Eidgenössische
Technische Hochschule
Zürich

Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Swiss Federal Institute of Technology Zurich

Second Order Accurate Boundary Treatment for Cartesian Grid Methods

H. Forrer

Research Report No. 96-13
September 1996

Seminar für Angewandte Mathematik
Eidgenössische Technische Hochschule
CH-8092 Zürich
Switzerland

Second Order Accurate Boundary Treatment for Cartesian Grid Methods

H. Forrer

Seminar für Angewandte Mathematik
Eidgenössische Technische Hochschule
CH-8092 Zürich
Switzerland

Research Report No. 96-13 September 1996

Abstract

The Euler equations describe the flow phenomena of compressible inviscid gas dynamics. We simulate such flows using a higher order Cartesian grid method together with a special treatment for the cells cut by the boundary of a body.

We describe a new method for the treatment of the boundary where these cut boundary cells are maintained as whole cells rather than as cut cells, thus avoiding stability problems. The method is second order accurate but not strictly conservative, but we can show that this error in the conservation does not lead to spurious phenomena on some representative test calculations.

The advantages of the new boundary treatment are that it is second order accurate, that it is independent of the applied method, and that it can easily be extended to three-dimensional calculations.

Keywords: Cartesian grid methods, gas dynamics, boundary treatment, CLAWPACK package

Subject Classification: 35, 65, 76

1 Introduction

A Cartesian grid method consists of a standard method for the regular cells and a special treatment for the boundary cells. An advantage of using Cartesian grids is that they can rely on fast computer architectures like vector or parallel computers. With their boundary treatment they are flexible, i.e. they can be used for flow simulations around complicated geometries. In the literature we can find Cartesian grid methods with different approaches for the boundary treatment, which are of first order only. Berger and LeVeque [1] use rotated boxes to get stability, Colella [2] uses flux-redistribution procedures, and Quirk [3] uses merging procedures. We have described these approaches briefly in [4]. The main problem is to avoid instability due to small boundary cells and to achieve high order of accuracy along the boundary. In [4] we presented such a scheme. In this paper we want to give a new method for the boundary treatment which is stable, second order accurate and simpler than our method given in [4]. Especially it will be much easier to extend this new approach to three space dimensions.

The compressible inviscid flow in two space dimensions is described by the Euler equations:

$$U_t + F_x + G_y = 0, \tag{1}$$

$$U = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho e \end{pmatrix}, \quad F = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(\rho e + p) \end{pmatrix}, \quad G = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(\rho e + p) \end{pmatrix},$$

$$p = (\gamma - 1)(\rho e - \frac{1}{2}\rho(u^2 + v^2)).$$

Where ρ is the mass density, $(u, v)^T$ the velocity vector, e the energy density, p the pressure and $\gamma = 1.4$.

It is a system of nonlinear hyperbolic equations. Thus one has to use a method which is able to treat shocks. As a discretization of the domain, we use a Cartesian grid. To do this, let h be a grid parameter and set the points $\mathbf{z}_{ij} = (x_i, y_j)$ with $x_i = x_0 + h \cdot i, \quad y_j = y_0 + h \cdot j, \quad i, j \in \mathbb{Z}$. The regular grid cell C_{ij} is then given by:

$$C_{ij} = [x_i, x_{i+1}] \times [y_j, y_{j+1}]\,. \tag{2}$$

Let the vector U_{ij}^0 be a weighted mean over the cell C_{ij} of the initial data $U = (\rho, \rho u, \rho v, \rho e)^T$ for the differential equation (1) at time t_0:

$$U_{ij}^0 = \frac{1}{h^2} \int_{C_{ij}} U(x, y, t_0) dx\, dy. \tag{3}$$

An integration of the equation (1) over the cell C_{ij} yields a method to advance an approximate solution U_{ij}^n at time t_n into an approximate solution U_{ij}^{n+1} at time

$t_{n+1} = t_n + \Delta t$.

$$U_{ij}^{n+1} = U_{ij}^n + \frac{\Delta t}{h}(F_{ij}^n - F_{i+1,j}^n + G_{ij}^n - G_{i,j+1}^n), \tag{4}$$

which together with a suitable flux solver represents a standard finite volume method capable of treating shocks.

A common method to calculate the fluxes F_{ij}, G_{ij} (F_{ij} is the flux between the cells $C_{i-1,j}$ and C_{ij} and G_{ij} is the flux between the cells $C_{i,j-1}$ and C_{ij}) is LeVeque's multidimensional method [5], which is of second order accuracy and stable up to a Courant-number $cfl = 1.0$ because of taking into account transverse fluxes. The Courant-number is defined by:

$$cfl = \frac{\Delta t \cdot v_{max}}{h}, \tag{5}$$

where v_{max} is the maximum characteristic speed occurring in the solution. We will use this scheme for the numerical experiments.

If some body is put into a Cartesian grid, then this makes it necessary to treat the cells which are cut by the boundary of the body. Colella [2] gave a method which made it possible to treat these boundary cells like regular cells, thus avoiding stability problems for small cut cells. The method is first order accurate along the boundary. Our new method also treats boundary cells as whole cells and is second order accurate also along the boundary.

Our new approach is motivated by the fact that for an inviscid flow a straight reflecting wall boundary behaves like a symmetry line. Suppose the straight boundary line goes through the point $\mathbf{z}$ and has a normal vector $\mathbf{n} = (-\sin\alpha, \cos\alpha)^t$ pointing into the flow field. Now we can reflect a point $\mathbf{p} = (x, y)^t$ from the physical area into a point $\mathbf{r}_{\mathbf{z},\alpha}(\mathbf{p})$ in the area beyond the boundary by

$$\mathbf{r}_{\mathbf{z},\alpha}(\mathbf{p}) = \mathbf{p} - 2(\mathbf{p} - \mathbf{z}, \mathbf{n}) \cdot \mathbf{n}, \text{ where } \mathbf{n} = (-\sin\alpha, \cos\alpha)^t, \tag{6}$$

and $(.,.)$ denotes the scaler product. Introducing the reflection matrix

$$R_\alpha = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(2\alpha) & \sin(2\alpha) & 0 \\ 0 & \sin(2\alpha) & -\cos(2\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{7}$$

we can extrapolate the flow field $U(\mathbf{p}, t)$ from the point $\mathbf{p}$ to the point $\mathbf{r}_{\mathbf{z},\alpha}(\mathbf{p})$ with

$$U(\mathbf{r}_{\mathbf{z},\alpha}(\mathbf{p}), t) = R_\alpha \cdot U(\mathbf{p}, t), \tag{8}$$

such that the extrapolated solution fulfils the governing equations (cf. Figure 1).

This is possible, because for the velocity vector $\mathbf{u}$ along the reflecting wall boundary we have $(\mathbf{u}, \mathbf{n}) = 0$.

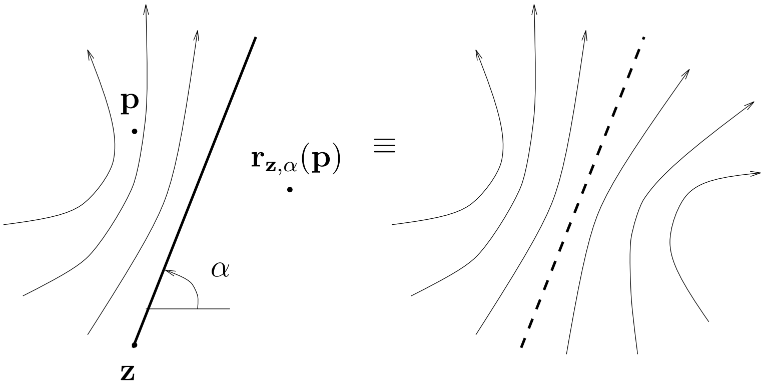

Figure 1: *Symmetry line of a wall boundary for an inviscid flow.*

In the next section we describe the new boundary treatment for the case of a reflecting boundary in one space dimension and give some numerical results concerning the order of accuracy and the conservation. In the third section we explain the method for wall boundaries in two space dimensions and describe a standard treatment for the outer domain boundary. We will also show some numerical results for a flow around a cylinder or a NACA0012 airfoil.

2 One space dimension

The new algorithm for the boundary treatment is easier to explain in one space dimension. So we first consider the Euler equations in 1D.

$$U_t + F_x = 0, \tag{9}$$

$$U = \begin{pmatrix} \rho \\ \rho u \\ \rho e \end{pmatrix}, \quad F = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ u(\rho e + p) \end{pmatrix}, \quad p = (\gamma - 1)(\rho e - \frac{1}{2}\rho u^2). \tag{10}$$

For a discretization we divide the real axes into intervals:

$$C_i = [x_i, x_{i+1}], \quad x_i = x_0 + h \cdot i. \tag{11}$$

We represent the initial data at time t_0 by weighted means:

$$U_i^0 = \frac{1}{h} \int_{C_i} U(x, t_0) dx. \tag{12}$$

An integration of equation (9) yields a finite volume method

$$U_i^{n+1} = U_i^n + \frac{\Delta t}{h}(F_i^n - F_{i+1}^n) \tag{13}$$

to advance the numerical solution U_i^n at time $t_n = t_0 + \Delta t \cdot n$ by a time step Δt. A standard method for the numerical calculation of the flux F_i^n which is second order accurate (and therefore needs some limiters for the gradients) usually needs flow variables of the cells $C_{i-2}, C_{i-1}, C_i, C_{i+1}$.

