JOURNAL OF COMPUTATIONAL PHYSICS **140,** 259–277 (1998)
ARTICLE NO. CP985891

A Higher-Order Boundary Treatment for Cartesian-Grid Methods

Hans Forrer[*] and Rolf Jeltsch[†]

[*]*Courant Institute of Mathematical Sciences, New York, New York 10012;* [†]*Eidgenössische Technische Hochschule, Zürich, Switzerland*
E-mail: forrer@cims.nyu.edu

Received September 19, 1996; revised November 13, 1997

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 an object. A new method for the treatment of the boundary is described 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 in one dimension and higher-order accurate in two dimensions but not strictly conservative; however, we 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 higher-order accurate, that it is independent of the applied method, and that it is simple. © 1998 Academic Press

Key Words: finite-volume schemes; hyperbolic conservation laws; boundary treatment; CLAWPACK package; Euler equations.

1. INTRODUCTION

A Cartesian-grid method consists of a standard method for the regular cells and a special treatment for the boundary cells. Cartesian-grid methods can take full advantage of fast computer architectures like vector or parallel computers. Cartesian-grid methods are flexible, i.e., they can be used for flow simulations around complex geometries. In the literature Cartesian-grid methods with different approaches for the boundary treatment can be found which are of first order. Berger and LeVeque [1] use rotated boxes to get stability, Colella [9] uses flux-redistribution procedures, and Quirk [10] uses merging procedures. The main problem is to avoid instability arising from small boundary cells and to achieve a high order of accuracy along the boundary. In [4] we presented such a scheme. In this paper a new method for the boundary treatment is given which is stable, higher-order accurate and simpler than the one given in [4]. Above all it is easy to extend this new approach to three-dimensional calculations.

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

$$\mathbf{U}_t + \mathbf{F}_x + \mathbf{G}_y = 0, \tag{1}$$

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

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

where ρ is the mass density, $(u, v)^T$ is the velocity vector, ρe is the energy density, p is 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 shock waves. Let h be a grid parameter and set $x_i = x_0 + ih$, $y_j = y_0 + jh$, $i, j \in \mathbb{Z}$. The regular Cartesian grid cell C_{ij} is then given by

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

The numerical solution at time t_n is given by approximations of the cell averages of the exact solution $\mathbf{U}(x, y, t_n)$ over the grid cells:

$$\mathbf{U}_{ij}^n \approx \frac{1}{h^2} \int_{C_{ij}} \mathbf{U}(x, y, t_n)\, dx\, dy. \tag{3}$$

An integration of Eq. (1) over the cell C_{ij} yields a method to advance these approximations $\mathbf{U}_{ij}^n$ to a new approximate solution $\mathbf{U}_{ij}^{n+1}$ at time $t_{n+1} = t_n + \Delta t$,

$$\mathbf{U}_{ij}^{n+1} = \mathbf{U}_{ij}^n + \frac{\Delta t}{h}\left(\mathbf{F}_{ij}^n - \mathbf{F}_{i+1,j}^n + \mathbf{G}_{ij}^n - \mathbf{G}_{i,j+1}^n\right), \tag{4}$$

where the fluxes $\mathbf{F}_{ij}$, $\mathbf{G}_{ij}$ ($\mathbf{F}_{ij}$ is the flux between the cells $C_{i-1,j}$ and C_{ij}, and $\mathbf{G}_{ij}$ is the flux between the cells $C_{i,j-1}$ and C_{ij}) are given by the numerical method.

One method to calculate the fluxes $\mathbf{F}_{ij}$, $\mathbf{G}_{ij}$ is LeVeque's multidimensional method [7] which is of second-order accuracy and stable up to a Courant number $cfl = 1.0$, because it takes into account transverse fluxes. The Courant number is defined by

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

where $v_{\max}$ is the maximum characteristic speed occurring in the solution. This shock-capturing scheme is used for the numerical experiments.

