Staggered Finite Difference Schemes for Conservation Laws

Gabriella Puppo[*], Giovanni Russo[†]

November 9, 2004

Key words. Conservation laws, balance laws, finite difference schemes, high-order accuracy, central schemes.

Abstract

Here we show how to construct finite-difference shock-capturing central schemes on staggered grids. Staggered schemes may have better resolution of the corresponding unstaggered schemes of the same order. They are based on high order non oscillatory reconstruction (ENO or WENO), and a suitable ODE solver for the computation of the integral of the flux. Although they suffer a more severe stability restriction, they do not require a numerical flux function. A comparison between central finite volume and finite difference, on staggered and non staggered grids, is reported.

1 Introduction

The purpose of this work is to introduce a family of high order finite difference (FD) central schemes on staggered grids for the solution of systems of conservation laws.

The original central scheme of Nessyahu and Tadmor [11], and the subsequent high order extension [10], [2], [9] are based on a finite volume (FV) discretization. Central schemes have the advantage over upwind schemes that they do not require the knowledge of the characteristic structure of the system. This feature arises naturally when the schemes are constructed on a staggered grid, as in the papers cited above. Central schemes can be constructed also on nonstaggered grids, as is shown in [8], where a de-staggering procedure is used to reconstruct the field on the original mesh, or in [5], where a semi-discrete central scheme has been derived. Several improvements and high order extension of semidiscrete central schemes have been developed (see, for example, [4], [6]). Although semi-discrete schemes have the advantage of greater flexibility in time discretization, and they allow a simpler treatment of boundary conditions, staggered schemes are attractive since they can give sharper resolution for the same cell size.

Conservative FD schemes for conservation laws have been introduced by Osher and Shu [17]. In such schemes the basic unknown is the pointwise value of the field variable. High order schemes (both FV and FD) can be obtained by combining high order nonoscillatory reconstruction (such as ENO or WENO, see [16]), with high order ODE solvers, such as Runge-Kutta. FD schemes are more efficient than FV for conservation laws in higher dimension, since the reconstruction step in FD schemes can be done dimension by dimension. Both FD and FV methods can be extended to treat balance laws, i.e. hyperbolic systems with a source term. If the source is *stiff* then it is desirable to treat the source term by an implicit scheme, thus avoiding unnecessary stability restrictions on the time step; the flux term can be treated explicitly, and the overall time discretization takes the form of an implicit-explicit (IMEX) scheme (see, for example, [1], [13], [3]). If schemes of order three or more are desired, then FD discretization appears more natural than FV, since in a high order FV scheme the source term couples the cells, making the implicit step more expensive than in the case of FD schemes.

[*] Dipartimento di Matematica, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy; gabriella.puppo@polito.it,http://calvino.polito.it/ puppo

[†] Dipartimento di Matematica ed Informatica, Università di Catania, Viale Andrea Doria 6, 95125 Catania, Italy. russo@dmi.unict.it

The main motivation of this paper is to show how to construct FD central schemes on staggered grids, and provides the framework for the construction of FD schemes for systems with stiff source on staggered grids. In this way one can benefit from the compactness and resolution of staggered schemes.

Hyperbolic systems with stiff source have the form

$$u_t + f_x(u) = \frac{1}{\epsilon} g(u) \tag{1.1}$$

Here, $u(x,t)$ is a function from $\mathbb{R} \times \mathbb{R}^+$ to $\mathbb{R}^m$, f and g are functions from $\mathbb{R}^m$ to $\mathbb{R}^m$, and $\epsilon \geq 0$ is the stiffness parameter. Moreover, we suppose that the Jacobian of f has real eigenvalues and a complete set of eigenvectors for each $u \in \mathbb{R}^m$.

Systems of this kind arise in a large number of applications, such as convection-diffusion equations, reacting gases, kinetic equations, to name just a few. In the case of convection-diffusion equation, the stiffness appear from the discretization of the diffusion operator on the right hand side, and it is not too severe, since $\epsilon \sim h$. For genuine relaxation, the small parameter is independent of the grid spacing, and it can be arbitrarily small.

If one is not interested in resolving the fast transients that arise in these models, when $\epsilon << 1$, it is thus necessary to use implicit schemes for the source term. An effective technique to deal with the different time scales of the convective and the source term is to use IMEX time discretizations. In this approach, an explicit Runge-Kutta scheme is used for the convective term $f_x(u)$, and a suitable diagonally implicit Runge-Kutta scheme is applied to the source term, [13].

For accuracy higher than second order, FV space discretizations are not suited for the construction of implicit methods, because the computation of the average of the source couples the cells and makes an implicit treatment of the source less efficient.

Finite differences do not carry this burden, because only pointwise values of u are involved, the source term does not couple cells, and the implicit step reduces to N_c nonlinear systems of size at most m, where N_c is the number of space cells and m is the number of equations.

However, FD appear to be less robust than standard FV schemes. For example they are more sensitive to the use of characteristic variables in the reconstruction. For a review on high order FD and FV schemes, see [16]. For a comparison between high order staggered and unstaggered FV schemes see [15], where the effect of the use of characteristic variables in the reconstruction is shown. It appears that staggered FV schemes are less sensitive to the use of characteristic variables than their unstaggered counterpart.