2.1 Description of the boundary treatment

Suppose, that we have a reflecting wall from the left at the position $x = a$, where $x_{i_w} < a < x_{i_w+1}$ for a certain $i_w \in \mathbb{Z}$. The intervals (11) are divided into four types as follows. $C_{i_w-3}, C_{i_w-4}, \ldots$ are empty intervals. C_{i_w-2} and C_{i_w-1} are ghost intervals, C_{i_w} is a boundary interval and $C_{i_w+1}, C_{i_w+2}, \ldots$ are regular intervals (cf. Figure 2).

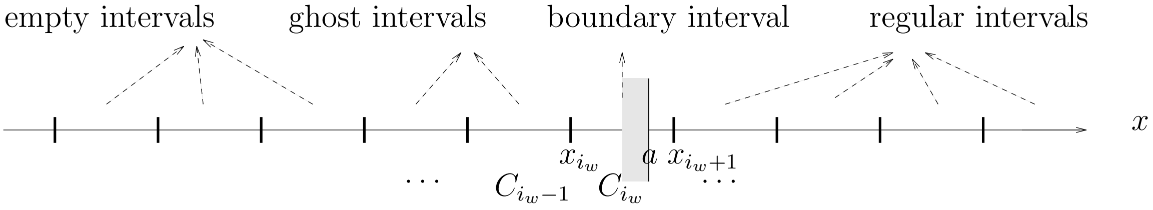

Figure 2: *Definition of the different intervals near a wall boundary.*

The reflection matrix R in one space dimension is given by

$$\begin{pmatrix} \rho \\ -\rho u \\ \rho e \end{pmatrix} = R \cdot \begin{pmatrix} \rho \\ \rho u \\ \rho e \end{pmatrix}, \; R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{14}$$

We define the following auxiliary intervals (cf. Figure 3):

$$\begin{aligned} C_{i_w}^1 &= [a, x_{i_w+1}] & (15) \\ C_{i_w}^2 &= [x_{i_w}, a] & (16) \\ (C_{i_w}^2)_r &= [a, 2a - x_{i_w}] & (17) \\ (C_{i_w-1})_r &= [2a - x_{i_w}, 2a - x_{i_w-1}] & (18) \\ (C_{i_w-2})_r &= [2a - x_{i_w-1}, 2a - x_{i_w-2}] & (19) \end{aligned}$$

At time t_0 we assign to the boundary interval C_{i_w} the following weighted mean:

$$U_{i_w}^0 = \frac{1}{h}\left(\int_{C_{i_w}^1} U(x, t_0) dx + \int_{(C_{i_w}^2)_r} R \cdot U(x, t_0) dx\right). \tag{20}$$

Given an approximation of the solution at time $t_n = t_0 + n \cdot \Delta t$ as a step function over the regular and the boundary intervals $C_{i_w}, C_{i_w+1}, \ldots$, then we want to advance this solution with the method (13). Because of the 4-point stencil for the calculation of F_i^n, this can be done for the values of the regular intervals $U_{i_w+2}^n, U_{i_w+3}^n, \ldots$. In order to advance the values $U_{i_w}^n$ and $U_{i_w+1}^n$, however, we need the fluxes $F_{i_w}^n$ and $F_{i_w+1}^n$. We want to calculate these two fluxes in the same way as all the other ones. Thus we need values $U_{i_w-1}^n, U_{i_w-2}^n$ for the ghost cells. Given some approximation $U^*(x, t_n)$ of the true solution $U(x, t_n)$, we calculate the values $U_{i_w-1}^n, U_{i_w-2}^n$ by the following weighted means:

$$U_{i_w-1}^n = \frac{1}{h}\int_{(C_{i_w-1})_r} R \cdot U^*(x, t_n)dx, \tag{21}$$

$$U_{i_w-2}^n = \frac{1}{h}\int_{(C_{i_w-2})_r} R \cdot U^*(x, t_n)dx. \tag{22}$$

For the approximate solution U^*, we take the piece-wise constant step function given by $U_{i_w}^n, U_{i_w+1}^n, \ldots$. Now we can calculate the fluxes $F_{i_w+1}^n(U_{i_w-1}^n, \ldots, U_{i_w+2}^n)$ and $F_{i_w}^n(U_{i_w-2}^n, \ldots, U_{i_w+1}^n)$ and advance also the solution $U_{i_w}^n$ and $U_{i_w+1}^n$ by a time step Δt.

As a finite volume method our boundary treatment is written in conservation form. In order that the method is conservative however, it is required that the flux along the boundary is given by the wall pressure p_w as

$$F_{boundary} = \begin{pmatrix} 0 \\ p_w \\ 0 \end{pmatrix}. \tag{23}$$

No mass, no energy, and no momentum not coming from pressure flows into the domain. For the new boundary treatment, we do not calculate a flux at the boundary but at cell interfaces close to the boundary. Thus for wall boundaries not aligned to the grid interfaces condition (23) is only fulfilled approximately. Conservation is one feature to guarantee that a numerical solution approximates the physically relevant weak solution to the Euler equations [6]. In [7](pp.123) LeVeque shows, how a non conservative method can lead to a wrong propagation speed of shocks. Therefore

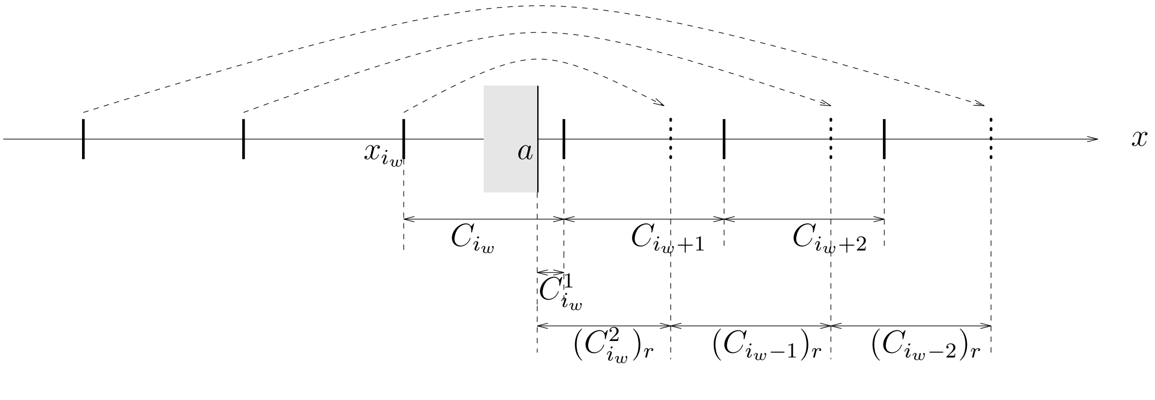

Figure 3: *Auxiliary intervals for the calculation of the weighted means.*

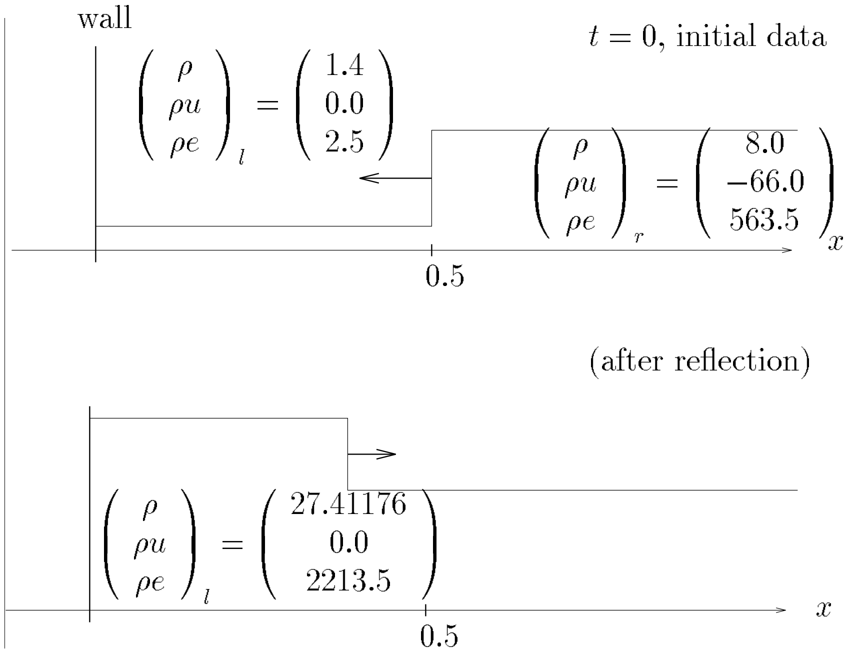

Figure 4: *Strong shock reflection.*

conservative method can lead to a wrong propagation speed of shocks. Therefore in the next section we show the numerical results for a strong shock reflection of a solid wall and it turned out that there are no noticeable spurious effects.

2.2 Numerical results

In this section we show numerical results of flows involving reflecting wall boundaries. As integrator we used the second order accurate finite volume integrator of CLAWPACK [8], where the approximate Riemann-solver of Roe [9] is used.

First we show the results for the strong Mach 10.0 shock reflection with a shock starting at $x = 0.5$ approaching a wall on the left and being reflected there. The discretization used is given by $x_i = h \cdot i$, $C_i = [x_i, x_{i+1}]$, where h is the grid parameter. The wall is located at $x_w = \alpha \cdot h$, $\alpha \in [0, 1[$. Thus C_0 is a boundary interval. The initial data and the reflected state are given in Figure 4. For the first calculation (Figure 5 on the left), the wall is situated at $x = 0.0$. In this special case the boundary treatment is conservative. We compare this result with a calculation where the wall is situated at $x = \alpha \cdot h$, $\alpha = 0.8$ (Figure 5 on the right), where the boundary treatment is not conservative. For both calculations a Courant number of $cfl = 0.8$ was used. The comparison shows that the reflected shock location comes out correctly in the ”eye norm”. (The loss in density close to the wall appears in both cases and is known as wall heating in the literature).

