CrossMark

Adaptive Mesh Refinement for Hyperbolic Systems Based on Third-Order Compact WENO Reconstruction

M. Semplice[1] · A. Coco[2] · G. Russo[3]

Abstract In this paper we generalise to non-uniform grids of quad-tree type the Compact WENO reconstruction of Levy et al. (SIAM J Sci Comput 22(2):656–672, 2000), thus obtaining a truly two-dimensional non-oscillatory third order reconstruction with a very compact stencil and that does not involve mesh-dependent coefficients. This latter characteristic is quite valuable for its use in h-adaptive numerical schemes, since in such schemes the coefficients that depend on the disposition and sizes of the neighbouring cells (and that are present in many existing WENO-like reconstructions) would need to be recomputed after every mesh adaption. In the second part of the paper we propose a third order h-adaptive scheme with the above-mentioned reconstruction, an explicit third order TVD Runge–Kutta scheme and the entropy production error indicator proposed by Puppo and Semplice (Commun Comput Phys 10(5):1132–1160, 2011). After devising some heuristics on the choice of the parameters controlling the mesh adaption, we demonstrate with many numerical tests that the scheme can compute numerical solution whose error decays as $\langle N \rangle^{-3}$, where $\langle N \rangle$ is the average number of cells used during the computation, even in the presence of shock waves, by making a very effective use of h-adaptivity and the proposed third order reconstruction.

Keywords High order finite volumes · h-adaptivity · Numerical entropy production

Mathematics Subject Classification 65M08 · 65M12 · 65M50

✉ M. Semplice
matteo.semplice@unito.it

A. Coco
Armando.Coco@bristol.ac.uk

G. Russo
russo@dmi.unict.it

[1] Dipartimento di Matematica "G. Peano", Università di Torino, Via C. Alberto, 10, 10123 Turin, Italy

[2] School of Earth Sciences, Bristol University, Queens Road, Bristol BS8 1RJ, UK

[3] Dipartimento di Matematica ed Informatica, Università di Catania, Viale Andrea Doria, 6, 95125 Catania, Italy

1 Introduction

We consider numerical approximations of the solution of an initial value problem for hyperbolic system of m conservation laws in a domain $\Omega \subset \mathbb{R}^d$

$$\partial_t u + \nabla \cdot f(u) = 0 \tag{1}$$

with suitable boundary conditions. The finite volume formulation of (1) reads

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{\Omega_j} u(t, x)\mathrm{d}x + \oint_{\partial \Omega_j} f(u(t, x)) \cdot \mathbf{n}\, \mathrm{d}l = 0, \tag{2}$$

where $\{\Omega_1, \Omega_2, \ldots, \Omega_N\}$ denotes a partition of Ω. A semidiscrete finite volume scheme tracks the evolution in time of the cell averages

$$U_j = \frac{1}{|\Omega_j|} \int_{\Omega_j} u(t, x)\, \mathrm{d}x$$

and, in order to approximate the boundary integrals in (2) one would need to know the point values of the approximate solution at suitable quadrature points along the boundary of each cell Ω_j. This is accomplished by a reconstruction procedure that computes point values of u at quadrature nodes along $\partial \Omega_j$ from the cell average U_j and the cell averages of the neighbours of Ω_j.

In order to obtain a convergent scheme, the reconstruction procedure must satisfy requirements of mass conservation, accuracy (on smooth data) and have some non-oscillatory property, controlling the boundedness of the total variation (on non-smooth data).

A robust method that yields high order accuracy and possesses non-oscillatory properties is the so-called WENO reconstruction. In its original formulation [35], in the case of a uniform grid in one space dimension, one starts by considering all the possible $k + 1$ stencils (from downwind, to central, to upwind) of width $k + 1$ containing the cell Ω_j and the corresponding polynomials $p_1(x), \ldots, p_k(x)$ of degree k that interpolate the data in the sense of the cell averages on each of the stencils. One can also determine the so-called linear weights $\alpha_\gamma^\pm$ such that $\sum_\gamma \alpha_\gamma^\pm p_\gamma(x)$ yields $2k + 1$ order accuracy when evaluated at cell boundaries $x_{j\pm1/2}$ (that is the accuracy given by interpolating in the sense of the cell averages $U_{j-k}, \ldots, U_{j+k}$ with a single polynomial of degree $2k$). The reconstruction is then taken to be a convex linear combination $p(x) = \sum_\gamma \tilde{\alpha}_\gamma p_\gamma(x)$, where the nonlinear weights $\tilde{\alpha}_\gamma$ are chosen according to the smoothness of the data in the γ-th stencil and in such a way that, $\tilde{\alpha}_\gamma \simeq \alpha_\gamma$ for all values of γ if the solution is smooth in the union of all the stencils and $\tilde{\alpha}_\gamma \simeq 0$ if the solution is not smooth in the γ-th stencil. Note that there will be two different sets of linear weights $\{\alpha_\gamma^+\}$ and $\{\alpha_\gamma^-\}$ respectively for the right- and left- boundary of the cell $\Omega_j = [x_{j-1/2}, x_{j+1/2}]$, but this can be handled efficiently in one space dimensions or even in higher dimensional setting if one has a Cartesian grid and employs dimensional splitting [35].

The situation is far more complicated for non-Cartesian grids, especially for higher order schemes. In fact, in two or more dimensions, in order to approximate the boundary integral in (2), one uses a suitable quadrature formula for each face of $\partial \Omega_j$. Up to schemes of order 2, the midpoint rule can be employed, but for schemes of higher order, one has to use a quadrature formula with more than one node per face and this increases the cardinality of the sets of linear weights that should be used.

One approach, typical of the ClawPack software, is to build the computational grids from regular Cartesian patches. We point out that ClawPack now includes higher order solvers (SharpClaw) [20], but their code has been so far released only in one space dimension [26].

The more traditional approach is to employ a single grid composed of cells of different sizes. In this framework, WENO schemes on arbitrary triangular meshes were constructed in [18]. For efficiency, the third order reconstruction suitable for a two-point quadrature formula on each edge, requires the precomputation and storage of 90 linear weights and 180 other constants for the smoothness indicators for each triangle in the mesh. Moreover, upon refinement or coarsening, one would have to recompute all the above mentioned constants in all cells that include the newly-created one in their stencil (usually 10 cells). Additionally, in the general case, the linear weights are not guaranteed to be positive: see [34] for the adverse effects of this non-positivity and for ways to circumvent this. Similar issues are expected to arise when using locally refined quadrangular grids, like those of Fig. 2 as there are many possible configurations for the number, size and relative position of the neighbours. More recently Dumbser and Käser [13] considered a WENO scheme that combines central and directional polynomials of degree k, requiring very large stencils.

Many of these difficulties are linked to the otherwise desirable aim of achieving enhanced accuracy of order $2k + 1$ at the quadrature nodes. Relaxing this requirement, a reconstruction of order $k + 1$ can obviously be achieved by combining polynomials of order up to k on each stencil using a single set of linear weights for every quadrature node of every face and for all dispositions of cells in the neighbourhood. In fact, this procedure gives a k-degree polynomial that represents a uniform approximation of order $k + 1$ on the cell and thus can just be evaluated at all quadrature nodes. The accuracy would thus be similar to the accuracy of a ENO procedure, but using a nontrivial linear combination of all polynomials avoids the difficulties of ENO reconstructions on very flat solutions [32].

Another problem arising is related to the stencil widths. In fact, in order to compute a polynomial of degree k interpolating the cell averages of the numerical solution using a fully upwind stencil in some direction, one needs to use data that is quite far away from the cell Ω_j. This poses a question on how efficiently one can gather the information of the local topology of the triangulation and, since this may change at every timestep in an adaptive mesh refinement code, it would be a clear advantage to employ a reconstruction that can achieve order $k + 1$ on smooth solutions by using a compact stencil.

A step in this direction was set by Levy et al. [23], that introduced the Compact WENO (CWENO) reconstruction of order 3 for regular Cartesian grids. The main idea is to consider a central stencil composed only by the cells that share at least a vertex with Ω_j and to construct an "optimal" reconstruction polynomial P_{OPT} of degree 2 that interpolates exactly the cell averages in this stencil. In order to control the total variation increase possibly caused by the presence of a discontinuity in the stencil, this polynomial is decomposed as

$$P_{OPT} = \alpha_0 P_0 + \sum_{\gamma=1}^{D(d)} \alpha_\gamma P_\gamma,$$

where the P_γ's for $\gamma > 0$ constitute a finite set of $D(d) = 2^d$ first degree polynomials interpolating in the sense of the cell averages in a directional sector of the central stencil in the direction of each of the 2^d vertices of the d-dimensional cubic cell. P_0 is a second degree polynomial computed by difference from P_{OPT}. The reconstruction polynomial is then given by

$$P = \tilde{\alpha}_0 P_0 + \sum_{\gamma=1}^{D(d)} \tilde{\alpha}_\gamma P_\gamma,$$

where the coefficients $\tilde{\alpha}_\gamma$ are computed from the linear weights α_γ as usual, with the help of smoothness indicators. This procedure allows to obtain third order accuracy in smooth regions, degrades to a linear reconstruction near discontinuities, but employs a very compact stencil. Notice that a natural choice is to take the $\alpha_\gamma = (1 - \alpha_0)/D(d)$ for each $\gamma = 1, \ldots, D(d)$. Therefore the set of linear weights depends on a single parameter $\alpha_0 \in (0, 1)$. In practice we use $\alpha_0 = 1/2$ in the whole paper. In our experience, the use of higher values of α_0 is beneficial only in problems without discontinuities.

This reconstruction does not aim at computing directly the point values at specific quadrature points on the cell boundary, but gives a single reconstruction polynomial that can then be evaluated at any point in the cell with uniform accuracy. The degree of accuracy is of course the same than the one achievable by the standard WENO technique with the same stencils, but this approach seems more suitable in the context of non-uniform and adaptive grids, because it requires a single set of linear weights. This feature turns out to be quite useful in the case of balance laws with source terms, where the reconstruction at quadrature points inside the cell is required for the well-balanced quadrature of the source term, see [31].

In order to construct a third order scheme suitable for an adaptive mesh refinement setting, in this paper we extend the idea of the CWENO reconstruction to nonuniform meshes. In one space dimension the extension is straightforward, while in higher dimensions complications arise due to the variable cardinality of the set of first neighbours of a given cell. These are solved in Sect. 2 by considering an interpolation procedure that enforces exact interpolation of the cell average in the central cell but only a best fit in the least squares sense of the other cell averages in the stencil. Section 3 is devoted to analysing the behaviour of an adaptive scheme based on some error indicator and the employment of cells of sizes between H and $H/2^L$, in order to provide guidance on the number of "levels" L that need to be used and on the response of the scheme to the choice of the refinement threshold. The case of global timestepping is considered in this paper. Finally, Sect. 4 presents numerical tests in one and two space dimensions that demonstrate both the properties of the reconstruction introduced in Sect. 2 and corroborate the ideas on the behaviour of adaptive schemes illustrated in Sect. 3. Section 5 contains the conclusions and perspectives for future work.

2 Third Order, Compact Stencil Reconstruction

In our approach, we try to combine the simplicity of the Cartesian meshes, the compactness of the CWENO reconstruction and the flexibility of the quad-tree local grid refinement for the construction of h-adaptive schemes. We thus restrict ourselves to grids where each cell Ω_j is a d-dimensional cube of edge h_j and may be refined only by splitting it in 2^d equal parts. Such grids are saved in binary/quad-/oct-trees. Let us introduce some notation. Let Ω be the entire domain, $\{\Omega_j, j = 1, \ldots, N\}$ the set of all cells, $\mathbf{x}_j$ the centre of cell Ω_j, and $|\Omega_j| = h_j^d$.

2.1 One Space Dimension

In one space dimension the reconstruction is a straightforward extension of the reconstruction described in [23] to the case of non-uniform grids. Let $u(x)$ be a function defined in $\Omega = [a, b]$ and let us suppose we know the mean values of u in each cell Ω_j, i.e. we know

$$U_j = \frac{1}{h_j} \int_{\Omega_j} u(x) dx, \quad j = 1, \ldots, N,$$

with $h_j = |\Omega_j|$. Now let us fix a cell Ω_j. In order to compute the reconstruction in that cell, we use a compact stencil, namely we use only the mean values of u in that cell, U_j, and in the first neighbours, U_{j-1} and U_{j+1}. If $u(x)$ is locally smooth, one may choose the third order accurate reconstruction given by the parabola that interpolates the data U_{j+i} in the sense of cell-averages to enforce conservation. We shall call such a parabola the *optimal* polynomial $P_{\mathrm{OPT}}(x)$:

$$\frac{1}{h_{j+i}} \int_{\Omega_{j+i}} P_{\mathrm{OPT}}(x)dx = U_{j+i}, \quad i = -1, 0, 1.$$

The optimal polynomial $P_{\mathrm{OPT}}(x)$ is completely determined by these conditions, and its expression is given by:

$$P_{\mathrm{OPT}}(x) = U_j + p_x(x - x_j) + \frac{1}{2}p_{xx}\left((x - x_j)^2 - \frac{h_j}{12}\right),$$

with

$$p_x = \frac{(h_j + 2h_{j-1})U[j - 1; j] + (h_j + 2h_{j+1})U[j; j + 1]}{2(h_{j-1} + h_j + h_{j+1})},$$

$$p_{xx} = \frac{3(2h_j + h_{j-1} + h_{j+1})U[j - 1; j; j + 1]}{2(h_{j-1} + h_j + h_{j+1})},$$

where

$$U[j - 1; j] = \frac{U_j - U_{j-1}}{x_j - x_{j-1}}, \quad U[j; j + 1] = \frac{U_{j+1} - U_j}{x_{j+1} - x_j},$$

$$U[j - 1; j; j + 1] = \frac{U[j; j + 1] - U[j - 1; j]}{x_{j+1} - x_{j-1}}.$$

If $u(x)$ was not smooth in $\cup_{i=-1}^{1}\Omega_{j+i}$, then this reconstruction would be oscillatory. Following the idea of [23], we compute two linear functions P_γ, $\gamma = 1, 2$ in such a way P_γ matches the cell averages U_j and $U_{j+2\gamma-3}$, namely:

$$P_\gamma(x) = U_j + U[j - 2 + \gamma; j - 1 + \gamma](x - x_j), \quad \gamma = 1, 2,$$

and a parabola P_0 determined by the relation:

$$P_{\mathrm{OPT}} = \alpha_0 P_0 + \sum_{\gamma=1}^{2} \alpha_\gamma P_\gamma,$$

where the coefficients α_γ can be chosen arbitrarily, provided $\alpha_\gamma > 0$ and $\sum_{\gamma=0}^{2} \alpha_\gamma = 1$. In practice we use $\alpha_0 = 1/2$ and $\alpha_\gamma = 1/4$, $\gamma = 1, 2$. The final reconstruction polynomial is:

$$P = \tilde{\alpha}_0 P_0 + \sum_{\gamma=1}^{2} \tilde{\alpha}_\gamma P_\gamma, \tag{3}$$

with

$$\tilde{\alpha}_\gamma = \frac{\omega_\gamma}{\sum_{\delta=0}^{2} \omega_\delta}, \quad \omega_\gamma = \frac{\alpha_\gamma}{(\epsilon + \beta_\gamma)^2}, \quad \gamma = 0, 1, 2.$$

