic field, belonging to the eigenvalue $\mu = v - c$ (here c is the sound speed). Thus we see that, even if the *CP* algorithm corresponds to a simple local linearization of the system of equations, still it is quite effective in separating all waves thus preventing the oscillatory behaviour of the *CW* scheme.

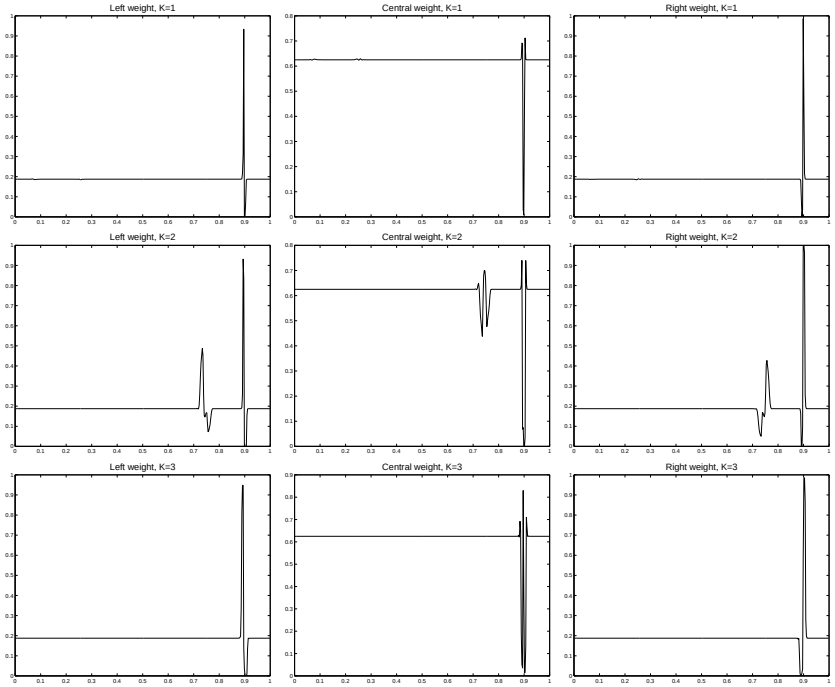

Fig. 3. Weights computed with characteristic projection for $N = 400$. From left to right: Left, Central and Right weight. From top to bottom: first, second and third characteristic field.

4 Adaptive characteristic projection

Table 1. CPU time

N	*CW*	*CP*	Adaptive
200	5.69	15.74	6.50
400	22.45	62.74	25.44
800	90.34	250.34	101.90
1600	359.24	1002.66	409.32

We have seen that the *CP* implementation is quite effective in the prevention of spurious oscillations. However this procedure is quite costly, as is apparent from reading the first two columns of Table 1. Here the CPU time required for the solution of Lax' Riemann Problem is reported for the *CW* and the *CP* algorithms for various grid sizes. These data were obtained running the code on a Pentium II 266 Mz processor, with the F77 Linux

compiler. It is clear that *CP* is roughly 3 times as expensive as *CW*. However Fig. 1 and Fig. 2 show that the *CW* algorithm needs to be improved only close to discontinuities.

