European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS 2000
Barcelona, 11-14 September 2000

COMPUTATION OF COMPRESSIBLE FLOWS ON UNSTRUCTURED STAGGERED GRIDS

Ivo Wenneker*, Guus Segal and Piet Wesseling

Faculty of Information Technology and Systems, Delft University of Technology,
Mekelweg 4, 2628 CD Delft, Holland e-mail: i.wenneker@math.tudelft.nl
* Supported by The Netherlands Organization for Scientific Research (NWO)

Key words: computational fluid dynamics, compressible flows, unstructured grids, staggered grids.

Abstract. *In this paper the discretisation of the 2D Euler equations on unstructured grids is discussed. A staggered positioning of the variables is used, which means that the scalar variables are located at the cell centers whereas the normal component of the vector variables are positioned at the midpoints of the cell faces. Cell face fluxes are obtained from simple upwind and central approximations. The Euler equations are solved in a decoupled way. Numerical results for 1D Riemann problems show that our scheme selects the correct entropy solution. A flow in a channel with a bump is computed to show the capability of the method to compute 2D subsonic flows.*

1 MOTIVATION TO WORK WITH STAGGERED, UNSTRUCTURED GRIDS

In industrial applications, flow computations usually have to be performed in domains of complicated shape. Experience has shown that the generation of unstructured grids is less man-hour consuming than the generation of structured grids. A second advantage of the use of unstructured grids is the relative ease of local refinement. Local (or adaptive) refinement, i.e. refining the grid in the regions where it is necessary, is very difficult on structured grids. As a consequence, at this moment much effort is devoted by industries and universities to develop computing technology using unstructured grids.

In e.g. aeronautics the main focus lies on the development of methods for computing compressible flows. As far as we know, all these methods use a colocated placement of the unknowns in the grid. In many other application fields such as automotive and chemical industries, flows can be considered as incompressible. Using a colocated discretisation and a straightforward discretisation, however, leads to odd-even decoupling of the pressure. To remedy this, artificial stabilizing measures have to be taken. The most popular method that has evolved is the pressure-weighted interpolation of Rhie and Chow [1]. This difficulty does not arise in the compressible case, which is why colocated schemes are prevalent in this area. By pressure-weighted interpolation artificial pressure diffusion is introduced in the mass conservation law. Circumventing this problem of spurious oscillations is possible by employing a staggered placement of the variables, which is why staggered schemes are often used for computing incompressible flows.

Staggered schemes only recently have caught on compressible flows, see e.g. [2] and [3] and references quoted there. This gives a unified method by which both compressible and incompressible regions are computed accurately and efficiently. However, in these references structured grids were used. This present work is a contribution to the development of unstructured grid methods, where we have decided to use a staggered positioning of the variables.

Section 2 focuses on the staggered discretisation of the Euler equations on 2D unstructured grids. Numerical results for both 1D and 2D problems are given in Section 3. The conclusions are summarized in Section 4.

2 DISCRETISATION OF THE EULER EQUATIONS ON A STAGGERED, UNSTRUCTURED 2D GRID

In this section the discretisation of the Euler equations on a 2D staggered and unstructured grid is discussed. First, the 2D Euler equations, in their conservative form, are summarized in Section 2.1. The solution procedure, together with the time integration method, is explained in Section 2.2. The staggered positioning of the variables in the grid is given in Section 2.3. The spatial discretisation of the Euler equations, for which the finite volume method is employed, is the subject of Sections 2.4 to 2.7.

2.1 The 2D Euler equations

The 2D Euler equations are given by:

$$\begin{aligned}\frac{\partial \rho}{\partial t} + \nabla \cdot \mathbf{u}\rho &= 0, &(1)\\ \frac{\partial \rho E}{\partial t} + \nabla \cdot \mathbf{u}(\rho H) &= 0, &(2)\\ \frac{\partial m^\alpha}{\partial t} + (u^\beta m^\alpha)_{,\beta} &= -p_{,\alpha} &(3)\end{aligned}$$

where $(m^x, m^y) = \mathbf{m}$ and $(u^x, u^y) = \mathbf{u}$ are the 2D momentum and velocity vectors. For the momentum equation (3) the Einstein summation-convention is used, so $m^\alpha_{,\alpha} = \nabla \cdot \mathbf{m}$. The other variables have their usual meaning, and the familiar relations

$$\begin{aligned}\mathbf{m} &= \rho \mathbf{u}, &(4)\\ H &= h + \frac{1}{2}(\mathbf{u}\cdot\mathbf{u}), &(5)\\ E &= e + \frac{1}{2}(\mathbf{u}\cdot\mathbf{u}), &(6)\\ h &= \gamma e. &(7)\end{aligned}$$

will be used. The system of equations is closed by the pressure equation (the equation of state):

$$p = \frac{\gamma - 1}{\gamma}\left[\rho H - \frac{1}{2}\rho(\mathbf{u}\cdot\mathbf{u})\right]. \tag{8}$$

2.2 Solution procedure

In contrast with what is customary, we do not make explicit use of the fact that the Euler equations form a coupled system of equations. Instead, we solve for the primary variables one after another, i.e. the Euler equations are considered as evolution equations for these variables. As the primary variables we take $\mathbf{m}$, ρ and ρH. By using first order upwind or central interpolations for the convection term in each of these equations, a very simple scheme results. Note that there is no need to determine numerical fluxes at control volume boundaries by flux-splitting or approximate Riemann solvers. Although such a decoupled approach is common in the fields of incompressible CFD and shallow-water equations, it is rarely seen in the field of compressible CFD. The most conspicuous one is the Jameson-Schmidt-Turkel scheme introduced in [4]. As mentioned before, another major difference with common practice in compressible CFD is that we use a staggered positioning of the variables.