The smoothness indicators, β_γ, are responsible for detecting large gradients or discontinuities and to automatically switch to the stencil that generates the least oscillatory reconstruction in such cases. On the other hand, if $u(x)$ is smooth the smoothness indicators should be

expected to be close to each other and thus $\tilde{\alpha}_\gamma \approx \alpha_\gamma$ and $P(x) \approx P_{\text{OPT}}(x)$ for every set of coefficients α_γ. Following [23,35], we define the smoothness indicators as:

$$\beta_\gamma = \sum_{l=1}^{2} \int_{\Omega_j} |\Omega_j|^{2l-1}(P_\gamma^{(l)}(x))^2 dx, \quad \gamma = 0, 1, 2. \tag{4}$$

Let us rewrite the central polynomial P_0 as:

$$P_0 = U_j + p_x^0(x - x_j) + \frac{1}{2}p_{xx}^0\left((x - x_j)^2 - \frac{h_j^2}{12}\right).$$

A direct computation of (4) yields:

$$\beta_0 = \frac{13}{12}h_j^4\left(p_{xx}^0\right)^2 + h_j^2\left(p_x^0\right)^2,$$
$$\beta_1 = U[j-1; j]^2 h_j^2, \quad \beta_2 = U[j; j+1]^2 h_j^2.$$

It is known that the role of ϵ goes beyond the simple avoidance of zero denominators in the computation of the nonlinear weights, but its choice can influence the order of convergence of the method. In fact, a fixed value of ϵ is not suitable to achieve the theoretical order of accuracy both on coarse and fine grids. Namely, a high value for ϵ may yield oscillatory reconstructions, while low ϵ values yields an estimated order of convergence that matches the theoretical one only asymptotically and often only for very fine grid spacings. Among the different solutions proposed in the literature, the mappings of [14,17] do not apply in a straightforward way to our compact WENO reconstruction technique, while taking an h-dependent ϵ as in [2] yields an improvement of the reconstruction, but we found experimentally that the scaling $\epsilon \propto h$ works better in our situation (see the numerical tests). Our choice is further supported by the work of Kolb [22] that analysed the optimal convergence rate of CWENO schemes depending on the choice of $\epsilon(h)$ on a uniform mesh. While on a uniform grid one can choose a constant value of ϵ that guarantees good numerical results, when using adaptive grids, with cell size h varying of several orders of magnitude, selecting the proper dependence $\epsilon(h)$ is of paramount importance. For a more detailed analysis of the non-uniform grid setting, see [11].

2.2 Higher Space Dimension

Although our reconstruction can be performed in any space dimension, we describe here for simplicity the two-dimensional case, giving hints for its generalisation. We start describing the structure of the adaptive grid.

2.2.1 Quad-Tree Grid in 2D

The adaptive grid is recursively generated starting by a coarse uniform Cartesian mesh of grid size H at level $l = 0$. Then each cell is possibly (according to some criterion) recursively subdivided into four equal squares. At the end of the recursive subdivision, the grid structure is described by a quad-tree.

Two examples of subdivision of an original cell with $L = 3$ levels of refinement are illustrated in Fig. 1, together with the corresponding quad-tree. The cell corresponding to the level $l = 0$ is the root of the quad-tree. Each cell of level $l \in \{1, \ldots, L\}$ has a father cell, which corresponds to its neighbour node in the quad-tree at level $l - 1$. The four nodes connected
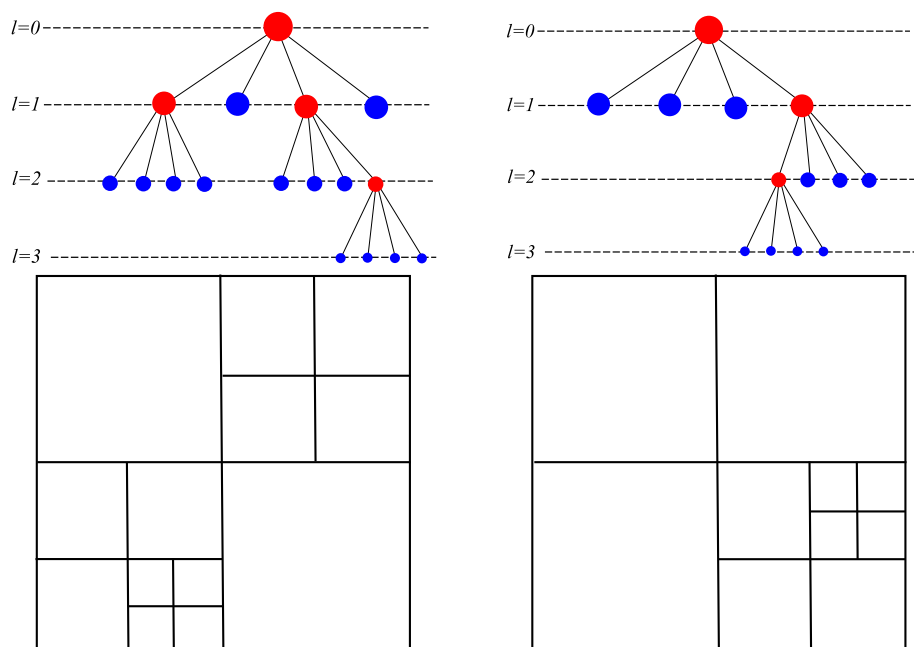
Fig. 1 Two examples (*left* and *right*) of subdivision of an original cell with $L = 3$ levels of refinement: quad tree (*up*) and grid (*down*)

to the father node are called the children of the node. The children of each subdivided cell are given some prescribed ordering (e.g. counter-clockwise starting from the upper-right as in the figure).

2.2.2 CWENO Reconstruction in 2D Adaptive Grid

In two dimensions the extension of the reconstruction described in [23] to the case of adaptive grids is not straightforward, so we describe it here in some detail. First, we define the set $\mathcal{N}_j$ of neighbours of Ω_j as (see two examples in Fig. 2):

$$\mathcal{N}_j = \left\{ k \neq j : \ \partial\Omega_k \cap \partial\Omega_j \neq \emptyset \right\}. \tag{5}$$

The optimal polynomial P_{OPT} is chosen among the quadratic polynomials matching the cell average U_j:

$$
\begin{aligned}
P_{\text{OPT}} = U_j &+ p_x(x - x_j) + p_y(y - y_j) \\
&+ \frac{1}{2} p_{xx} \left((x - x_j)^2 - \frac{h_j^2}{12} \right) + \frac{1}{2} p_{yy} \left((y - y_j)^2 - \frac{h_j^2}{12} \right) + p_{xy}(x - x_j)(y - y_j).
\end{aligned}
\tag{6}
$$

The coefficients p_x, p_y, p_{xx}, p_{xy} and p_{yy} are determined imposing that P_{OPT} fits the cell averages of the cells in $\mathcal{N}_j$ in a least-square sense. This can be realised if $|\mathcal{N}_j| \geq 5$. Such condition is always satisfied in a quad tree mesh. In fact, cell Ω_j has at least 3 neighbours among the children of its father cell and two edges on the boundary of the father cell and thus
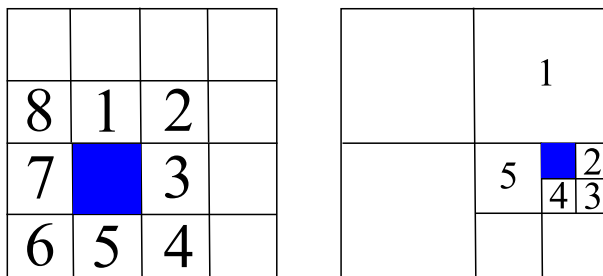
Fig. 2 Two particular configurations of the grid around a cell Ω_j. Cell Ω_j is *colour-filled*, while the neighbours are *numbered*. In these examples cell Ω_j has eight (*left* uniform grid) or five (*right* adaptive grid) neighbours, the latter being the case with the minimum number of neighbours (Color figure online)

at least two more neighbours outside the father cell (see Fig. 2, in particular the figure on the right for the case in which the cardinality of $\mathcal{N}_j$ is as minimum as possible). This result can be generalised in d dimensions, where a cell has at least $2^d - 1$ neighbours among its brother cells and shares at least d neighbours with its father cell, making the minimum number of neighbours equal to $2^d + d - 1$, while the unknown coefficients of a second degree polynomial in d variables matching the cell average U_j are $\binom{2+d}{d} - 1$. It can be easily proved that $2^d + d - 1 \geq \binom{2+d}{d} - 1$ for all $d \geq 1$.

The equations to be satisfied in a least-square sense are:

$$\frac{1}{h_k^2} \int_{\Omega_k} P_{\text{OPT}} \, dx = U_k, \quad k \in \mathcal{N}_j. \tag{7}$$

From (6) and (7) we obtain the possibly over-determined system $A\mathbf{c} = \mathbf{r}$, where $c = [p_x, p_y, p_{xx}, p_{xy}, p_{yy}]^T$, $r_k = U_k - U_j$ denotes the component of $\mathbf{r}$ corresponding to neighbour k, while A is a $|\mathcal{N}_j| \times 5$ matrix whose row corresponding to neighbour k is:

$$\begin{pmatrix} x_k - x_j \\ y_k - y_j \\ \frac{1}{2}\left((x_k - x_j)^2 + \frac{1}{12}(h_k^2 - h_j^2)\right) \\ (x_k - x_j)(y_k - y_j) \\ \frac{1}{2}\left((y_k - y_j)^2 + \frac{1}{12}(h_k^2 - h_j^2)\right) \end{pmatrix}^T$$

We note in passing that the idea of employing least squares fitting of the data in the same context has been exploited, in different fashions, at least in [1,15,24].

Following the same procedure as in the one dimensional case, we introduce four linear functions P_γ, $\gamma = 1, \ldots, 4$ in such a way that P_γ matches the cell average U_j exactly

$$P_\gamma = U_j + p_x^\gamma (x - x_j) + p_y^\gamma (y - y_j) \tag{8}$$

and the cell averages of a suitable subset of $\mathcal{N}_j$ in a least-square sense. In practice, we choose these four subsets (stencils) along the direction of each of the four vertices of the central cell. Let us start describing the choice of these stencils made in [23] for the uniform grid case. We introduce the following sets:

$$\mathcal{N}_j^E = \{k \in \mathcal{N}_j \colon \ x_k \geq x_j\}, \quad \mathcal{N}_j^W = \{k \in \mathcal{N}_j \colon \ x_k \leq x_j\},$$
$$\mathcal{N}_j^N = \{k \in \mathcal{N}_j \colon \ y_k \geq y_j\}, \quad \mathcal{N}_j^S = \{k \in \mathcal{N}_j \colon \ y_k \leq y_j\}. \tag{9}$$

and define the four stencils:

$$\mathcal{N}_j^{\alpha\beta} = \mathcal{N}_j^{\alpha} \cap \mathcal{N}_j^{\beta}, \quad \text{with} \ \ \alpha \in \{N, S\}, \ \beta \in \{W, E\}. \tag{10}$$

Referring to the left grid of Fig. 2, sets (9) are:

$$\mathcal{N}_j^E = \{\Omega_k \colon \ k = 1, 2, 3, 4, 5\}, \quad \mathcal{N}_j^W = \{\Omega_k \colon \ k = 1, 5, 6, 7, 8\},$$
$$\mathcal{N}_j^N = \{\Omega_k \colon \ k = 1, 2, 3, 7, 8\}, \quad \mathcal{N}_j^S = \{\Omega_k \colon \ k = 3, 4, 5, 6, 7\},$$

and stencils (10) are:

$$\mathcal{N}_j^{NE} = \{\Omega_k \colon \ k = 1, 2, 3\}, \quad \mathcal{N}_j^{NW} = \{\Omega_k \colon \ k = 1, 7, 8\},$$
$$\mathcal{N}_j^{SE} = \{\Omega_k \colon \ k = 3, 4, 5\}, \quad \mathcal{N}_j^{SW} = \{\Omega_k \colon \ k = 5, 6, 7\}.$$

Observe that if we apply the same procedure in the case of adaptive grids, we could have some stencil with only one cell. In fact, referring to the adaptive structure of the right grid of Fig. 2, we would have:

$$\mathcal{N}_j^E = \{\Omega_k \colon \ k = 2, 3, 4\}, \quad \mathcal{N}_j^N = \{\Omega_k \colon \ k = 1, 2\} \Longrightarrow \mathcal{N}_j^{NE} = \{\Omega_k \colon \ k = 2\}.$$

Since having only one cell in a stencil is not enough to determine a linear function (8), we slightly modify the definition of the four sets (9) (in such a way we include cell Ω_1 in the $\mathcal{N}_j^E$ stencil of the right grid of Fig. 2). The new definition is:

$$\mathcal{N}_j^E = \left\{k \in \mathcal{N}_j \colon \ x_k + \frac{h_k}{2} \geq x_j\right\}, \quad \mathcal{N}_j^W = \left\{k \in \mathcal{N}_j \colon \ x_k - \frac{h_k}{2} \leq x_j\right\},$$
$$\mathcal{N}_j^N = \left\{k \in \mathcal{N}_j \colon \ y_k + \frac{h_k}{2} \geq y_j\right\}, \quad \mathcal{N}_j^S = \left\{k \in \mathcal{N}_j \colon \ y_k - \frac{h_k}{2} \leq y_j\right\}.$$

and the four stencils are defined as in (10). Now, referring to the adaptive structure of the right grid of Fig. 2, we have:

$$\mathcal{N}_j^E = \{\Omega_k \colon \ k = 1, 2, 3, 4\}, \quad \mathcal{N}_j^N = \{\Omega_k \colon \ k = 1, 2, 5\} \Longrightarrow \mathcal{N}_j^{NE} = \{\Omega_k \colon \ k = 1, 2\}.$$

Let us rename the four stencils as $\mathcal{N}_j^{\gamma}$, $\gamma = 1, \ldots, 4$. The pairs of coefficients p_x^{γ} and p_y^{γ} of (8) are determined solving the system $A^{\gamma} c^{\gamma} = r^{\gamma}$ in a least-square sense, with $c^{\gamma} = [p_x^{\gamma}, p_y^{\gamma}]^T$, $r^{\gamma} = [U_k - U_j, \ k \in \mathcal{N}_j^{\gamma}]$, and A^{γ} is a $|\mathcal{N}_j^{\gamma}| \times 2$ matrix whose kth row is:

$$\begin{pmatrix} x_k - x_j \\ y_k - y_j \end{pmatrix}^T$$

we observe that also in this case the least-square problem is well-determined because $|\mathcal{N}_j^{\gamma}| \geq 2$, since we have at least one cell on the other side of each cell's edge (see Fig. 2).