We have made an analysis of the unphysical mass production or cancellation. For a time t_n such that the shock already got reflected at the wall but still is in the

h	Δmass	$\Delta\sigma$	α
$h_0 = 0.25$	-0.7086	-0.0365	0.75
$\frac{h_0}{2}$	-0.3598	-0.0185	$\vdots$
$\frac{h_0}{4}$	-0.1577	-0.0081	$\vdots$
$\frac{h_0}{8}$	-0.0867	-0.0045	$\vdots$
$\frac{h_0}{16}$	-0.0407	-0.0021	$\vdots$
$\frac{h_0}{32}$	-0.0202	-0.0010	$\vdots$

α	Δ mass	α	Δ mass	h
0.0	0.0000	0.6	-0.0121	$\frac{h_0}{32}$
0.1	-0.0141	0.7	-0.0177	$\vdots$
0.2	-0.0238	0.8	-0.0225	$\vdots$
0.3	-0.0236	0.9	-0.0168	$\vdots$
0.4	-0.0174	1.0	0.0000	$\vdots$
0.5	-0.0000			$\vdots$

Table 1: *Unphysical mass production or cancellation Δ mass and the resulting error in the shock location $\Delta\sigma$ for a Mach 10 reflection.*

computational domain, we obtain

$$\Delta\text{mass} = \Delta x \cdot (\sum_{i=1}^{N} \rho_i^n + (1-\alpha)\cdot\rho_0^n) - \int_{\alpha h}^{(N+1)h} \rho_{\text{exact}}(x, t_n). \tag{24}$$

A change in the total mass results in an error of the shock location. Using the equal area rule ([7] pp.35) the error in the shock location $\Delta\sigma$ is given by

$$\Delta\sigma = \frac{\Delta\text{mass}}{\rho_l - \rho_r}, \tag{25}$$

where ρ_l, ρ_r are the densities of the exact solution on the left and on the right of the reflected shock. In Table 1 we see that for the Mach 10 shock reflection Δmass and therefore $\Delta\sigma$ as well are linear in the grid parameter h and that the method is conservative for the special cases of a wall along a grid line or through the middle of a grid interval.

For one dimensional calculations this unphysical mass production or cancellation only happens for the short time when a shock gets reflected at the boundary. For the rest of the time the solution will be smooth at the boundary and the mass

h	ρ_{wall}	ρ_{half}	Δ_{wall}	order	Δ_{half}	order
h_0=0.1	1.035478	1.088415	0.015277		0.014760	
$\frac{h_0}{2}$	1.023884	1.100191	0.003683	2.05	0.002985	2.3
$\frac{h_0}{4}$	1.021120	1.102493	0.000920	2.00	0.000683	2.13
$\frac{h_0}{8}$	1.020432	1.103011	0.000232	1.99	0.000164	2.06
$\frac{h_0}{16}$	1.0202562	1.1031362	0.0000558	2.06	0.0000399	2.04
$\frac{h_0}{128}$	1.0202003	1.1031761				

Table 2: *Convergence history for the smooth 1D test example.*

production is then globally quadratic in h as the global accuracy of the method for smooth flows (see below).

As a next test case, we took some arbitrary smooth initial data for $x \in [0,1]$ with solid wall boundary conditions on both sides at $x = 0.0$ and $x = 1.0$, such that the flow field stays smooth at least until time $t = 1.0$ (cf. Figure 6). As there is no way to get the exact solution to this problem, we took a calculation with a fine grid of 1280 grid points as "exact" solution. For this test example we made an error analysis to evaluate the order of accuracy of the method. Away from the boundary the method should be of second order even if the accuracy of the boundary treatment is locally one order lower than in the rest of the domain [10]. We will check the global accuracy looking at the convergence of the flow variables at the point $x = 0.5$. To calculate the order of the boundary treatment, we used a grid such that the first interval C_0 is a boundary interval. Using grid parameters $h = \frac{1}{n}$, $n \in I\!N$ and $\alpha \in]0,1[$ the grid is defined by

$$
\begin{aligned}
C_0 &= [\alpha h - h, \alpha h]\,, \\
C_1 &= [\alpha h, \alpha h + h]\,, \\
C_2 &= [\alpha h + h, \alpha h + 2h]\,. \\
&\vdots
\end{aligned}
$$

We made a calculation with fixed Courant number to proceed the initial data $\rho_i^{ini}, u_i^{ini}, e_i^{ini}$ to get the values at time $t = 1.0$ $\rho_i^{end}, u_i^{end}, e_i^{end}$. Thus ρ_0^{end} and ρ_1^{end} are approximate values for density at the points $x = |\alpha - 0.5| \cdot h$ and $x = |\alpha + 0.5| \cdot h$, respectively. We used these two values together with $\frac{\partial}{\partial x}\rho(x = 0)$ to get ρ_{wall}, the wall density at $t = 1.0$.

$$
\rho_{wall} = \rho^{end}(x = 0) = \frac{\rho_0^{end}(\alpha + \frac{1}{2})^2 - \rho_1^{end}(\alpha - \frac{1}{2})^2}{2\alpha}. \tag{26}
$$

Analogously, we calculated a value of the density ρ_{half} at time $t = 1.0$ and $x = 0.5$. The convergence history on Table 2 for the density values ρ_{wall} and ρ_{half} for a grid with $\alpha = 0.25$ demonstrates that the overall method and the implemented boundary treatment is of second order.

3 Two space dimension

Now we want to extend the method to handle reflecting wall boundaries in two space dimensions. Also in 2D we want the boundary treatment to be independent of the method for the regular cells, but we demand that the method can be written in conservation form (4).

3.1 Description of the boundary treatment

Without loss of generality, we assume from now on that in order to calculate the flux F_{kl}, we need the flow variables of the cells $C_{k-2,l}, C_{k-1,l}, C_{k,l}, C_{k+1,l}$. But the method has to take into account that the the neighbouring layers of cells $C_{k-2,l-1}$, $C_{k-1,l-1}$, $C_{k,l-1}$, $C_{k+1,l-1}$ and $C_{k-2,l+1}, C_{k-1,l+1}, C_{k,l+1}, C_{k+1,l+1}$ are needed as well. Either by using correction terms as LeVeques method [5] (see Remark 1 at the end of this section) or in a genuinely multidimensional way as Feys method [11].

First we have to determine, which cells are cut by the boundary. We denote them as boundary cells and we approximate the boundary for these cells as a straight line segments. The other cells are either regular cells or empty cells (cf. Figure 7).

The straight line, which approximates the curved boundary in the boundary cell C_{kl} goes through a point $\mathbf{z}_{kl}$ and has a normal vector $\mathbf{n} = (-\sin \alpha_{kl}, \cos \alpha_{kl})^t$ pointing into the computational domain (cf. Figure 8). The initial data of a regular cell is given by (3). To obtain some initial data for a boundary cell C_{kl} as well, we divide this cell into two parts C^1_{kl} and C^2_{kl}, where C^1_{kl} is the part of C_{kl} lying in the computational domain and C^2_{kl} is the remaining part, such that $C^1_{kl} \cup C^2_{kl} = C_{kl}$ (cf. Figure 8).

In (6) we have defined how to reflect a point $\mathbf{p}$ at a straight line through the point $\mathbf{z}$ with normal vector $\mathbf{n} = (-\sin \alpha, \cos \alpha)^t$. Analogously we define the reflection of a polygon C with corner points $\mathbf{p}_1, \dots, \mathbf{p}_m$ as polygon $r_{\mathbf{z},\alpha}(C)$ with the reflected corner points $\mathbf{r}_{\mathbf{z},\alpha}(\mathbf{p}_1), \dots, \mathbf{r}_{\mathbf{z},\alpha}(\mathbf{p}_m)$. If we reflect a polygon at the straight boundary line going through the boundary cell C_{kl}, we simplify the notation by

$$r_{kl}(C) := r_{\mathbf{z_{kl}},\alpha_{kl}}(C). \tag{27}$$

In Figure 9 we show the reflection of C^2_{kl} and $C_{k+1,l}$ onto $r_{kl}(C^2_{kl})$ and $r_{kl}(C_{k+1,l})$. Using the reflection matrix R_α from (7), we can define initial data for a boundary cell as weighted mean at time t_0 as follows,

$$U^0_{kl} = \frac{1}{h^2}\Big(\int_{C^1_{kl}} U(x,y,t_0)dx\,dy + \int_{r_{kl}(C^2_{kl})} R_{\alpha_{kl}} \cdot U(x,y,t_0)dx\,dy\Big). \tag{28}$$

Now we can proceed as in 1D. If we have an approximation of the solution at time $t_n = t_0 + n \cdot \Delta t$ as a piece-wise constant function over the regular and the boundary cells U_{ij}^n, then we want to advance this solution with the method (4) also for the boundary cells and regular cells near the boundary. Suppose, that the l-th horizontal layer of grid cells consists of regular cells $C_{k-1,l}, C_{k-2,l}, \dots$, of the boundary cell C_{kl}, and of the empty cells $C_{k+1,l}, C_{k+2,l}$. With a 4-point stencil for the F-flux calculation we can evaluate the F-fluxes $F_{k-1,l}, F_{k-2,l}, \dots$. But in order to update the variables in the cells $C_{k-1,l}$ and C_{kl} using (4), we need also the F-fluxes F_{kl} and $F_{k+1,l}$. To calculate these two fluxes with the same stencil as the other fluxes, we need flow variables in the cells $C_{k+1,l}, C_{k+2,l}$. We can obtain these variables as weighted means over the reflected cells as follows,

$$U_{k+1,l}^n = \int_{r_{kl}(C_{k+1,l})} R_{\alpha_{kl}} \cdot U^*(x, y, t_n) dx\, dy, \tag{29}$$

$$U_{k+2,l}^n = \int_{r_{kl}(C_{k+2,l})} R_{\alpha_{kl}} \cdot U^*(x, y, t_n) dx\, dy. \tag{30}$$

For the approximate solution $U^*(.,.,t_n)$, we take the piece-wise constant function given by U_{ij}^n for the regular and the boundary cells. Now we can calculate all the fluxes and advance the solution U_{ij}^n also for the boundary cells and regular cells near the boundary by a time step Δt.