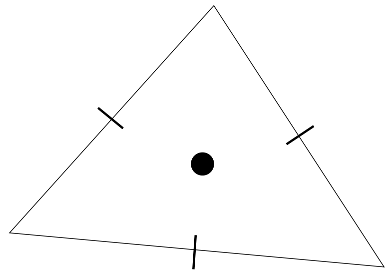

Figure 1: Staggered grid

For the time integration, an explicit or implicit Euler scheme is employed. Supposing that we have the solution vectors at time-level n at our disposal and that the implicit Euler scheme is used, the solution at the next time-level is determined by the following method:

1. $\mathbf{m}^{n+1}$ follows from the discretised momentum equation, where use is made of $\mathbf{m}^n$, $\mathbf{u}^n$ and p^n;

2. ρ^{n+1} follows from the discretised continuity equation, where use is made of ρ^n and $\mathbf{m}^{n+1}$;

3. $(\rho H)^{n+1}$ follows from the discretised energy equation, where use is made of $(\rho H)^n$, $\mathbf{m}^{n+1}$ and ρ^{n+1};

4. the pressure follows from the equation of state.

Since a time-marching scheme (without preconditioning of the time derivative) is used, both stationary and non-stationary problems can be considered. This will be illustrated with the numerical experiments as shown in Section 3. In each of the first three stages, a linear system has to be solved. This is done by a Krylov subspace iterative method (GMRES), preconditioned by ILU.

2.3 Staggered placement of unknowns

We consider only 2D grids consisting of triangles. In Figure 1 the employed staggered placement of variables in the grid is shown. At the cell centroids the scalar variables are located. The projected momentum m, i.e. the component of the momentum vector parallel to the normal of the cell face, is stored at the midpoint of each face. This placement of the variables is similar to the classic staggered scheme on structured grids with quadrilateral cells as introduced by Harlow and Welch [5] and which is used by our group in, for example, [2], [6] and [3].

At every face e there are two normalized normal vectors $\mathbf{n}_e$, one pointing in the opposite direction of the other. By some unambiguous procedure we select at each face one of these

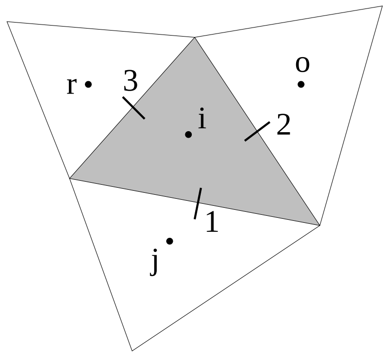

Figure 2: The control volume for the scalar variable in cell i is shaded, and 1, 2 and 3 indicate its faces. The adjacent cells are labeled j, o and r

which will be called the unique normal vector $\mathbf{N}_e$. Since $\mathbf{n}_e$ points either in the same or in the opposite direction of $\mathbf{N}_e$, i.e. $(\mathbf{n}_e \cdot \mathbf{N}_e) = \pm 1$, we have

$$\mathbf{N}_e = (\mathbf{n}_e \cdot \mathbf{N}_e)\mathbf{n}_e; \qquad \mathbf{n}_e = (\mathbf{n}_e \cdot \mathbf{N}_e)\mathbf{N}_e. \tag{9}$$

These identities will be used frequently.

2.4 Continuity equation: discretisation on a 2D unstructured grid

The continuity equation is considered as an evolution equation for the density. Integrating (1) over its control volume T_i, for which we choose triangle i itself, see Figure 2, is done as follows:

$$\begin{aligned} \int_{T_i} \frac{\partial \rho}{\partial t}\, d\mathbf{x} + \int_{T_i} \nabla \cdot \mathbf{u}\rho\, d\mathbf{x} &= \frac{d}{dt}\int_{T_i} \rho\, d\mathbf{x} + \oint_{\partial T_i} (\mathbf{u}\cdot\mathbf{n})\rho\, d\Gamma \approx \\ &\approx \Omega_i \frac{\rho_i^{n+1} - \rho_i^n}{\Delta t} + \sum_{e=1}^{3} u_e \rho_e \bar{l}_e = 0, \end{aligned} \tag{10}$$

with $\mathbf{n}$ the outwards normal at the boundary ∂T_i of the control volume. Superscripts n and $n+1$ refer to the time level, and Δt is the time step. The area of the triangle equals Ω_i, the summation over e is over the three faces of cell i, the normal velocity is given by $u_e = (\mathbf{u}_e \cdot \mathbf{N}_e)$ and

$$\bar{l}_e = l_e(\mathbf{n}_e \cdot \mathbf{N}_e). \tag{11}$$

with l_e the length of the face. Notice that the velocity u_e appearing in the convection term is already located at the cell face, hence no interpolation of the velocity is required.