Remark 1 we observe that in the uniform grid case the scheme does not reduce to the 2D CWENO described in [23]. In fact here each plane is obtained solving an over-determined system of four equations (in an adaptive grid $|\mathcal{N}_j^{\gamma}| + 1$ equations, but $|\mathcal{N}_j^{\gamma}| = 3$ in the uniform case) and three coefficients, while in [23] each plane was obtained by imposing only three conditions.

The parabola P_0 is determined by the relation:

$$P_{\text{OPT}} = \alpha_0 P_0 + \sum_{i=1}^{4} \alpha_\gamma P_\gamma. \tag{11}$$

By the same argument of the one dimensional case, the choice of coefficients α_γ is arbitrary. In practice we use $\alpha_0 = 1/2$ and $\alpha_\gamma = 1/8$, $\gamma = 1, \ldots, 4$. The final reconstruction is:

$$P = \tilde{\alpha}_0 P_0 + \sum_{\gamma=1}^{4} \tilde{\alpha}_i P_\gamma, \tag{12}$$

with

$$\tilde{\alpha}_\gamma = \frac{\omega_\gamma}{\sum_{\delta=0}^{4} \omega_\delta}, \quad \omega_\gamma = \frac{\alpha_\gamma}{(\epsilon + \beta_\gamma)^2}, \quad \gamma = 0, \ldots, 4.$$

The smoothness indicators, β_γ, are [23]:

$$\beta_\gamma = \sum_{|\alpha|=1}^{2} \int_{\Omega_j} h_j^{2(|\alpha|-1)} (P_\gamma^\alpha)^2 d\Omega, \quad \gamma = 0, \ldots, 4, \tag{13}$$

where $\alpha = (\alpha_x, \alpha_y)$ is a multi-index denoting the derivatives. Let us rewrite the polynomials $P_\gamma, \gamma = 0, \ldots, 4$ as

$$P_\gamma = U_j + p_x^\gamma (x - x_j) + p_y^\gamma (y - y_j) + \frac{1}{2} p_{xx}^\gamma \left((x - x_j)^2 - \frac{h_j^2}{12} \right)$$

$$+ \frac{1}{2} p_{yy}^\gamma \left((y - y_j)^2 - \frac{h_j^2}{12} \right) + p_{xy}^\gamma (x - x_j)(y - y_j).$$

A direct computation of (13) yields:

$$\beta_\gamma = h_j^2 \left(p_x^\gamma \right)^2 + h_j^2 \left(p_y^\gamma \right)^2 + \frac{13}{12} h_j^4 \left(p_{xx}^\gamma \right)^2 + \frac{7}{6} h_j^4 \left(p_{xy}^\gamma \right)^2 + \frac{13}{12} h_j^4 \left(p_{yy}^\gamma \right)^2.$$

Observe that $p_{xx}^\gamma = p_{xy}^\gamma = p_{yy}^\gamma = 0$ for $\gamma = 1, \ldots, 4$.

2.3 Treatment of the Boundaries

Boundary conditions are treated as follows. We create one layer of ghost cells around the computational domain. The size of each ghost cell matches the size of the adjacent cell in the domain Ω. The value of the cell average of the field variables in the ghost cell is determined by the boundary conditions (e.g. free flow BC, reflecting BC, and so on). Because of the compactness of the scheme, only one layer is necessary in order to perform the reconstruction in the physical cell near the boundary. The field variable on the outer side of the physical boundary, necessary to compute the numerical flux, is computed from the field variable on the inner side by applying the boundary conditions again. For example, in free flow boundary conditions the inner and outer values of the field variables are the same, while for reflecting BC in gas dynamics, density, pressure and tangential velocity on both side are equal, while the outer value of the normal velocity is the opposite of the inner one.
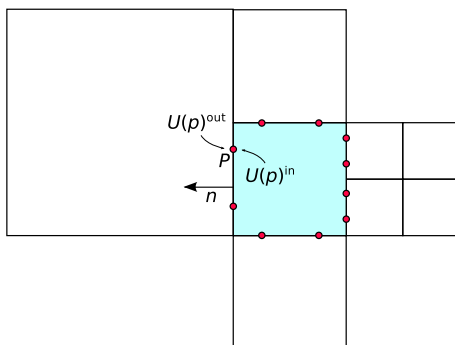
3 Fully Discrete Scheme and Adaptivity

In this section we describe the fully discrete scheme that we use to test the CWENO reconstruction, that is we specify the time integration procedure and the strategy for adaptive mesh refinement. First we introduce some notation for semidiscrete schemes.

Denote with U_j^n the approximate cell average of the solution at time t^n in cell Ω_j. Then a first order (in time) scheme for system (2) may be written as

$$U_j^{n+1} = U_j^n - \frac{\Delta t}{|\Omega_j|} \mathcal{Q}(\partial \Omega_j; \mathcal{F}),$$

where $\mathcal{Q}$ is a suitable quadrature formula and $\mathcal{F}$ are the numerical fluxes. Let $\mathcal{N}_j^A$ denote the set of *adjacent* neighbours of cell j (cells that share a segment with Ω_j). Then we split the boundary $\partial \Omega_j$ as

$$\partial \Omega_j = \bigcup_{i \in \mathcal{N}_j^A} \partial \Omega_j \cap \partial \Omega_i.$$

We choose a Gauss quadrature on each segment of this decomposition and compute the numerical fluxes at each quadrature node as $\mathcal{F}(f \cdot \mathbf{n}; U(x)^{\text{in}}, U(x)^{\text{out}})$, where $f : \mathbb{R}^d \to \mathbb{R}^m$ is the exact flux function, $U(x)^{\text{in}}$ is the reconstruction in cell j evaluated at x and $U(x)^{\text{out}}$ is obtained evaluating at x the reconstruction in the nearby cell. Each numerical flux computed on a quadrature node is used on the two adjacent cells it belongs, thus automatically guaranteeing conservation. Note that the concept of nearby cell is unambiguous if the quadrature nodes are not on the vertices of the cell, as is the case for Gauss-type formulas. An example is depicted in Fig. 3.

An important issue in the construction of an efficient adaptive method concerns the time stepping: because of hyperbolic CFL restriction, time step has to be of the order of the local mesh size, therefore one would have $\Delta t = O(H)$ in smooth regions, and $\Delta t = O(h)$ near singular regions. Time advancement procedures that can employ different time step length in different regions of the computational domain are called local timestepping schemes. Originally local time stepping methods have been developed on uniform grids, for problems with highly non uniform propagation speeds, in order to guarantee an optimal CFL condition on the whole domain (see [28]). Later they have been adopted to improve the efficiency of schemes based on non uniform grids. Different strategies have been adopted to implement local time stepping. In ClawPack, for example, the computational domain is discretized with a coarse uniform grid of mesh size H, which contains patches of more refined grid where

necessary. At macroscopic time step $\Delta t = O(H)$ the solution is updated on the coarse grid, and refined on the finer grid and on the coarse cells adjacent to the fine grid region, making sure that conservation is guaranteed (see [7] for details). More direct second order AMR methods with adaptive time step have been adopted, see for example [10,21], where a detailed analysis is performed of second order local time step methods, [30] for an application of local timestepping in a very similar setting, or [25], where an asynchronous time step strategy is adopted, in which the cell with the smallest time is the one that is advanced first. In the present paper we shall not discuss local time step, since the main point of the paper is to extend and analyse Compact WENO discretization to adaptive grids. Local time stepping for third order schemes is non-trivial, and will be the subject of a future paper. The time step Δt will thus be chosen in order to satisfy the CFL condition everywhere in the domain of the PDE. Thus our fully discrete scheme will be written as

$$U_j^{n+1} = U_j^n - \frac{\Delta t}{|\Omega_j|} \sum_{i=0}^{\sigma} b_i \mathcal{Q}\left(\partial\Omega_j; \mathcal{F}^{(i)}\right) \tag{14}$$

where the stage fluxes $\mathcal{F}^{(i)}$ are computed by applying the same formula as above to the stage values of the explicit Runge–Kutta scheme

$$U_j^{(i)} = U_j^n - \frac{\Delta t}{|\Omega_j|} \sum_{k=0}^{i-1} a_{ik} \mathcal{Q}\left(\partial\Omega_j; \mathcal{F}^{(k)}\right).$$

Here (a_{ij}, b_i) denote the coefficient of the Butcher tableaux of a Strong Stability Preserving Runge–Kutta scheme with σ stages (see [16]). In all the tests of this paper we employ the Local Lax–Friedrichs numerical fluxes.

3.1 Error Estimators/Indicators

Several techniques can be adopted to decide where to refine or derefine locally the mesh. Most of them are based on the use of local error indicators, such as, for example, discrete gradients and discrete curvature [3], interpolation error [7], residuals of the numerical solution [19] or of the entropy [27,29]. In the present paper we employ the numerical entropy production, that was introduced in [29] for central schemes and later extended in [30] to unstaggered finite volume schemes of arbitrary order. The motivation for this choice is that the numerical entropy production is naturally available for any system of conservation laws with an entropy inequality, it scales as the truncation error in the regular regions, and its behaviour allows to distinguish between contact discontinuities and shocks. In order to construct the indicator, one considers an entropy pair (η, ψ) and chooses a numerical entropy flux Ψ compatible with the exact entropy flux ψ in the usual sense that $\Psi(u, u) = \psi(u)$ and Ψ is at least Lipshitz-continuous in each entry. Then one forms the quantity

$$S_j^n = \frac{1}{\Delta t}\left[\langle \eta(U^{n+1}) \rangle_j - \langle \eta(U^n) \rangle_j + \frac{\Delta t}{|\Omega_j|} \sum_{i=0}^{\sigma} b_i \mathcal{Q}\left(\partial\Omega_j; \Psi^{(i)}\right) \right] \tag{15}$$

where we have denoted with $\langle \cdot \rangle_j$ the operation of averaging on the cell Ω_j and $\Psi^{(i)}$ denote the numerical entropy fluxes computed using the reconstruction of the i-th stage value.

In one spatial dimension, [30] shows that, if the solution is locally smooth, $S_j^n = O(h^r)$ with r equal to the minimum between the order of the scheme and the order of the quadrature formulas used to compute the cell averages of the entropy; on the other hand, $S_j^n = O(h)$ (resp. $O(1/h)$) if there is a contact discontinuity (resp. shock) in Ω_j. Of course, for a second
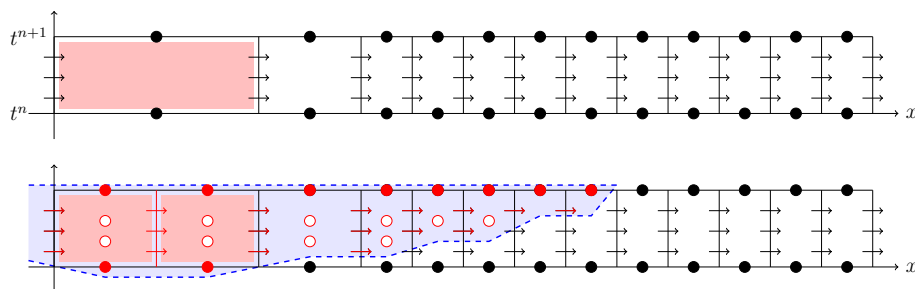
Fig. 4 Recomputation after refinement. The *top row* depicts the tentative computation of the time advancement after which the *shaded cell* is marked for refinement. *Circles* represents the cell averages and the *arrows* the numerical fluxes computed during the three Runge–Kutta stages. The *bottom row* depicts the recomputation performed after splitting the shaded cell. The *dashed line* indicates the numerical domain of dependence originating from the split cell. The stage values of the Runge–Kutta scheme that are recomputed are indicated by *empty circles*. The other stage values are not indicated for easier reading

order scheme it is enough to employ the midpoint rule $\langle \eta(u^n) \rangle_j = \eta(U_j^n)$ as in [30], whereas in order to observe a third order scaling of S_j^n as $h \to 0$, one has to employ a quadrature formula of higher order and thus evaluate the reconstructions of u^n and u^{n+1} at the quadrature points. Since the CWENO reconstruction yields a polynomial with uniform accuracy in the whole Ω_j, this entails simply an evaluation of the already computed polynomial and does not involve other reconstruction steps or extra sets of weights. Of course the reconstruction of u^n is already available and the reconstruction of u^{n+1} would be computed in any case at the beginning of the next time step.