If an object is put into a Cartesian grid, then a special treatment of the cells cut by the boundary of the object is necessary. Pember *et al.* [9] give a method which makes it possible to treat these boundary cells like regular cells, thus avoiding instability problems for small

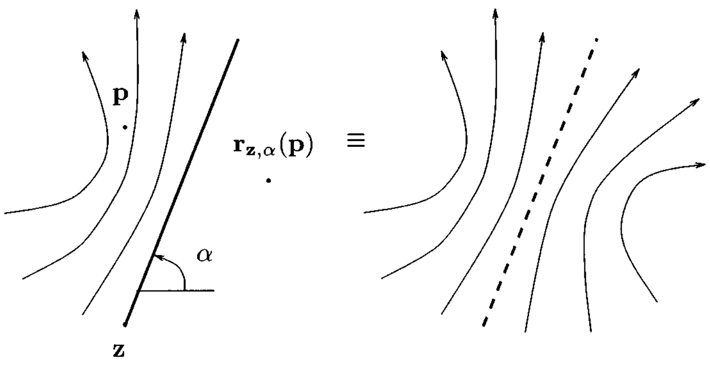

FIG. 1. Symmetry line of a wall boundary for an inviscid flow.

cut cells. The method is first-order accurate along the boundary. Our new method also treats boundary cells as whole cells but is of higher-order accuracy along the boundary.

The 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. A point $\mathbf{p} = (x, y)^T$ can be reflected from the physical area into a point $\mathbf{r}_{\mathbf{z},\alpha}(\mathbf{p})$ in the area beyond the boundary by

$$\mathbf{r}_{\mathbf{z},\alpha}(\mathbf{p}) = \mathbf{p} - 2\mathbf{n}(\mathbf{p} - \mathbf{z})^T\mathbf{n}. \tag{6}$$

Introducing the reflection matrix

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

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

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

such that the extrapolated solution fulfills the governing equations (cf. Fig. 1). This is possible because for the velocity vector $\mathbf{u}$ along the reflecting wall boundary $\mathbf{u}^T\mathbf{n} = 0$.

In Section 2 the new boundary treatment is described for the case of a reflecting boundary in one dimension and some numerical results are given concerning the order of accuracy and the conservation. In Section 3 the method is described for wall boundaries in two dimensions with some numerical results.

2. ONE DIMENSION

In this section the boundary treatment is described for the one-dimensional Euler equations,

$$\mathbf{U}_t + \mathbf{F}_x = 0, \tag{9}$$

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

For a discretization the real axis is divided into cells:

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

In a finite-volume method the numerical solution at time t_n is given by approximations of the cell averages,

$$\mathbf{U}_i^n \approx \frac{1}{h}\int_{C_i} \mathbf{U}(x, t_n)\, dx, \tag{12}$$

where $\mathbf{U}(x, t_n)$ is the exact solution at time t_n. These values are then integrated in time by

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n + \frac{\Delta t}{h}\left(\mathbf{F}_i^n - \mathbf{F}_{i+1}^n\right), \tag{13}$$

where $\mathbf{F}_i^n$ are approximations of the exact time-averaged fluxes. A standard method for the numerical calculation of the fluxes $\mathbf{F}_i^n$ 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} by a flux solver

$$\mathbf{F}_i^n = \mathcal{F}\left(\mathbf{U}_{i-2}^n, \mathbf{U}_{i-1}^n, \mathbf{U}_i^n, \mathbf{U}_{i+1}^n\right). \tag{14}$$

2.1. *Description of the Boundary Treatment*

Suppose there is a reflecting wall at the left-hand side of a flow field at the position $x = a$, where $x_{i_w} < a < x_{i_w+1}$ for a certain $i_w \in \mathbb{Z}$. The grid cells (11) are divided into four types as follows: $C_{i_w-3}, C_{i_w-4}, \ldots$ are empty cells; C_{i_w-2} and C_{i_w-1} are ghost cells; C_{i_w} is a boundary cell and $C_{i_w+1}, C_{i_w+2}, \ldots$ are regular cells (cf. Fig. 2). The reflection matrix $\mathbf{R}$ in one dimension is given by

$$\begin{pmatrix} \rho \\ -\rho u \\ \rho e \end{pmatrix} = \mathbf{R}\begin{pmatrix} \rho \\ \rho u \\ \rho e \end{pmatrix}, \quad \mathbf{R} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{15}$$

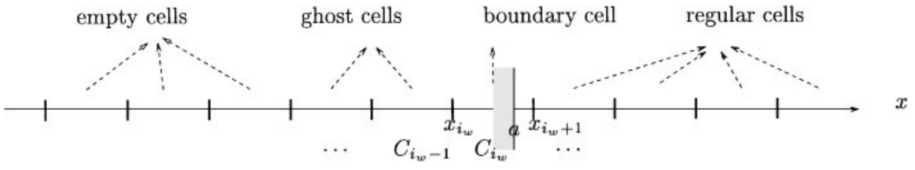

FIG. 2. Definition of the different cells near a wall boundary.