Remark 1: With the method of LeVeque [5] stability is achieved by splitting vertically the waves of the flux F_{kl} and using these split components to modify the fluxes $G_{k-1,l}, G_{kl}, G_{k-1,l+1}, G_{k,l+1}$. Suppose the flux $F_{k,l-1}$ is not calculated because $C_{k-1,l-1}$ and $C_{k,l-1}$ are empty cells, then the fluxes $G_{k-1,l}, G_{kl}$ would get no modification. Therefore in this case the flux $F_{k,l-1}$ has to be calculated as well.

Remark 2: If the cells $C_{k+1,l}, C_{k+2,l}$ are empty cells next to a boundary cell C_{kl}, we should not use their memory location to store the values $U_{k+1,l}^n$ and $U_{k+2,l}^n$ of (29)(30). We illustrate the reason for this using a NACA0012 airfoil discretization. Suppose, that the cells $C_{k-1,l}$ and C_{kl} are lying on the trailing edge of an airfoil as in Figure 10.

At the cell interface between these two cells, we have to calculate two different F-fluxes. We use a flux F_{kl}^1 to use (4) for the cell $C_{k-1,l}$ and a flux F_{kl}^2 to use (4) for the cell C_{kl}. Using the 4-point stencil $F^{num}(.,.,.,.)$ for the F-fluxes we obtain these two values as follows,

$$F_{kl}^1 = F^{num}(U_{k-2,l}^n, U_{k-1,l}^n, \overline{U}_{kl}^n, \overline{U}_{k+1,l}^n), \tag{31}$$

$$F_{kl}^2 = F^{num}(\overline{U}_{k-2,l}^n, \overline{U}_{k-1,l}^n, U_{kl}^n, U_{k+1,l}^n), \tag{32}$$

where $U_{k-2,l}^n, U_{k-1,l}^n, U_{kl}^n, U_{k+1,l}^n$ are the values of the corresponding regular or boundary cells and $\overline{U}_{k-2,l}^n, \overline{U}_{k-1,l}^n, \overline{U}_{kl}^n, \overline{U}_{k+1,l}^n$ are weighted means over reflected cells, as e.g.

$$\overline{U}_{k-1,l}^n = \int_{r_{kl}(C_{k-1,l})} R_{\alpha_{kl}} \cdot U^*(x, y, t_n) dx\, dy, \tag{33}$$

$$\overline{U}_{kl}^n = \int_{r_{k-1,l}(C_{kl})} R_{\alpha_{k-1,l}} \cdot U^*(x, y, t_n) dx\, dy. \tag{34}$$

Thus we have to store $\overline{U}_{k-2,l}^n, \overline{U}_{k-1,l}^n, \overline{U}_{kl}^n, \overline{U}_{k+1,l}^n$ in separate memory locations.

3.2 Domain boundary treatment

At the domain boundary we have to determine variables for the two (because of the 4-point stencil) outermost cells such that disturbances generated inside the computational domain leave the domain without reflections. Consider the left hand side of the l-th horizontal layer of grid cells, where we have to determine variables U^n_{0l}, U^n_{1l} in the cells C_{0l}, C_{1l}. If in the cell C_{2j} at time t_n the horizontal velocity component u^n_{2j}, and the local speed of sound c^n_{2j}, is such that $u^n_{2j} > c^n_{2j}$, we treat the boundary like a supersonic inflow. This means that we fill U^n_{0j}, U^n_{1j} with conservative variables given by the asymptotic conditions. On the other hand, if $u^n_{2j} < -c^n_{2j}$, we use supersonic outflow conditions and copy $U^n_{0j} = U^n_{2j}$ and $U^n_{1j} = U^n_{2j}$. For $0 < u^n_{2j} < c^n_{2j}$ we have subsonic inflow conditions, and for $0 > u^n_{2j} > -c^n_{2j}$ we have subsonic outflow conditions. There we use a characteristic treatment of the outer boundary, assuming locally homentropic flow and that derivatives along the boundary can be neglected, $\frac{\partial(.)}{\partial y} \to 0$ (cf. [12]). Using the entropy $s = \frac{p}{\rho^\gamma}$, the speed of sound $c = \sqrt{(\frac{\gamma p}{\rho})}$, and the Riemann variables $R^\pm = u \pm \frac{2c}{\gamma-1}$ we determine

$$\left.\begin{array}{c} s^n_{1/0,l} \\ v^n_{1/0,l} \\ R^{+n}_{1/0,l} \\ R^{-n}_{1/0,l} \end{array}\right\} = \left\{\begin{array}{l} s_\infty \\ v_\infty \\ R^+_\infty \\ R^{-n}_{2j} \end{array}\right. : \text{(for inflow)}, \; = \left\{\begin{array}{l} s_{2j} \\ v_{2j} \\ R^+_{2j} \\ R^{-n}_\infty \end{array}\right. : \text{(for outflow)}. \tag{35}$$

For the calculation of a transonic steady-state flow around an airfoil the asymptotic conditions are given by the Mach number M_∞, the density ρ_∞, the pressure p_∞, and the angle of attack α_∞. As often done in the literature, we add correction terms to the velocity given by the circulation following the ideas of Thomas and Salas [12]. Drag and lift are nondimensionalized as follows:

$$c_d = \frac{2}{\rho_\infty u^2_\infty L} \oint (\rho u(\mathbf{u}\cdot\mathbf{n}) + p n_x) dS, \tag{36}$$

$$c_l = \frac{2}{\rho_\infty u^2_\infty L} \oint (\rho v(\mathbf{u}\cdot\mathbf{n}) + p n_y) dS, \tag{37}$$

where L is the chord length of the airfoil and the integration is taken along an arbitrary closed contour S enclosing the airfoil with outward-facing normal vector $\mathbf{n} = (n_x, n_y)$. Linearizing the far field in a coordinate system $(\overline{x}, \overline{y})$ aligned with the free stream velocity and centred at the airfoil quarter chord the velocity disturbance is given as

$$\Delta u = \frac{\Gamma}{2\pi} \cdot \frac{\beta \overline{y}}{\overline{x}^2 + (\beta \overline{y})^2}, \tag{38}$$

$$\Delta v = \frac{-\Gamma}{2\pi} \cdot \frac{\beta \overline{x}}{\overline{x}^2 + (\beta \overline{y})^2}, \tag{39}$$

where $\beta = \sqrt{1 - M_\infty^2}$, and Γ is the circulation. Using the Joukowski formula [13] one gets

$$\Gamma = -\frac{|\mathbf{u}_\infty| L}{2}(c_l \cos\alpha_\infty - c_d \sin\alpha_\infty) \tag{40}$$

Thus, we get a feedback algorithm between far field circulation and lift coefficient.

3.3 Numerical results

For the following numerical examples we used CLAWPACK of LeVeque [8] as the underlying method for the regular cells together with our new reflecting boundary treatment.

We present five computational examples illustrating the use of the boundary treatment in 2D. In the first and second example the geometry is given by a circular cylinder. First we let an incident shock reflect off a cylinder. For this example we can check the conservation. In the second example we use a subcritical inflow of Mach 0.38 presented at a GAMM workshop [14]. We compare the results with other methods which use body-fitted grids. The third example is a flow over a ramp, an example introduced by Woodward and Colella resulting in a double Mach reflection [15]. Next we will calculate a transonic flow around a NACA0012 airfoil described also [14]. But we point out that for this example Cartesian grids are not at all well fitted, because of the smallness of the airfoil compared to the necessary computational domain. The last example is a calculation of a Prandtl-Meyer expansion. As in [1][2] we use this example to measure the order of accuracy of the method.

In the first example we study the behaviour of an incident shock travelling at a relative Mach number 3.0 and its reflection off a cylinder. The computational domain is the unit square. The cylinder is located in the middle of the domain and has a radius of 0.2. The initial shock location is just in front of the cylinder. We use a Courant number (5) of $cfl = 0.9$. In Figure 11 on the left we see the density for a calculation with a 300×300 mesh at time $t = 0.15$.