In this work we propose a family of high order FD schemes based on staggered grids. We find that the use of a staggered grid improves the robustness of the schemes.

To justify the construction of FD schemes on staggered cells, we compare the new schemes with FV schemes on staggered and unstaggered grids, and also with traditional FD schemes on unstaggered grids. All schemes considered in this work are *black box* schemes, that is we are considering schemes that can, in principle, be applied to complex systems of equations very easily, because they use very little information on the structure of the system. In fact all these schemes require only the definition of the physical flux function, and an estimate of the characteristic velocities.

The structure of the paper is the following. We start in §2 setting the notation for Runge-Kutta schemes and the reconstruction algorithms. Next we describe FD schemes on unstaggered grids, in order to be able to introduce the new FD schemes on staggered grids. Finally in §3 we compare four fourth order accurate schemes for conservation laws. The errors and the shock capturing properties of these schemes are shown, discussing also the efficiency of each scheme.

2 Finite difference schemes for conservation laws

In this paper, we describe the construction of high order finite difference schemes for conservation laws. Before describing the new schemes, proposed in this work, we briefly review the construction of standard FD schemes. In this way we can emphasize the main differences between the two families of schemes, and better specify the actual implementation used in the tests. The section ends with a description of the new FD schemes on staggered grids.

In this section we will consider the case $g(u) \equiv 0$. The extension to problems with stiff sources will be considered in a forthcoming paper [14].

System (1.1) reduces to the following hyperbolic system of conservation laws:

$$u_t + f_x(u) = 0. \tag{2.1}$$

Consider, for the sake of simplicity, a uniform grid, with grid spacing h and time step Δt. Let x_j denote the grid points, $x_{j+1} - x_j = h$, while $x_{j+1/2}$ denotes the point $x_j + h/2$.

We will specialize the description to the construction of fourth order schemes. But the same technique can be applied to schemes of any order of accuracy.

2.1 Notation

Before describing the actual schemes for system (2.1), we need to specify our notation for Runge-Kutta schemes and for the reconstruction algorithms.

2.1.1 Explicit Runge-Kutta schemes

Consider the autonomous ODE problem:

$$\begin{cases} y'(t) = f(y(t)), \\ y'(0) = y_0. \end{cases}$$

The solution is evaluated at the instants t^n, while $y^n \approx y(t^n)$ will denote the approximate solution at time t^n. A ν stage Runge-Kutta scheme applied to the ODE above can be written as:

$$y^{n+1} = y^n + \Delta t \sum_{i=1}^{\nu} b_i K^{(i)}.$$

The quantities $K^{(i)}$ will be called *Runge-Kutta fluxes*, and are defined by the equations:

$$K^{(i)} = f(y^{(i)}), \tag{2.2}$$

where $y^{(i)}$ are called *stage values* and are given by:

$$y^{(1)} = y^n$$

$$y^{(i)} = y^n + \Delta t \sum_{l=1}^{i-1} a_{i,l} K^{(l)}, \quad i = 2 \dots \nu.$$

Thus the matrix $A = (a_{i,l})$ and the vector $b = (b_i)$ uniquely define the Runge-Kutta scheme. For conservation laws, we consider explicit Runge-Kutta schemes, which are characterized by a strictly lower triangular matrix A.

2.1.2 WENO reconstruction

A reconstruction is a piecewise polynomial function:

$$R(x) = \sum_j P_j(x) \chi_{I_j}(x)$$

with possible jumps at the end points of the interval $I_j = (x_j - h/2, x_j + h/2]$. The purpose of the reconstruction is to use information about the field u (for instance, the point values u_j or the cell averages $\bar{u}_j$ of the numerical solution u of (2.1)) to obtain highly accurate information about the field at some other location, according to the structure of the scheme. In this work, we use the piecewise parabolic WENO reconstruction. Here each polynomial $P_j(x)$ is the result of the superposition of three parabolas, with weights chosen in order to maximize accuracy and prevent the onset of spurious oscillations:

$$P_j(x) = \sum_{l=-1}^{1} \omega_j^l P_j^l(x). \tag{2.3}$$

The parabolas $P_j^l(x)$ are computed solving a suitable interpolation problem on the cells $I_{j-1+l}; I_{j+l}; I_{j+1+l}$ for $l = -1, 0, 1$. Therefore the parabolas $P_j^l(x)$ can be written as:

$$P_j^l(x) = \mathcal{R}[u_{j-1+l}, u_{j+l}, u_{j+1+l}]\,(x). \tag{2.4}$$

In other words, the quantity $\mathcal{R}$ is an operator that associates its input data to the interpolation polynomial P_j^l. The structure of $\mathcal{R}$ depends on the particular interpolation requirement that motivates the computation of P_j^l: in the present case, either interpolation in the sense of point values or interpolation in the sense of cell-averages.