The velocity u_e is computed using the momentum at the new time level and the velocity at the old time level, i.e. $u_e = m_e^{n+1}/\rho_{e,av}^n$, where $\rho_{e,av}$ follows from a weighted interpolation. For example, with the situation as sketched in Figure 2 we get for $\rho_{1,av}$

$$\rho_{1,av} = \frac{\Omega_i}{\Omega_i + \Omega_j}\rho_j + \frac{\Omega_j}{\Omega_i + \Omega_j}\rho_i. \tag{12}$$

Central interpolations give for the convected quantity ρ at face 1

$$\rho_1 = \frac{1}{2}(\rho_i + \rho_j), \tag{13}$$

and similar expressions hold for ρ_2 and ρ_3. Remark that we use a factor 1/2 in (13) and not, for example, a weighted averaging. By doing this the symmetry of the underlying operator is preserved, i.e. the convective derivative should be approximated by a skew-symmetric operator as was demonstrated by Veldman and Rinzema [7]. The first order upwind approximation for ρ_1 is

$$\rho_1 = \begin{cases} \rho_i & \text{if } (\mathbf{u}_1 \cdot \mathbf{n}_1) = u_1(\mathbf{n}_1 \cdot \mathbf{N}_1) > 0, \text{ i.e. a flow from } i \text{ to } j, \\ \rho_j & \text{if } (\mathbf{u}_1 \cdot \mathbf{n}_1) = u_1(\mathbf{n}_1 \cdot \mathbf{N}_1) < 0, \text{ i.e. a flow from } j \text{ to } i, \end{cases} \tag{14}$$

and similar approximations hold for ρ_2 and ρ_3.

2.5 Energy equation: discretisation on a 2D unstructured grid

The energy equation is considered as an evolution equation for ρH. A completely similar derivation, on the same control volume, as given in the previous section leads to a discretised energy equation of the form

$$\Omega_i \frac{(\rho E)_i^{n+1} - (\rho E)_i^n}{\Delta t} + \sum_e u_e (\rho H)_e \bar{l}_e = 0. \tag{15}$$

The velocity u_e is computed from the momentum and density at the new time level using the method as described in the previous section. The convected energy term ρH is treated completely similar as the convected density term in the continuity equation, cf. equations (13) and (14). In order to be able to solve (15) for primary variable $(\rho H)_i^{n+1}$, the relation

$$(\rho E)_i = \frac{1}{\gamma}(\rho H)_i + \frac{\gamma - 1}{2\gamma}\rho_i (\mathbf{u} \cdot \mathbf{u})_i, \tag{16}$$

which can be derived from (5), (6) and (7), is used. The computation of the term $\rho_i(\mathbf{u} \cdot \mathbf{u})_i$ will be discussed in Section 2.7.

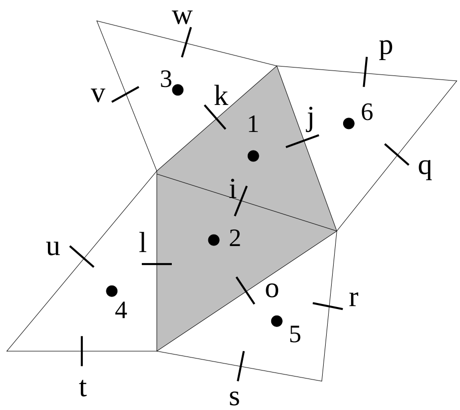

Figure 3: The control volume for the normal momentum component at face i is shaded. The numbers denote the cells, while the faces of the stencil are indicated by letters

2.6 Momentum equation: discretisation on a 2D unstructured grid

As said before, at all faces in the grid the momentum component parallel to the normal vector at this face is stored. This is different from the usual approach, in which at each gridpoint one stores and solves for both the x- and y-component of the momentum vector. Another aspect that is different from common practice in compressible CFD, although more recent schemes like the AUSM scheme [8] and Jameson's CUSP scheme [9], [10] utilize it as well, is the splitting of the flux in a separate convection and pressure term.

We choose the control volume T_i for the normal momentum component $m_i = (\mathbf{m}_i \cdot \mathbf{N}_i)$ at face i to consist of the two triangles adjacent to this face, see Figure 3.