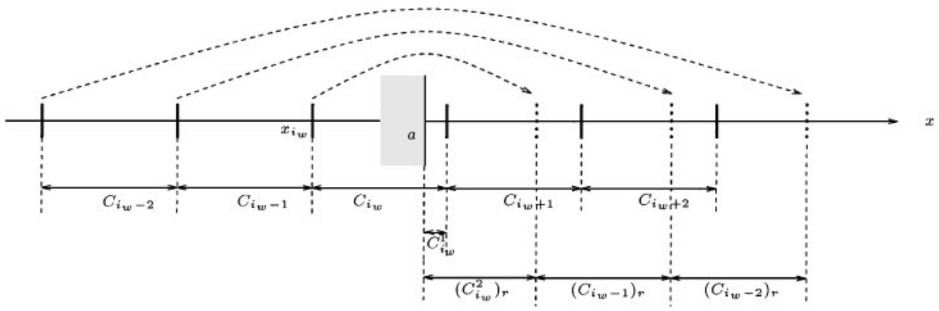

FIG. 3. Auxiliary cells for the calculation of the cell averages.

We define the following auxiliary cells (cf. Fig. 3):

$$C^1_{i_w} = [a, x_{i_w+1}], \tag{16}$$

$$C^2_{i_w} = [x_{i_w}, a], \tag{17}$$

$$\left(C^2_{i_w}\right)_r = [a, 2a - x_{i_w}], \tag{18}$$

$$(C_{i_w-1})_r = [2a - x_{i_w}, 2a - x_{i_w-1}], \tag{19}$$

$$(C_{i_w-2})_r = [2a - x_{i_w-1}, 2a - x_{i_w-2}]. \tag{20}$$

Using the reflecting extrapolation of the flow beyond the boundary, a cell average of the exact solution over the boundary cell C_{i_w} can be defined. Let $\mathbf{U}^n_{i_w}$ be an approximation for this generalized cell average:

$$\mathbf{U}^n_{i_w} \approx \frac{1}{h}\left(\int_{C^1_{i_w}} \mathbf{U}(x, t_n)\, dx + \int_{(C^2_{i_w})_r} \mathbf{RU}(x, t_n)\, dx\right). \tag{21}$$

Given $\mathbf{U}^n_{i_w}, \mathbf{U}^n_{i_w+1}, \mathbf{U}^n_{i_w+2}, \ldots,$ the cell averages of the boundary cell C_{i_w} and the regular cells $C_{i_w+1}, C_{i_w+2}, \ldots$ at time t_n, the fluxes $\mathbf{F}^n_{i_w+2}, \mathbf{F}^n_{i_w+3}, \ldots$ can be calculated with a 4-point stencil flux solver (14). With these fluxes and the finite-volume method (13) the cell averages $\mathbf{U}^n_{i_w+2}, \mathbf{U}^n_{i_w+3}, \ldots$ can be updated.

In order to advance the values $\mathbf{U}^n_{i_w}$ and $\mathbf{U}^n_{i_w+1}$, however, the fluxes $\mathbf{F}^n_{i_w}$ and $\mathbf{F}^n_{i_w+1}$ are needed. We want to calculate these two fluxes in the same way as the fluxes between regular cells, i.e., with the same flux solver (14). Thus, ghost cell values $\bar{\mathbf{U}}^n_{i_w-1}, \bar{\mathbf{U}}^n_{i_w-2}$ must be provided, where the bars signify ghost cell values. Given some approximation $\mathbf{U}^*(x, t_n)$ of the exact solution $\mathbf{U}(x, t_n)$, values $\bar{\mathbf{U}}^n_{i_w-1}, \bar{\mathbf{U}}^n_{i_w-2}$ can be obtained by the following cell averages:

$$\bar{\mathbf{U}}^n_{i_w-1} = \frac{1}{h}\int_{(C_{i_w-1})_r} \mathbf{RU}^*(x, t_n)\, dx, \tag{22}$$

$$\bar{\mathbf{U}}^n_{i_w-2} = \frac{1}{h}\int_{(C_{i_w-2})_r} \mathbf{RU}^*(x, t_n)\, dx. \tag{23}$$

For the approximate solution $\mathbf{U}^*$, we take the piece-wise constant step function. Now the fluxes $\mathbf{F}^n_{i_w+1} = \mathcal{F}(\bar{\mathbf{U}}^n_{i_w-1}, \mathbf{U}^n_{i_w}, \mathbf{U}^n_{i_w+1}, \mathbf{U}^n_{i_w+2})$ and $\mathbf{F}^n_{i_w} = \mathcal{F}(\bar{\mathbf{U}}^n_{i_w-2}, \bar{\mathbf{U}}^n_{i_w-1}, \mathbf{U}^n_{i_w}, \mathbf{U}^n_{i_w+1})$ can be calculated and the cell averages $\mathbf{U}^n_{i_w}$ and $\mathbf{U}^n_{i_w+1}$ can be advanced by a time step Δt. The conservative interpolation for the ghost cells is obtained by an overlapping of the reflected ghost cells and the Cartesian-grid cells. Since these cells are of the same size, this interpolation is linearity-preserving.