The weights ω_j^l are given by [16]:

$$\omega_j^l = \frac{\alpha_j^l}{\sum_{k=-1}^{1} \alpha_j^k}, \qquad \alpha_j^l = \frac{d_l}{(\epsilon + \beta_j^l)^2}. \tag{2.5}$$

The constants d_l are the accuracy constants, and they depend on the particular quantity that must be reconstructed with high accuracy, see [16] and [9]. The quantities β_j^l are the smoothness indicators. Their task is to bias the scheme towards smooth stencils, thus preventing the onset of spurious oscillations. For systems of equations, the performance of the scheme improves if the smoothness indicators are computed globally, summing up the contributions due to each component, namely:

$$\beta_j^l = \frac{1}{m} \sum_{r=1}^{m} \frac{1}{||\bar{u}_r||_2^2} \left(\sum_{k=1}^{2} \int_{I_j} h^{2k-1} \left(\frac{d^k P_{j,r}^l}{dx^k} \right)^2 dx \right) \qquad l = -1, 0, 1, \tag{2.6}$$

where r denotes the r-th component of the solution. Due to the finite speed of propagation, the smoothness indicators do not change qualitatively within one time step. For this reason, we distinguish between a *heavy* reconstruction step, in which the reconstruction is computed together with the corresponding smoothness indicators, and a *light* reconstruction step, in which the reconstruction utilizes previously computed smoothness indicators.

When time integration is performed through a fourth order Runge-Kutta scheme, as in the tests presented in this work, at least four reconstructions are needed. If only the first reconstruction is *heavy*, the performance of the scheme does not change, while the CPU time is reduced considerably.

The piecewise parabolic WENO reconstruction is fifth order accurate for the evaluation of point values for unstaggered schemes, while it is fourth order accurate for central schemes based on staggered grids, unless a splitting of the weights in their positive and negative parts is used, see [12].

2.2 Finite difference scheme on non-staggered cells

In this section we sketch the construction of standard FD schemes based on unstaggered cells. More details appear for instance in the review [16].

In the FD approach, the equation (2.1) is transformed in a system of ODE's in time. Each ODE is evaluated at a grid point x_j, namely:

$$\frac{d}{dt} u_j(t) = -\frac{1}{h} \left(F(u(x_j + h/2, t)) - F(u(x_j - h/2, t)) \right), \tag{2.7}$$

where the function F is the cell primitive of the flux, in the sense that:

$$f(u(x,t)) = \frac{1}{h} \int_{x-h/2}^{x+h/2} F(u(\tilde{x}, t))\, d\tilde{x} \qquad \Longrightarrow \qquad \partial_x f|_{x_j}(t) = \frac{1}{h} \left[F(x_j + h/2, t) - F(x_j - h/2, t) \right].$$

In this fashion, the system of ODE's is naturally written in conservation form, see [17].

In order to build a numerical scheme, it is necessary to construct numerical fluxes $\hat{F}_{j+1/2}$, starting from the point values of the unknown function $u_j(t)$, which must be consistent approximations of the real fluxes $F(x_j + h/2, t)$. The semidiscrete numerical scheme will read as:

$$\frac{d}{dt} u_j(t) = -\frac{1}{h} \left(\hat{F}_{j+1/2} - \hat{F}_{j-1/2} \right). \tag{2.8}$$

To enforce stability, it is necessary to introduce upwinding in the numerical flux. For this reason, the physical flux is split into a positive and a negative part:

$$f(u(x_j,t)) = f^+(u(x_j,t)) + f^-(u(x_j,t)),$$

where, f^+ and f^- are chosen so that their Jacobians have only non-negative and non-positive eigenvalues respectively. Moreover, to preserve high accuracy, the splitting must be a smooth function of its arguments. In this work, we used the local Lax-Friedrichs flux splitting, namely:

$$\begin{array}{rcl} f^+(u(x_j,t)) & = & \frac{1}{2}\left(f(u(x_j,t)) + \alpha\, u(x_j,t)\right) \\ f^-(u(x_j,t)) & = & \frac{1}{2}\left(f(u(x_j,t)) - \alpha\, u(x_j,t)\right) \end{array} \qquad \alpha = \rho(f'(u)), \tag{2.9}$$

where $\rho(f'(u))$ denotes the spectral radius of the Jacobian matrix of f. As a consequence, the numerical flux will also be split as:

$$\hat{F}_{j+1/2} = \hat{F}^+_{j+1/2} + \hat{F}^-_{j+1/2}.$$

To compute $\hat{F}^{\pm}_{j+1/2}$, two separate reconstruction algorithms are used: $R^+(x)$ and $R^-(x)$ which interpolate the data $f^+(u_j)$ and $f^-(u_j)$ respectively in the sense of cell averages, namely:

$$f^+(u_j) = \frac{1}{h}\int_{I_j} R^+(x)\,dx, \qquad f^-(u_j) = \frac{1}{h}\int_{I_j} R^-(x)\,dx.$$

Both reconstructions are piecewise polynomial: let $P^+_j(x)$ and $P^-_j(x)$ be the two polynomials computed on the interval I_j. Now we can finally define the two numerical fluxes as:

$$\begin{array}{rcccl} \hat{F}^+_{j+1/2} & = & R^+(x^-_{j+1/2}) & = & P^+_j(x_{j+1/2}), \\ \hat{F}^-_{j+1/2} & = & R^-(x^+_{j+1/2}) & = & P^-_{j+1}(x_{j+1/2}). \end{array} \tag{2.10}$$

This completes the recipe to compute the numerical flux $\hat{F}_{j+1/2}$, starting from the point values u_j. Next, the system of ODE's (2.8) will be integrated with a ν stage Runge-Kutta scheme. In this case, the Runge-Kutta fluxes $K^{(i)}$ of (2.2) are given by:

$$K^{(i)}_j = -\frac{1}{h}\left[\hat{F}^{(i)}_{j+1/2} - \hat{F}^{(i)}_{j-1/2}\right], \tag{2.11}$$

where the quantities $\hat{F}^{(i)}_{j+1/2}$ are the numerical fluxes computed using the $i-$th intermediate state of the Runge-Kutta scheme.