Following ideas from [30] and extending them to the case of d space dimensions, we construct an adaptive mesh refinement scheme as follows:

– we start from a uniform coarse grid with N_0 cells, called the level-0 grid.
– any single cell of the grid may be replaced by 2^d equal cells and this operation may be performed recursively, obtaining a computational grid that is conveniently stored in a binary, quad- or oct-tree depending on d. Note that these grids will have in general hanging nodes, but this poses no problem to finite-volume discretizations. Let's call *level* of a cell its depth in the tree; obviously a cell of level l has diameter 2^l times smaller than its level-0 ancestor.
– at the end of each timestep, the quantity S_j^n is computed in every cell. If it is bigger than a threshold S_{ref} and if the level of refinement of Ω_j does not equal the maximum refinement level allowed in the grid, the cell is refined. The cell averages in the newly created cells are set by averaging the reconstruction of $\{U_i^n\}$[1] and the timestep recomputed locally, i.e. U_i^{n+1} is recomputed only in the numerical domain of dependence of the cell Ω_j (see Fig. 4).
– the solution $\{U_j^{n+1}\}$ is accepted when no refinement is required nor possible by the conditions on the size of S_j^n and on the level of the cells. Note that since S_j^n diverges on shocks when the grid is refined, fixing a maximum refinement level is necessary.
– a coarsening pass checks if all 2^d direct children of a previously refined cell have an entropy production lower than a given coarsening threshold, i.e. $S_j^n < S_{\text{coa}}$ and if so, replaces the 2^d children with their ancestor cell, where it sets U_j^{n+1} equal to the average

---

[1] As an exception, during the first time step, if the initial condition is known analytically, it is more accurate to use the analytic expression to set the cell averages in the newly created cells.

of the cell averages in the children. As in [30] we employ $S_{coa} = S_{ref}/2^{p+1}$. At this point a new timestep starts.

Our code makes use of the DUNE interface [4–6] to achieve grid-independent and dimension-independent coding of the numerical scheme in C++. Such interface is able to adopt several kinds of grid packages, including the ALUGRID library [8], which is the one that has been adopted in the two dimensional simulations. More precisely, a *dune-module* called `dune-fv` was written by the first author to provide generic interfaces to explicit Runge–Kutta, numerical fluxes, adaptive strategy, reconstructions, as well as the implementation of the classical schemes; the CWENO reconstruction was incorporated in `dune-fv` by the second author. The source code, licensed under GPL terms, allows to build adaptive mesh refinement schemes in one or two space dimensions, with spatial and temporal order up to three. Most components are easily interchangeable and the source allows for easy experimentation with different combinations of time-stepper, reconstructions, numerical fluxes, error indicator, etc [33].

4 Order of Accuracy

In this section we provide a scaling argument in support of the use of the third order scheme. We distinguish between the computation of regular solutions and piece-wise smooth solutions. In the case of regular solutions, the observed order of accuracy of the method should be equal to the theoretical one, giving an error that scales like $N^{-r/d}$, where N is the total number of cells, r is the (space and temporal) order of the scheme and d the number of space dimensions.

In this case, h-adaptivity helps by automatically refining in the regions of smaller space scales (see e.g. the tests of Fig. 7 for one space dimension and Fig. 8 in two space dimensions). The ratio between the largest and the smallest mesh size depends on the ratio between the largest and the smallest macroscopic scale of the physical system. The CFL condition implies that the time discretization is bounded by the space mesh scale, which suggests that the optimal choice is obtained by using the same order of accuracy in both space and time.

The situation is very different when discontinuities are present. Let us discuss separately the cases of one and of higher spatial dimensions.

4.1 One Dimensional Case

Let us assume that we want to solve a problem whose solution is piece-wise smooth. For simplicity, we assume that the spatial and temporal order of accuracy are the same. More precisely, let us assume that we want to solve a generalised Riemann problem:

$$u_t + f(u)_x = 0,$$
$$u(x, 0) = u_0(x) + H(x)\delta u(x),$$

where u_0 and δu are smooth functions and H is the Heaviside function

$$H(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}.$$

For some time the solution of such problem will consist of a piece-wise smooth function, in which singularities are localised in a few points. For example, for Euler equations in gas dynamics we have a jump in all quantities at the shock, a jump in density at contact

discontinuity and a jump in the first space derivatives at the end of the rarefaction region induced by the initial discontinuity.

Let r denote the order of the scheme employed, H the mesh size of the grid that we would use on a smooth region, h the smallest mesh that we allow by refinement. During the evolution, the adaptive algorithm will refine around the singularity points. We can identify two regions: a smooth region, bounded away from singularities, and a singular region, whose size is $O(H)$ around the singularities. Assume that a non adaptive scheme resolves each singularity in ν grid points.[2] Then the size of the singular region is $\mathcal{L}_s = \nu H$. Near the singularity the method reduces to first order. If we want that the lack of accuracy caused by the low order scheme does not pollute the solution in the smooth region, we have to require that $h = O(H^r)$. The motivation for this choice is the following. The solution in the regular region is affected by the presence of the singularity, with an error which is the maximum between the truncation error of size $O(H^r)$ due to the method and the error induced by the first order treatment of the singularity, which is $O(h)$. Therefore the error, even in smooth regions, is $\max(O(H^r), O(h))$, which is the motivation for suggesting the optimal choice $O(h) = O(H^r)$. In this way the overall accuracy of the solution in the regular regions should not be affected by the presence of the singularity, in the sense that the error introduced by the singularity is of the same order of the local truncation error and one expects that the error after a finite time is $O(H^r)$. This is true in local timestepping schemes where $\Delta t_{\max} = O(H)$ and even more so in the present case where we employ global time stepping, i.e. $\Delta t = O(h)$.

Let us check the dependence of the error on the total number of cells. Let $\mathcal{L}$ be the length of the computational domain. Then the number of cells in the large region is $N_R = \dfrac{\mathcal{L} - N_s \nu H}{H} = \dfrac{\mathcal{L}}{H} - N_s \nu$, where N_s denotes the number of singular points. The number of cells in the singular region depends on the number of refinement levels $L = \log_2 \dfrac{H}{h}$. In fact, if we assume that we have a shock located at a given point, we keep refining the cells until we reach the minimum cell size. From the numerical experiments (see Fig. 5 for the case of the Burgers equation after shock formation) it appears that the number of refined cells around one shock is approximately equal to L plus a number of smallest cells located at the shock. Such a number appears to be mildly increasing with L, so that the total number of cells refined due to the presence of a shock is much smaller than the total number of cells. Since $h = cH^r +$ h.o.t. for some constant c, we have:

$$L \simeq \log_2 \frac{H}{cH^r} = \log_2 \frac{1}{c} - (r-1)\log_2 H.$$

The number of cells in the singular region N_I is therefore negligible with respect to the number of cells in the smooth region, $N_I = O(L) \ll N_R$. Therefore, the total number of cells $N = N_R + N_I \approx \mathcal{L}/H$. As a consequence, the loglog plot of Errors vs. N should show the classical slope close to $-r$. Notice, however, the slight dependence of the number of levels on H. In practice, if we change H with $H/2$, in order to obtain an error decay rate of r, we also have to increase the number of levels by an amount $r - 1$. This is because the error in the smooth region becomes 2^r times smaller, so h has to be divided by 2^r (Table 1).

4.2 Higher Dimensional Case

The situation is different in more space dimensions. Let us first analyse in detail the two dimensional case, which is the one relevant to the numerical tests of this paper. Let us assume

---

[2] For simplicity we assume ν is independent on the nature of the singularity, which of course is not true in general.
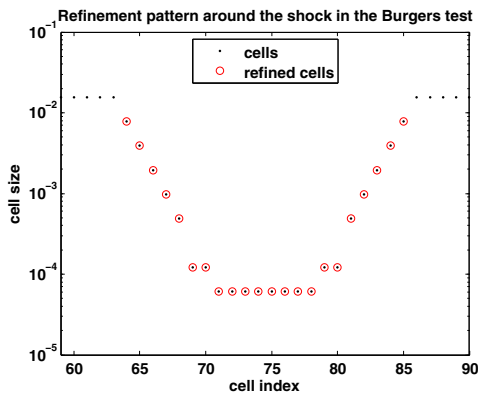
Fig. 5 Refined cells around the shock at $x = 0$ for the Burgers test with standing shock (at the final time of the simulation). Cell size versus index when using $L = 9$ refinement levels (*left*); number of cells involved in refinement versus refinement levels (*right*)

Table 1 How the maximum number of levels has to change when we increase the resolution on the coarse grid by a factor 2

Largest ΔX	Smallest Δx	No. of levels
$H_1 = H$	$h_1 = h$	$L_1 = \log_2(H/h)$
$H_2 = H/2$	$h_2 = h/2^r$	$L_2 = \log_2(2^{r-1}H/h) = (r-1) + L_1$

that the singularities are concentrated on a line that evolves (see e.g. Fig. 12). Let N denote the number of cells per direction on the coarsest grid. Without refinement there would be a total of $N_{\text{TOT}} = N^2$ cells. Assume now that we adopt a grid refinement until the finest cell becomes of size h. The cell size on the coarsest grid is $H = \mathcal{L}/N$. Since near the discontinuity the accuracy degrades to first order, it is natural to choose $h = cH^r$ for some constant c. The number of cells of size h^2 is proportional to the length of the discontinuity line divided by h, so $N_I = O(1/h)$, while $N_R = O(1/H^2)$. The total number of cells, is thus $N_{\text{TOT}} \approx C_I/h + C_R/H^2$, for some constants C_I and C_R. Hence $N_{\text{TOT}} \approx C_I/(cH^p) + C_R/H^2$ and we have the following scaling:

$r < 2$: most cells lie in the regular region;
$r = 2$: the number of cells in the regular and in the singular region have the same scaling;
$r > 2$: most cells are near the singular line.

This argument suggests that, in the presence of discontinuities, it is not possible to observe an asymptotic behaviour of the error better than $O(1/N_{\text{TOT}})$ even using higher order reconstruction in the smooth region. However, although the asymptotic behaviour is the same, the use of a higher order reconstruction may produce, as we shall see, a considerably smaller error for the same coarsest mesh, or, alternatively, the same error can be obtained with a considerably smaller number of cells. That this is the case in practical tests can be appreciated in Fig. 8 for a smooth test and in Figs. 12 and 13 in the presence of shocks.

In three dimensions the singularities are in general concentrated on a two dimensional manifold. Therefore, even with a second order accurate method most of the cells will be of size h, since $N_{\text{TOT}} \approx \dfrac{C_I}{h^2} + \dfrac{C_R}{H^3} = \dfrac{C_I}{cH^{2r}} + \dfrac{C_R}{H^3}$. Therefore if $r = 1$ most cells will be in the regular region, while with $r = 2$ the majority of cells will be near the singular surface.

Table 2 Reconstruction errors for the smooth function u_s on a uniform grid

Smooth test, $\epsilon = h_j$					Smooth test, $\epsilon = 10^{-6}$				
N	$\|E\|_1$	Rate	$\|E\|_\infty$	Rate	N	$\|E\|_1$	Rate	$\|E\|_\infty$	Rate
20	5.63e−03		2.17e−02		20	1.30e−02		4.17e−02	
40	5.04e−04	3.48	1.49e−03	3.87	40	2.26e−03	2.53	1.02e−02	2.03
80	4.97e−05	3.34	1.20e−04	3.64	80	3.55e−04	2.67	2.52e−03	2.02
160	5.40e−06	3.20	1.32e−05	3.18	160	4.95e−05	2.84	5.89e−04	2.10
320	6.22e−07	3.12	1.65e−06	3.00	320	4.96e−06	3.32	5.44e−05	3.44
640	7.43e−08	3.07	2.06e−07	3.00	640	4.36e−07	3.51	2.25e−06	4.60
1280	9.07e−09	3.03	2.57e−08	3.00	1280	3.91e−08	3.48	1.31e−07	4.10
2560	1.12e−09	3.02	3.22e−09	3.00	2560	3.27e−09	3.58	8.95e−09	3.87

Table 3 Reconstruction errors for the discontinuous function u_d on a uniform grid

Discontinuous test, $\epsilon = h_j$					Discontinuous test, $\epsilon = 10^{-6}$				
N	$\|E\|_1$	Rate	$\|E\|_\infty$	Rate	N	$\|E\|_1$	Rate	$\|E\|_\infty$	Rate
20	5.14e−03		7.51e−02		20	5.42e−03		1.02e−01	
40	2.70e−03	0.93	7.98e−02	−0.09	40	2.56e−03	1.08	1.00e−01	0.02
80	9.69e−04	1.48	4.74e−02	0.75	80	7.29e−04	1.81	5.78e−02	0.80
160	4.51e−04	1.11	4.99e−02	−0.08	160	3.61e−04	1.01	5.77e−02	0.00
320	2.08e−04	1.11	5.28e−02	−0.08	320	1.80e−04	1.00	5.77e−02	0.00
640	4.37e−06	5.57	9.82e−04	5.75	640	6.42e−09	14.78	1.63e−06	15.11
1280	6.36e−07	2.78	2.84e−04	1.79	1280	6.91e−10	3.22	4.06e−07	2.00
2560	8.59e−08	2.89	7.64e−05	1.89	2560	8.12e−11	3.09	1.01e−07	2.01

The discontinuity is on a grid interface from N = 640 onward

5 Numerical Tests

In the first set of tests we want to assess the convergence order of the CWENO reconstruction and of the fully discrete numerical scheme on uniform and non-uniform grids. We focus mainly on the choice of the parameter ϵ and on its dependence on the local grid size h. At variance with the case of a uniform grid, in which one can use a constant value of ϵ, here it is crucial to incorporate its dependence on the local grid size, which may vary by orders of magnitude.

One-dimensional reconstruction tests We set up U_j with the cell averages of the smooth function $u_s(x) = \sin(2\pi x - \sin(2\pi x)/\pi)$ on the domain $[0, 1]$ and of the discontinuous function $u_d(x) = e^{-x^2} + 0.1H(x)$ on $[-0.5, 0.5]$, where $H(x)$ is the Heavyside function. Then boundary extrapolated data $U_{j+1/2}^\pm$ are computed with the reconstruction and compared with limits of the function u_s (resp. u_d) for $x \to x_{j+1/2}^\pm$.

