A Bi–Hyperbolic Finite Volume Method on Quadrilateral Meshes

H. Joachim Schroll and Fredrik Svensson*

July 6, 2004

— preprint Lund University 2004 —

Abstract

A non–oscillatory, high resolution reconstruction method on quadrilateral meshes in 2D is presented. It is a two–dimensional extension of Marquina's hyperbolic method. The generalization to quadrilateral meshes allows the method to simulate realistic flow problems in complex domains. An essential point in the construction of the method is a second order accurate approximation of gradients on an irregular, quadrilateral mesh. The resulting scheme is *optimal* in the sense that it is third order accurate and the reconstruction requires only nearest neighbour information.

Numerical experiments are presented and the computational results are compared to experimental data.

Key words

Conservation law, hyperbolic reconstruction, high resolution finite volume scheme, quadrilateral mesh.

1 Introduction and Background

Finite volume schemes approximate local averages within computational cells. These averages are determined by the flow balance across the cell boundary. Thus, the accurate and efficient computation of fluxes along the boundary is crucial for FV-schemes. A dilemma is that the available data during the simulation are *average* quantities of the state variables. Accurate *point values* of the flux along the boundary need to be *reconstructed*

*Centre for Mathematical Sciences, Lund University, Box 118, SE–221 00 Lund, Sweden achim.schroll@na.lu.se and fredrik.svensson@na.lu.se

from these averages. During the 80's, piecewise polynomial reconstructions were developed (MUSCLE, PPM, ENO, WENO, etc...).

By the nature of *hyperbolic* conservation laws, lacking natural diffusion, small disturbances of a solution are preserved but not damped in time — solutions are *sensitive* to artificial oscillations. It is the building principle and the achievement of ENO-type reconstructions to select from several higher degree polynomials the one of lowest variation. Obviously, by this selection process, the stencil defining *all* the various interpolation problems increases. Moreover, the question arises: Is a polynomial the preferred ansatz-function for approximating the solution of a conservation law? As usual, the answer is problem dependent. Oscillatory phenomena, like acoustic waves may be well described by polynomials. Piecewise linear solutions of 1D Riemann problems however, are better represented by piecewise monotone functions like, for example, Marquina's piecewise hyperbolic reconstruction [12].

Given monotone data in only three neighbouring cells, the three parameters a, b and c defining a hyperbola

$$h(x) = a + \frac{b}{(x - x_j) + c} \tag{1}$$

can be adjusted in order to establish a third order accurate reconstruction of the unknown profile ξ from its local cell averages

$$v_j = \int_{C_j} \xi(x)\ dx,\ \ C_j = [x_{j-1/2},\ x_{j+1/2}],\ \ x_j = j \cdot \Delta x,\ \ \Delta x > 0,\ \ j \in \mathbb{Z}.$$

The building principle [12, 1] is to preserve the cell average v_j

$$\frac{1}{\Delta x} \int_{C_j} h(x)\ dx = v_j \tag{2}$$

and to approximate the lateral gradients up to second order

$$h'(x_{j+1/2}) = \xi'(x_{j+1/2}) + \mathcal{O}(\Delta x^2). \tag{3}$$

In the case of a uniform mesh, a second order approximation of $\xi'(x_{j+1/2})$ is easily obtained by a central difference of neighbouring averages

$$\frac{1}{\Delta x}(v_{j+1} - v_j) = \xi'(x_{j+1/2}) + \mathcal{O}(\Delta x^2).$$

The so obtained hyperbolic reconstruction is third order accurate [1], local in the sense that only nearest neighbour information is required and it is known for its capability to resolve sharp corners especially well.

This feature of the *piecewise hyperbolic method* (PHM) was already observed and documented in Marquina's original paper [12]. The compactness of the stencil (five points in 1D) gives better resolution of corners compared to ENO and TVD upwind schemes of the *same order* (and with larger stencils). We refer to Example 2 (linear advection) and Example 4 (non-convex, scalar conservation) in [12] for a comparison of third order ENO and PHM.

We would also like to argue that in some sense a third order five-point FV-scheme (like PHM) is the *optimal* method for a hyperbolic conservation law in one space dimension: By definition of *hyperbolicity* such an evolution is a *local* process. The most local scheme that is feasible would be a three-point FV-scheme. However, a third order three-point scheme is not TV-stable. In fact, any three-point TVD-scheme is at most first order accurate [18]. The next bigger symmetric stencil contains five points. Achieving third order accuracy and TV stability on a five point stencil is the best one can hope for. A third order ENO scheme is not optimal in this sense, as it requires a seven point stencil.

To take advantage of Marquina's successful concept in realistic flow simulations, it is tempting to establish a similar reconstruction in two space dimensions defined on non-uniform, quadrilateral meshes. The key ingredient for achieving this goal is to establish a second order approximation of the lateral gradients based on local averages in non-rectangular meshes — the next section deals with that problem. Once again it is advantageous to start out with a local scheme like PHM, otherwise this numerical differentiation process would be even more technically involved and hence less efficient.

The bi-hyperbolic reconstruction in Section 3 is based on these gradients. Flux balancing, see Section 4, is then straight forward and SSP-Runge Kutta time stepping can be applied. The resulting bi-hyperbolic finite volume scheme is validated in Section 5. In a shock/bubble experiment, we compare our numerical results to experimental data (Section 5.2). Two other test cases, flow in a compression channel (Section 5.3) and flow past a triangular obstacle (Section 5.4), illustrates the performance of the new scheme in non-rectangular geometries.

2 Numerical gradients on quadrilateral meshes in two dimensions

As outlined above, lateral gradients of second order accuracy along the cell boundary are required for reconstruction. In analogy to one dimensional finite difference formulas, the idea is to establish a linear combination of function values from neighbouring cells, approximating the desired partial derivative up to second order accuracy. The coefficients in this linear combination are obtained by Taylor expansion.

For illustration we consider the rectangular case first, where the mesh is aligned with the coordinate system and spatial derivatives can be approximated by one-dimensional difference formulas. To achieve second order accuracy, it is sufficient to sample three function values and the coefficients are spelled out in Section 2.1.

This concept is then extended to non-rectangular cells in two space dimensions, where a six-point stencil is required for second order accuracy. The six coefficients depend on the local (mesh-)geometry. They may be computed (numerically) from local, linear systems. Another issue is how to select an appropriate six point stencil in two dimensions, see Section 2.2.

2.1 Rectangular cells

In a rectangular mesh the approximation of normal derivatives along the boundary is a one-dimensional problem. For the ease of notation, we omit the second variable in this subsection. Consider non-uniform cells $C_i = [\underline{x}_i, \overline{x}_i]$, $i \in \mathbb{Z}$ with boundaries $\overline{x}_i = \underline{x}_{i+1}$ and variable width $\Delta x_i = \overline{x}_i - \underline{x}_i$, see Fig. 1. The goal is to find a linear combination of known averages

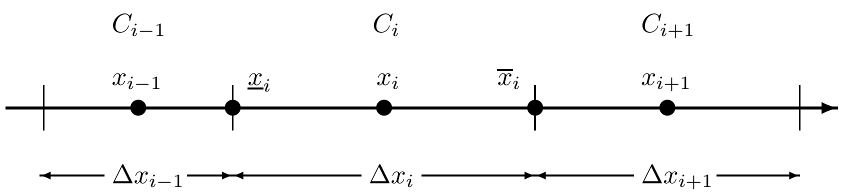

Figure 1: Non-uniform, rectangular mesh.