Remark. Following section 2.1.2, we note that here we need two heavy reconstructions per time step, one to compute $\hat{F}^+$ and one to compute $\hat{F}^-$, at the beginning of each step, because the fluxes f^+ and f^- can be very different. Next, for a 4 stages Runge-Kutta scheme, 6 light reconstructions are required (3 for f^+ and 3 for f^-).

2.3 Staggered finite difference schemes

We now describe how FD schemes based on a staggered grid can be constructed. We cover the computational domain with grid points of the form (x_j, t^n) for even values of n. For odd values of n we consider a staggered grid, with grid points of the form $(x_{j+1/2}, t^n)$, with $x_{j+1/2} - x_j = h/2$.

The evolution of the point values of the solution u on the staggered grid points is given by:

$$\frac{d}{dt}u_{j+1/2}(t) = -\frac{1}{h}\left(F(u(x_{j+1},t)) - F(u(x_j,t))\right),$$

where, as before, F denotes the primitive of the physical flux f.

As in all central schemes based on staggered grids, see [18], the main idea is to construct piecewise polynomial interpolants which are smooth in the intervals I_j centered around the grid points x_j on which the numerical solution is known. Thanks to grid staggering, the numerical fluxes need to be computed at the points x_j, where the interpolants are smooth. This feature will make upwinding un-necessary. Thus the computation of the numerical flux will not require flux splitting. The semidiscrete numerical scheme will be given by:

$$\frac{d}{dt}u_{j+1/2}(t) = -\frac{1}{h}\left(\hat{F}_{j+1} - \hat{F}_j\right). \tag{2.12}$$

Again, the time discretization is performed with a ν stage Runge-Kutta scheme. Thus, if we start from the unstaggered grid (even n), the updated solution can be written as:

$$u^{n+1}_{j+1/2} = u^n_{j+1/2} - \frac{\Delta t}{h}\sum_{i=1}^{\nu} b_i\left[\hat{F}^{(i)}_{j+1} - \hat{F}^{(i)}_j\right], \tag{2.13}$$

The Runge-Kutta fluxes in this case are given by:

$$K^{(i)}_{j+1/2} = -\frac{1}{h}\left[\hat{F}^{(i)}_{j+1} - \hat{F}^{(i)}_j\right].$$

Note the parallel with equation (2.11). To compute the Runge-Kutta fluxes, we need the intermediate values $u^{(i)}_j$ of the solution at the grid points x_j. Since the reconstruction polynomials are smooth at the grid points x_j, these quantities can be computed using the differential form of the PDE, as in CRK schemes [12]:

$$\begin{array}{ll} u^{(1)}_j = u^n_j & \\ u^{(i)}_j = u^n_j + \Delta t\sum_{l=1}^{i-1} a_{i,l}\hat{K}^{(l)}_j & \text{where now} \quad \hat{K}^{(l)}_j = -\partial_x f(u^{(l)})|_j \qquad i = 2,\ldots,\nu. \end{array} \tag{2.14}$$

Once the intermediate state $\{u^{(i)}_j\}$ for all j is known, the intermediate point values of the fluxes are immediately obtained: $f^{(i)}_j = f(u^{(i)}_j)$. With these data, a non-oscillatory interpolant can be computed, in order to obtain high accurate values for $\partial_x f(u^{(i)})|_j$. Here we use the Central WENO interpolation to compute derivatives, see [9]. Finally, the primitive of the fluxes are computed with a deconvolution step, applied to the data $f^{(i)}_j$, thus yielding $\hat{F}^{(i)}_j$.

We still need to specify how to compute the term $u^n_{j+1/2}$ in (2.13). This is done through deconvolution. First, we solve the following interpolation problem: given a smooth function $U(x)$, compute a non-oscillatory piecewise polynomial approximation to the primitive of U. More precisely, find a piecewise polynomial function $v(x)$ such that:

$$\begin{array}{ll} 1) & U(x_j) = \frac{1}{h}\int_{x_{j-1/2}}^{x_{j+1/2}} v(x)\,dx \\ 2) & \frac{1}{h}\int_{x_j}^{x_{j+1}} v(x) = U(x_{j+1/2}) + O(h)^r, \end{array} \tag{2.15}$$

where r is the accuracy of the scheme. This problem can be solved using the central WENO reconstruction of [9]. Thus, given the point values of the numerical solution u^n_j, we compute the primitive $v(x)$, and we set:

$$u^n_{j+1/2} = \frac{1}{h}\int_{x_j}^{x_{j+1}} v(x)\,dx.$$

Remark. To implement a scheme of order 4, based on the central WENO reconstruction and a 4 stage RK scheme, 8 interpolation steps are needed for a single time step. More precisely, one reconstruction step is needed to compute $u^n_{j+1/2}$ starting from the data u^n_j, 3 interpolation steps are required to compute $\partial_x f|^{(i)}_j$, starting from the values of $f(u^{(i)}_j)$, and 4 deconvolution steps are needed to obtain each $\hat{F}^{(i)}$ from the data $f(u^{(i)}_j)$.