As in 1D, we have made an analysis of the unphysical mass production or cancellation. For a time t_n such that the shock is still in the computational domain, we obtain

$$\Delta\text{mass}\,(t_n) = \sum_{ij} |C_{ij}^1| \cdot (\rho_{ij}^n - \rho_{ij}^0) \ - \ v_S(\rho_l - \rho_r) \cdot t_n, \tag{41}$$

where the summation $\sum_{ij}$ goes over the regular and boundary cells and $|C_{ij}^1|$ denotes the area of the cell C_{ij} lying in the computational domain. v_S is the shock speed and $(\rho_l - \rho_r)$ is the jump of density across the shock. In Table 3 we show the loss in conservation for different meshes. We see that the conservation is violated linearly in the grid spacing.

For the second example a circular cylinder with radius 0.5 is placed in the middle of a square with length 10.0. We study a subcritical inflow of Mach 0.38 and use a 400×400 mesh. The grid cells have a length of $1/40 = 0.025$ such that we can compare the results with methods using a 128×32 O-mesh, where the cells of the

mesh	Δmass
100×100	18.850E-03
200×200	12.250E-03
400×400	7.659E-03
800×800	3.914E-03

Table 3: *Unphysical mass production or cancellation for a shock reflection off a cylinder at time* $t = 0.15$.

first layer around the cylinder have a length of $\pi/128 = 0.0245$. We use a constant flow as initial condition and reach steady state after a sufficiently large time. In Figure 11 on the right we show the density at time $t = 60$. We get a maximum value for the entropy deviation, $\Sigma = (p/p_\infty)/(\rho/\rho_\infty)^\gamma - 1$, of $\Sigma_{max} = 0.0028$ and a maximum value for the Mach number of $M_{max} = 0.9094$ which compares well with the results found in literature using O-meshes [16] [14].

The next example was introduced in [15] to compare different numerical schemes. It is a reflection of a Mach 10 shock at a ramp. The resulting configuration of a double Mach reflection contains some fluid dynamical features as a moving shock, a quasi steady state shock, a contact discontinuity, and two reflected shocks, all of which make a good test example out of it. We used this example by tilting the reflecting wall using our boundary treatment to model the reflecting boundary oblique to the mesh rather than by tilting the incident shock as done in [15]. We used a grid with $\Delta x = \Delta y = 1/120$ and 30^0 angle of the ramp as in [15] to be able to compare the results. The Mach 10 shock starts a few layers of grid cells in front of the origin of the ramp and the computation is stopped at time $t = 0.2$. In Figure 12

the results are shown using a 300×240 grid. The results compare well with [15]. It is very encouraging that the first reflected shock is not smeared at all as it hits the boundary.

Our next example was chosen as an example of more practical importance, namely a transonic flow around a NACA0012 airfoil. The geometry is described in [14] but we scaled the chord length to 1.0. The calculation was done with the far-field correction described in the previous section. We took a 400×400 mesh with a side length of 8 chords and placed the airfoil at a distance of 3 chords from the inflow boundary. We compare the results with the results of a 256×64 O-mesh with a boundary extent of 100 chords of Thomas and Salas [12]. Our Cartesian grid has $\Delta x = \Delta y = 1/50$ resulting in about 100 grid cells around the surface of the airfoil. We do not refine the grid at the leading edge. Thus we can not expect that our results can compete with [12]. In Figure 13

we show details of the mesh and of a C_p contour plot, where $C_p = (p - p_\infty)/(\frac{1}{2}\rho_\infty u_\infty^2)$. The pressure C_p and the entropy deviation Σ are shown in Figure 14. The forces

h	entropy		stagnation enthalpy	
	$error_{total}$	order	$error_{total}$	order
$\frac{1}{200}$	0.0007390	-	0.004332	-
$\frac{1}{400}$	0.0001849	1.998	0.001758	1.301
$\frac{1}{700}$	0.00005895	2.043	0.0008545	1.289
	$error_{boundary}$		$error_{boundary}$	
$\frac{1}{200}$	0.009998	-	0.033900	-
$\frac{1}{400}$	0.003786	1.401	0.018621	0.866
$\frac{1}{700}$	0.001688	1.479	0.016400	0.227

Table 4: Convergence history for the Prandtl-Meyer expansion.

on the airfoil c_d and c_l are computed to $c_d = 0.0205$ and $c_l = 0.3118$ compared with $c_d = 0.0226$ and $c_l = 0.358$ of [12]. The numerical entropy production at the leading edge is of the order of the physical entropy production across the shock on the upper surface in our calculation (cf. Figure 14 on the right). Thus we expect to get more exact results for this example if we refine the grid at the leading edge.

The last example is a steady-state calculation of a Prandtl-Meyer expansion for a Mach 1.83 flow over a 35^0 bend. The exact solution is a smooth flow with constant entropy $S = p/\rho^\gamma$ and constant enthalpy $H = (p + \rho e)/\rho$. In Figure 15 we see a density contour plot of a calculation with a 200×200 mesh. As introduced by Berger and LeVeque [1][2], we can use this example to analyse the order of accuracy of a method by taking the L_1-norm of the error in entropy and enthalpy in the whole flow area or only along the boundary:

$$error_{total} = \frac{\sum^1 |u_{ij} - u_{ij}^c|}{\sum^1 |C_{ij}|}, \tag{42}$$

$$error_{boundary} = \frac{\sum^2 |u_{ij} - u_{ij}^c|}{\sum^2 |C_{ij}|}, \tag{43}$$

where u_{ij}^c is the exact solution (here entropy or stagnation enthalpy), u_{ij} is the numerical solution, $\sum^1$ is a summation over all the grid cells, and $\sum^2$ is a summation over the boundary cells only. Under the assumption that the error can be expressed as $error = C \cdot h^p$, we can calculate the order p of the method in the whole area or only along the boundary (cf. Table 4). As in [2], the results for entropy suggest that the method is of second order accuracy in the whole area. But the value of $p = 1.48$ for the order in entropy along the boundary is an improvement comparing with other Cartesian grid methods. The results for stagnation enthalpy are less satisfying, as in [2] as well.

4 Conclusions

For Cartesian grid methods simulating the Euler equations, a boundary treatment has been developed based on reflecting locally the velocity field at a straight boundary line. The formulation is easily implemented and the extension to three space dimensions is straightforward.

Numerical results in one space dimension indicate that second order accurate results can be obtained for smooth flows also at the boundary. The violation of the conservation at the boundary does not lead to wrong solutions for shock reflections off the boundary in one dimension.

In two dimensions it could be shown that the numerical entropy production along the boundary of a circular cylinder is comparable to other time dependent methods using body-fitted grids. For a shock reflection off a cylinder it could be shown that the violation of the conservation is of order one in the grid spacing h, and that this violation does not lead to spurious solutions. Shock smearing at the boundary could be shown to be negligible for a shock wandering along the boundary oblique to the mesh in the case of a double Mach reflection off a ramp. For the calculation of a transonic flow past a NACA0012 airfoil the accuracy of the drag and lift coefficients was limited due to the high curvature of the boundary at the leading edge of the airfoil. By means of a smooth steady flow calculation, we measured the order of accuracy of the method in the whole domain and along the boundary. In the whole domain the method is of second order and along the boundary we got an order of 1.48 which is an improvement comparing with other boundary treatments.

References

[1] R.J. LeVeque M.J. Berger. *A Rotated Difference Scheme for Cartesian Grids in Complex Geometries.* AIAA Paper CP-91-1602, 1991.

[2] R.B. Pember J.B. Bell P. Colella W.Y. Crutchfield M.L. Welcome. *An Adaptive Cartesian Grid Method for Unsteady Compressible Flow in Irregular Regions.* Journal of Comp. Phys., 120, pp 278-304, 1995.

[3] J.J Quirk. *An alternative to unstructered grids for computing gas dynamic flows around arbitrarily complex two-dimensional bodies.* Computers Fluids, Vol. 23 No. 1, pp 125-142, 1994.

[4] H. Forrer. *Boundary Treatment for a Cartesian Grid Method.* ETH Research-Report No. 96-04, 1996.

[5] R.J. LeVeque. *Simplified multi-dimensional flux limiter methods.* Numerical methods for fluid dynamics, 4, pp. 175-190, 1993.

[6] P.D. Lax. *Hyperbolic Systems of Conservation Laws and the Mathematical Theory of Shock Waves.* SIAM Regional Conference Series in Applied Mathematics, No.11, 1972.

[7] R.J. LeVeque. *Numerical Methods for Conservation Laws.* Birkhäuser Verlag, 1992, ETH Zürich.

[8] R.J. LeVeque. *CLAWPACK software.* available from `netlib.att.com` in `netlib/pdes/claw`.

[9] P.L. Roe. *Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes.* Journal of Comp. Phys., 43, pp 357-372, 1981.

[10] B. Wendroff A.White. *On irregular grids.* Proc. 2nd Intl. Conf. Nonlinear Hypeerbolic Problems, Notes on Num. Fluid Mech, Vol. 24, 1988.

[11] M. Fey. *Ein echt mehrdimensionales Verfahren zur Lösung der Eulergleichungen.* Dissertation ETH No. 10034 am Seminar für angewandte Mathematik, 1993.

[12] M.D. Salas J.L. Thomas. *Far-Field Boundary Conditions for Transonic Lifting Solutions to the Euler Equations.* AIAA Journal, vol. 24, no. 7, 1986.

[13] E.M Lifschitz L.D. Landau. *Hydrodynamik, Lehrbuch der theoretischen Physik, Bd. 6, 5. Auflage.* Verlag Harri Deutsch, 1991.

[14] B. van Leer A. Dervieux. *GAMM workshop on Numerical Solution of the Euler Equation.* Notes on Numerical Fluid Mechanics, Vol. 26, Vieweg, 1989.