$\overline{f}_{C_i} = \left(\int_{C_i} f(x)\ dx\right) / \Delta x_i$, approximating the right edge derivative $f(\overline{x}_i)$ up to second order accuracy

$$\alpha \overline{f}_{C_{i+1}} + \beta \overline{f}_{C_i} + \gamma \overline{f}_{C_{i-1}} = f'(\overline{x}_i) + \mathcal{O}(\Delta x^2), \tag{4}$$

where $\Delta x = \sup_i \Delta x_i$.

Lemma 2.1 Assume $f \in \mathcal{C}^3(\mathbb{R})$. The linear combination (4) with coefficients α, β and γ satisfying

$$\begin{pmatrix} 1 & 1 & 1 \\ \frac{\Delta x_{i+1}}{2} & -\frac{\Delta x_i}{2} & \Delta x_i + \frac{\Delta x_{i-1}}{2} \\ \Delta x_{i+1}^2 & \Delta x_i^2 & 3\Delta x_i^2 + 3\Delta x_i \Delta x_{i-1} + \Delta x_{i-1}^2 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \tag{5}$$

is a second order approximation of $f'(\overline{x}_i)$.

Proof: Expand f around the right edge in C_i and integrate

$$\overline{f}_{C_i} = f(\overline{x}_i) - \frac{1}{2}\Delta x_i f'(\overline{x}_i) + \frac{1}{6}\Delta x_i^2 f''(\overline{x}_i) + \mathcal{O}(\Delta x^3).$$

Similarly,

$$\overline{f}_{C_{i+1}} = f(\overline{x}_i) + \frac{1}{2}\Delta x_{i+1} f'(\overline{x}_i) + \frac{1}{6}\Delta x_{i+1}^2 f''(\overline{x}_i) + \mathcal{O}(\Delta x^3)$$

and

$$\begin{aligned} \overline{f}_{C_{i-1}} &= f(\overline{x}_i) + (\Delta x_i + \frac{\Delta x_{i-1}}{2}) f'(\overline{x}_i) \\ &- \frac{1}{6}\frac{\Delta x_i^3 - (\Delta x_i + \Delta x_{i-1})^3}{\Delta x_{i-1}} f''(\overline{x}_i) + \mathcal{O}(\Delta x^3). \end{aligned}$$

Now the system (5) ensures that

$$\alpha \overline{f}_{C_{i+1}} + \beta \overline{f}_{C_i} + \gamma \overline{f}_{C_{i-1}} = f'(\overline{x}_i) + \alpha \cdot \mathcal{O}(\Delta x^3) + \beta \cdot \mathcal{O}(\Delta x^3) + \gamma \cdot \mathcal{O}(\Delta x^3).$$

As the coefficients α, β and γ themselves are of order Δx^{-1}, the error sums up to second order and the lemma is proved.

$\square$

Remark 2.2 In the case of an equidistant mesh we have $\Delta x_i = \Delta x$. In this case the system (5) reads

$$\begin{pmatrix} 1 & 1 & 1 \\ \frac{\Delta x}{2} & -\frac{\Delta x}{2} & -\frac{3\Delta x}{2} \\ \Delta x^2 & \Delta x^2 & 7\Delta x^2 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} .$$

The (unique) solution is $\alpha = 1/\Delta x$, $\beta = -1/\Delta x$ and $\gamma = 0$ and the well-known central difference formula is recovered

$$\frac{\overline{f}_{C_{i+1}} - \overline{f}_{C_i}}{\Delta x} = f'(\overline{x}_i) + \mathcal{O}(\Delta x^2) \ .$$

Remark 2.3 A formula for the left edge derivative can be obtained analogously. The defining system reads

$$\begin{pmatrix} 1 & 1 & 1 \\ \frac{\Delta x_{i+1}}{2} + \Delta x_i & \frac{\Delta x_i}{2} & -\frac{\Delta x_{i-1}}{2} \\ 3\Delta x_i^2 + 3\Delta x_i \Delta x_{i+1} + \Delta x_{i+1}^2 & \Delta x_i^2 & \Delta x_{i-1}^2 \end{pmatrix} \begin{pmatrix} \alpha' \\ \beta' \\ \gamma' \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} .$$

Observe the symmetry with respect to $x = x_i$.

2.2 Quadrilateral cells

In this section, non-rectangular convex cells bounded by four linear edges are considered. Given cell averages $\overline{f}_{C_j} = (\int_{C_j} f(x,y)\ dxdy)/\mathrm{vol}(C_j)$ of a sufficiently smooth function $f \in \mathcal{C}^3(\Omega)$, $\Omega \subset \mathbb{R}^2$, the plan is to make a second order accurate approximation of the gradient of f in a given point $\overline{p} = (\overline{x}, \overline{y})$ on one of the edges (see Fig. 4 for illustration). To this end we expand f in $p = (x,y) \in \Omega$ around $\overline{p}$

$$\begin{aligned} f(x,y) \quad = \quad & f(\overline{p}) + f_x(\overline{p})(x-\overline{x}) + f_y(\overline{p})(y-\overline{y}) + \\ & \tfrac{1}{2} f_{xx}(\overline{p})(x-\overline{x})^2 + f_{xy}(\overline{p})(x-\overline{x})(y-\overline{y}) + \\ & \tfrac{1}{2} f_{yy}(\overline{p})(y-\overline{y})^2 + \mathcal{O}(\|p-\overline{p}\|^3). \end{aligned}$$

Next, we integrate $f(x,y)$ in C_j, divide by the cell volume $\mathrm{vol}(C_j)$ and find for the average

$$\overline{f}_{C_j} = f(\overline{p}) + \alpha_j f_x(\overline{p}) + \beta_j f_y(\overline{p}) + \frac{1}{2}\gamma_j f_{xx}(\overline{p}) + \delta_j f_{xy}(\overline{p}) + \frac{1}{2}\varepsilon_j f_{yy}(\overline{p}) + \mathcal{O}(\Delta_j^3) \ ,$$

where $\Delta_j = \mathrm{diam}(C_j) = \max_{p,q \in C_j} \|p-q\|$. Note that by averaging, third order is preserved. Moreover, the coefficients

$$\alpha_j = \frac{1}{\mathrm{vol}(C_j)} \iint_{C_j} (x-\overline{x})\ dxdy \ , \quad \beta_j = \frac{1}{\mathrm{vol}(C_j)} \iint_{C_j} (y-\overline{y})\ dxdy \ ,$$

$$\gamma_j = \frac{1}{\mathrm{vol}(C_j)} \iint_{C_j} (x-\overline{x})^2\ dxdy \ , \quad \delta_j = \frac{1}{\mathrm{vol}(C_j)} \iint_{C_j} (x-\overline{x})(y-\overline{y})\ dxdy \ ,$$

and

$$\varepsilon_j = \frac{1}{\mathrm{vol}(C_j)} \iint_{C_j} (y-\overline{y})^2\ dxdy$$

are determined from the mesh and $\overline{p}$ only. The integrals can be worked out explicitly, which is avoided in this text however.