However, if we use only one heavy reconstruction at the beginning of each time step, the number of deconvolution steps needed to compute $\hat{F}_j$ reduces substantially. In fact the numerical flux at the i-th stage can be written as:

$$\hat{F}_j^{(i)} = P_j^{(i)}(x_j) = \left(\sum_{l=-1}^{1} \omega_j^l P_j^l(x_j) \right)^{(i)},$$

see (2.3). The parabolas P_j^l are computed through linear interpolation, namely:

$$(P_j^l)^{(i)}(x) = \mathcal{R}\left[f(u_{j-1+l}^{(i)}), f(u_{j+l}^{(i)}), f(u_{j+1+l}^{(i)}) \right](x)$$

Since the smoothness indicators remain constant within the time step, the weights ω_j^l do not depend on the level i. Thus the numerical flux can be computed exploiting the linearity of $\mathcal{R}$:

$$\begin{aligned} \textstyle\sum_{i=1}^{\nu} b_i \hat{F}_j^{(i)} &= \textstyle\sum_{l=-1}^{1} \omega_j^l \left(\sum_{i=1}^{\nu} b_i (P_j^l)^{(i)}(x_j) \right) \\ &= \textstyle\sum_{l=-1}^{1} \omega_j^l \mathcal{R} \left(\sum_{i=1}^{\nu} b_i f_{j-1+l}^{(i)}, \sum_{i=1}^{\nu} b_i f_{j+l}^{(i)}, \sum_{i=1}^{\nu} b_i f_{j+1+l}^{(i)} \right)(x_j). \end{aligned}$$

With this formulation only one deconvolution step is needed to compute the global numerical flux, and therefore the total number of interpolation steps per time step is reduced to five.

3 Numerical results

In this section, we compare our new FD scheme on a staggered grid with other high resolution schemes for conservation laws.

In our experience, fourth order schemes seem to be particularly effective whenever high resolution is called for [12]. For this reason we will compare four different fourth order schemes. All schemes considered here are based on the piece-wise parabolic WENO reconstruction, coupled with the standard 4th order Runge-Kutta scheme for time discretization.

The acronym FDS denotes the scheme proposed in this work (Finite Difference on Staggered grid). FDS-5 is again a FD scheme on staggered grid, but it is based on the 5th order accurate reconstruction for point values, obtained with a splitting into positive and negative weights: this scheme differes from FDS only in the evaluation of the primitive of the flux at the end of each time step. However, it has a stricter CFL limit than plain FDS, see [12]. FDU is the Finite Difference scheme on Unstaggered grids described above and found in [16]. We also consider three FV schemes. FVS is the 4th order Central Runge Kutta scheme described in [12] (Finite Volume on Staggered grid). FVS is based on the 4th order Central WENO reconstruction . It has the same predictor step of the present FDS method to compute the intermediate states $u^{(i)}$, but it does not require the computation of the primitive of the fluxes. FVS-5 is the FVS scheme with the 5th order reconstruction described, as before, in [12]. Finally FVU (Finite Volume on Unstaggered grid) has again been drawn from [16]. It has the 5th order piecewise parabolic WENO reconstruction, while the numerical flux is the local Lax-Friedrichs flux. Thus at the cell interface $x_{j+1/2}$, the numerical flux is given by:

$$F_{j+1/2} = F(u_{j+1/2}^+, u_{j+1/2}^-) = \frac{1}{2}\left[f(u_{j+1/2}^+) + f(u_{j+1/2}^-) - \alpha \left(u_{j+1/2}^+ - u_{j+1/2}^- \right) \right]. \tag{3.16}$$

Here α is the stabilization parameter, based on the characteristic velocities: for these tests it was chosen as $\alpha = \max(\rho(f'(u_{j+1/2}^-)), \rho(f'(u_{j+1/2}^+)))$, while $u_{j+1/2}^+$ and $u_{j+1/2}^-$ are the right and left boundary extrapolated data. Therefore this scheme requires two flux evaluations for each computation of the numerical flux.

Remark. Note that all schemes considered in this section are central type schemes, in the sense that the evaluation of the numerical fluxes does not ask for the computation of the wave structure of the local solution. Thus all four schemes can be easily applied to complex systems of equations.