Remark 1. The calculation of the ghost cell values $\bar{\mathbf{U}}^n_{i_w-1}$ and $\bar{\mathbf{U}}^n_{i_w-2}$ using a piece-wise constant reconstruction for $\mathbf{U}^*$ is equivalent to

$$\bar{\mathbf{U}}^n_{i_w-1} = \mathbf{R}\mathbf{U}^{**}\left(2a - \left(x_{i_w-1} + \frac{1}{2}h\right), t_n\right), \tag{24}$$

$$\bar{\mathbf{U}}^n_{i_w-2} = \mathbf{R}\mathbf{U}^{**}\left(2a - \left(x_{i_w-2} + \frac{1}{2}h\right), t_n\right), \tag{25}$$

where $\mathbf{U}^{**}(x, t_n)$ now is a piece-wise linear reconstruction obtained by linearly connecting the cell averages $\mathbf{U}^n_i$ at the cell centers $x_i + \frac{1}{2}h$. Note that $2a - (x_{i_w-1} + \frac{1}{2}h)$ and $2a - (x_{i_w-2} + \frac{1}{2}h)$ are the cell centers of the reflected cells $(C_{i_w-1})_r$ and $(C_{i_w-2})_r$, respectively. This form is useful for extensions of the boundary treatment to more than one dimension, because it is simpler to evaluate a piece-wise linear function at some point than to integrate a piece-wise constant function over a domain given by the intersection of cells.

As a finite-volume method our boundary treatment is written in conservation form. For the method to be conservative, however, the flux along the boundary must be given by the wall pressure p_w as follows:

$$\mathbf{F}_{\text{boundary}} = \begin{pmatrix} 0 \\ p_w \\ 0 \end{pmatrix}. \tag{26}$$

No mass, no energy, and no advected momentum flows into the domain. For the new boundary treatment no flux at the wall boundary is calculated but at cell interfaces close to the wall boundary. Thus, for wall boundaries not aligned with the grid interfaces, condition (26) is only fulfilled approximately. LeVeque shows in [6, p. 123] how a nonconservative method can lead to a wrong propagation speed of shock waves. Therefore, in the next section a strong shock reflection off a solid wall is studied. It turns out that a proper reflection of the shock is obtained.

2.2. *Numerical Results*

In this section numerical results of flows involving a reflecting wall boundary are shown. The second-order accurate finite-volume method CLAWPACK [5] is used, with the approximate Riemann-solver of Roe [11].

The first calculation shows a strong Mach 10.0 shock reflection with a shock wave starting at $x = 0.5$ hitting a wall on the left and being reflected there. The discretization is given by $x_i = ih$, $C_i = [x_i, x_{i+1}]$, where $h = 1/N$ is the grid parameter. The wall is located at $x_w = \alpha h$, $\alpha \in [0, 1)$. Thus, C_0 is a boundary cell. The initial data and the reflected state are given in Fig. 4. For the first calculation (Fig. 5 on the left), the wall is situated at $x = 0.0$. In this special case the boundary treatment is conservative. This result is compared with a calculation where the wall is situated at $x = \alpha h$, $\alpha = 0.8$ (Fig. 5 on the right), where the