By a linear combination of averages in neighbouring cells, the goal is to approximate e.g. $f_x(\overline{p})$ by

$$\sum_l \mu_l \overline{f}_{C_l} = f_x(\overline{p}) + \sum_l \mu_l \mathcal{O}(\Delta_l^3) \ , \tag{6}$$

where l is a local index to be specified below, see Fig. 3. To achieve this, the following linear systems have to hold for the weights μ_l

$$\begin{aligned}
&\text{Constant terms :} && \sum_l \mu_l = 0 \\
&\text{First order integral terms :} && \sum_l \alpha_l \mu_l = 1 \ , \quad \sum_l \beta_l \mu_l = 0 \\
&\text{Second order integral terms :} && \sum_l \gamma_l \mu_l = \sum_l \delta_l \mu_l = \sum_l \varepsilon_l \mu_l = 0 \ .
\end{aligned} \tag{7}$$

These equations form a linear system

$$\mathbf{A}\boldsymbol{\mu} = \mathbf{1}_2 \tag{8}$$

from which the weights μ_l and thus the formula (6) is determined. The l-th column of $\mathbf{A}$ is $\mathbf{A}_l = (1,\ \alpha_l,\ \beta_l,\ \gamma_l,\ \delta_l,\ \varepsilon_l)^T$ and $\mathbf{1}_2 = (0, 1, 0, \ldots, 0)^T \in \mathbb{R}^6$. The corresponding formula for the y-derivative

$$\sum_l \nu_l \overline{f}_{C_l} = f_y(\overline{p}) + \sum_l \nu_l \mathcal{O}(\Delta_l^3) \tag{9}$$

is determined from

$$\mathbf{A}\boldsymbol{\nu} = \mathbf{1}_3 \ ,$$

correspondingly. To fulfill the six conditions (7) in general meshes, a six-cell stencil is required. A tempting choice is to use the six nearest neighbour cells to $\overline{p}$, as illustrated in Fig. 2. However in uniform meshes the resulting system (8) for $\boldsymbol{\mu}$ is under determined. In this case $\gamma_l = \gamma$ is the same constant in all six cells, thus $\sum \mu_l = 0$ implies $\sum \gamma_l \mu_l = 0$. Indeed, in this situation the central difference $\frac{1}{\Delta x}\left(\overline{f}_{C_3} - \overline{f}_{C_0}\right) = f_x(\overline{p}) + \mathcal{O}(\Delta x^2)$ is a second order approximation. The coefficients μ_i for $i = 1, 2, 4$ and 5 may, but need not, be set to zero. Thus, in the general case the most compact stencil (Fig. 2) is not practical, as the formula becomes ambiguous in uniform meshes.

The problem is avoided by moving C_3 in Fig. 2 to the west side, see Fig. 3. Note the renumbering of cells. On uniform quadratic cells with cell side length h, the formulas using this stencil are uniquely defined, $\det \mathbf{A} = 8h^8$. It turns out that $\mu_0 = \mu_5 = 0$ and the partial derivatives are

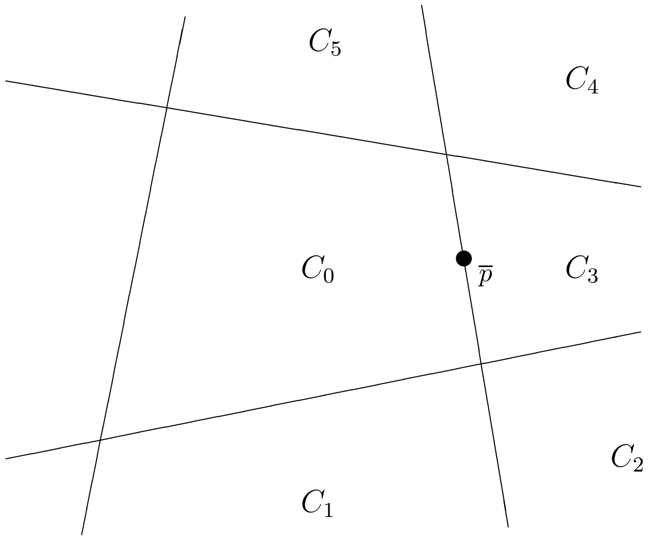

Figure 2: The most compact stencil.

approximated by an average of two central differences

$$f_x(\overline{p}) = \frac{1}{2}\left(\frac{\overline{f}_{C_2} - \overline{f}_{C_1}}{h} + \frac{\overline{f}_{C_3} - \overline{f}_{C_4}}{h}\right)$$

and

$$f_y(\overline{p}) = \frac{1}{2}\left(\frac{\overline{f}_{C_4} - \overline{f}_{C_1}}{2h} + \frac{\overline{f}_{C_3} - \overline{f}_{C_2}}{2h}\right) \ .$$

In the general case however, it is a hopeless enterprise to spell out explicit formulas for the coefficients $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$. Therefore, in practical computations, the coefficients are computed and stored in each cell before the main simulation. Note that the matrix $\mathbf{A}$ and thus the formulas (6) and (9) depend on the local geometry of the mesh.

In the context of bi-hyperbolic reconstructions, as described in the next section, x-derivatives and therefore μ-coefficients are required on east and west cell edges, while y-derivatives and ν-coefficients are stored for use on the north and south edges of every cell. The derivation of numerical derivatives on the west, north and south edges is completely analogous to those on the east side (as described above), and is therefore omitted.

Finally, it would be interesting to prove the regularity of the coefficient matrix $\mathbf{A}$ in the general case for an arbitrary quadrilateral cell. However,

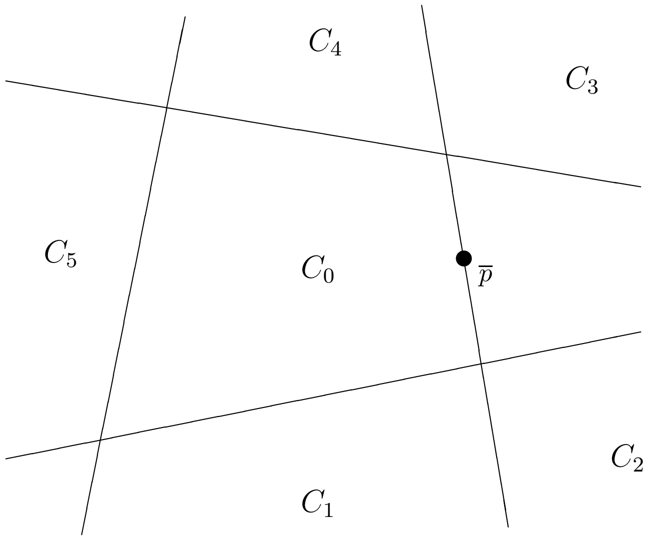

Figure 3: Chosen stencil.

even that is tedious due to the many parameters involved. In any case, due to the continuity of the determinant these systems are regular at least for "almost" rectangular meshes. Also, in the numerical simulations presented in Section 5, all linear systems defining the coefficients μ_l and ν_l were regular, and the formulas (6) and (9) are uniquely defined.

3 The bi–hyperbolic reconstruction

Marquina proposed in [12] a hyperbolic reconstruction of type (1). To establish the conservation property (2), the ansatz reads

$$r_j^M(x) = v_j + d_j \frac{\Delta x}{\alpha_j^2} \left(\ln \left(\frac{2-\alpha_j}{2+\alpha_j} \right) - \frac{\Delta x}{(x-x_j) - \frac{\Delta x}{\alpha_j}} \right), \tag{10}$$

where v_j denotes the given average in cell C_j. The remaining two parameters d_j and α_j are adjusted by both lateral gradients.

In the case of monotone data $(v_j - v_{j-1}) \cdot (v_{j+1} - v_j) > 0$, there exists a unique reconstruction (10) which preserves the average v_j and picks up both lateral gradients

$$r_j^M(x_{j\pm 1/2}) = \pm \frac{v_{j\pm 1} - v_j}{\Delta x}.$$

Otherwise, third order reconstruction by a monotone hyperbola is not possible and *order reduction* is the consequence (a globally third order, double logarithmic reconstruction is currently being developed [2]). The resulting algorithm is known as local hyperbolic reconstruction: LHR, see [12] for details.