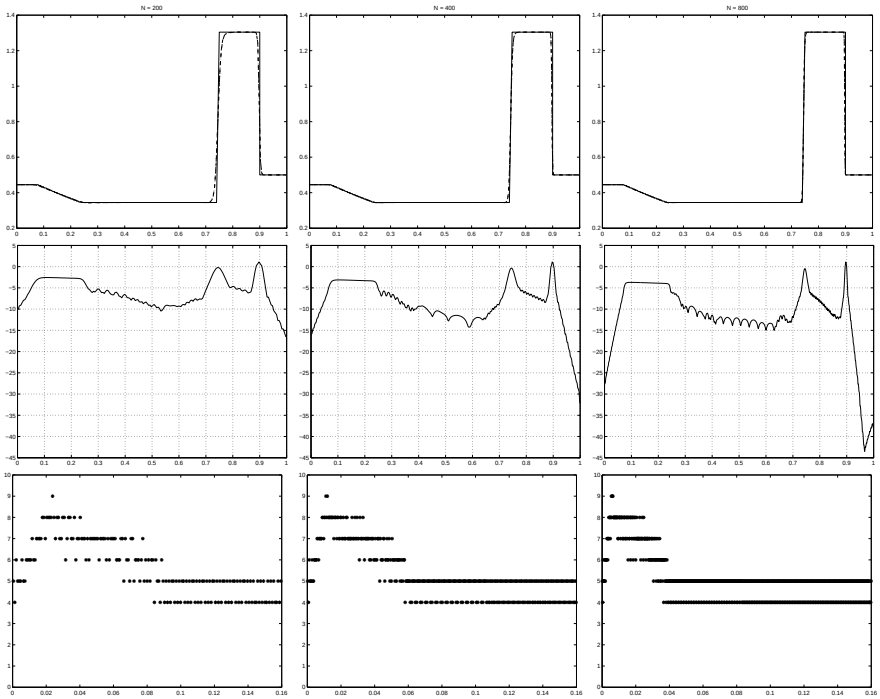

Fig. 4. Top: solution with selective application of characteristic projection, for $N = 200$, $N = 400$ and $N = 800$. **Middle:** Logarithmic plot of $\mathrm{IS}_j^{\mathrm{TOT}}$ vs x at final time. **Bottom:** Number of cells where characteristic projection is used vs time.

Thus it appears that the *CP* scheme can be made much more efficient, if projection along characteristic directions is applied selectively, only in those cells close to discontinuities, and computing the solution in the remaining cells with the *CW* recipe.

The adaptive scheme we propose computes the Global Smoothness indicator in each cell, at the beginning of each time step. Then we obtain a measure of the smoothness of the data in the whole stencil, by evaluating:

$$\mathrm{IS}_j^{\mathrm{TOT}} = \sum_{l=-1}^{1} \mathrm{IS}_j^l,$$

where IS_j^l is given in (10). As for the Smoothness Indicator, IS_j^l, this quantitity is $O(h^2)$ in smooth regions, while it is $O(1)$ if a jump discontinuity is

present somewhere in the whole stencil on which the reconstruction is based. Thus, we apply the *CP* scheme only in those cells for which:

$$\mathrm{IS}_j^{\mathrm{TOT}} \geq 1, \tag{11}$$

while we use the *CW* scheme in the remaining cells. The resulting scheme will be called Adaptive scheme.

We show the results obtained with this strategy in Fig. 4 for Lax' Riemann Problem. In the first row, we show a comparison of the solution computed with the *CP* (dashed line) and the Adaptive (dotted line) schemes. The solid line is the exact solution. As is apparent from the figure, for all grids tested, the two numerical solutions almost coincide.

The second row in Fig. 4 shows plots of the logarithm of $\mathrm{IS}_j^{\mathrm{TOT}}$ for various grid sizes at final time. Compairing with the exact solution in the row above, it is apparent that the peaks of $\mathrm{IS}_j^{\mathrm{TOT}}$ coincide with the location of the discontinuities.

Finally, the third row of Fig. 4 shows the number of cells $N_{CP}(t)$ on which projection along characteristic directions was actually used, at time t, as a function of time for various grid sizes. It is interesting to note that $N_{CP}(t)$ does not depend on the number of grid points. The quantity $N_{CP}(t)$ is larger for small values of t, when the waves issueing form the initial discontinuity have not yet separated, and it decreases as the waves separate. It is clear from these results that a considerable saving in computing time can be obtained through an adaptive application of projection along characteristic directions.