Obtaining an equation for m_i is done by projection of (3) on N_i^α and integration over T_i. With $\mathbf{N}_e$ the unique normal at face e, and $\mathbf{n}_e$ the outwards pointing normal with respect to T_i, the convection term is handled as follows:

$$\begin{aligned}\int_{T_i} (u^\beta m^\alpha N_i^\alpha)_{,\beta}\, d\mathbf{x} &= \int_{T_i} \nabla \cdot [\mathbf{u}(\mathbf{m}\cdot\mathbf{N}_i)]\, d\mathbf{x} = \oint_{\partial T_i} (\mathbf{u}\cdot\mathbf{n})(\mathbf{m}\cdot\mathbf{N}_i)\, d\Gamma \approx \\ &\approx \sum_e (\mathbf{u}_e\cdot\mathbf{n}_e)(\mathbf{m}_e\cdot\mathbf{N}_i) l_e = \sum_e u_e (\mathbf{m}_e\cdot\mathbf{N}_i)\bar{l}_e \qquad (17)\end{aligned}$$

where we used the Gauss theorem and relations (9) and (11). The summation is over the faces $e \in \{k, l, o, j\}$. This results in the following discretised momentum equation:

$$\Omega_i \frac{m_i^{n+1} - m_i^n}{\Delta t} + \sum_e u_e (\mathbf{m}_e \cdot \mathbf{N}_i)\bar{l}_e = -\Omega_i (\nabla p \cdot \mathbf{N})_i, \qquad (18)$$

with $\Omega_i = (\Omega_1 + \Omega_2)$ the area of the control volume.

Picard linearisation of the convection term is used, which means that u_e is computed using the momentum and velocity at the old time level. For the discretisation of the convection term the so-called reconstruction procedure is introduced. With $\mathbf{t}_k$ the tangential vector at face k, there exists unique solutions with respect to the reconstruction coefficients ξ for the relations

$$\mathbf{t}_k = \xi_i \mathbf{N}_i + \xi_j \mathbf{N}_j \qquad \text{and} \qquad \mathbf{t}_k = \xi_v \mathbf{N}_v + \xi_w \mathbf{N}_w. \tag{19}$$

The momentum vector at face k can be decomposed as

$$\mathbf{m}_k = m_k \mathbf{N}_k + \tilde{m}_k \mathbf{t}_k. \tag{20}$$

where $\tilde{m}_k = (\mathbf{m}_k \cdot \mathbf{t}_k)$ is the tangential momentum component and m_k denotes, as usual, the normal momentum component. Hence we can write

$$\mathbf{m}_k \cdot \mathbf{N}_i = m_k(\mathbf{N}_k \cdot \mathbf{N}_i) + \tilde{m}_k(\mathbf{t}_k \cdot \mathbf{N}_i). \tag{21}$$

The tangential momentum component $\tilde{m}_k$ can be deduced from a central or first order upwind interpolation. Considering a first order upwind interpolation and assuming that the flow is from cell 1 to 3, the tangential momentum follows from

$$\tilde{m}_k = \mathbf{m}_k \cdot \mathbf{t}_k = \mathbf{m}_k \cdot (\xi_i \mathbf{N}_i + \xi_j \mathbf{N}_j) \approx \xi_i m_i + \xi_j m_j \tag{22}$$

and, if the flow moves in the opposite direction, this term is evaluated from

$$\tilde{m}_k \approx \xi_v m_v + \xi_w m_w. \tag{23}$$

In the case of central interpolation, the tangential momentum component follows from

$$\tilde{m}_k \approx \frac{1}{2}(\xi_i m_i + \xi_j m_j + \xi_v m_v + \xi_w m_w). \tag{24}$$

The contributions from the other faces is computed similarly.

For the numerical evaluation of the projected pressure gradient $(\nabla p \cdot \mathbf{N})_i$, the path-integral formulation as discussed in [11] is used. This formulation is based on the idea that first an approximation of the gradient vector $(\nabla p)_i$ is made, after which the inner product with $\mathbf{N}_i$ is taken. The heart of the path-integral formulation is formed by the integral identity

$$p_b - p_a = \int_a^b \nabla p \cdot d\mathbf{x}. \tag{25}$$

Discretisation of this expression leads to

$$p_b - p_a \approx (\nabla p)_{ab} \cdot (\mathbf{x}_b - \mathbf{x}_a), \tag{26}$$

where point ab is assumed to be located somewhere between $\mathbf{x}_a$ and $\mathbf{x}_b$. For linear pressure fields of the form $p = p_0 + p_x x + p_y y$ the '$\approx$'-sign can be replaced by a '$=$'-sign for every point in the domain, hence the method will be exact for linear pressure fields. We will show how to proceed from (26) to arrive at an approximation for $(\nabla p \cdot \mathbf{N})_i$.

Application of (26) on the path from cell-center 1 to cell-center 2, see Figure 3, gives

$$p_2 - p_1 \approx (\nabla p)_i \cdot (\mathbf{x}_2 - \mathbf{x}_1), \tag{27}$$

where we note that in smooth grids the midpoint of face i is located in the vicinity of the line between $\mathbf{x}_1$ and $\mathbf{x}_2$. Equation (27) alone is not sufficient to obtain $(\nabla p)_i$; for this one additional relation is required. Using (26) on the path from 5 to 3 leads to

$$p_3 - p_5 \approx (\nabla p)_i \cdot (\mathbf{x}_3 - \mathbf{x}_5), \tag{28}$$

and, similarly,

$$p_4 - p_6 \approx (\nabla p)_i \cdot (\mathbf{x}_4 - \mathbf{x}_6). \tag{29}$$

Combination of these two expressions leads to

$$p_3 - p_6 + p_4 - p_5 \approx \nabla p_i \cdot (\mathbf{x}_3 - \mathbf{x}_6 + \mathbf{x}_4 - \mathbf{x}_5). \tag{30}$$