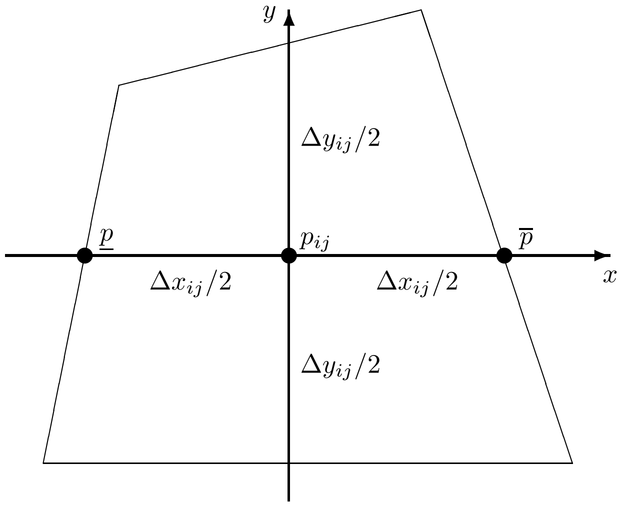

Figure 4: The geometry of a generic cell.

The natural extension of (10) to non-uniform meshes in two space dimensions reads

$$\begin{aligned} r_{ij}(x,y) \quad = \quad v_{ij} \quad &+ \quad d_{ij}\frac{\Delta x_{ij}}{\alpha_{ij}^2} \cdot \left(\ln \frac{2-\alpha_{ij}}{2+\alpha_{ij}} - \frac{\Delta x_{ij}}{(x-x_{ij}) - \frac{\Delta x_{ij}}{\alpha_{ij}}} \right) \\ &+ \quad e_{ij}\frac{\Delta y_{ij}}{\beta_{ij}^2} \cdot \left(\ln \frac{2-\beta_{ij}}{2+\beta_{ij}} - \frac{\Delta y_{ij}}{(y-y_{ij}) - \frac{\Delta y_{ij}}{\beta_{ij}}} \right). \end{aligned} \tag{11}$$

Just like in (10), the bi-hyperbolic functions r_{ij} are supported only in the corresponding cells C_{ij}, so that each computational cell has its own specific reconstruction. Obviously, Δx_{ij} and Δy_{ij} are the local cell sizes according to Fig. 4. Further, v_{ij} is set to the desired cell specific average. The four parameters α_{ij}, β_{ij}, d_{ij} and e_{ij} are adjusted by the lateral gradients on all four edges, which are determined as in the previous section. In fact, we apply Marquina's LHR-algorithm coordinate wise so that the x-derivatives on the east and west edges determine the reconstruction in x-direction i.e. α_{ij} and d_{ij}; analogous in y-direction.

For computational convenience we define

$$\eta(\gamma; p, p_{ij}, h) = \frac{1}{\gamma^2}\left(\ln\frac{2-\gamma}{2+\gamma} - \frac{h}{(p-p_{ij}) - \frac{h}{\gamma}}\right).$$

The ansatz (11) then reads

$$r_{ij}(x,y) = v_{ij} + d_{ij}\Delta x_{ij}\eta(\alpha_{ij}; x, x_{ij}, \Delta x_{ij}) + e_{ij}\Delta y_{ij}\eta(\beta_{ij}; y, y_{ij}, \Delta y_{ij}).$$

Note, the function η has a removable singularity at $\gamma = 0$:

$$\lim_{\gamma\to 0}\eta(\gamma; p, p_{ij}, h) = \frac{p-p_{ij}}{h}.$$

At this point a remark is in place: On rectangular — not necessarily uniform — meshes, the cell average of r_{ij} in fact is v_{ij}

$$\iint_{C_{ij}} r_{ij}(x,y)\ dxdy = v_{ij}\cdot \mathrm{vol}(C_{ij}), \tag{12}$$

where $\mathrm{vol}(C_{ij})$ denotes the volume of the cell. This is easily checked using the one-dimensional conservation principle (2). In non-rectangular cells however, (12) does not hold due to the nonlinearity of the bi-hyperbolic ansatz r_{ij}. Consequently, third order accuracy can not be expected in those cells and once again *order reduction* occurs.

In principle, conservation and hence third order accuracy can be restored by adjusting the parameter v_{ij} such that the desired cell average is achieved by the reconstruction. This means in practice that the bi-hyperbolic ansatz (11) needs to be integrated in the quadrilateral cell. However, an appropriate numerical quadrature in 2D is expensive, as it has to be performed cell wise. Our numerical experience (Section 5) shows that this effort is not justified and we accept local order reduction instead.

4 Flux Balancing and Time Update

Consider the conservation law which is simulated cell-wise

$$\frac{d}{dt}\int_{C_{ij}} U(x,y,t)\ dxdy + \int_{\partial C_{ij}} F(U(x,y,t))\cdot \mathbf{n}\ dS = 0. \tag{13}$$

To determine the dynamics of the cell average

$$\overline{U}_{ij}(t) = \frac{1}{\mathrm{vol}(C_{ij})}\int_{C_{ij}} U(x,y,t)\ dxdy$$

the flux balance along the cell boundary is required

$$\frac{d\overline{U}_{ij}(t)}{dt} = -\frac{1}{\mathrm{vol}(C_{ij})}\int_{\partial C_{ij}} F(U(x,y,t)) \cdot \mathbf{n}\, dS. \tag{14}$$

The flux $F(U(\cdot,\cdot),t)|_{\partial C_{ij}}$ on the boundary is computed via flux-splitting and upwinding, i.e. van-Leer's splitting [21] for the Euler equations or Marquina's splitting [13] in case of two component gas.

The boundary integral in (14) is evaluated by a quadrature formula of suitable order e.g. the fourth order Gauss formula or, for simplicity, just the midpoint rule. Therefore the reconstructed flux functions need to be evaluated at nodal points along ∂C_{ij}. The quadrature formula applied to the reconstructed normal fluxes $F(U(\cdot,\cdot),t)|_{\partial C_{ij}}$, summed over all four edges of the cell gives the right hand side in (14). Finally the third order, non-linear SSP-Runge Kutta scheme ([5], Section 4.1) is applied for stable time integration.

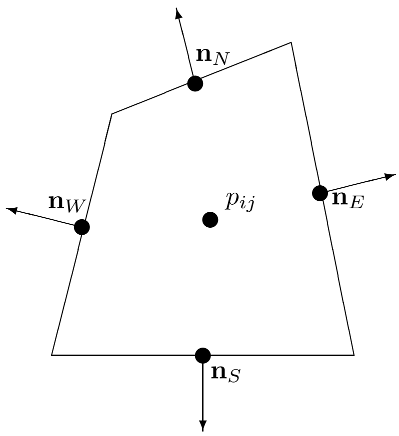

Figure 5: The normal vectors for a generic cell and nodal points for the mid point rule.

5 Numerical Tests

In order to test and validate the newly developed bi-hyperbolic scheme, we simulate four different gas flows, three of them governed by the well known

Euler system in two dimensions [15, 8, 17]

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x + \mathbf{G}(\mathbf{U})_y = \mathbf{0}, \tag{15}$$

with

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix}, \quad \mathbf{F}(\mathbf{U}) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E+p) \end{pmatrix}, \quad \mathbf{G}(\mathbf{U}) = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E+p) \end{pmatrix}. \tag{16}$$

In air, $\gamma = 1.4$ and the pressure is $p = (\gamma - 1)(E - \frac{\rho}{2}(u^2 + v^2))$.