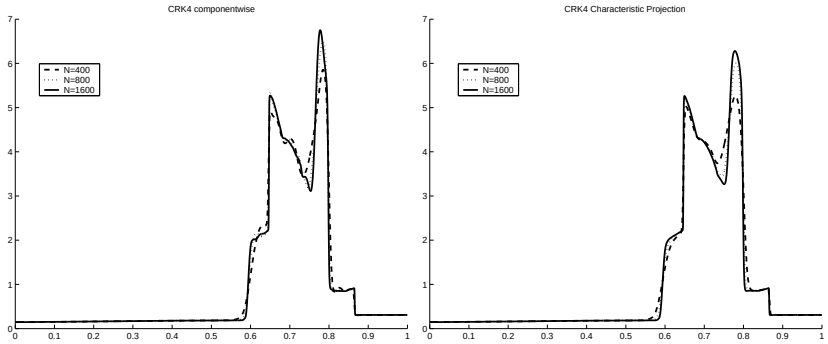

Fig. 5. Solution to Woodward and Colella blast problem, at $T = 0.038$. **Left:** *CW* scheme. **Right:** Adaptive *CP* scheme.

The last column of Table 1 shows the CPU time required by the Adaptive strategy just described. The computational cost of the Adaptive scheme is comparable to the *CW* case, but now the control of spurious oscillations is analogous to what is achieved by the *CP* scheme.

To check the robustness of the adaptive strategy, we compute the solution of Woodward and Colella blast wave problem. The density obtained at $T =$

0.038 is shown in Fig. 5, for the CW (left) and the Adaptive (right) strategies. Again, the CW scheme results in spurious oscillations, which disappear when the Adaptive characteristic projection is used. The numerical results obtained by the Adaptive and the CP scheme are indistinguishable.

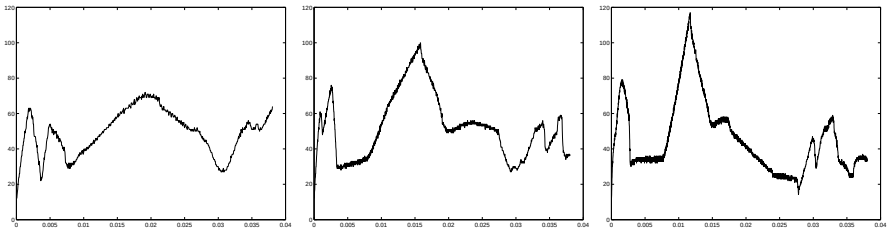

Fig. 6. Woodward and Colella blast problem. Number of cells labelled for projection along characteristic directions as a function of time. From left to right: $N = 400$, $N = 800$, $N = 1600$.

Finally, Fig. 6 shows the values of $N_{CP}(t)$ in the case of Woodward and Colella problem as a function of time, for various grid sizes. These plots are quite irregular, due to the complex behaviour in time of the solution. However we note that even in this case $N_{CP}(t)$ depends only weekly on the number of grid points. This is even more apparent if we consider the average of $N_{CP}(t)$, $< N_{CP} >$, over time. We find: $< N_{CP} >= 48.6$, $< N_{CP} >= 51.9$, $< N_{CP} >= 46.5$ for $N = 400$, $N = 800$ and $N = 1600$ respectively.

We conclude that the adaptive strategy based on applying characteristic projection only in those cells for which: $\mathrm{IS}_j^{\mathrm{TOT}} \geq 1$ is effective in preventing the onset of spurious oscillations, without increasing significantly the computational cost of the componentwise algorithm. Moreover the strategy just described is robust, since it remains effective even on a problem involving a complex solution, with several wave interactions.

References

1. Jiang G.-S., Shu C.-W.: (1996): Efficient Implementation of Weighted ENO Schemes. J. Comput. Phys., **126**, pp.202–228.
2. Levy D., Puppo G., Russo G.: (1999) Central WENO Schemes for Hyperbolic Systems of Conservation Laws. Math. Model. and Numer. Anal., **33**, no. 3, pp.547–571.
3. Pareschi L., Puppo G., Russo G., (2002): Central Runge-Kutta Schemes for Conservation Laws, in preparation.
4. Qiu J, and Shu C.W.: (2002) On the construction, comparison, and local characteristic decomposition for high order central WENO schemes, submitted to J. Comput. Phys.

5. Tadmor E.: (1998): Approximate Solutions of Nonlinear Conservation Laws. In: Quarteroni A. (ed) Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, Lecture Notes in Mathematics, Springer, Berlin.