Solution of the system (27)–(30) results in a discretisation of $(\nabla p)_i$. The inner product with $\mathbf{N}_i$ is easily taken, leading to an expression of the form

$$(\nabla p \cdot \mathbf{N})_i = \sum_{j=1}^{6} \gamma_j p_j. \tag{31}$$

With $\mathbf{x}_j = (x_j, y_j)$, with $\mathbf{N}_i = (N_x, N_y)$ and defining

$$\begin{aligned} a_{11} &= x_2 - x_1; & a_{12} &= y_2 - y_1; \\ a_{21} &= x_3 - x_6 + x_4 - x_5; & a_{22} &= y_3 - y_6 + y_4 - y_5, \end{aligned}$$

the gradient coefficients γ_j follow from

$$\begin{aligned}\gamma_1 = -\gamma_2 &= \frac{a_{21}N_y - a_{22}N_x}{a_{11}a_{22} - a_{12}a_{21}},\\ \gamma_3 = \gamma_4 = -\gamma_5 = -\gamma_6 &= \frac{a_{11}N_y - a_{12}N_x}{a_{11}a_{22} - a_{12}a_{21}}.\end{aligned} \tag{32}$$

A special situation occurs when vector $(\mathbf{x}_2 - \mathbf{x}_1)$ is parallel to $\mathbf{N}_i$. The path-integral method then reduces to:

$$(\nabla p \cdot \mathbf{N})_i = \frac{p_2 - p_1}{|\mathbf{x}_2 - \mathbf{x}_1|}. \tag{33}$$

When one or more cells in the stencil of Figure 3 are absent, as occurs in the vicinity of a boundary, the corresponding paths in (28)–(29) have to be modified.

2.7 Pressure equation: discretisation on a 2D unstructured grid

The discretised equation of state reads:

$$p_i = \frac{\gamma - 1}{\gamma}\left[(\rho H)_i - \frac{1}{2}\rho_i(\mathbf{u}\cdot\mathbf{u})_i\right]. \tag{34}$$

The computation of the kinetic energy term is not trivial since the velocity is not located at cell-center i. Rewriting the kinetic energy in terms of primary variables gives

$$\frac{1}{2}\rho_i(\mathbf{u}\cdot\mathbf{u})_i = \frac{1}{2}\frac{(\mathbf{m}\cdot\mathbf{m})_i}{\rho_i}, \tag{35}$$

where the difficulty now lies in finding an approximation for $(\mathbf{m}\cdot\mathbf{m})_i$. In order to compute this term, we will restrict ourselves to the use of the normal momentum components at the three faces of cell i. With $\mathbf{m}_i$ the momentum vector at cell-center i, we know that $m_e = (\mathbf{m}_e \cdot \mathbf{N}_e) \approx (\mathbf{m}_i \cdot \mathbf{N}_e)$ holds for its three faces e. We choose $\mathbf{m}_i$ such that the least squares functional

$$\mathcal{F}(\mathbf{m}_i) = \sum_e l_e^2[m_e - (\mathbf{m}_i \cdot \mathbf{N}_e)]^2 \tag{36}$$

is minimal.

3 NUMERICAL RESULTS

In this section we will present numerical results obtained with the staggered method as discussed above. In Section 3.1 we will show that the decoupling of the equations does not prevent the scheme from converging to the entropy solution for 1D Riemann problems. In Section 3.2 results for a subsonic 2D problem, namely a channel with bump, will be discussed.

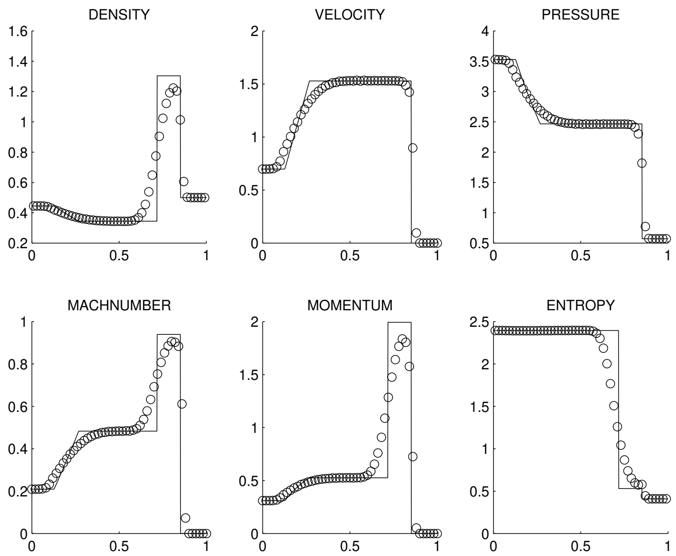

Figure 4: Test case of Lax. Comparison of the exact solution with the staggered first order upwind results for $t = 0.140$ and $\lambda = 0.12$

3.1 Numerical results for 1D Riemann problems

As already mentioned in Section 2.2, it is customary to consider the Euler equations as a coupled system. Therefore one might wonder whether convergence to the entropy solution is ensured for a scheme like ours. As for most schemes, no convergence proof is available, hence we try to validate it by doing numerical experiments. For an important 1D standard test case for compressible codes, namely the Riemann problem, we will compare the numerical solution obtained using the 1D staggered method with the exact solution. The Riemann problem consists of solving the 1D Euler equations for a special kind of initial conditions, namely a constant left state for $x < x_d$ and a constant right state for $x > x_d$.