Moreover we only consider the basic componentwise implementation of these schemes, with the global smoothness indicators computed only at the beginning of each time step (see [9]). Thus no projection

L^1 norm of the error

N	FVS	FVS-5	FVU	FDS	FDS-5	FDU
10	0.58291	0.62727	0.61649	0.58077	0.62635	0.62292
20	0.15184	0.20264	0.30372	0.15204	0.20210	0.30729
40	0.87974E-02	0.11303E-01	0.18575E-01	0.87680E-02	0.11275E-01	0.19677E-01
80	0.42768E-03	0.40547E-03	0.69332E-03	0.42481E-03	0.40516E-03	0.75106E-03
160	0.24118E-04	0.12825E-04	0.23136E-04	0.24026E-04	0.12824E-04	0.24484E-04
320	0.14698E-05	0.40048E-06	0.83244E-06	0.14670E-05	0.40049E-06	0.85879E-06
640	0.91271E-07	0.12331E-07	0.36373E-07	0.91180E-07	0.12331E-07	0.36240E-07

Table 3.1: L^1 error on point values for several fourth order schemes

along characteristic directions is required. The only information that these schemes need is the physical flux function and an estimate of the maximum characteristic velocity to satisfy the CFL condition. The schemes based on unstaggered grids require sharper information on the size of characteristic velocities to compute a stable and not too diffusive numerical flux, so that α in (2.9) and (3.16) must not overestimated.

Note that the addition of characteristic information improves the quality of the results even for staggered schemes, especially for higher order schemes. However, staggered schemes are less sensitive to the use of characteristic variables in the reconstruction. These effects are studied with some detail in [15], [12].

3.1 Linear advection

We consider the initial value problem:

$$u_t + Au_x = 0 \qquad \text{on} \qquad [0,1],$$

with periodic boundary conditions. The matrix A and its eigenvalues μ are:

$$A = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & -1 \end{pmatrix} \qquad \Rightarrow \qquad \mu = -1, 1, 3.$$

The initial condition is:

$$u_0(x) = (\sin(2\pi x), \sin(4\pi x), \cos(2\pi x))^T.$$

The integration is stopped at $T = 3$. In Table 3.1 we report the error in the L^1 norm for the point values at the center of each cell. We report only the error computed on the second equation of the system, since here the error is largest. The mesh ratio is $\lambda \equiv \Delta t/h = 0.3$ for the schemes based on unstaggered grids, while it is $\lambda = 0.3 * 0.5$ for FVS and FDS, since staggered grids require a stricter CFL condition, see [18]. FVS-5 and FDS-5 require $\lambda = 0.3 * 0.4$, see [12].

The behaviour of the errors is comparable for all six methods. The FDS and FVS display smaller errors on coarse grids, but they loose ground on fine grids with respect to unstaggered schemes. This is due to the fact that the Central WENO reconstruction used by these staggered schemes is only fourth order accurate on point values, while the reconstruction of the unstaggered schemes is fifth order accurate on point values. The staggered schemes with the 5th order accurate in space reconstruction, FDS-5 and FVS-5, have smaller errors than the corresponding unstaggered schemes on almost all grids studied: the gap increases as the grid is refined. A comparison of Table 3.1 and Table 3.1 seems to indicate that, for a given CPU time, the staggered schemes achieve smaller errors than their unstaggered version, if the reconstruction and time integration used have the same accuracy.

Next, we compare the efficiency of these schemes in Table 3.1. The CPU time is sampled from the accuracy runs that produced Table 3.1. The number of time steps corresponds to the finest grid of Table 3.1. It is really difficult to give reliable CPU times for different methods, since the CPU time depends heavily on the particular implementation considered. This is why the data reported here should be considered only qualitatively. Specifically, they simply show that the execution time is comparable for all schemes tested, even though the methods based on staggered grids require at least twice as many time steps.

Efficiency

	FVS	FVS-5	FVU	FDS	FDS-5	FDU
CPU time	48.75	68.20	56.59	55.55	72.56	47.29
Number of time steps	12800	16000	6400	12800	16000	6400
Flux evaluations (per time step)	4	4	8	4	4	4
Heavy reconstructions (per time step)	1	1	1	1	1	2
Light reconstructions (per time step)	3	3	3	4	4	6
Numerical fluxes (per time step)	0	0	4	0	0	8
Cell staggering (per time step)	1	1	0	1	1	0

Table 3.2: CPU time and operation count for several fourth order schemes

The remaining information reported in Table 3.1 tries to explain why the schemes based on staggered grids have a faster time step, although all schemes considered are central type schemes. In particular, we note that FVU requires 8 flux function evaluations per time step, since the flux must be computed for both the left and the right boundary extrapolated data, at each intermediate step of the RK scheme. On the other hand, FDU requires the splitting of the flux into its positive and negative parts. This doubles the number of reconstructions and the number of numerical flux computations required at each step. In both FDU and FVU the heavy reconstructions are computed the first time the numerical flux is needed in each time step.

For the staggered schemes, the heavy reconstruction is applied when the deconvolution step is needed to obtain $u^n_{j+1/2}$ from u^n_j. Beside the deconvolution, the evaluation of $u^n_{j+1/2}$ requires a quadrature on the cell. This simple operation is accounted for as "cell staggering" in the Table. Note that no numerical flux is needed for the staggered schemes. This is due to the fact that the flux is computed only in regions of smoothness of the solution, thus the numerical flux can be chosen to coincide with the physical flux. All staggered schemes require 3 light reconstructions, to compute the 3 intermediate values of $f_x(u^{(i)})_j$ from the knowledge of $f(u^{(i)}_j)$. The FD version needs one more light reconstruction step to compute the primitive of the assembled flux, as specified in the remark at the end of §2.3. FVS-5 and FDS-5 differ from FVS and FDS only in the evaluation of point values, which occurs only once per time step: the higher value of the CPU time for these schemes is due to the fact that they require a larger number of time steps.