In the second test case (Section 5.2) a two component gas consisting of air and helium is simulated. The governing equations (see [7, 13]) are (15) with

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \\ \rho\phi \end{pmatrix}, \quad \mathbf{F}(\mathbf{U}) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E+p) \\ \rho\phi u \end{pmatrix}, \quad \mathbf{G}(\mathbf{U}) = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E+p) \\ \rho\phi v \end{pmatrix}. \tag{17}$$

Here ϕ models the mass fraction of the first component in the mixture and hence $1 - \phi$ is the mass fraction of the second component. Furthermore γ is the ratio of specific heats in the mixture

$$\gamma(\phi) = \frac{C_p}{C_v} = \frac{C_{p_1}\phi + C_{p_2}(1-\phi)}{C_{v_1}\phi + C_{v_2}(1-\phi)}.$$

The pressure is determined as usual

$$p = (\gamma(\phi) - 1)(E - \frac{\rho}{2}(u^2 + v^2)). \tag{18}$$

5.1 Shock interacting with a bubble in air

Shock/bubble interactions have been studied both in experiments [6] and by impressing numerical simulations [7, 9, 13]. The fascinating images generated by these problems are due to the generation of vorticity, turbulence and mixing. Photographs of the experiments by Haas and Sturtevant [6] with a bubble of helium immersed in air are presented in the Gallery of Fluid Motion [14].

Inspired by these experiments Langseth and LeVeque [9] formulated a single gas experiment: A 2.95 Mach shock wave propagates into a bubble of low density air. A colored Schlieren image of this interaction became the CLAWPACK logo [23]. The original 3D problem is cylindrically symmetric and may be formulated as the 2D Euler equations (15) and (16), however with a source term. Very nice results both in 2D and 3D computed by the CLAWPACK package can be found in [9].

Here we simulate the homogenous Euler equations in 2D, with the initial data from [9] but ignoring the source term. Figs. 6, 7 and 8 show Schlieren type images of the density computed on a 1600×500 rectangular mesh on the computational domain $\Omega = [-0.1, 1.5] \times [-0.5, 0]$. The Figures do not reflect the full computational resolution due to data compression. Anyway, compared to both the first order Front Tracking simulation [22] and the second order accurate CLAWPACK results reported in the preprint [9] we do observe very well resolved vortex formulation, which indicates low numerical viscosity — as expected for a third order scheme.

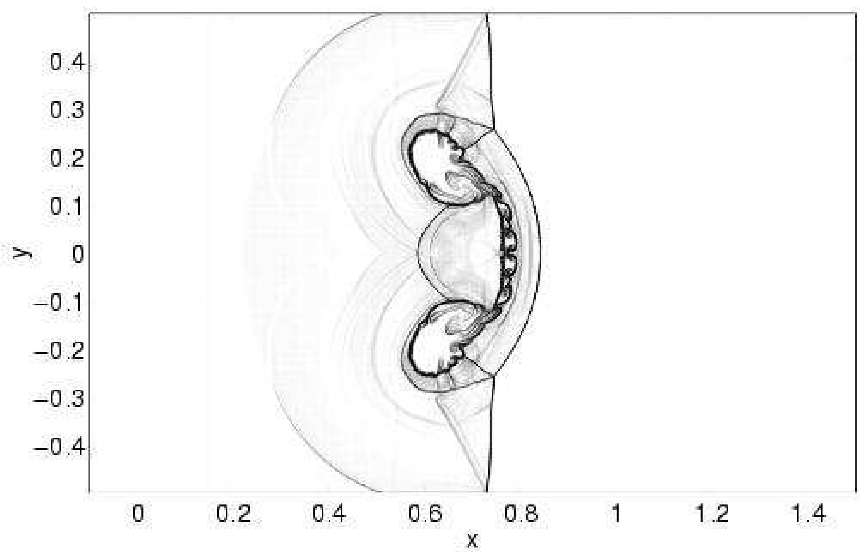

Figure 6: Shock-bubble interaction in air at $t = 0.21$ s.

5.2 A shock wave in air hits a bubble of helium

The next simulation addresses the two gas experiment by Haas and Sturtevant [6]: A 1.22 Mach shock in air hits a helium bubble contaminated with 28% of air. We find this an attractive test case as well documented data both from experiments and simulations are available. Following Marquina and Mulet [13], the flow is governed by the homogenous 2D two component Euler equations (15) and (17). The initial location of the shock and the bubble is shown in Fig. 9.

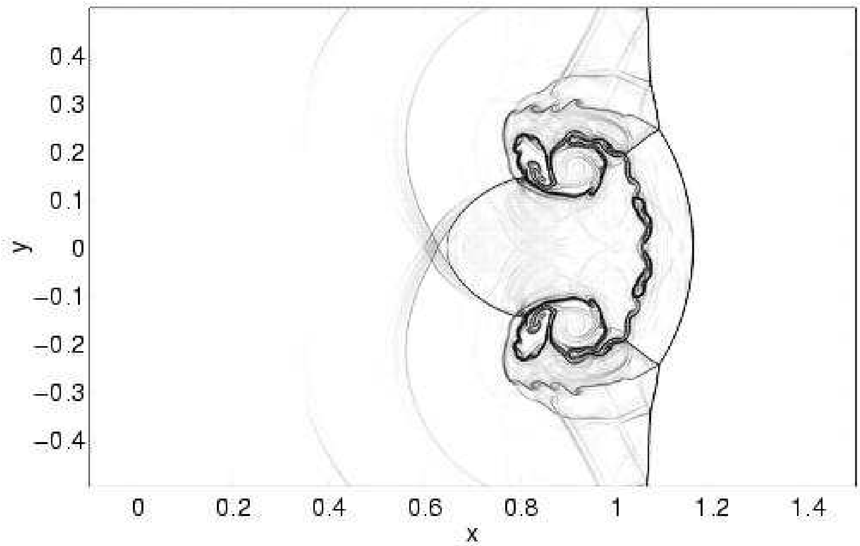

Figure 7: Shock-bubble interaction in air at $t = 0.30$ s.

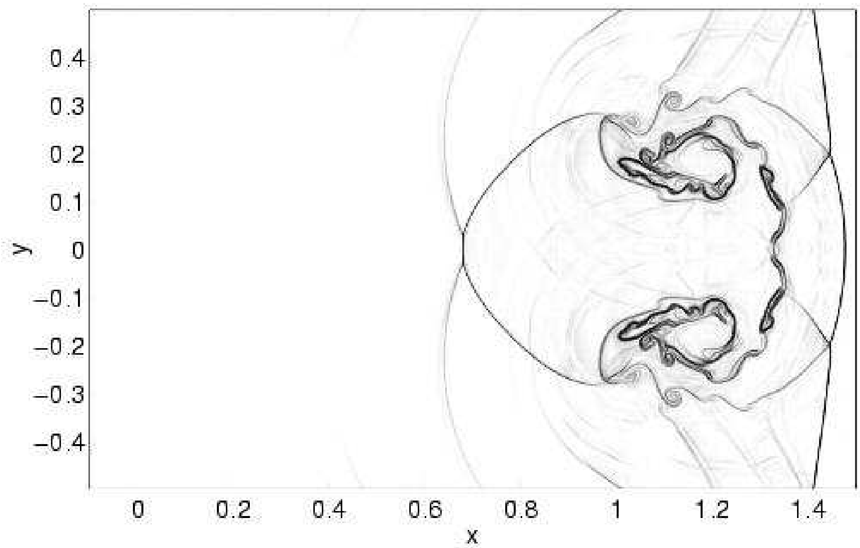

Figure 8: Shock-bubble interaction in air at $t = 0.40$ s.