To this aim we have discretised the 1D Euler equations on a 1D equidistant staggered grid and used an explicit Euler time integration scheme together with the solution algorithm discussed in Section 2.2.

In the results given here we have put the location of the initial discontinuity at $x_d = 0.5$ and the number of grid cells is set to 50. A first order upwind scheme is used for all three conservation laws.

Frequently used is the test case of Lax [12], with initial state

$$U_l = [u_l, p_l, \rho_l] = [0.698,\ 3.528,\ 0.445] \qquad U_r = [u_r, p_r, \rho_r] = [0.0,\ 0.571,\ 0.5]. \quad (37)$$

In Figure 4 the results of the staggered scheme are compared with the exact solution at $t = 0.140$. The continuous line represents the analytical solution, whereas the small

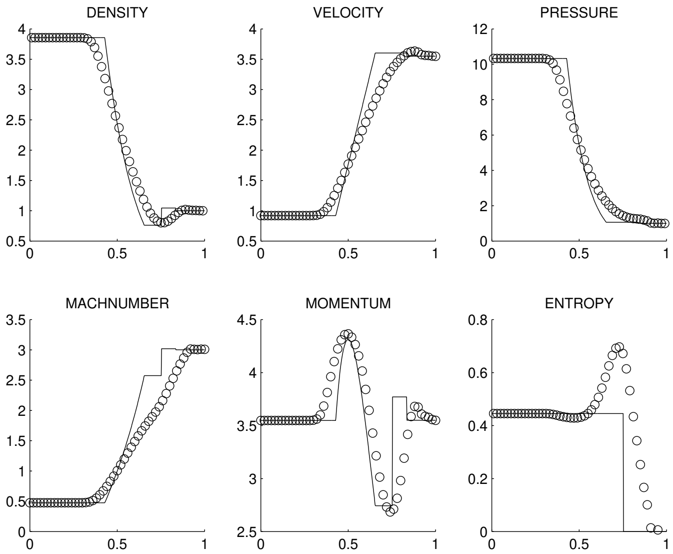

Figure 5: Mach 3 test case. Comparison of the exact solution with the staggered first order upwind results for $t = 0.068$ and $\lambda = 0.13$

circles are the solution computed using the staggered scheme. The main difficulties in computing the correct solution to this test case are the contact discontinuity and the strong shock, which are relatively close to each other. We see that our scheme converges to the correct weak solution. The contact discontinuity is, as is common for first order upwind methods, smeared due to divergence of the characteristics. Since characteristics converge into shocks, this produces a steepening effect resulting in a crisp shock resolution. Due to the use of relatively few grid points and the small distance between the contact discontinuity and the shock, the density and momentum do not reach the analytical maximum value. Increasing the number of grid points enhances this (results not shown). Using the eye-ball norm the (explicit) scheme turns out to be stable for a CFL-number $\sigma = \max(|u| + a)\frac{\Delta t}{\Delta x} = \max(|u| + a)\lambda$ smaller than approximately 0.5.

In the preceding test case, the flow remained subsonic. Supersonic flow may bring additional numerical difficulties. In the supersonic flow problem posed by Arora and Roe [13], the Mach number rises to a value of approximately three. This test case is a notorious one. For this problem the Roe scheme [14] violates the entropy condition. The sonic entropy fix, introduced by Harten [15], is necessary to cure this problem. The price one has to pay is the appearance of a sonic glitch. Also the results from the Osher scheme, the Van Leer scheme and the AUSM scheme show a sonic glitch, although for these schemes no entropy fix is necessary. The sonic glitches in the Van Leer and AUSM scheme can be seen at pages 392 – 395 of Laney [16]. The initial left and right state are defined for the Mach 3 test case as:

$$U_l = [u_l, p_l, \rho_l] = [0.92,\ 10.333,\ 3.857] \qquad U_r = [u_r, p_r, \rho_r] = [3.55,\ 1.0,\ 1.0]. \quad (38)$$

The exact solution together with the numerical one is shown in Figure 5. One observes that the flow is dominated by a huge expansion fan followed by a contact discontinuity and a small shock. We observe that our method converges to the entropy solution and that no sonic glitch has appeared. Using the eyeball-norm the (explicit) scheme turns out to be stable for a CFL-number smaller than approximately 0.6.

3.2 Numerical results for a channel with bump

The problem of the flow in a channel with a 10 % circular arc bump was chosen to evaluate the code for subsonic steady state modeling. The grid used is shown in Figure 6. The boundaries of the channel are divided into 72 × 24 nodal points, leading to a grid consisting of 3982 cells, 6069 faces and 2088 vertices. At the upper and lower boundaries of the wall, the solid-wall boundary condition is applied. At the inflow boundary, on the left side, the Mach number is put to 0.5. An implicit Euler time integration scheme is used. For all equations the first order upwind scheme is used.