3.2 Shock tube problem

To assess the shock capturing properties of these schemes and their ability to prevent the onset of spurious oscillations, we consider the classical Lax' Riemann problem for gas dynamics, [7]. Here the initial condition is $u = u_L$ for $x \leq 0.5$, and $u = u_R$ for $x > 0.5$. The computational domain is $[0, 1]$, with free-flow boundary conditions. The left (L) and right (R) states are given by:

$$u_L = \begin{pmatrix} 0.445 \\ 0.311 \\ 8.928 \end{pmatrix}, \qquad u_R = \begin{pmatrix} 0.5 \\ 0. \\ 1.4275 \end{pmatrix},$$

where u is the vector of conservative variables, namely: density, momentum and total energy per unit volume. The computation is arrested at $T = 0.16$, and the mesh ratio is $\lambda = 0.2$ for unstaggered schemes, and $\lambda = 0.1$ for the staggered schemes.

A detail of the density component of the solution can be seen in Fig. 3.1 for the case with $N = 200$ grid points and in Fig. 3.2 for $N = 400$. Both figures show the profiles obtained with the staggered schemes on the left, and the solution given by the unstaggered schemes on the right. In the case of staggered schemes, FVS and FDS give very similar solutions, and the two profiles cannot be distinguished. On the contrary, the FD unstaggered scheme FDU (dashed solution) seems more oscillatory than its FV analogue. This is particularly apparent in Fig. 3.1.

As the grid is refined, see Fig. 3.2, the amplitude of the spurious oscillations decreases fast for all schemes considered.

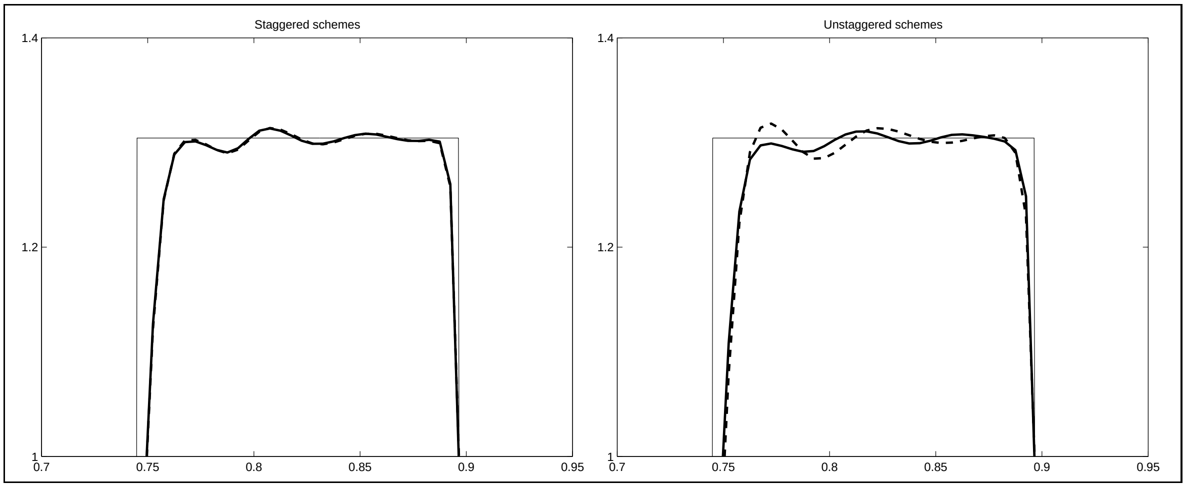

Figure 3.1: Lax' Riemann problem $\lambda = 0.2$. Solution for $N = 200$ for the staggered schemes (left) and the unstaggered schemes(right). Solid line: FV schemes, Dashed line: FD schemes.