The initial data is described as follows: To the left of the shock both gases are at rest, i.e. $u = v = 0\ m/s$. The pressure is uniformly 101325 Pa and the density of air (outside the bubble) is $\rho_l = 1225\ g/m^3$. From the ideal gas law $p = \rho RT$ assuming constant temperature it follows for the density inside the bubble $\rho_b = \rho_l \frac{R}{R_b} = 222.8\ g/m^3$. The gas constants for air and helium containing 28% of air are $R = 0.287\ J/gK$ and $R_b = 1.578\ J/gK$. The total energy is determined from the pressure and the specific heats by using (18). So both states to the left of the shock are completely determined by specifying the ratios of specific heats for air $\gamma = 1.4$ and the mixture $\gamma_b = 1.648$. It remains to determine the state to the right of the shock wave. From standard shock relations [20] (3.59) and (3.61) we find $\rho_r = 1686\ g/m^3$, $u_r = -113.52\ m/s$, $v_r = 0$, and $p_r = 159060\ Pa$ (note the misprint in Table Ib in [13]).

The remaining two plots in Fig. 9 show density contours at later times. We see how the shock deforms the bubble, the contact line curls up and vorticity is created. These solutions are computed on a 3000×300 grid. Marquina et al used a 8000×800 grid for their simulations. Here, the intention is not to show highly resolved Schlieren pictures, but to compare certain time-averaged wave speeds with available data. Table I is actually taken from [13] but augmented with the figures from the present computation.

	V_s	V_r	V_t	$V_{ui^{(1)}}$	$V_{ui^{(2)}}$	V_{di}	V_j
Haas and Sturtevant	410	900	393	170	113	145	230
Bi-hyp scheme	414	936	374	181	118	143	230
Percentage error	-1.0	-3.9	5.1	-6.1	-4.2	1.4	0
Marquina and Mulet	414	943	373	176	111	153	229
Bi-hyp scheme	414	936	374	181	118	143	230
Percentage error	0	0.8	-0.3	-2.8	-5.9	7.0	-0.4

Table I.

In this Table V_t, for example, stands for the velocity of the transmitted shock wave and V_{di} is the velocity of the downstream border of the bubble, both measured along $y = 0$. For further details we refer to [13]. In any case, we can conclude that our figures agree perfectly (up to the measurement error) with the experiment and also quite well with the independent numerical simulation. In this test case, our code is clearly verified.

5.3 A compression channel

Here, the bi-hyperbolic method is applied to a flow in a non-rectangular domain, i.e. a channel of decreasing width. The mesh consists of 900×600

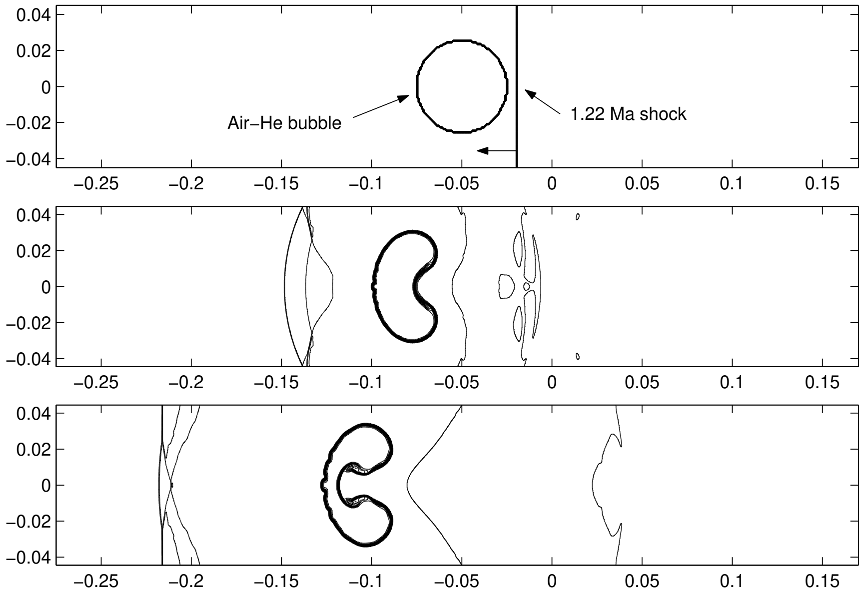

Figure 9: Initial configuration and computed solution (density contours) at $t = 261\mu$s and $t = 449\mu$s.

cells and the CFL number is set to 0.8. In Fig. 10 we observe the incident 2.95 Mach shock wave in air rightwards propagating in the cannel. By the compression within the channel, secondary shocks form at the walls and propagate in vertical direction into the channel. Observe the three-shock interaction with the incident wave at $t = 0.2$ s. The secondary shocks pass each other in the middle of the channel and get reflected at the opposite wall ($t = 0.4$ s). They bounce back into the channel and a periodic wave pattern forms ($t = 0.5$ s and $t = 0.6$ s). The initial shock has passed through the channel at these times.

5.4 Triangular obstacle

A 1.36-Mach shock wave in pure air passes a triangular obstacle with reflective properties. Like all previous experiments, also this problem is obviously symmetric with respect to the x-axis. The computational domain is the lower half only as shown schematically in Fig. 11. This domain is divided into four blocks as shown in Fig. 12. By this division each block has only one type of boundary condition along each edge and, of course, parallel computation speeds up the turn around time. The mesh size is 200×200 cells in each block, i.e. there are 160 kcells in the computational domain.