Table 2 compares the choice $\epsilon = h$ and the classical choice $\epsilon = 10^{-6}$ in the case of the smooth function. It is clear that our choice yields lower errors at all grid resolutions and a much more regular convergence pattern. The results for $\epsilon = h^2$ (not reported) are in between the two.

Table 4 Reconstruction errors for the smooth function u_s on non-uniform grids

Smooth test, quasi-uniform grid					Smooth test, random grid				
N	$\|E\|_1$	Rate	$\|E\|_\infty$	Rate	N	$\|E\|_1$	Rate	$\|E\|_\infty$	Rate
20	5.63e−03		2.17e−02		20	5.36e−03		1.87e−02	
40	8.65e−04	2.70	3.85e−03	2.50	40	4.97e−04	3.43	1.35e−03	3.79
80	1.08e−04	3.01	5.26e−04	2.87	80	5.07e−05	3.29	1.35e−04	3.32
160	1.26e−05	3.10	5.36e−05	3.30	160	5.46e−06	3.22	1.67e−05	3.02
320	1.47e−06	3.09	7.00e−06	2.94	320	6.25e−07	3.13	2.08e−06	3.01
640	1.75e−07	3.08	8.85e−07	2.98	640	7.46e−08	3.07	2.64e−07	2.98
1280	2.11e−08	3.05	1.11e−07	3.00	1280	9.11e−09	3.03	3.15e−08	3.07
2560	2.59e−09	3.03	1.39e−08	3.00	2560	1.12e−09	3.02	4.19e−09	2.91

The quasi uniform grids have cell centres at $x_j = j/N + 0.1 \sin(20\pi j/N)/10$ and the random ones at $x_j = j/N + 0.25/N r_j$ where the r_j are uniformly distributed in $[-0.5, 0.5]$

Table 3 is about the same test for the discontinuous function. Here we can observe two very different regimes, depending on whether the discontinuity is located exactly at a grid interface or not. In our tests the discontinuity is located at $x = 1/320$, so that it is located exactly at a cell interface only from N=640 onward, while being inside a cell for coarser grids. The convergence rates are obviously degraded to 1 and 0 (resp. in the 1- and the ∞-norm) when the discontinuity is not located exactly at a cell interface. We notice that both choices for ϵ yield very similar errors in both norms, until $N = 320$. From $N = 640$ onward, since the discontinuity is located exactly at an interface, the errors improve greatly. In this regime, $\epsilon = 10^{-6}$ can exploit more the favourable situation and produce very small errors.

This situation is however quite rare in an evolutionary problem and thus we can conclude that the choice $\epsilon = h$ is a better one, on the grounds that it is more accurate for the smooth parts and comparable to $\epsilon = 10^{-6}$ around discontinuities. Finally, in Table 4, the CWENO reconstruction with $\epsilon = h_j$, i.e. the local mesh size, is tested on non-uniform grids, both of quasi-regular and random type (see the caption of Table 4 for the definition of these grids), showing remarkable robustness in the order of convergence and errors very close to those obtained on uniform grids, suggesting that the choice $\epsilon \propto h$ is a good one, and that the reconstruction algorithm works very well even on non-uniform grids.

Two-dimensional reconstruction tests We apply the reconstruction to the exact cell averages of the smooth function $u^{\text{exa}}(x, y) = \sin(\pi x) \cos(\pi y)$ on the unit square, with periodic boundary conditions. The error E is computed by comparing the reconstructed function u^{rec} with u^{exa} on a uniform grid of reference $\mathcal{G}_{\text{ref}}$, in both the 1-norm and the ∞-norm. The reconstruction u^{rec} is defined by $u^{\text{rec}}|_{\Omega_j} = P$, for $j = 1, \ldots, N$, with P being the reconstruction polynomial (3) or (12) on the cell Ω_j. Cell averages U_j are set using the exact computation of the integral, namely $U_j = \int_{\Omega_j} u^{\text{exa}} d\Omega$. In details:

$$\|E\|_q = \left(\frac{\sum_{G \in \mathcal{G}_{ref}} |u^{\text{rec}}(G) - u^{\text{exa}}(G)|^q}{|\mathcal{G}_{\text{ref}}|} \right)^{1/q}, \quad q = 1, \infty.$$

The reference grid $\mathcal{G}_{\text{ref}}$ is a uniform grid such that each cell is finer than the smallest cell allowed by the adaptive algorithm. Although it is sufficient to have an accurate reconstruction only on the boundary of each cell (since the reconstruction is needed to compute numerical

Table 5 Reconstruction errors for a smooth function on a uniform grid

Uniform grid, $\epsilon = 10^{-6}$					Uniform grid, $\epsilon = h_j$				
N	$\|E\|_1$	Rate	$\|E\|_\infty$	Rate	N	$\|E\|_1$	Rate	$\|E\|_\infty$	Rate
8^2	7.19e−02		3.11e−01		8^2	5.32e−02		1.90e−01	
16^2	1.36e−02	2.41	9.61e−02	1.69	16^2	8.51e−03	2.64	3.74e−02	2.34
32^2	2.16e−03	2.65	2.51e−02	1.94	32^2	9.56e−04	3.15	3.79e−03	3.30
64^2	2.72e−04	2.99	4.11e−03	2.61	64^2	9.45e−05	3.34	2.55e−04	3.89
128^2	2.68e−05	3.34	4.33e−04	3.25	128^2	1.09e−05	3.12	3.20e−05	3.00

The average order of accuracy is 2.84 for $\|E\|_1$ and 2.35 for $\|E\|_\infty$ with $\epsilon = 10^{-6}$, and 3.10 for $\|E\|_1$ and 3.23 for $\|E\|_\infty$ with $\epsilon = h$

Table 6 Reconstruction errors for a smooth function on an adaptive grid

Adaptive grid, $\epsilon = 10^{-6}$					Adaptive grid, $\epsilon = h_j$				
N_C	$\|E\|_1$	Rate	$\|E\|_\infty$	Rate	N_C	$\|E\|_1$	Rate	$\|E\|_\infty$	Rate
8^2	6.19e−02		3.11e−01		8^2	4.48e−02		1.90e−01	
16^2	1.11e−02	2.48	9.61e−02	1.69	16^2	7.00e−03	2.68	3.74e−02	2.34
32^2	1.75e−03	2.67	2.53e−02	1.93	32^2	8.16e−04	3.10	3.79e−03	3.30
64^2	2.33e−04	2.91	6.40e−03	1.98	64^2	8.91e−05	3.20	2.55e−04	3.89

The first column shows the number of cells on the coarsest grid. For each test we choose $L = 3$ levels of refinement. The average order of accuracy is 2.68 for $\|E\|_1$ and 1.87 for $\|E\|_\infty$ with $\epsilon = 10^{-6}$, and 3.00 for $\|E\|_1$ and 3.19 for $\|E\|_\infty$ with $\epsilon = h$

fluxes on quadrature points, see Fig. 3), by choosing this grid we show that the method indeed provides a uniform accuracy all over the domain. Table 5 reports the reconstruction errors observed on uniform grids. Next, an h-adapted grid with 3 levels of refinement was generated by recursively refining the cells of a $N_{START} \times N_{START}$ uniform grid where $\|P_{OPT}^1 - P_{OPT}\|_2^2 > 0.01 h_j^2$, with P_{OPT}^1 being the optimal first degree polynomial computed with the same least square procedure of P_{OPT}. We choose $N_{START} = 8$ and denote this grid as $\mathcal{G}_0$. From this grid we generate grids $\mathcal{G}_k$, for $k = 1, \ldots,$ by subdividing each cell of $\mathcal{G}_0$ into 4^k equal cells. For each grid $\mathcal{G}_k$, $k = 0, \ldots,$ we define $N_C = N_{START} \cdot 2^k = 2^{3+k}$. The result of the convergence test on non-uniform meshes is reported in Table 6. The two-dimensional results are in line with those obtained in one dimension, confirming the good quality of the choice $\epsilon \propto h$.

Convergence tests on adaptive grids: notation The following tests will compare the numerical schemes on uniform and adaptive grids. We will consider a second order scheme (minmod reconstructions, Heun timestepping, numerical entropy evaluated on the cell averages) and the third order scheme described in this paper (CWENO reconstruction, three-stage third order SSP-RK and numerical entropy indicator computed with Gauss quadrature with two points per direction). The order will be denoted in the legends with the letter r.

In all tests the numerical fluxes are the Local Lax Friedrichs ones, i.e.

$$F(u, v) = \frac{1}{2} \left[f(u) + f(v) - \alpha(u - v) \right],$$

where α is the largest eigenvalue among those of $f'(u)$ and $f'(v)$. The numerical entropy fluxes are chosen accordingly as

$$\Psi(u,v) = \frac{1}{2}\left[\psi(u) + \psi(v) - \alpha(\eta(u) - \eta(v))\right],$$

with the same value for α (see [30]). For the computation of the numerical entropy production we have employed $\eta(u) = u^2$ for scalar tests and the physical entropy for the gas-dynamics tests. The convergence tests are conducted by performing sequences of computations with a coarse grid of size $N_0 = M2^k$ for $k = 0, \ldots, K$ for some integer M. We say that a simulation employs ℓ cell levels if the scheme is allowed to chose among ℓ different cell sizes, i.e. $H, H/2, \ldots, H/2^{\ell-1}$, where H is the size of the uniform coarse mesh. Let $\ell(k)$ be the number of levels employed in the k-th test of a sequence. In the legends of the convergence graphs, the number of levels is indicated by an integer L if $\ell(k) = $ L, by L+ (respectively L++) if $\ell(k) = $ L $+ k$ (respectively $\ell(k) = $ L $+ 2k$). Obviously L $= 1$ corresponds to uniform grids. The refinement threshold is set to $S_{\text{ref}} = S_0 s^{-k}$ in the k-th computation for some scaling factor $s \geq 1$, where S_0 is the threshold used in the coarsest computation of the sequence. The choice of the parameter s should be guided by the features of the solution on which one wants h-adaptivity to act. Since the numerical entropy production scales as $O(h^r)$ on smooth flows (see Sect. 3), we conduct the tests using $s = 2^r$ if the solution is smooth everywhere and h-adaptivity is expected to distinguish between high and low frequency regions. On the other hand, the numerical entropy production scales as $O(1/h)$ on shocks, $O(1)$ on contacts and $O(h)$ on corner points and thus we take $s = 2$ on flows with singularities so that h-adaptivity acts on rarefaction corners and stronger discontinuities. These choices for s ensure that the level of refinement induced by the choice of S_0 on the first grid is replicated in the k-th test. Since in h-adaptive tests the number of cells is variable during the evolution, in the graphs for the convergence tests we employ the time-average of the number of cells employed in each timestep.

One-dimensional linear transport test (1) We evolve the initial data $u(x,0) = \sin(\pi x - \sin(\pi x)/\pi)$ on $[-1, 1]$ up to $t = 1.0$ with the linear transport equation $u_t + u_x = 0$ and periodic boundary conditions.

The purpose of this test is to ensure that, on a single scale smooth solution, with a proper choice of the parameter s, the behaviour of the adaptive scheme is similar to that of a uniform grid method.

In the set of runs, the number of grid refinement levels was kept at L $= 3$, while the coarse grid size and the refinement threshold have been diminished (see Fig. 6). The refinement threshold for the coarsest run in the series (S_0) is taken to be 10^{-1} for the second order scheme and 10^{-2} for the third order scheme. As it appears from the figure, the behaviour of the adaptive scheme with the choice $s = 2^r$ is similar to the one given by the uniform grid method. We observe a slight improvement of the adaptive scheme for $r = 3$, while for $r = 2$ there is really no advantage in using an adaptive mesh in such a simple test case.

One-dimensional linear transport test (2) Next, we want to test the situation in which the adaptive scheme may compute on a finer grid the rapidly varying regions of a smooth solution. To this end, we evolve the initial data $u(x,0) = \sin(\pi x) + \frac{1}{4}\sin(15\pi x)e^{-20x^2}$ on $[-1, 1]$ up to $t = 2.0$ with the linear transport equation $u_t + u_x = 0$ and periodic boundary conditions (see Fig. 7). The top graph shows an example of numerical solution obtained with the adaptive scheme ($N_0 = 64$, L $= 3$, $S_{\text{ref}} = 2 \times 10^{-5}$), while the middle one shows the cell levels in use at final time, revealing that small cells are selected in the central region where the
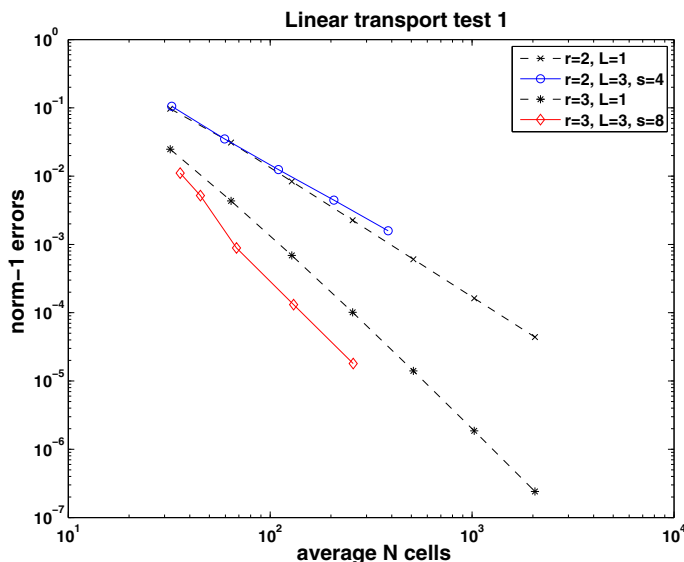
Fig. 6 Norm-1 errors for uniform and adaptive grids for the linear transport test (1). The number of levels is kept fixed, while the coarse cell size and the refinement threshold are progressively lowered. The *symbols in the legend* refer to the order of the scheme (r), the number of different cell sizes (L) and the scaling of the refinement threshold between runs (s)

solution contains higher frequencies. The dashed red line in the top graph is the error of the adaptive solution and should be read against the logarithmic scale on the right. It reveals that no spurious oscillations or other defects occur at grid discontinuities.