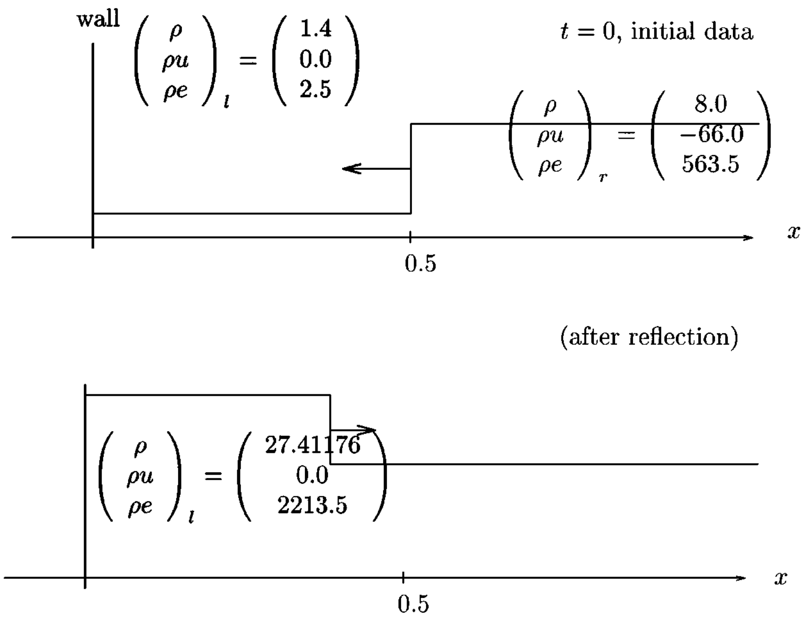
wall
$\begin{pmatrix}\rho\\ \rho u\\ \rho e\end{pmatrix}_l = \begin{pmatrix}1.4\\ 0.0\\ 2.5\end{pmatrix}$
$t = 0$, initial data
$\begin{pmatrix}\rho\\ \rho u\\ \rho e\end{pmatrix}_r = \begin{pmatrix}8.0\\ -66.0\\ 563.5\end{pmatrix}$
x
0.5
(after reflection)
$\begin{pmatrix}\rho\\ \rho u\\ \rho e\end{pmatrix}_l = \begin{pmatrix}27.41176\\ 0.0\\ 2213.5\end{pmatrix}$
x
0.5

FIG. 4. Strong shock reflection, exact solution.

boundary treatment is not conservative. For both calculations a Courant number of $cfl = 0.8$ is used. The comparison shows that the reflected shock location comes out correctly. (The loss in density close to the wall, which appears in both cases is related to the numerical wall heating).

An analysis of the nonphysical mass production or loss for a time $t_n = 0.25$ where the shock wave has already been reflected at the wall but is still in the computational domain gives

$$\Delta\text{mass} = h\left(\sum_{i=1}^{N-1} \rho_i^n + (1-\alpha)\rho_0^n\right) - \int_{\alpha h}^{1} \rho_{\text{exact}}(x, t_n)\, dx. \tag{27}$$

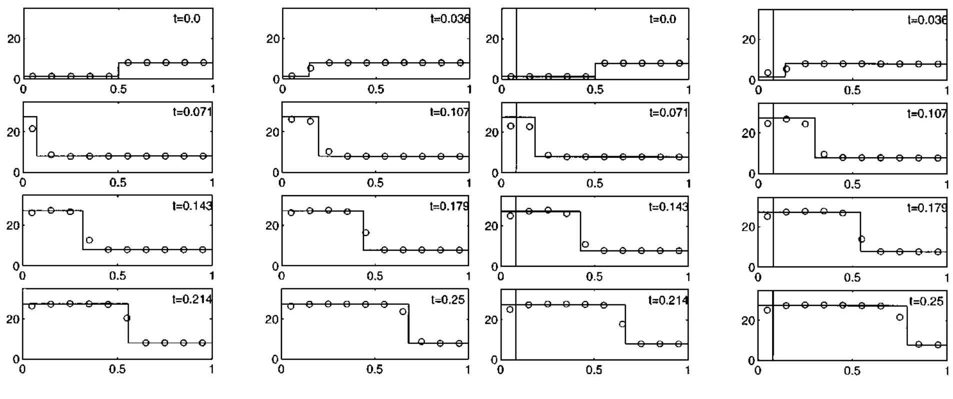

FIG. 5. Density plot of a Mach 10 shock reflection with a wall located at $x = 0.0$ (left) and at $x = 0.08$ (right) (circles denote the numerical solution, the solid line the exact solution).

TABLE 1
Nonphysical Mass Production or Loss Δmass and the Resulting Error in the Shock Location $\Delta\sigma$ for a Mach 10 Reflection at Time $t = 0.25$

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	Δmass	α	Δmass	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			$\vdots$

A change in the total mass results in an error of the shock location. Using the equal area rule [6, p. 35] a numerical shock location σ can be obtained. Subtracting from this the exact shock location, the error in the shock location $\Delta\sigma$ can be obtained

$$\Delta\sigma = \frac{\Delta\text{mass}}{\rho_l - \rho_r}, \tag{28}$$

where ρ_l, ρ_r are the densities of the exact solution on the left and on the right of the reflected shock wave. Table 1 shows 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 cell.