[15] P. Colella P. Woodward. *The Numerical Simulation of 2D Fluid Flow with Strong Shocks.* J. Comp. Phys. , Vol. 54, pp 115-173, 1984.

[16] N. Botta. *Numerical investigation of 2D Euler flows.* Dissertation ETH No. 10852 am Seminar für angewandte Mathematik, 1994.

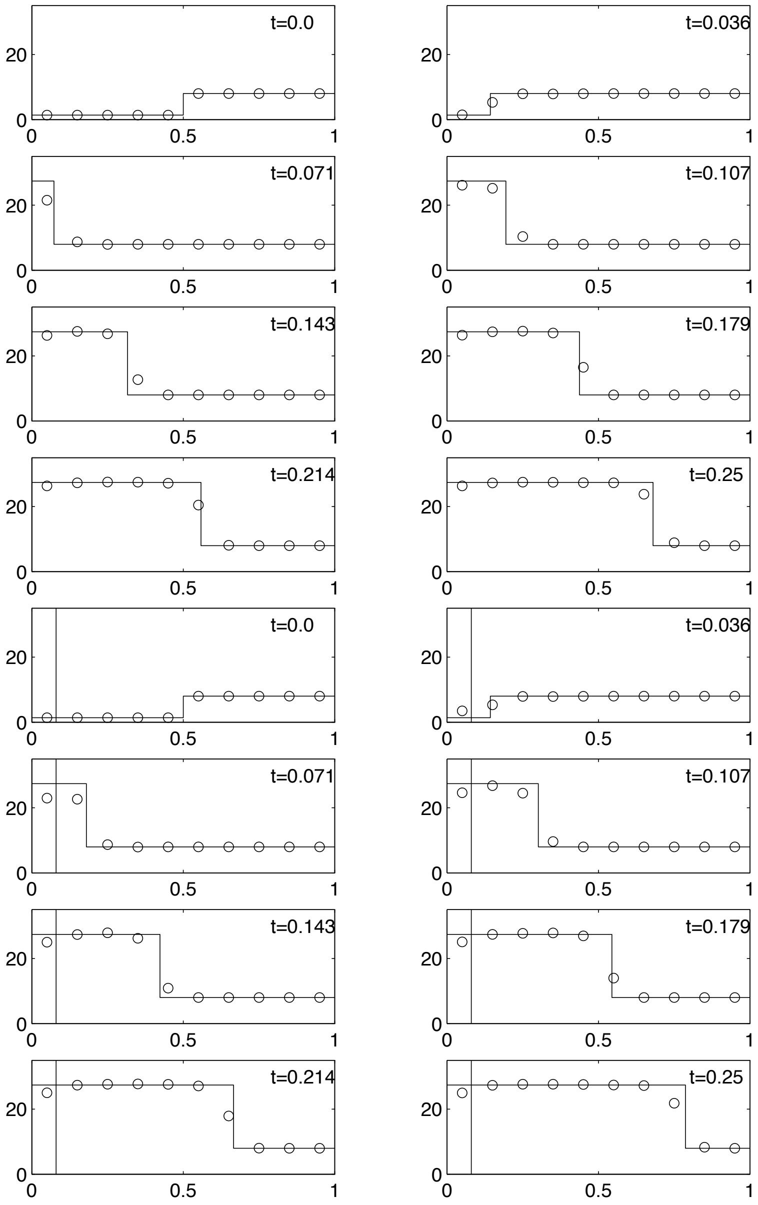

Figure 5: *Density plot of a Mach 10 shock reflection with a wall located at* $x = 0.0$ *(left) and at* $x = 0.08$ *(right) (circles denote the numerical solution, the solid line the exact solution).*

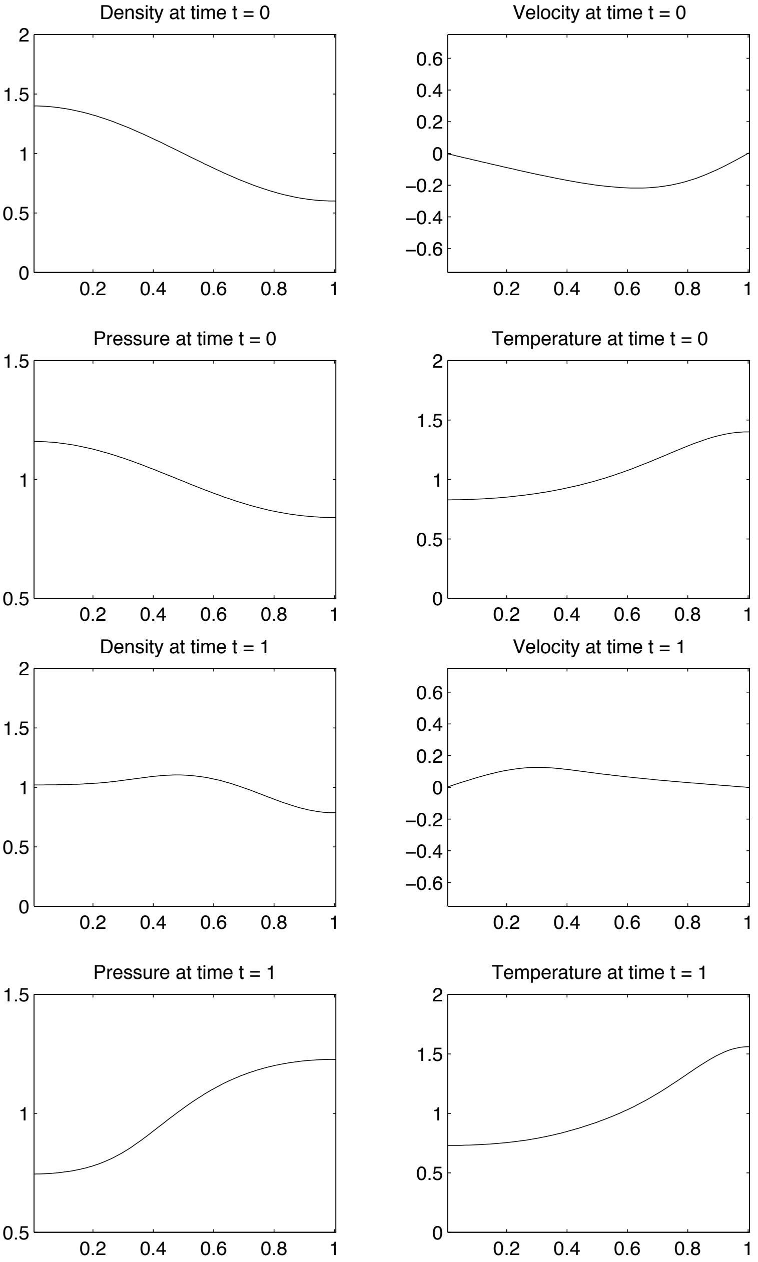

Figure 6: *The smooth flow test calculation with $h = 0.01$.*

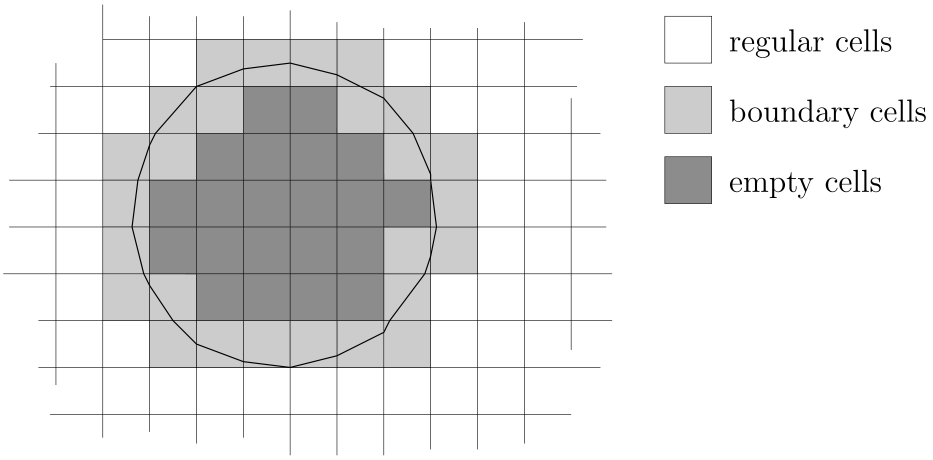

Figure 7: *Discretization of a circle.*

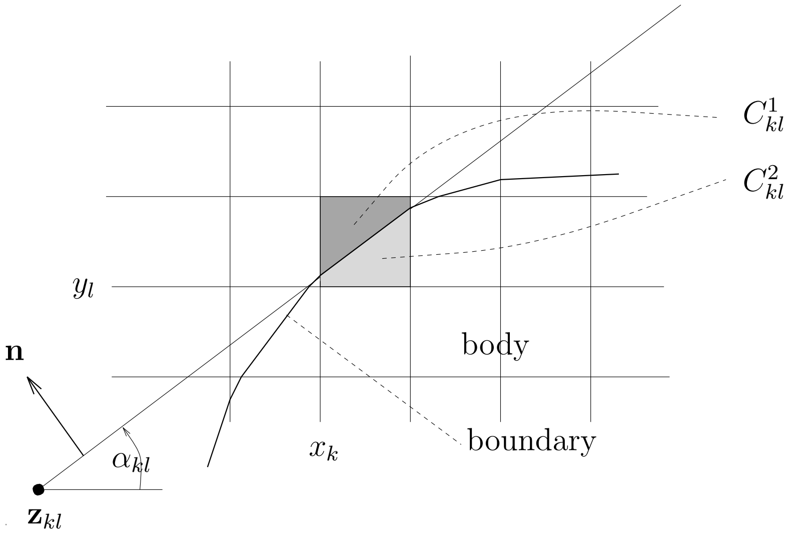

Figure 8: *Description of the boundary segment through the cell* C_{kl}.