The results of the convergence tests are shown on the bottom of Fig. 7. First, note that the runs on uniform grids cannot resolve the high frequencies in the central area and thus they do not show convergence until $N_0 \geq 128$. For example, the convergence rates estimated from the errors shown in the figure for the third order scheme are 0.44, 0.01, 1.09, 2.61, 3.02 and 2.98.

For the adaptive grid tests we used $N_0 = 16 \cdot 2^k$ and $L = 4$, so that even in the coarsest run the scheme was able to use some cells of size $1/128$ -th of the domain. S_0 was set to 0.005, 0.001 for the schemes of order $r = 2, 3$ respectively. These values were chosen in such a way that, for $k = 0$, only cells in the region of the 4 central peaks ($[-0.2, 0.2]$ in the initial/final data shown in the top panel of the figure) had a chance to be refined to the 4-th level. With respect to the previous test, obviously here an adaptive scheme has much room for improvement over uniform grid ones and all tested sequences show improvements over the uniform case.

Since the solution is smooth everywhere, we expect that scaling the refinement threshold with $s = 8$ in the scheme with $r = 3$ would cause every run to use the same refinement pattern. As a matter of fact this choice leads to the best convergence rates (1.06, 2.62, 2.99, 2.97).

two-dimensional scalar test with smooth solution Next we consider a scalar problem in two space dimensions with a smooth solution which is known in closed form, originally presented in [12]. It is sometimes referred to as "atmospheric instability problem", as it mimics the mixing of a cold and hot front. The equation is

$$u_t + \nabla \cdot (v(x, y)u) = 0 \quad \text{on} \quad \Omega = [-4, 4]^2 \tag{16}$$

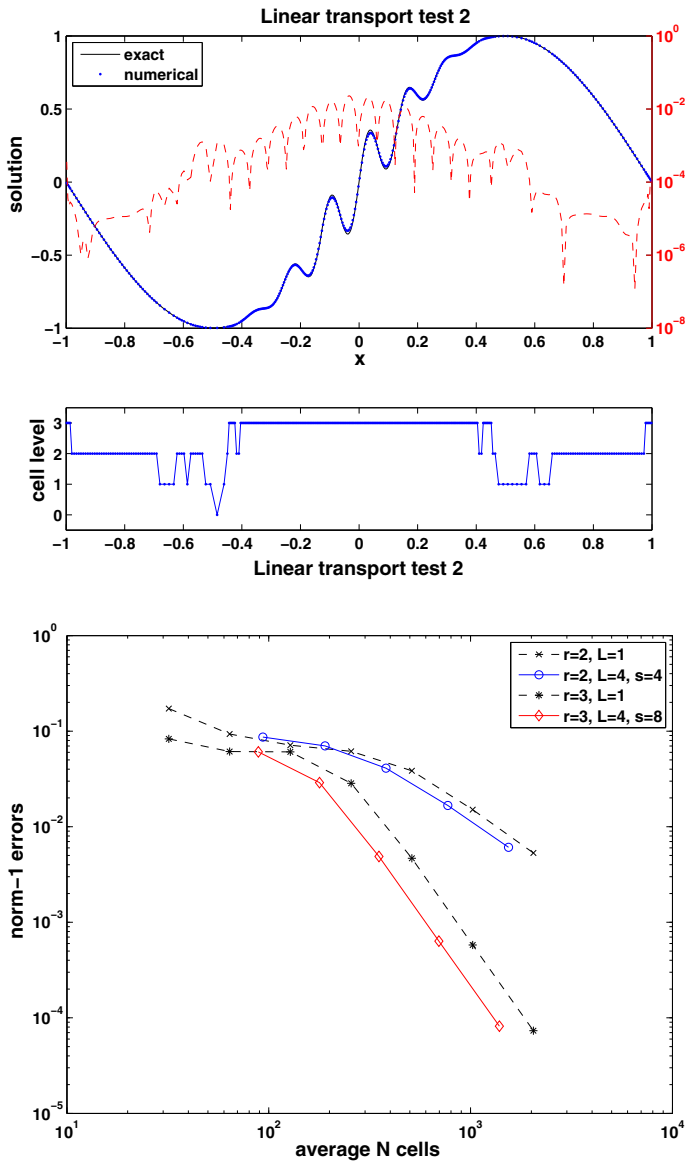with initial data $u_0(x, y) = -\tanh(y/2)$ and velocity

Fig. 7 Norm-1 errors for uniform and adaptive grids for the linear transport test (2). An example of a solution (*top*), with the grid levels employed (*middle*). In the *top panel* the *dashed red line* is the error of the numerical solution and should be read against the logarithmic scale on the right vertical axis. On the *bottom*, the convergence test (Color figure online)

$$v(x, y) = \left[-\frac{y}{r} \frac{f}{0.385}, \frac{x}{r} \frac{f}{0.385} \right],$$

where $r = \sqrt{x^2 + y^2}$ and $f = \tanh(r)/(\cosh(r))^2$. The exact solution at $t = 4$ is

$$u(4, x, y) = -\tanh\left(\frac{y}{2} \cos\left(\frac{4f}{0.385r} \right) - \frac{x}{2} \sin\left(\frac{4f}{0.385r} \right) \right).$$

Figure 8 shows the solution at final time and the results of the convergence test conducted with $M = 16$, $S_0 = 10^{-2}$ for both $s = 2, 3$. It is clear that third order schemes achieve far better efficiency, by computing solutions with lower errors and using fewer cells. Note that even the uniform grid third order scheme outperforms the second order adaptive scheme, which yields lower errors but also slightly lower convergence rates than the second order uniform grid scheme. The third order adaptive scheme shows a lower error constant than the corresponding uniform grid one.

The most striking difference between second and third order schemes, however, is the different refinement pattern employed. The bottom graphs of Fig. 8 show the meshes at final time for both schemes in the run with $N_0 = 64$, $L = 4$, $S_{\text{ref}} = 6 \times 10^{-4}$. Clearly the third order scheme is able to employ small cells in much narrower regions, thus leading to additional computational savings.

One dimensional Burgers equation The first tests involving shocks employ Burgers' equation

$$\partial_t u + \partial_x \left(\frac{1}{2} u^2 \right) = 0 \tag{17}$$

on the domain $[-1, 1]$ with periodic boundary conditions. The results in Fig. 9 are relative to the initial data $u_0(x) = -\sin(\pi x)$ and final time $t = 0.35$, that gives rise to a standing shock located at $x = 0$, while those of Fig. 10 refer to the initial condition $u_0(x) = -\sin(\pi x) + 0.2 \sin(5 \pi x)$, that gives rise to two moving shocks.

In the error plot of Fig. 9 first we note that the schemes with uniform meshes tend to become first order accurate, since most of the error is associated to the shock. For the adaptive runs we chose $M = 16$ and $S_0 = 10^{-2}$ for both $r = 2, 3$. Adaptive schemes with a fixed number of levels do not improve much with respect to the uniform mesh ones: they give a smaller error, asymptotically approaching first order convergence, although with an error 2^{L-1} times smaller than a uniform grid with the same number of cells. Allowing more levels when refining the coarse mesh yields an experimental order of convergence (EOC) of 2.0 (left). The third order scheme gives an EOC of 2.7 when increasing L by two every time the coarse mesh is refined (right). Note that the scaling factor for the refinement threshold can be taken as $s = 2$, since here one is interested in refining only in the presence of the shock. For comparison purposes, we also show the result of using $s = 4$ in the third order scheme, which gives a poorer work-precision performance due to excessive refinement.

Figure 10 refers to a more complex situation, since in this case the shocks are moving (and thus also coarsening comes into play) and there is a richer smooth structure away from the shocks. Here the advantage of third order schemes is more clear, even on uniform grids. This better performance is confirmed by the adaptive codes ($M = 16$, $S_0 = 0.5$ for the second order case and $S_0 = 0.1$ for the third order one) and the situation is quite similar to the smooth test cases, except that the best convergence rates are obtained, similarly to the previous test with a shock, with scaling factor $s = 2$ instead of 8 (the EOC of the adaptive schemes are 1.7 and 2.6).

One dimensional gas-dynamics: shock-acoustic interaction Next we consider the interaction of a shock wave with a standing acoustic wave from [36]. The conservation law is the one-dimensional Euler equations with $\gamma = 1.4$, the initial data is

$$(\rho, v, p) = \begin{cases} (3.857143, 2.629369, 10.333333), & x \in [0, 0.25] \\ (1.0 + 0.2 \sin(16 \pi x), 0.0, 1.0), & x \in (0.25, 1.0] \end{cases}$$
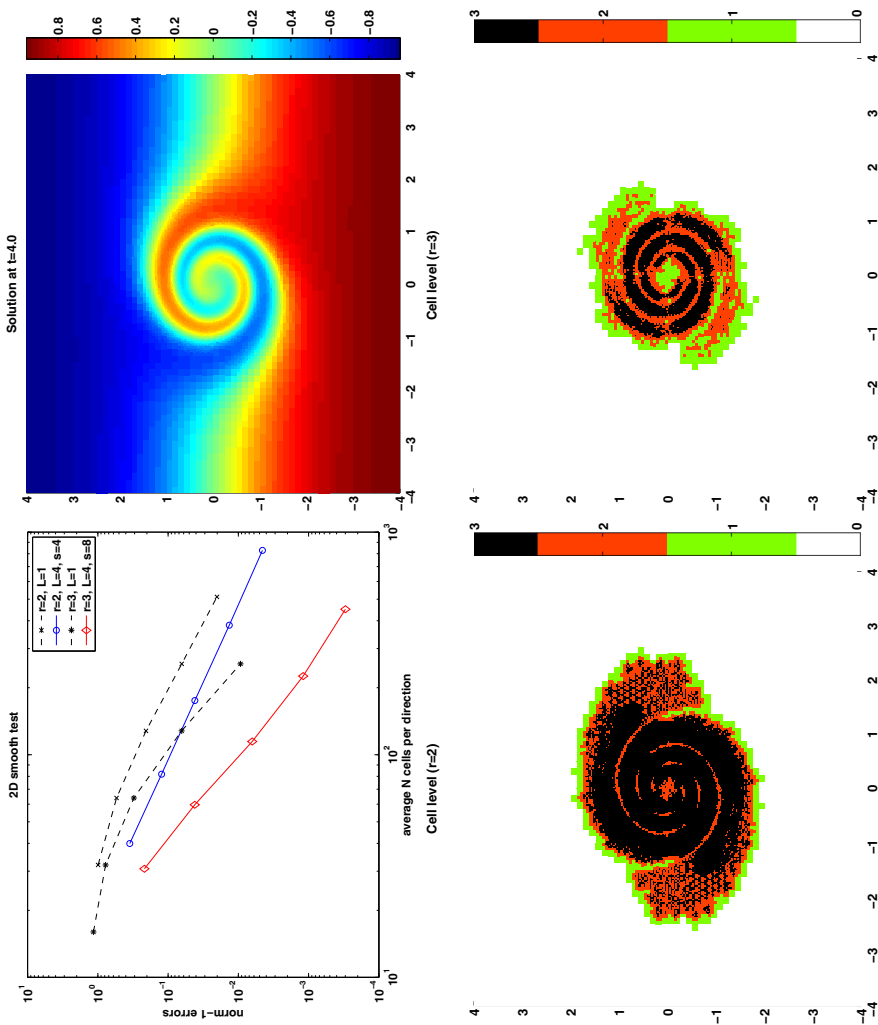
Fig. 8 Results of the 2d smooth test of Eq. (16). Norm-1 errors (*top-left*), solution at final time (*top-right*), grid refinement level at final time with the second and third order scheme with $N_0 = 64$, $L = 4$, $S_{ref} = 6 \times 10^{-4}$ (*bottom*) (Color figure online)
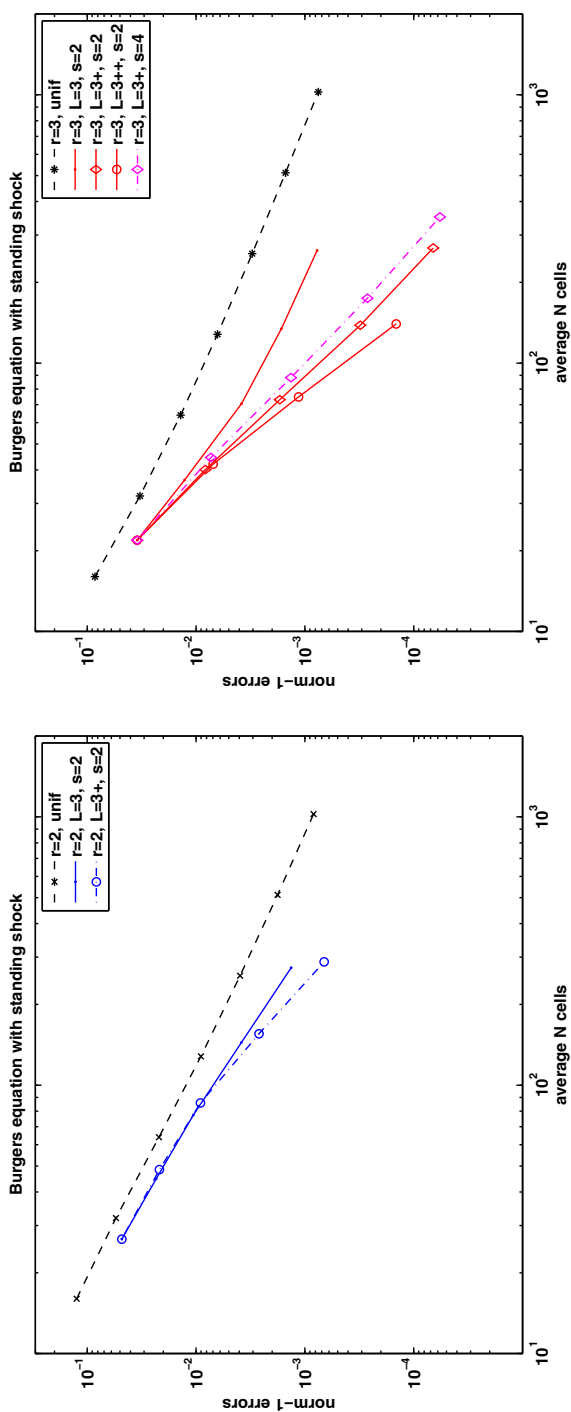
Fig. 9 Burgers equation after shock formation ($t = 0.35$): error versus number of cells. Second order (*left*) and third order (*right*) scheme. The errors were computed by comparing with the exact solution computed with characteristics
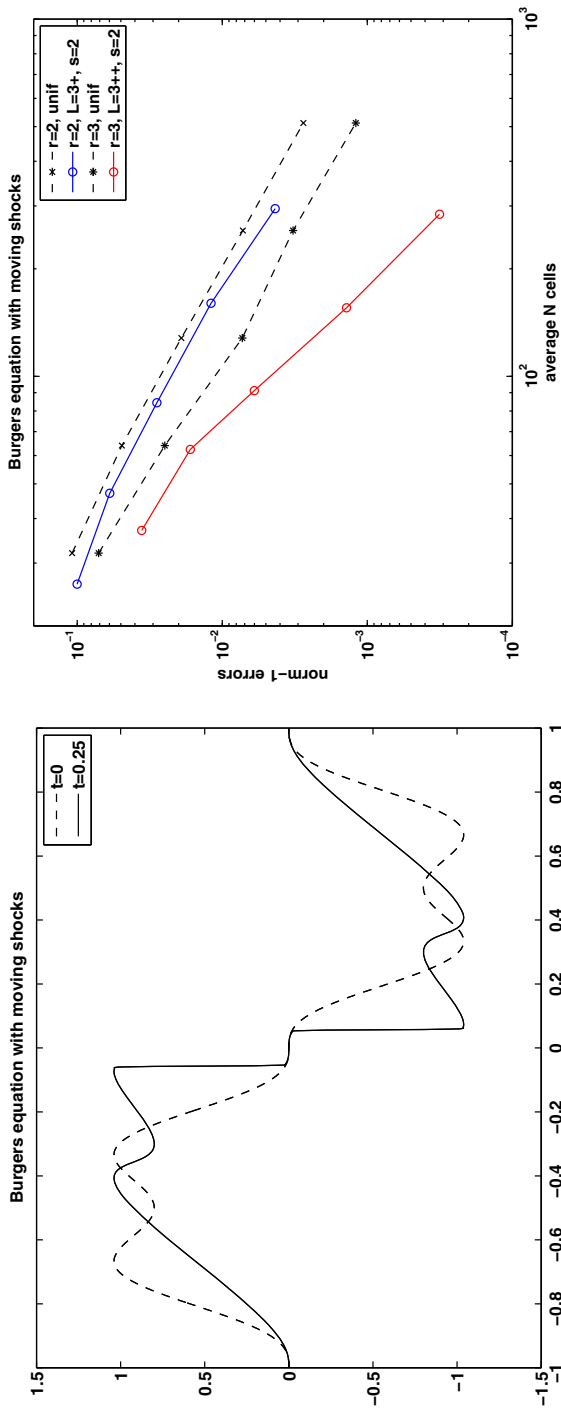
Fig. 10 Burgers equation with moving shocks: solution (*left*) and errors (*right*), obtained by comparing with a reference solution computed by the third order scheme on a uniform grid composed of 262144 cells