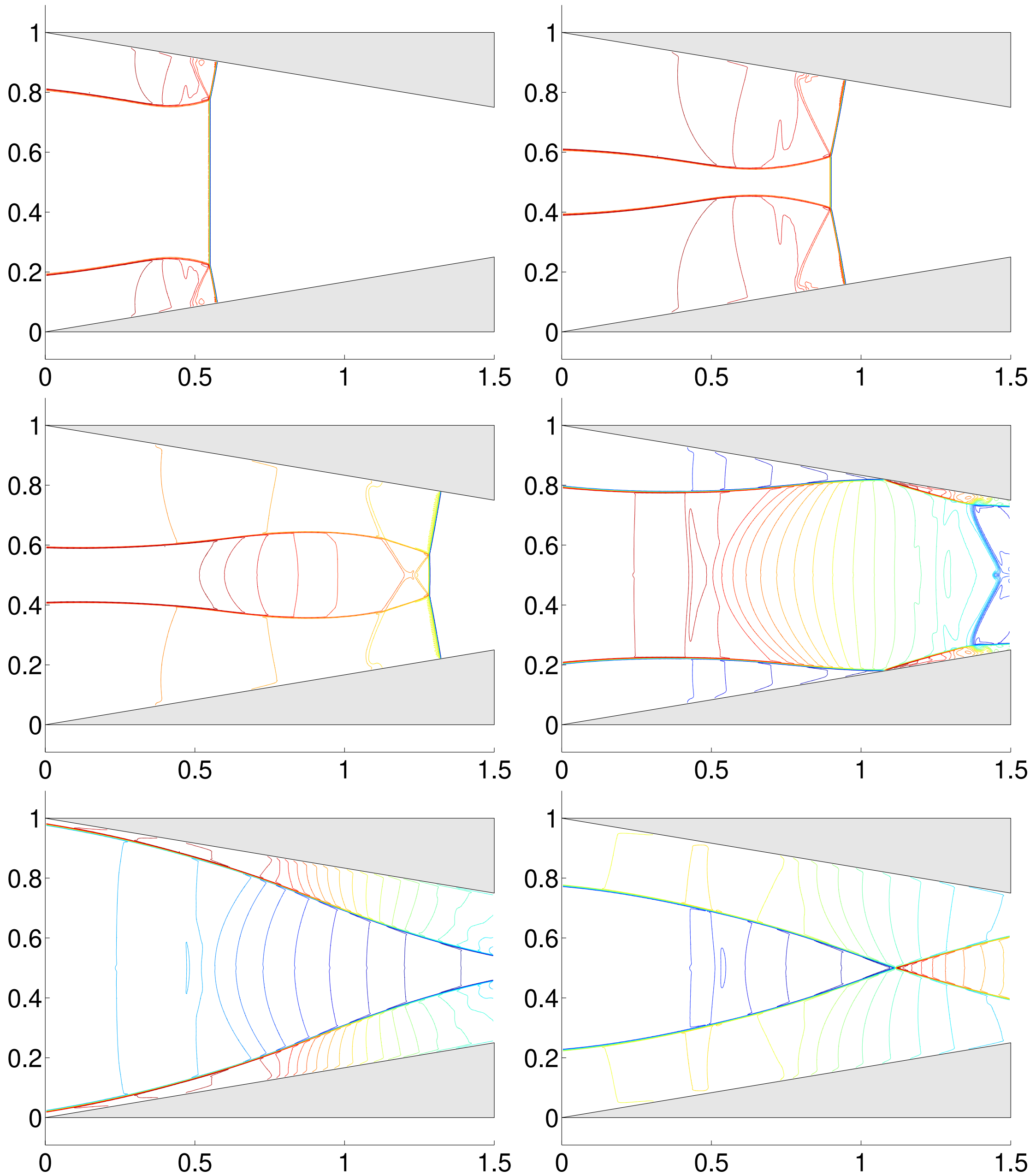

Figure 10: Density contours for a shock wave passing through a compression channel. Time is increasing row wise from 0.1 s to 0.6 s by 0.1 s steps.

How to implement boundary conditions is described in the textbook [10].

Density contours at four different times are shown in Figs. 13, 14 and 15. It is quite fascinating to observe how the contact lines in the wake of the triangle roll up and form vortices as time evolves.

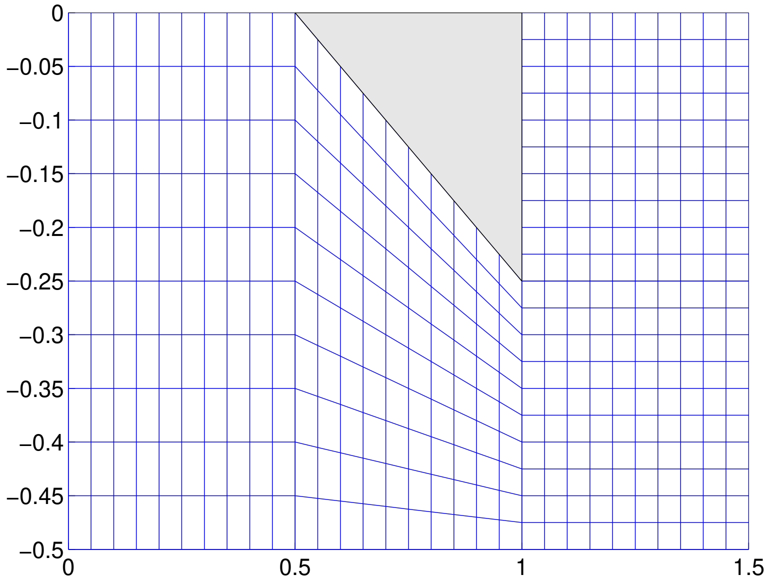

Figure 11: The computational domain.

6 Summary, discussion and outlook

We have presented a new high resolution FV-scheme based on bi-hyperbolic reconstruction on quadrilateral meshes in two space dimensions. The bi-hyperbolic reconstruction is a two-dimensional extension of Marquina's piecewise hyperbolic method (PHM). The formally third order accurate, bi-hyperbolic FV-scheme is validated with both computational and experimental data.

To formulate the basic FV-scheme on quadrilateral cells is straightforward. The difficulty was to establish a higher order reconstruction on irregular meshes. The key ingredients are the accurate approximations of lateral gradients developed in Section 2.

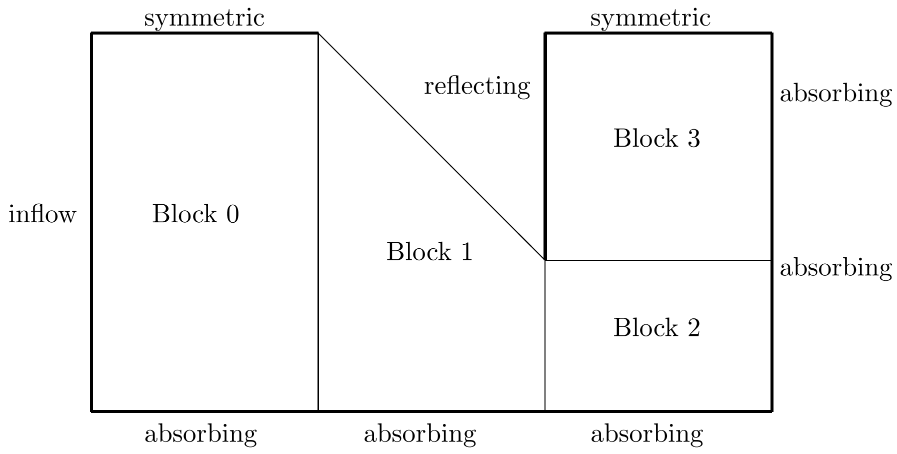

Figure 12: Block structure of the computational domain.

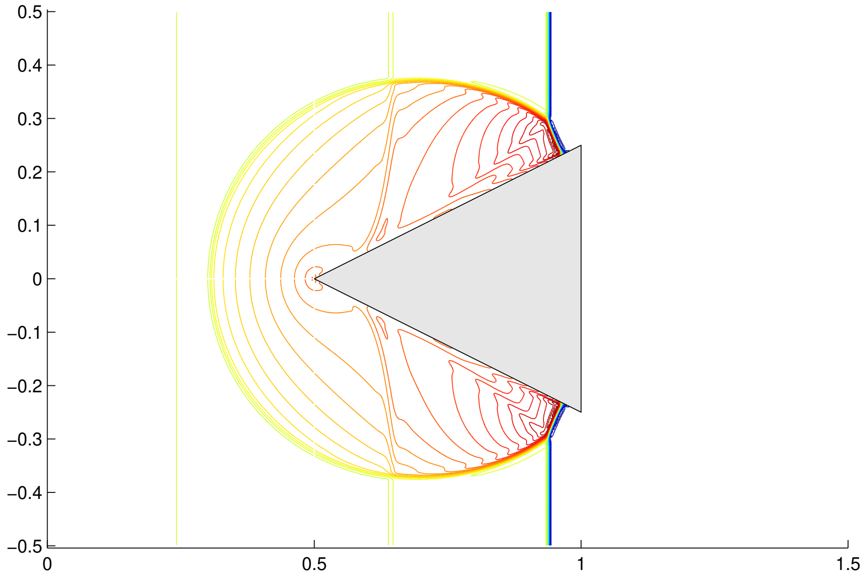

Figure 13: Flow past a triangle at $t = 0.30$ s.

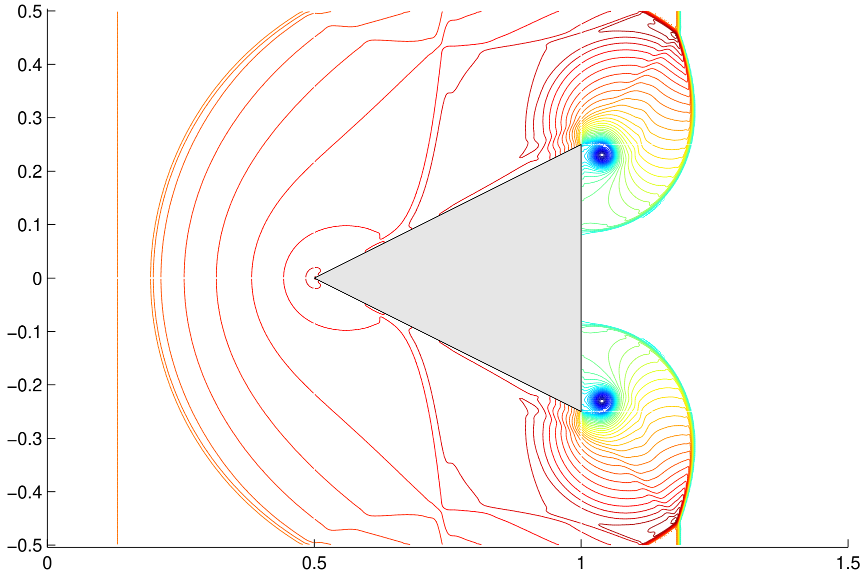

Figure 14: Flow past a triangle at $t = 0.45$ s.