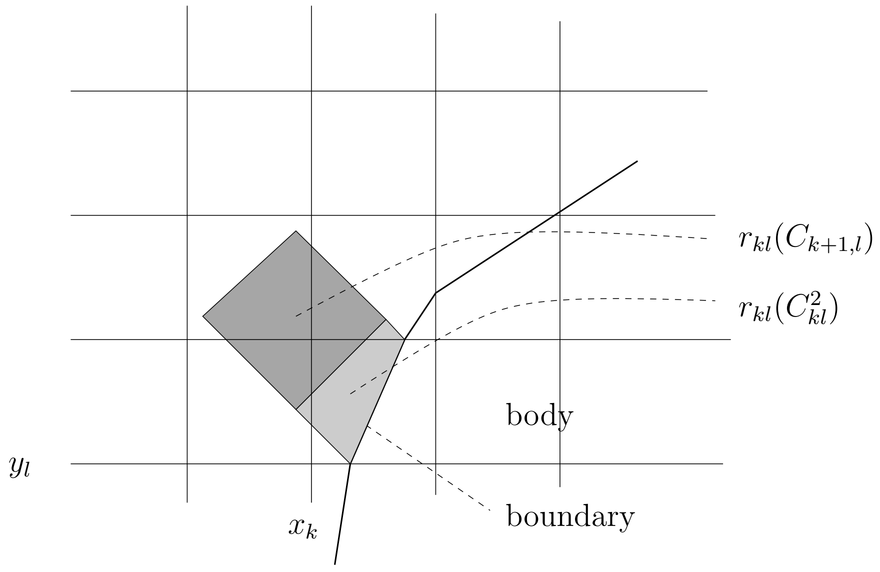

Figure 9: *Reflection at the boundary segment of cell* C_{kl}.

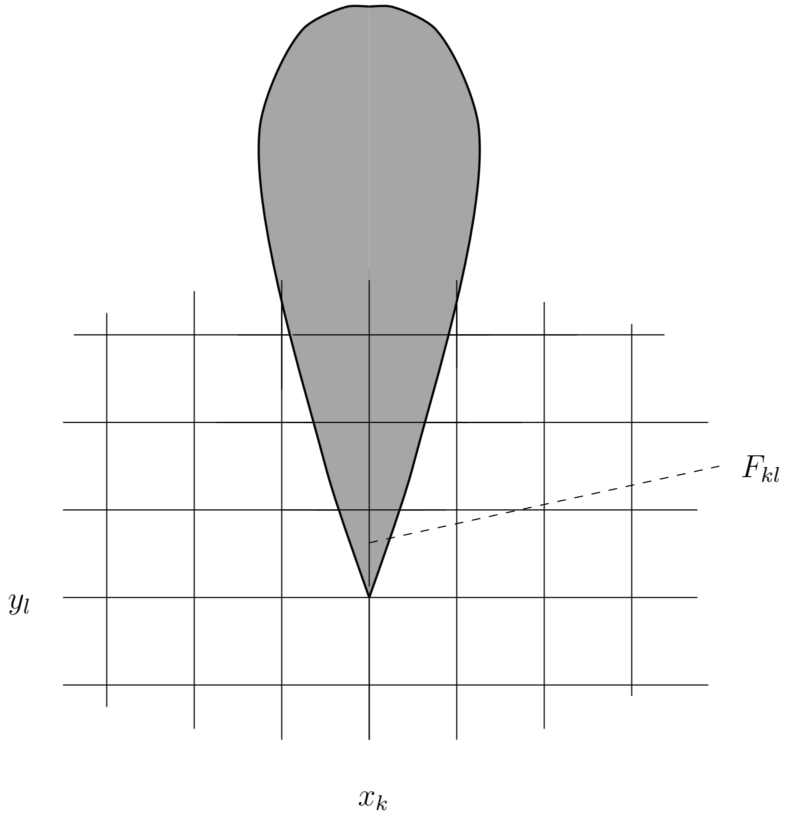

Figure 10: *Discretization of an airfoil.*

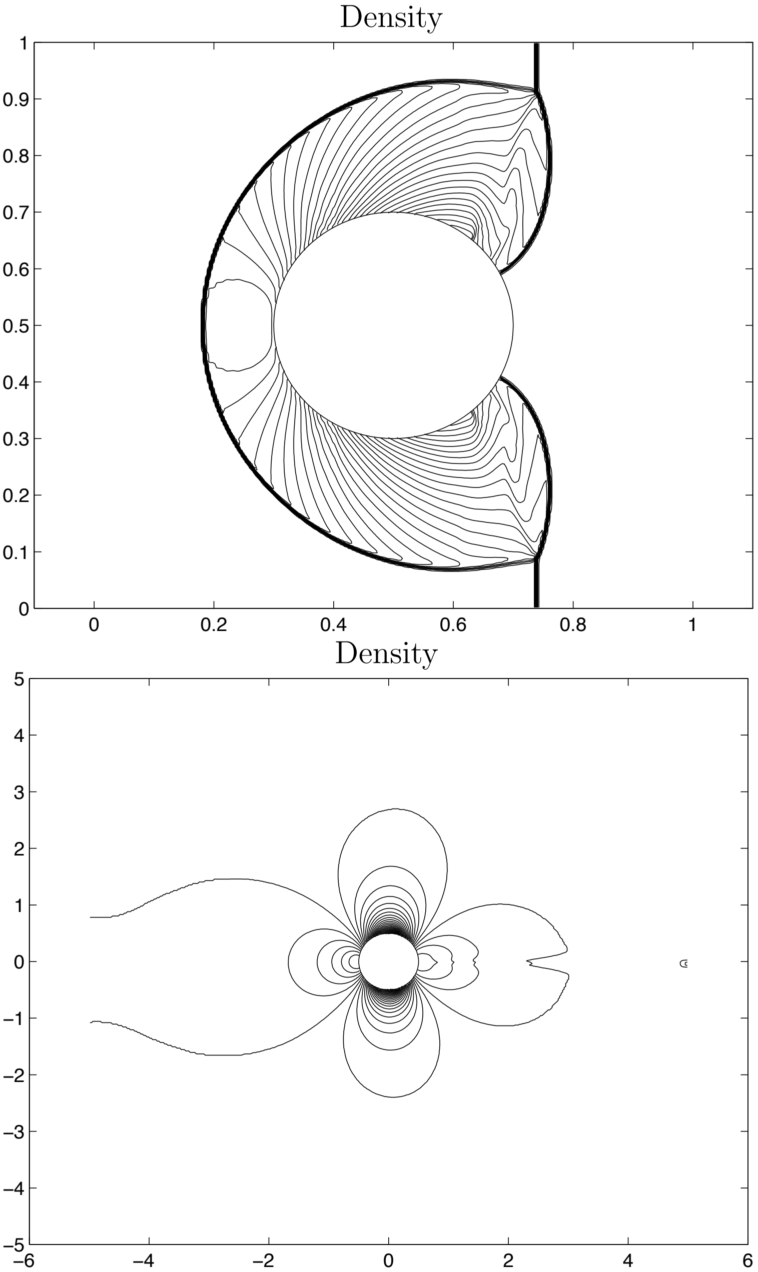

Figure 11: *Shock reflecting off a cylinder (left). Subcritical flow around a cylinder with Mach* 0.38 *(right).*