and the evolution was computed up to $t = 0.2$. As the right moving shock impinges in the stationary wave, a very complicated smooth structure emerges and then gives rise to small shocks and rarefactions (see Fig. 11).

The top-left panel of Fig. 11 shows the 1-norm of the errors versus the average number of cells used during the computation. Due to the high frequency of the waves behind the shock, uniform grid methods start showing some convergence only when using more than 1000 cells: before that they lack the spatial resolution required to resolve the smooth waves in the region $x \in [0.65, 0.85]$.

In the adaptive tests, the number of levels is 6 when the coarse grid has $N_0 = 32$ cells, so that even the smallest h-adaptive runs have the chance of using cells of size $1/1024$, and we took $S_0 = 10^{-1}$ for both $r = 2, 3$. As in the case of Burgers' equation, in order to observe high convergence rates, one has to increase the number of refinement levels each time that the coarse grid is doubled. Here we additionally observe that the scaling factor $s = 4$ gives better results than $s = 2$; although, given the presence of the shock, this is somewhat unexpected, it can be understood by looking at the refinement pattern in the top-right panel of the figure, which is representative of the grids employed in the h-adaptive tests. The smallest cells are used only at the shocks, but behind the main shock, there is a non-negligible smooth region where mid-sized cells are employed. These h-adapted grids are an hybrid situation between grids refined only at shocks (which would call for the choice $s = 2$) and grids refined in the smooth regions (which would favour the choice $s = 8$ in the case of the third order scheme). Despite the aforementioned difficulties in selecting optimal parameter values, we point out that the second order h-adaptive scheme gives convergence rates similar to those of the uniform grid case, but with lower error constants, while the third order also shows improved error decay rates: the EOC for the two segments shown in the figure for the run with L=6++, s=4 are 1.6 and 3.0.

In the lower part of Fig. 11 we compare the reference solution with those computed on uniform grids (left) and with the adaptive algorithm (right), restricted to the interval $x \in [0.2, 0.85]$. Solutions computed on uniform grids converge rather slowly, lacking resolution for both the frequency of the waves behind the shock and their amplitude. The adaptive solution captures perfectly the frequency of the small waves behind the shock and approximates reasonably well their amplitude even with 656 cells (on average during the time evolution), by making an effective use of adaptivity. In fact the cell sizes are distributed in the computational domain (top-right panel) by concentrating them around shocks and high frequency waves, while using larger cells in smooth regions, where the solution is efficiently resolved by the third order scheme.

2D Riemann problem For a two-dimensional test involving shocks, contact discontinuities and a smooth part, we consider the Euler equation of gas dynamics in two spatial dimensions and set up a Riemann problem in the unit square with initial data given by:

$$\begin{cases} \rho = 0.8, u = 0.1, v = 0.1, p = 1.0 & x < 0.5, y < 0.5 \\ \rho = 1.0222, u = -0.6179, v = 0.1, p = 1.0 & x < 0.5, y > 0.5 \\ \rho = 1.0, u = 0.1, v = 0.8276, p = 1.0 & x > 0.5, y < 0.5 \\ \rho = 0.531, u = 0.1, v = 0.1, p = 0.4 & x > 0.5, y > 0.5 \end{cases}$$

The solutions were compared to a reference one computed with the third order scheme on a uniform grid with 2048×2048 cells and the results are presented in Fig. 12, whose
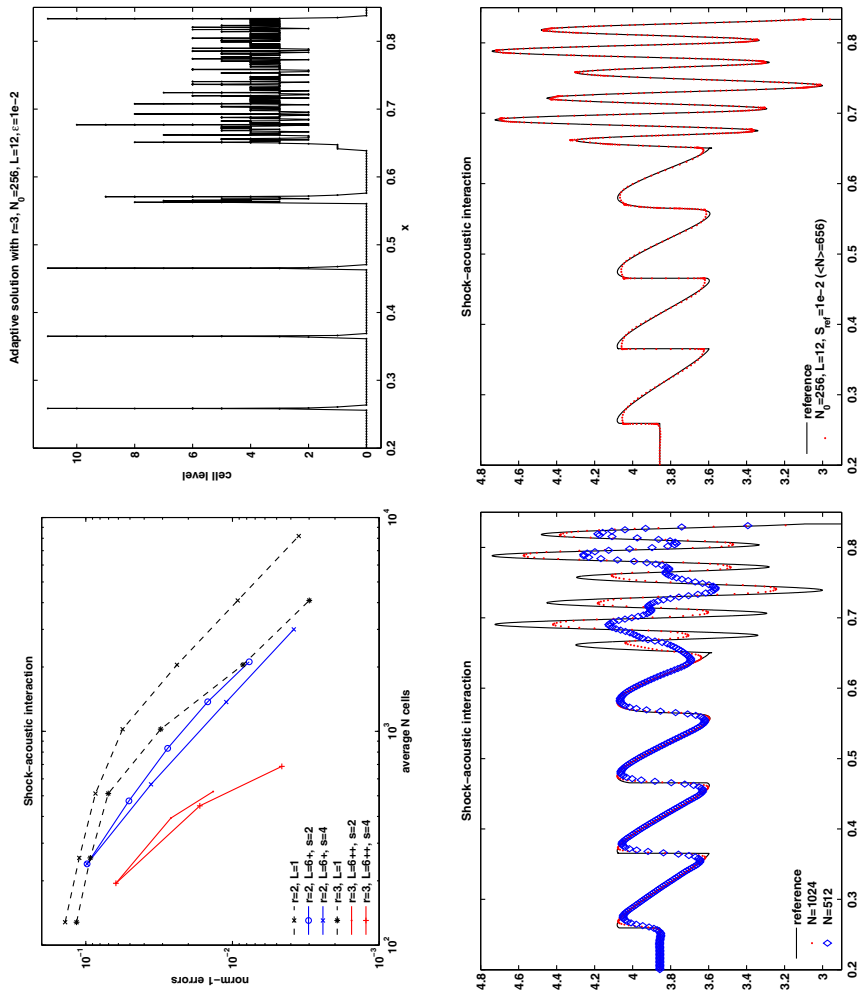
Fig. 11 Shock-acoustic interaction problem. (*Top-left*) Norm-1 errors versus average number of cells, using a reference solution computed by the third order scheme on a uniform mesh of 262144 cells. The solution on an h-adapted grid is shown with the cell levels (*top-right*) and with a zoom in the oscillatory region (*bottom-right*). For comparison we show two uniform grid solutions (*bottom-left*)
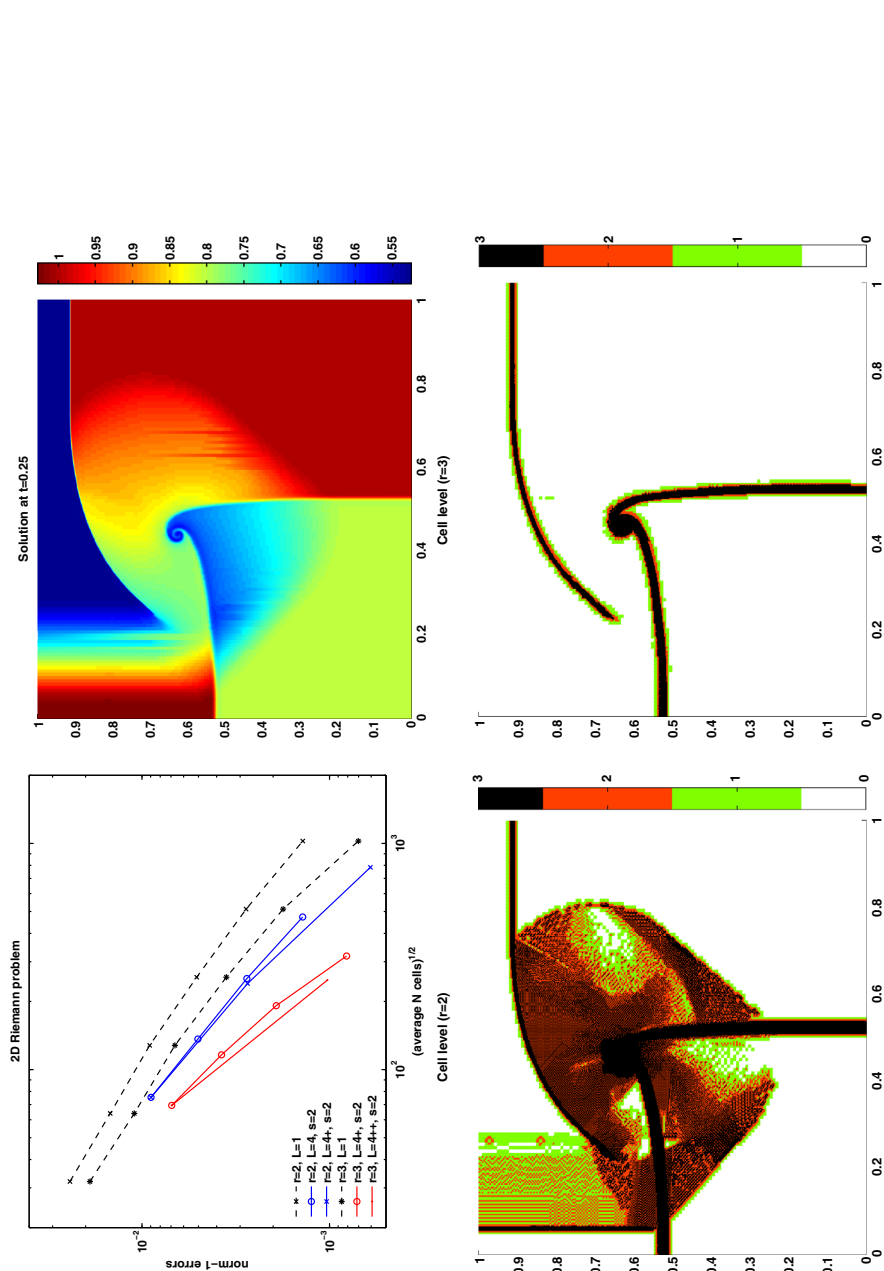
Fig. 12 Two dimensional Riemann problem for the Euler equations: errors (*top-left*), density at final time (*top-right*), grid refinement levels with $N_0 = 128$, $L = 4$, $S_{ref} = 2 \times 10^{-3}$ (*bottom*) (Color figure online)

top-right panel depicts the density at final time. In the error versus number of cells graph (top-left), we can observe that, due to the presence of the shocks, the performance of the two uniform grid schemes have little differences, with the third order one being char-acterised by approximately the same error decay rate of the second order scheme (1.0), only with a better error constant. The adaptive schemes ($M = 16$, $S_0 = 10^{-3}$) yield better results that the uniform grid ones, by exploiting small cells to reduce the error in the region close to the shocks, where the numerical scheme are obviously only first order accurate. The EOC are 1.0 and 1.2 for the `r=2`, `L=4+` series and 1.5 for the `r=3`, `L=4++` segment. We note however that the third order scheme is much more effec-tive also due to its ability to employ much coarser grids, as it is evident comparing the two computational grids in the lower panels of Fig. 12, that refer to the solutions com-puted with $N_0 = 128$, `L` $= 4$ and $S_{ref} = 2 \times 10^{-3}$. As a result the third order scheme can compute a solution with error in the 10^{-3} range with 6.18×10^4 cells, whereas the uniform grid one requires 1.05×10^6 and the second order adaptive scheme 6.14×10^5.