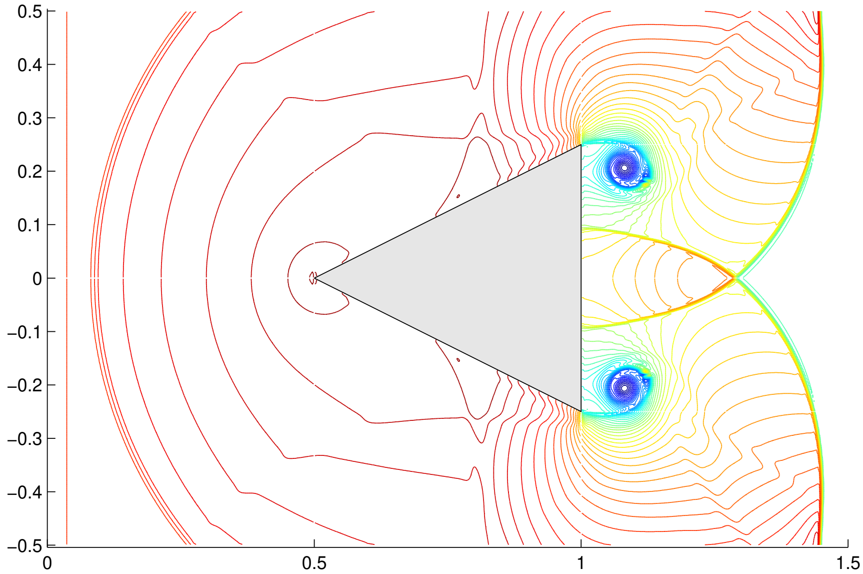

Figure 15: Flow past a triangle at $t = 0.60$ s.

It might be tempting to avoid this effort by transforming the quadrilateral mesh in the physical domain onto a logically rectangular and uniform computational mesh. The transformed conservation law, involving metric terms, could then be discretized on the rectangular mesh in the computational domain. However, this approach has several drawbacks:

First, in order to achieve good accuracy when discretizing the transformed equations, the grid mapping must be smooth. In practice however, for highly non-uniform meshes in the physical domain, this mapping typically is not smooth. Second, when discretizing in the computational domain, it is not obvious how to establish exact conservation with respect to the physical variables. Finally, even if the first two points may be handled in some way, then in order to simulate the transformed conservation law in the computational domain, transformed Riemann problems need to be solved. These artificial Riemann problems might be more difficult to deal with than the original one describing some natural process.

In combination with the mesh redistribution method by Tang and Tang [19], the bi-hyperbolic FV-scheme may be used as an efficient, robust and competitive, adaptive simulation tool for realistic flow simulations.

It might also be possible to use bi-hyperbolic reconstructions as basis functions in a discontinuous Galerkin framework [3]. Goal oriented error control and adaptivity via duality could be exploited.

Acknowledgement

Support by the European network HYKE, funded by the EC as contract HPRN-CT-2002-00282, is acknowledged.

References

[1] Artebrant R. and Schroll J. (2003) *Conservative logarithmic reconstructions and finite volume methods*, submitted.

[2] Artebrant R. and Schroll J. (2004) *A limiter free, globally third order reconstruction*, in preparation.

[3] Cockburn B., Karniadakis G.E. and Shu C.-W. (eds.) (2000) *Discontinuous Galerkin methods. Theory, computation and applications.*, Lecture Notes in Computational Science and Engineering. Springer, Berlin.

[4] Donat R. and Marquina A. (1996) *Capturing shock reflections: An improved flux formula*, J. Comput. Phys., v. 125 (1996) pp 42-58.

[5] Gottlieb S., Shu C.W. and Tadmor E. (2001) *Strong Stability-preserving high order time discretization methods*, SIAM Review **43** pp. 89–112.

[6] Haas J.-F. and Sturtevant B. (1987) *Interaction of weak shock waves with cylindrical and spherical gas inhomogeneities*, J. Fluid. Mech. **181** pp. 41–76.

[7] Karni S. (1994) *Multicomponent flow calculations by a consistent primitive algorithm*, J. Comput. Phys. 112 (1994), pp 31–43.

[8] Kröner D. (1997) *Numerical schemes for conservation laws*, Advances in Numerical Mathematics, Wiley-Teubner, Chichester & Stuttgart.

[9] Langseth J. O. and LeVeque R. J. (1997) *A Wave Propagation Method for Three-Dimensional Hyperbolic Conservation Laws*, Preprint 1997-003 available from Conservation laws preprint server `http://www.math.ntnu.no/conservation/`. Published in revised form in J. Comput. Phys., **165**, pp 126–166.

[10] LeVeque R.J. (2002) *Finite volume methods for hyperbolic problems*, Cambridge University Press, Cambridge.

[11] Li S. and Petzold L. (1997) *Moving mesh methods with upwinding schemes for timedependent PDEs*, J. Comput. Phys., v. 131 (1997) pp 368-377.

[12] Marquina A. (1994) *Local piecewise hyperbolic reconstruction of numerical fluxes for nonlinear scalar conservation laws*, SIAM J. Sci. Comput. **15**, no. 4, pp. 892–915.

[13] Marquina A. and Mulet P. (2003) *A flux split algorithm applied to conservative models for multicomponent compressible flows*, J. Comput. Phys. **185** pp. 120–138.

[14] M. Samimy, K. S. Breuer, L. G. Leal and P. H. Steen (eds.) (2003), *A Gallery of Fluid Motion*, Cambridge University Press, Cambridge.

[15] Schroll H.J. (2003) *High resolution schemes for hyperbolic conservation laws*, to appear in J. Sci. Comp.

[16] Shu C.W. (1998) *Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws*, In Quateroni A. (ed.) *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*, Lecture Notes in Mathematics, Springer, Berlin, pp. 325–432.

[17] Schultz-Rinne C.W. (1993) *Classification of the Riemann problem for two-dimensional gas dynamics*, SIAM J. Math. Anal. **24**, no. 1, pp. 76–88.

[18] Tadmor E. (1984) *Numerical viscosity and the entropy condition for conservative difference schemes*, Mathematics of Computation **43** (1984), pp. 369–381.

[19] Tang H. and Tang T. (2003) *Adaptive mesh methods for one- and two-dimensional hyperbolic conservation laws*, SIAM J. Numer. Anal. **41**, no. 2, pp 487–515.

[20] Toro E. F. (1999) *Riemann Solvers and Numerical Methods for Fluid Dynamics, 2nd Edition*, Springer, Berlin.

[21] van Leer B. (1982) *Flux-vector splitting for the Euler equations*, Lecture Notes in Physics, E. Krause (ed.), Springer-Verlag. **170**, p. 507.

[22] `http://www.math.ntnu.no/~andreas/fronttrack/gas/sb/`

[23] `http://www.amath.washington.edu/~claw`