For one-dimensional calculations this nonphysical mass production or loss only happens at the precise moment when the shock wave is reflected at the boundary. For the rest of the time the solution will be smooth at the boundary and the mass production is then quadratic in h as the global accuracy of the method for smooth flows (see below).

As a next test case, smooth initial data for $x \in [0, 1]$ with solid wall boundary conditions on both sides at $x = 0.0$ and $x = 1.0$ are taken such that the flow field stays smooth, at least until time $t = 1.0$ (cf. Fig. 6). In this example the order of accuracy of the boundary treatment is evaluated in an error analysis. We compare the higher-order boundary treatment with a conservative first-order accurate boundary treatment (h-box method of Berger and LeVeque [1]).

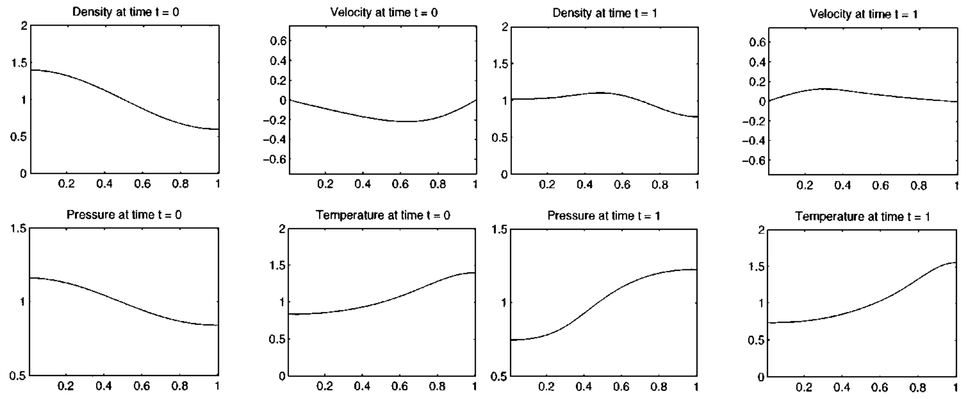

FIG. 6. The smooth flow test calculation with $h = 0.01$.

We use a grid such that the first cell C_0 is a boundary cell. With grid parameters $h = 1/n, n \in \mathbb{N}$, and $\alpha \in (0, 1)$ the grid is defined by

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

For the calculation the Courant number is fixed and the initial data is advanced from time $t = 0.0$ to time $t = 1.0$. Since the solution stays smooth for this time interval, limiting of gradients is not necessary and second-order accuracy is expected also at the boundary.

The final densities ρ_0 and ρ_1 in the cells C_0 and C_1 are approximate values for the density at the points $x = |\alpha - 0.5|h$ and $x = |\alpha + 0.5|h$, respectively. These two values, together with $(\partial/\partial x)\rho$ $(x = 0)$, give ρ_{wall}, the wall density at time $t = 1.0$ by quadratic interpolation:

$$\rho_{\text{wall}} = \rho(x = 0) = \frac{\rho_0\left(\alpha + \frac{1}{2}\right)^2 - \rho_1\left(\alpha - \frac{1}{2}\right)^2}{2\alpha}. \tag{29}$$

Using linear interpolation, a final density value ρ_{half} at time $t = 1.0$ and at $x = 0.5$ is obtained.

As the exact solution is not available in closed form, we calculate it with a fine grid of 1280 grid points. For this "exact" solution we obtain $\rho_{\text{wall}} = 1.0202003$ and $\rho_{\text{half}} = 1.1031761$.

The global accuracy is analyzed by looking at the convergence of the density variable ρ_{half}. The convergence history in Table 2 (top) for the density values ρ_{wall} and ρ_{half} for a grid with $\alpha = 0.25$ shows that using a first-order treatment of the boundary and a second-order method elsewhere, the order of accuracy is still second-order in the whole domain, as predicted in [13]. On the other hand, the convergence history in Table 2 (bottom) demonstrates that using our higher-order boundary treatment, together with a second-order finite-volume method, the convergence is of second order, not only for the whole domain, but also along the boundary.