Shock–bubble interaction in 2D gas-dynamics For the final test we consider again the 2D Euler equations of gas dynamics and set up an initial datum with a right-moving shock that impinges on a standing bubble of gas at low pressure, as in [9]. In particular, the domain is $[-0.1, 1.6] \times [-0.5, 0.5]$ and in the initial datum we distinguish three areas: the left region (A) for $x < 0$, the bubble (B) of centre $(0.3, 0.0)$ and radius 0.2, the right region (C) of all points with $x > 0$ that are not inside the bubble. The initial datum is

$$\begin{cases} \rho = 3.6666666666666666, u = 2.7136021011998722, v = 0.0, p = 10.0 & (x, y) \in A \\ \rho = 0.1, u = v = 0.0, p = 1.0 & (x, y) \in B \\ \rho = 1.0, u = v = 0.0, p = 1.0 & (x, y) \in C \end{cases}$$

Boundary conditions are of Dirichlet type on the left, free-flow on the right, reflecting on $y = \pm 0.5$. Due to the symmetry in the y variable, the solution was computed on the half domain with $y > 0$, with reflecting boundary conditions on $y = 0$.

For this test we used coarse grids with multiples of 51×15 square cells. Figure 13 shows a comparison of the solution at $t = 0.4$ and the h-adapted computational grid at final time. Each panel depicts a solution in the whole domain, using data from the third order scheme in the upper half and data from the second order one in the lower half. Both solutions depicted in the figure have been computed with a coarse grid of 204×60 cells, `L` $= 4$ and $S_{ref} = 2.5 \times 10^{-2}$. The CPU times were comparable (18 h 44 m for $r = 2$ and 18 h 35 m for $r = 3$), but we can appreciate the much increased sharpness of the solution computed with the third order scheme: note for example the regions close to $y = 0.5$ and just behind the main shock, the region of the vortex and the areas just behind it. Also in this case, this better quality solution is obtained using a much coarser grids, i.e. using h-adaptivity in much smaller areas, quite concentrated around the prob-lematic features of the solution. In fact the computation with the second order scheme employed almost twice as much cells than the third order one: 141,539 versus 71,895 cells on average and 2,214,934 versus 123,114 at final time. Table 7 shows that the com-putation times compare favourably with the uniform grid schemes. Extra saving in CPU time may be achieved by implementing a local timestepping scheme for time advance-ment.
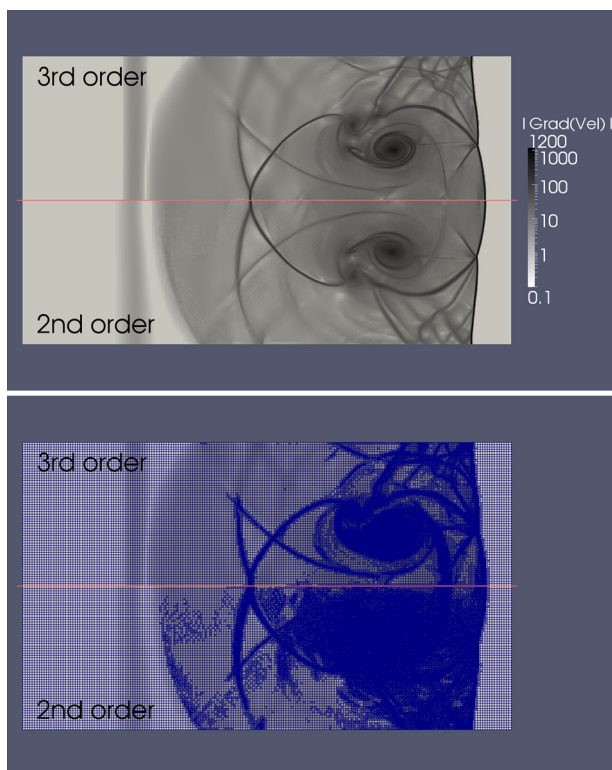
Fig. 13 Schlieren plots of the solutions (*top*) of the shock–bubble interaction problem computed with an adaptive grid with 4 levels from a coarse grid of 204 × 60 cells. The grids at final time are shown in the *bottom* graph. Since the solution is symmetric, each plot is split in two parts, with the third order solution in the *top half* and the second order one in the *bottom half*

Table 7 Comparison of norm-1 errors and CPU-times for the computation of the shock–bubble interaction problem

These tests were run on a workstation equipped with a 2.60 GHz Intel Xeon processor

Scheme	Norm-1 error	CPU time
$r = 2, L = 1, N_0 = 480 \times 1632$	2.97×10^{-2}	20 h 36′
$r = 3, L = 1, N_0 = 240 \times 816$	3.67×10^{-2}	19 h 12′
$r = 2, L = 4, N_0 = 60 \times 204$	2.97×10^{-2}	18 h 44′
$r = 3, L = 4, N_0 = 60 \times 204$	9.60×10^{-3}	18 h 35′

6 Conclusions

In this work we extended the Compact WENO reconstruction of [23] to the case of non-uniform meshes of quad-tree type and used it to build a third order accurate numerical scheme for conservation laws. Indications on how to extend the result to three-dimensional oct-tree grids are provided. Even in one space dimension, there are two advantages of the Compact WENO reconstruction over the standard WENO procedure. First, the reconstruction provides a uniformly-accurate reconstruction in the whole cell, which was exploited e.g. in [31] for the integration of source terms, and it is not obtained through dimensional splitting,

thus including also the xy terms. Secondly, the linear weights do not depend on the relative size of the neighbours and thus the generalisation to two-dimensional grids does not need to distinguish among very many cases of local grid geometry.

The simplicity in computing the reconstruction on non-uniform grids was exploited for coding an h-adaptive scheme that relies on the numerical entropy production as an error indicator. Our tests show that the third order scheme works better than second order ones, not only in the sense that a smaller error is produced in each cell, but also in the sense that, when employed in the h-adaptive scheme, it gives rise to much smaller meshes, that are refined only in a narrow region around shocks.

In order to evaluate the effectiveness of the adaptive schemes we presented some heuristics for the relation between error and number of cells in an h-adaptive scheme in the presence of isolated shocks. In one space dimension we can observe the predicted third order error decay rate, while in the two dimensional case we observe convergence rates slightly higher than those predicted by the heuristics.

Future work of this subject should certainly look at analysing more deeply the role of the parameter ϵ appearing in the nonlinear weights of the CWENO reconstruction on non-uniform grids, extending the reconstruction techniques to quad-tree grids composed of triangles split by baricentric subdivision and finally at improving the CPU-time efficiency by a third order local timestepping technique.

References

1. Abgrall, R.: On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation. J. Comput. Phys. **114**, 45–58 (1994)
2. Aràndiga, F., Baeza, A., Belda, A.M., Mulet, P.: Analysis of WENO schemes for full and global accuracy. SIAM J. Numer. Anal. **49**(2), 893–915 (2011)
3. Arvanitis, C., Delis, A.I.: Behavior of finite volume schemes for hyperbolic conservation laws on adaptive redistributed spatial grids. SIAM J. Sci. Comput. **28**(5), 1927–1956 (2006). doi:10.1137/050632853
4. Bastian, P., Blatt, M., Dedner, A., Engwer, C., Fahlke, J., Gräser, C., Klöfkorn, R., Nolte, M., Ohlberger, M., Sander, O.: DUNE web page (2011). http://www.dune-project.org
5. Bastian, P., Blatt, M., Dedner, A., Engwer, C., Klöfkorn, R., Kornhuber, R., Ohlberger, M., Sander, O.: A generic grid interface for parallel and adaptive scientific computing. Part II: Implementation and tests in DUNE. Computing **82**(2–3), 121–138 (2008)
6. Bastian, P., Blatt, M., Dedner, A., Engwer, C., Klöfkorn, R., Ohlberger, M., Sander, O.: A generic grid interface for parallel and adaptive scientific computing. Part I: Abstract framework. Computing **82**(2–3), 103–119 (2008)
7. Berger, M.J., LeVeque, R.J.: Adaptive mesh refinement using wave-propagation algorithms for hyperbolic systems. SIAM J. Numer. Anal. **35**, 2298–2316 (1998)
8. Burri, A., Dedner, A., Klöfkorn, R., Ohlberger, M.: An efficient implementation of an adaptive and parallel grid in dune. In: Notes on Numerical Fluid Mechanics, vol. 91, pp. 67–82 (2006). See also http://aam.mathematik.uni-freiburg.de/IAM/Research/alugrid
9. Čada, M., Torrilhon, M.: Compact third-order limiter functions for finite volume methods. J. Comput. Phys. **228**(11), 4118–4145 (2009). doi:10.1016/j.jcp.2009.02.020
10. Constantinescu, E.M., Sandu, A.: Multirate timestepping methods for hyperbolic conservation laws. J. Sci. Comput. **33**(3), 239–278 (2007). doi:10.1007/s10915-007-9151-y
11. Cravero, I., Semplice, M.: On the accuracy of WENO and CWENO reconstructions of third order on nonuniform meshes. Preprint arXiv:1503.00736
12. Davies-Jones, R.: Comments on "Kinematic analysis of frontogenesis associated with a nondivergent vortex". J. Atmos. Sci. **42**(19), 2073–2075 (1985)

13. Dumbser, M., Käser, M.: Arbitrary high order non-oscillatory finite volume schemes on unstructured meshes for linear hyperbolic systems. J. Comput. Phys. **221**(2), 693–723 (2007)
14. Feng, H., Hu, F., Wang, R.: A new mapped weighted essentially non-oscillatory scheme. J. Sci. Comput. **51**, 449–473 (2012)
15. Fürst, J.: A weighted least square scheme for compressible flows. Flow Turbul. Combust. **76**(4), 331–342 (2006)
16. Gottlieb, S., Ketcheson, D., Shu, C.W.: Strong Stability Preserving Runge–Kutta and Multistep Time Discretizations. World Scientific Publishing, Hackensack (2011)
17. Henrick, A.K., Aslam, T.D., Powers, J.M.: Mapped weighted essentially non-oscillatory schemes: achieving optimal order near critical points. J. Comput. Phys. **207**, 542–567 (2005)
18. Hu, C., Shu, C.W.: Weighted essentially non-oscillatory schemes on triangular meshes. J. Comput. Phys. **150**(1), 97–127 (1999)
19. Karni, S., Kurganov, A.: Local error analysis for approximate solutions of hyperbolic conservation laws. Adv. Comput. Math. **22**(1), 79–99 (2005). doi:10.1007/s10444-005-7099-8
20. Ketcheson, D.I., Parsani, M., LeVeque, R.J.: High-order wave propagation algorithms for hyperbolic systems. SIAM J. Sci. Comput. **35**(1), A351–A377 (2013)
21. Kirby, R.: On the convergence of high resolution methods with multiple time scales for hyperbolic conservation laws. Math. Comput. **72**(243), 1239–1250 (2003). doi:10.1090/S0025-5718-02-01469-2
22. Kolb, O.: On the full and global accuracy of a compact third order WENO scheme. SIAM J. Numer. Anal. **52**(5), 2335–2355 (2014)
23. Levy, D., Puppo, G., Russo, G.: Compact central WENO schemes for multidimensional conservation laws. SIAM J. Sci. Comput. **22**(2), 656–672 (2000)
24. Li, W., Ren, Y.X.: High-order k-exact weno finite volume schemes for solving gas dynamic euler equations on unstructured grids. Int. J. Numer. Methods Fluids **70**(6), 742–763 (2012)
25. Lörcher, F., Gassner, G., Munz, C.D.: A discontinuous Galerkin scheme based on a space-time expansion. I. Inviscid compressible flow in one space dimension. J. Sci. Comput. **32**(2), 175–199 (2007). doi:10.1007/s10915-007-9128-x
26. Mandli, K.T., Ketcheson, D.I., et al.: Pyclaw software (2011). http://numerics.kaust.edu.sa/pyclaw
27. Ohlberger, M.: A review of a posteriori error control and adaptivity for approximations of non-linear conservation laws. Int. J. Numer. Methods Fluids **59**, 333–354 (2009). doi:10.1002/fld.1686
28. Osher, S., Sanders, R.: Numerical approximations to nonlinear conservation laws with locally varying time and space grids. Math. Comput. **41**(164), 321–336 (1983). doi:10.2307/2007679
29. Puppo, G.: Numerical entropy production for central schemes. SIAM J. Sci. Comput. **25**(4), 1382–1415 (2003/04)
30. Puppo, G., Semplice, M.: Numerical entropy and adaptivity for finite volume schemes. Commun. Comput. Phys. **10**(5), 1132–1160 (2011)
31. Puppo, G., Semplice, M.: Well-balanced high order 1d schemes on non-uniform grids and entropy residuals. Preprint arXiv:1403.4112
32. Rogerson, A., Meiburg, E.: A numerical study of the convergence properties of ENO schemes. J. Sci. Comput. **5**(2), 151–167 (1990)
33. Semplice, M., Coco, A.: Dune-fv software. http://www.personalweb.unito.it/matteo.semplice/codes.htm
34. Shi, J., Hu, C., Shu, C.W.: A technique of treating negative weights in WENO schemes. J. Comput. Phys. **175**(1), 108–127 (2002)
35. Shu, C.: Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. In: Advanced Numerical Approximation of Nonlinear Hyperbolic Equations (Cetraro, 1997), Lecture Notes in Mathematics, vol. 1697, pp. 325–432. Springer, Berlin (1998)
36. Shu, C.W., Osher, S.: Efficient implementation of essentially nonoscillatory shock-capturing schemes. J. Comput. Phys. **77**(2), 439–471 (1988)