The convergence behaviour of the density residual, in the L_2 norm, is shown in Figure 7.

For inviscid and subsonic flow the Mach number contour plot, shown in Figure 8, should be symmetric. The asymmetry is due to the numerical diffusion introduced by the first order upwind scheme. A similar asymmetry can be observed in the first order Godunov method results that Eidelman et al. [17] obtained on a 99 × 33 grid. The Mach number at the lower surface near the outflow in our results is 0.475, whereas the Mach number is 0.425 at the same location in the results of Eidelman.

The computation is repeated using a 72 × 24 structured grid with the structured grid method described in [2]. Also in this computation a first order upwind scheme is used. The Mach number distribution at the upper and lower surface of the channel is compared for the structured and unstructured method in Figure 9. We clearly observe that the results obtained on the unstructured grid contain more numerical diffusion than the ones obtained on a structured grid.

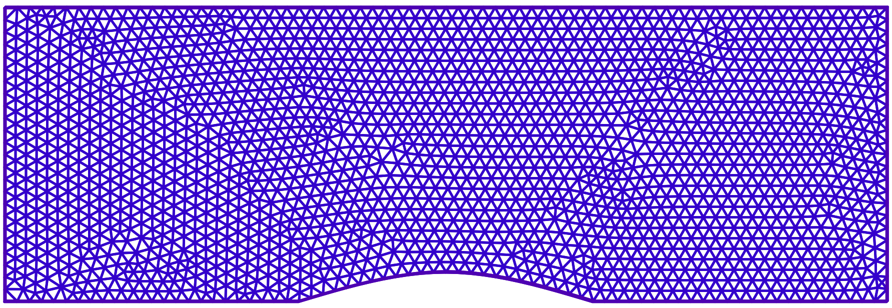

Figure 6: Grid for the channel with bump

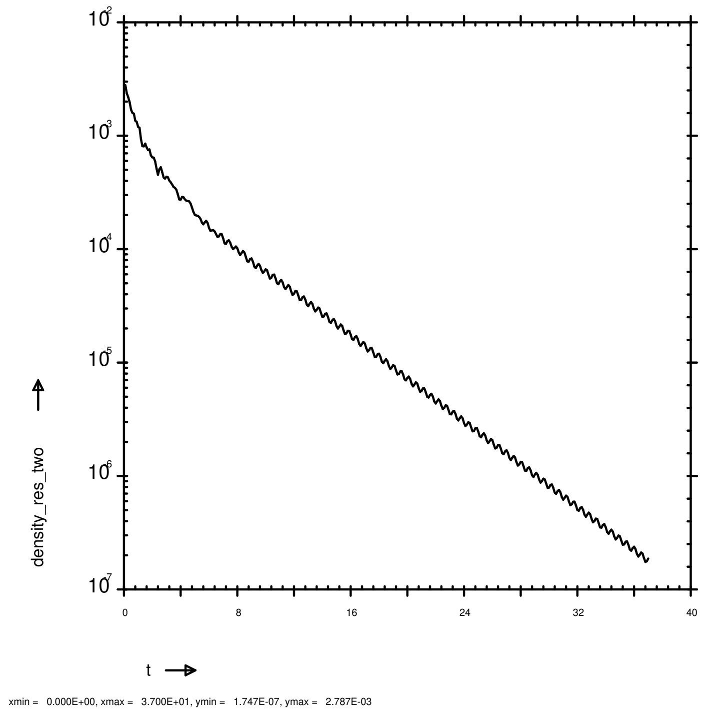

Figure 7: Convergence behaviour of the density residual

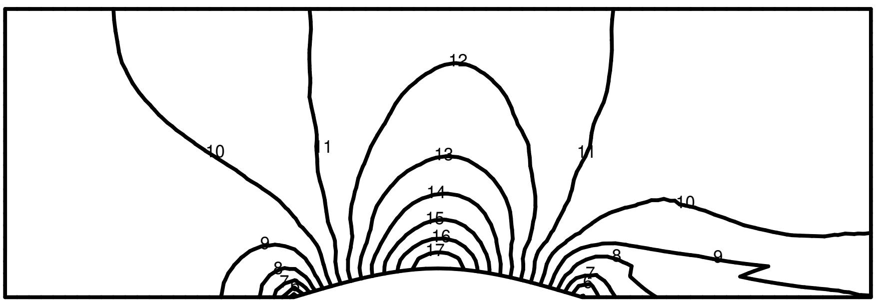

Figure 8: Mach contour plot. Level 10 corresponds to a Mach number of 0.5, and the difference between each level is 0.025