TABLE 2
Convergence History for the Smooth One-Dimensional Test Example with the Conservative h-Box Method (Top) and the New Nonconservative Boundary Treatment (Bottom)

h	ρ_{wall}	ρ_{half}	$\lvert\Delta\rho_{\text{wall}}\rvert$	Order	$\lvert\Delta\rho_{\text{half}}\rvert$	Order
$h_0 = 0.1$	1.024664	1.096459	0.004430		0.006720	
$\frac{h_0}{2}$	1.022522	1.101377	0.002290	0.95	0.001800	1.91
$\frac{h_0}{4}$	1.021261	1.102835	0.001030	1.15	0.000356	2.34
$\frac{h_0}{8}$	1.020710	1.103084	0.000479	1.11	0.000092	1.95
$\frac{h_0}{16}$	1.0204769	1.1031527	0.0002450	1.14	0.0000237	1.95
$h_0 = 0.1$	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

3. TWO DIMENSIONS

Now we extend the method to handle reflecting wall boundaries in two dimensions. Still, the boundary treatment is independent of the method for the regular cells as long as the method can be written in conservation form (4).

3.1. *Description of the Boundary Treatment*

Without loss of generality, let us assume that in order to calculate the flux $\mathbf{F}_{kl}$, the flow variables of the cells $C_{k-2,l}$, $C_{k-1,l}$, $C_{k,l}$, $C_{k+1,l}$ are used by the flux solver:

$$\mathbf{F}^n_{kl} = \mathcal{F}\left(\mathbf{U}^n_{k-2,l}, \mathbf{U}^n_{k-1,l}, \mathbf{U}^n_{kl}, \mathbf{U}^n_{k+1,l}\right). \tag{30}$$

But the method can take into account the neighboring layers of grid cells $C_{k-2,l-1}$, $C_{k-1,l-1}$, $C_{k,l-1}$, $C_{k+1,l-1}$ and $C_{k-2,l+1}$, $C_{k-1,l+1}$, $C_{k,l+1}$, $C_{k+1,l+1}$.

The cells which are cut by the reflecting wall are denoted as cut cells. For each cut cell the boundary of the object is approximated by a straight line. The other cells are either regular or empty (cf. Fig. 7). The numerical solution for a regular cell at time t_n is given by an approximation of the cell average (3). To define a numerical solution for a boundary cell C_{kl}, as well, let us first make some geometrical definitions.

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. Fig. 8). We divide the boundary cell C_{kl} 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. Fig. 8). In (6) it is 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 $\mathbf{p}_1, \ldots, \mathbf{p}_m$ as polygon $r_{\mathbf{z},\alpha}(C)$ with

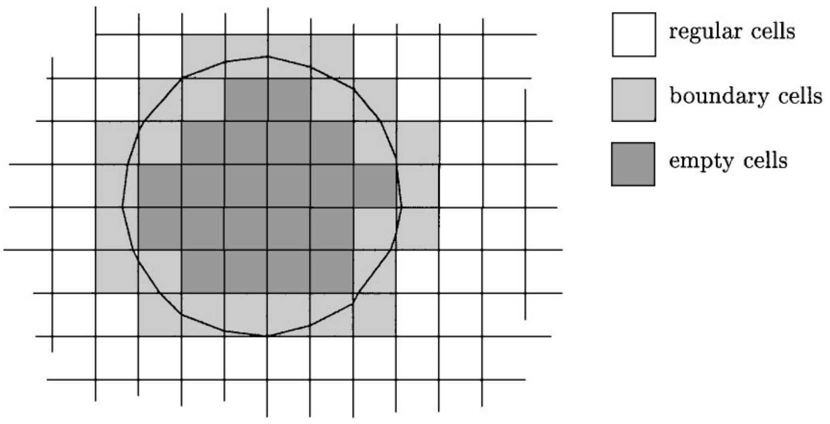

FIG. 7. Discretization of a circle.

the reflected corner points $\mathbf{r}_{\mathbf{z},\alpha}(\mathbf{p}_1), \ldots, \mathbf{r}_{\mathbf{z},\alpha}(\mathbf{p}_m)$. If a polygon is reflected at the straight boundary line going through the boundary cell C_{kl}, the notation is simplified by

$$r_{kl}(C) := r_{\mathbf{z}_{kl},\alpha_{kl}}(C). \tag{31}$$

Figure 9 shows the reflection of C^2_{kl} and $C_{k+1,l}$ onto $r_{kl}(C^2_{kl})$ and $r_{kl}(C_{k+1,l})$. The solution near the boundary cell C_{kl} can be extrapolated beyond the boundary using the reflection matrix $\mathbf{R}_{\alpha_{kl}}$ (7).