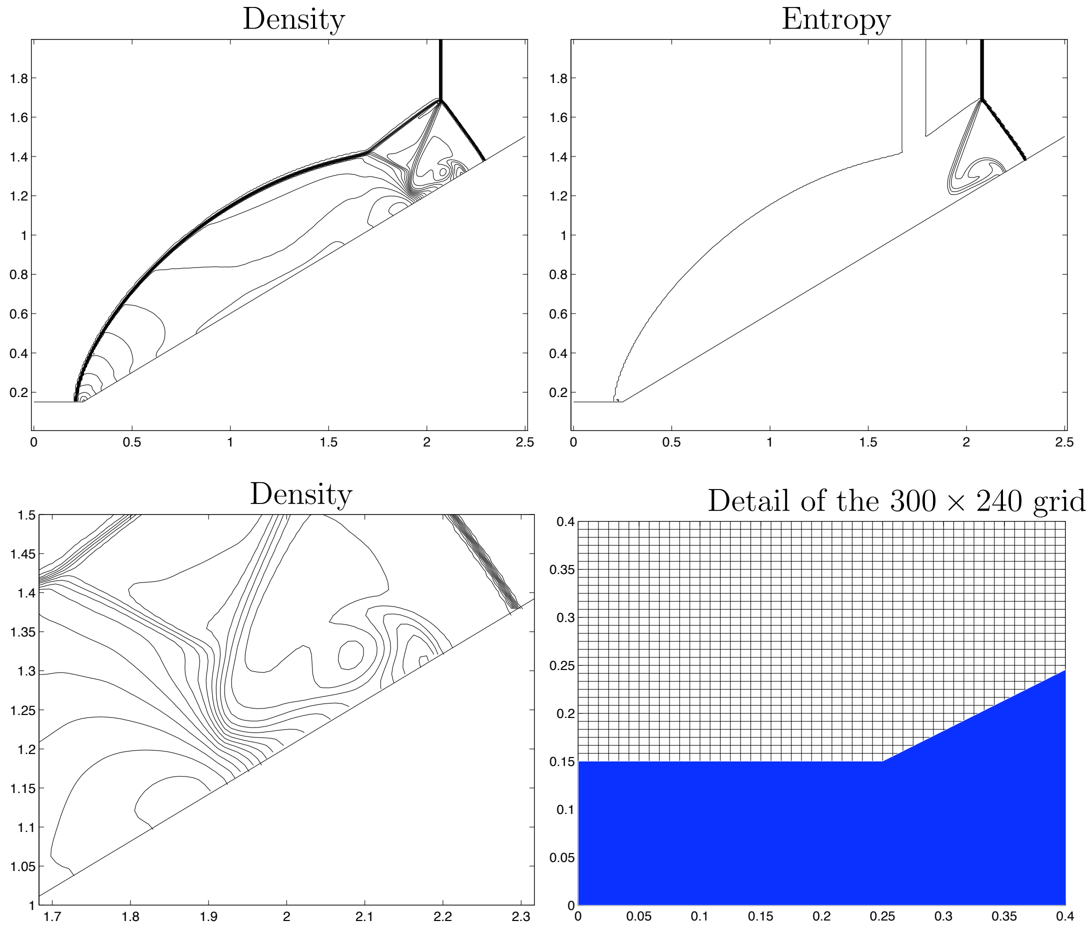

Figure 12: *Reflection of a Mach 10 shock at a* 30^0 *angle at time* $t = 0.2$ *using a grid with* $\Delta x = \Delta y = 1/120$.

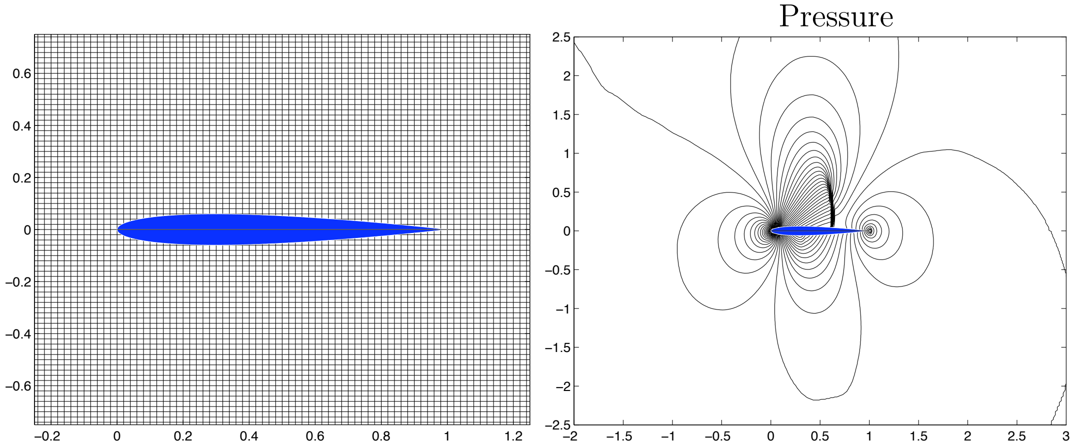

Figure 13: *Mesh and pressure contours for a Mach 0.8 flow around a NACA0012 airfoil with* 1.25^0 *angle of attack.* $\Delta x = \Delta y = 1/50$.

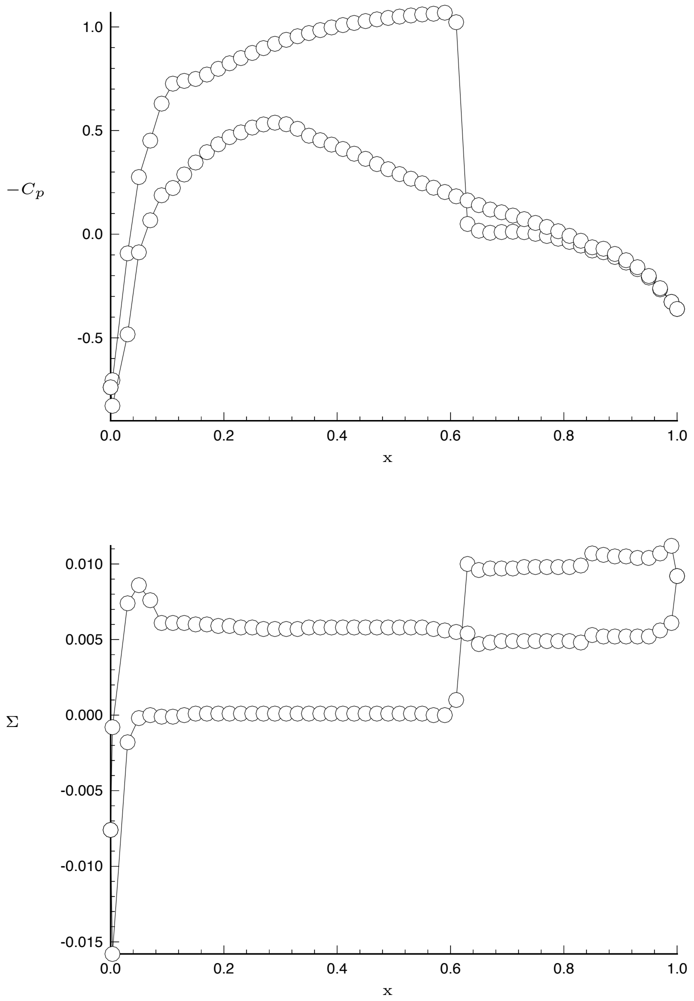

Figure 14: *Distribution of pressure and entropy deviation around the surface of the airfoil.*

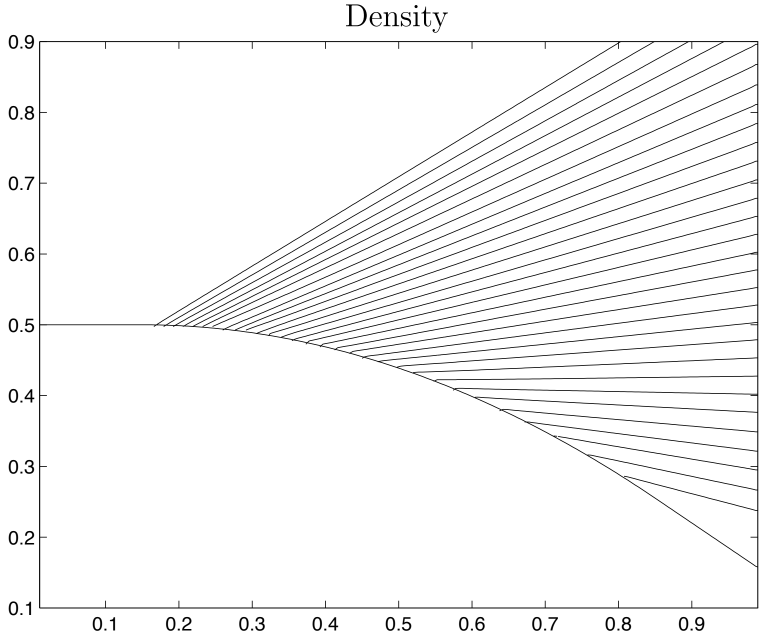

Figure 15: *Expansion of a Mach 1.83 flow over a* 35^0 *bend.* $h = \Delta x = \Delta y = 1/200$.

Research Reports

No.	Authors	Title
96-13	H. Forrer	Second Order Accurate Boundary Treatment for Cartesian Grid Methods
96-12	K. Gerdes, C. Schwab	Hierarchic models of Helmholtz problems on thin domains
96-11	K. Gerdes	The conjugated vs. the unconjugated infinite element method for the Helmholtz equation in exterior domains
96-10	J. Waldvogel	Symplectic Integrators for Hill's Lunar Problem
96-09	A.-T. Morel, M. Fey, J. Maurer	Multidimensional High Order Method of Transport for the Shallow Water Equations
96-08	A.-T. Morel	Multidimensional Scheme for the Shallow Water Equations
96-07	M. Feistauer, C. Schwab	On coupled problems for viscous flow in exterior domains
96-06	J.M. Melenk	A note on robust exponential convergence of finite element methods for problems with boundary layers
96-05	R. Bodenmann, H.J. Schroll	Higher order discretisation of initial-boundary value problems for mixed systems
96-04	H. Forrer	Boundary Treatment for a Cartesian Grid Method
96-03	S. Hyvönen	Convergence of the Arnoldi Process when applied to the Picard-Lindelöf Iteration Operator
96-02	S.A. Sauter, C. Schwab	Quadrature for hp-Galerkin BEM in $\mathbb{R}^3$
96-01	J.M. Melenk, I. Babuška	The Partition of Unity Finite Element Method: Basic Theory and Applications
95-16	M.D. Buhmann, A. Pinkus	On a Recovery Problem
95-15	M. Fey	The Method of Transport for solving the Euler-equations
95-14	M. Fey	Decomposition of the multidimensional Euler equations into advection equations
95-13	M.D. Buhmann	Radial Functions on Compact Support
95-12	R. Jeltsch	Stability of time discretization, Hurwitz determinants and order stars
95-11	M. Fey, R. Jeltsch, A.-T. Morel	Multidimensional schemes for nonlinear systems of hyperbolic conservation laws
95-10	T. von Petersdorff, C. Schwab	Boundary Element Methods with Wavelets and Mesh Refinement