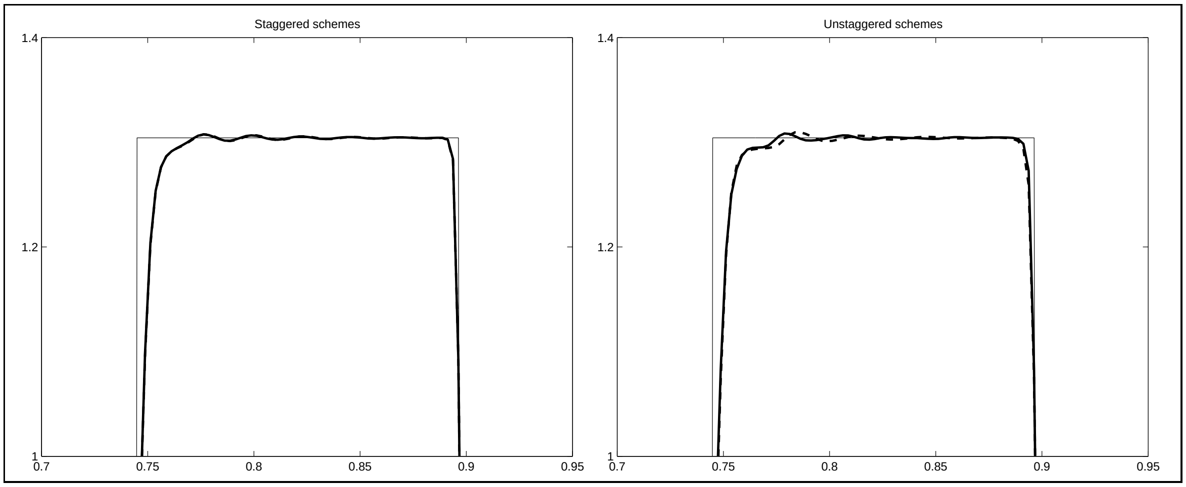

Figure 3.2: Lax' Riemann problem $\lambda = 0.2$. Solution for $N = 400$ for the staggered schemes (left) and the unstaggered schemes(right). Solid line: FV schemes, Dashed line: FD schemes.

3.3 Conclusions

In this work, we propose a fourth order accurate, essentially non-oscillatory FD scheme for conservation laws, based on a staggered grid (FDS).

Several comparisons, with analogous fourth order accurate central schemes, show that the FDS scheme is competitive with the other schemes as far as accuracy and computing complexity are concerned. Moreover the FDS scheme seems to have an edge over a FD scheme such as FDU on the control of spurious oscillations. This makes the FDS scheme an interesting tool for the construction of high order non oscillatory schemes for stiff balance laws, where finite differences, in many interesting cases, allow to write more efficient implicit methods than algorithms based on finite volumes. This problem is addressed in [14].

References

[1] U. Asher, S. Ruuth, R. J. Spiteri, *Implicit-explicit Runge-Kutta methods for time dependent Partial*

Differential Equations, Appl. Numer. Math. 25, (1997), pp. 151–167.

[2] Bianco F., Puppo G., Russo G., *High Order Central Schemes for Hyperbolic Systems of Conservation Laws*, SIAM J. Sci. Comp., 21 (1999) no 1, 294–322.

[3] C. A. Kennedy, M. H. Carpenter, *Additive Runge-Kutta schemes for convection-diffusion-reavtion equations*, *Appl. Numer. Math.* **44**, no. 1-2 (2003), 139–181.

[4] Kurganov A., Noelle S., Petrova G., *Semi-discrete central-upwind scheme for hyperbolic conservation laws and Hamilton Jacobi equations*, SIAM J. Sci. Comp., **23** (2001), 707-740.

[5] Kurganov A., Tadmor E., *New high-resolution central schemes for nonlinear conservation laws and convection-diffusion equations*, J. Comput. Phys. **160** (2000), no. 1, 241–282.

[6] Kurganov, A., Levy, D., *A third-order semidiscrete central scheme for conservation laws and convection-diffusion equations*, SIAM J. Sci. Comput. **22**, no. 4 (2000), 1461–1488.

[7] Lax P.D., *Weak Solutions of Non-Linear Hyperbolic Equations and Their Numerical Computation*, CPAM **7** (1954), 159-193.

[8] Jiang, G.-S., Levy, D., Lin, C.-T., Osher, S., Tadmor, E., *High-resolution nonoscillatory central schemes with nonstaggered grids for hyperbolic conservation laws*, SIAM J. Numer. Anal. **35**, no. 6 (1998), 2147–2168.

[9] Levy D., Puppo G., Russo G., *Central WENO Schemes for Hyperbolic Systems of Conservation Laws*, Math. Model. and Numer. Anal., **33**, no. 3 (1999), 547–571.

[10] Liu X.D., Tadmor E., *Third order nonoscillatory central scheme for hyperbolic conservation laws*, Numer. Math., **79** (1998), 397–425.

[11] Nessyahu H., Tadmor E., *Non-oscillatory Central Differencing for Hyperbolic Conservation Laws*, J. Comput. Phys., **87**, no. 2 (1990), 408–463.

[12] Pareschi L., Puppo G., Russo G., *Central Runge-Kutta Schemes for Conservation Laws*, to appear on SIAM J. Sci. Comp.

[13] Pareschi L., Russo G., *Implicit-explicit Runge-Kutta Schemes and Applications to Hyperbolic Systems with Relaxation*, accepted on Journal of Scientific Computing.

[14] Puppo G., Russo G., *Staggered Finite Difference Schemes for Balance Laws*, in preparation.

[15] Qiu J, and Shu C.W., *On the construction, comparison, and local characteristic decomposition for high order central WENO schemes*, Journal of Computational Physics, **183** (2002), pp.187-209.

[16] Shu C.-W., *Essentially Non-Oscillatory and Weighted Essentially Non-Oscillatory Schemes for Hyperbolic Conservation Laws* in Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, Lecture Notes in Mathematics (A. Quarteroni editor), Springer, Berlin, 1998.

[17] Shu C.-W., Osher S., *Efficient Implementation of Essentially Non-oscillatory Shock-Capturing Schemes*, JCP **77**, 1988, 439-471.

[18] Tadmor E., *Approximate Solutions of Nonlinear Conservation Laws*, in Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, Lecture Notes in Mathematics (editor: A. Quarteroni), Springer, Berlin, 1998.