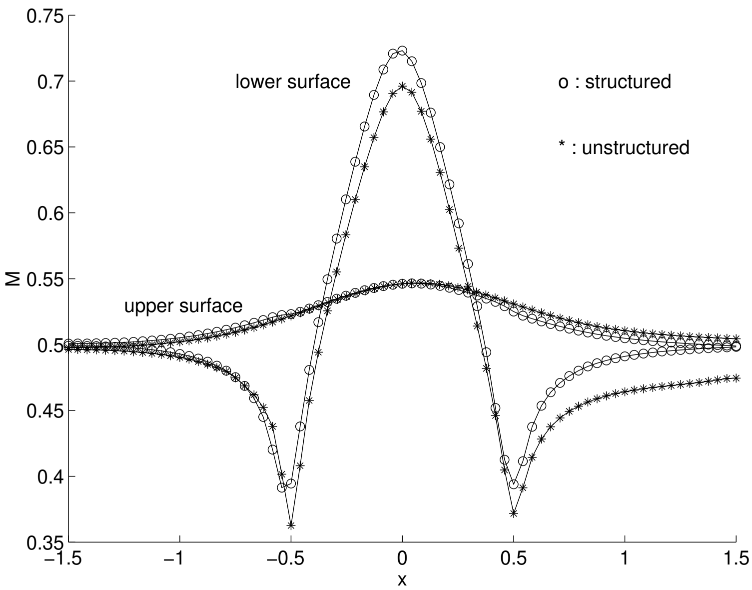

Figure 9: Mach number at the upper and lower boundary of the channel

4 CONCLUSIONS

In this paper, the discretisation of the 2D Euler equations on an unstructured grid is discussed. For the positioning of the variables, a staggered location is adopted. The scalar variables are located at the cell centers, whereas the normal momentum components are positioned at the cell faces. The Euler equations are solved in a decoupled way such that central or first order upwind interpolations suffice in each of the equations. This results in a scheme that is much simpler than the prevailing schemes for the Euler equations, since only central and/or upwind differences are required, thereby avoiding the necessity to determine numerical fluxes at control volume boundaries by flux-splitting or approximate Riemann solvers.

A time stepping algorithm is used such that both stationary as instationary problems can be solved.

Numerical solutions to 1D Riemann problems show that our staggered first order upwind scheme selects the correct entropy solution. A flow through a channel with a bump is computed to show the capability of the method to compute 2D subsonic flows.

REFERENCES

[1] C.M. Rhie and W.L. Chow, "Numerical study of the turbulent flow past an airfoil with trailing edge separation", *AIAA J.*, **21**, 1525–1532 (1983).

[2] H. Bijl and P. Wesseling, "A unified method for computing incompressible and compressible flows in boundary-fitted coordinates", *J. Comp. Phys.*, **141**, 153–173 (1998).

[3] D.R. van der Heul, C. Vuik and P. Wesseling, "A staggered scheme for hyperbolic conservation laws applied to unsteady sheet cavitation", *Computing and Visualization in Science*, **2**, 63–68, (1999).

[4] A. Jameson and W. Schmidt and E. Turkel, "Numerical Solution of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time Stepping Scheme", *AIAA Paper 81-1259*, (1981).

[5] F.H. Harlow and J.E. Welch, "Numerical calculation of time-dependent viscous incompressible flow of fluid with a free surface", *The Physics of Fluids*, **8**, 2182–2189 (1965).

[6] P. Wesseling and A. Segal and C.G.M. Kassels, "Computing flows on general three-dimensional nonsmooth staggered grids", *J. Comp. Phys.*, **149**, 333–362, (1999).

[7] A.E.P. Veldman and K. Rinzema, "Playing with nonuniform grids", *J. Eng. Math.*, **26**, 119–130, (1992).

[8] M.-S. Liou and C.J. Steffen, "A new flux splitting scheme", *J. Comp. Phys.*, **107**, 23–39, (1993).

[9] A. Jameson, "Analysis and design of numerical schemes for gas dynamics, 1: artificial diffusion, upwind biasing, limiters and their effect on accuracy and multigrid convergence", *Int. J. Comp. Fluid Dyn.*, **4**, 171–218, (1995).

[10] A. Jameson, "Analysis and design of numerical schemes for gas dynamics, 2: artificial diffusion and discrete shock structure", *Int. J. Comp. Fluid Dyn.*, **5**, 1–38, (1995).

[11] P. Wesseling and A. Segal and C.G.M. Kassels and H. Bijl, "Computing flows on general two-dimensional nonsmooth staggered grids", *J. Eng. Math.*, **34**, 21–44, (1998).

[12] P.D. Lax, "Weak solutions of nonlinear hyperbolic equations and their numerical approximation", *Comm. Pure Appl. Math.*, **7**, 159–193, (1954).

[13] M. Arora and P.L Roe, "A well-behaved TVD limiter for high-resolution calculations of unsteady flow", *J. Comp. Phys.*, **132**, 3–11, (1997).

[14] P.L. Roe, "Approximate Riemann solvers, parameter vectors, and difference schemes", *J. Comp. Phys.*, **43**, 357–372, (1981).

[15] A. Harten, "High resolution total-variation-stable finite-difference schemes", *SIAM J. Num. Anal.*, **21**, 1–23, (1984).

[16] C.B. Laney, *Computational Gasdynamics*, Cambridge University Press, UK, (1998).

[17] S. Eidelman and P. Colella and R.P. Shreeve, "Application of the Godunov method and its Second-Order Extension to Cascade Flow Modelling", *AIAA J.*, **22**, 1609–1615, (1984).