For a boundary cell C_{kl}, the numerical solution at time t_n is given by an approximation of the cell average over the whole cell using the reflected extrapolation of the exact solution:

$$\mathbf{U}^n_{kl} \approx \frac{1}{h^2}\left(\int_{C^1_{kl}} \mathbf{U}(x,y,t_n)\,dx\,dy + \int_{r_{kl}(C^2_{kl})} \mathbf{R}_{\alpha_{kl}}\mathbf{U}(x,y,t_n)\,dx\,dy\right). \tag{32}$$

With this definition, we can proceed as in one dimension. If an approximation of the solution at time $t_n = t_0 + n\Delta t$ is given as a piece-wise constant function over the regular

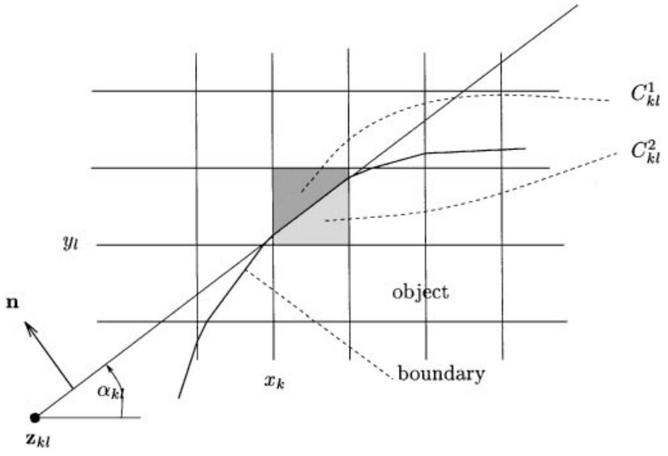

FIG. 8. Description of the boundary segment through the cell C_{kl}.

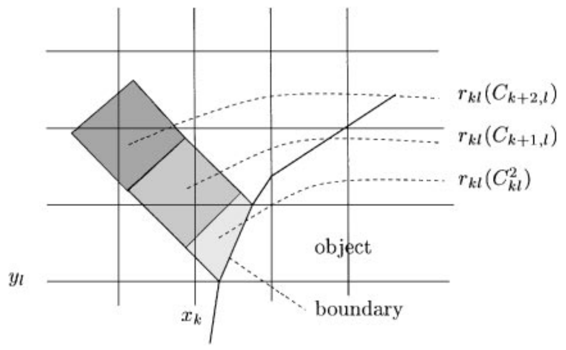

FIG. 9. Reflection at the boundary segment of cell C_{kl}.

and the boundary cells, this solution is advanced using (4). Suppose the lth horizontal layer of grid cells consists of regular cells $C_{k-1,l}, C_{k-2,l}, \ldots$, of the boundary cell C_{kl}, and of the empty cells $C_{k+1,l}, C_{k+2,l}$ as in Fig. 9. With the 4-point stencil flux solver (30) the **F**-fluxes $\mathbf{F}_{k-1,l}, \mathbf{F}_{k-2,l}, \ldots$, can be obtained and $\mathbf{U}_{k-2,l}, \mathbf{U}_{k-3,l}, \ldots$, can be updated. But in order to update also $\mathbf{U}_{k-1,l}$ and $\mathbf{U}_{kl}$ using (4) the **F**-fluxes $\mathbf{F}_{kl}$ and $\mathbf{F}_{k+1,l}$ are needed as well. To calculate these two fluxes with the same stencil, ghost cell variables $\bar{\mathbf{U}}_{k+1,l}$ and $\bar{\mathbf{U}}_{k+2,l}$ are necessary. These variables are obtained as integrals over the reflected cells as

$$\begin{aligned}\bar{\mathbf{U}}^n_{k+1,l} &= \int_{r_{kl}(C_{k+1,l})} \mathbf{R}_{\alpha_{kl}} \mathbf{U}^*(x, y, t_n)\, dx\, dy,\\ \bar{\mathbf{U}}^n_{k+2,l} &= \int_{r_{kl}(C_{k+2,l})} \mathbf{R}_{\alpha_{kl}} \mathbf{U}^*(x, y, t_n)\, dx\, dy.\end{aligned} \tag{33}$$

For the approximate solution $\mathbf{U}^*(.,.,t_n)$, we take the piece-wise constant step function. Then $\mathbf{F}^n_{kl} = \mathcal{F}(\mathbf{U}^n_{k-2,l}, \mathbf{U}^n_{k-1,l}, \mathbf{U}^n_{k,l}, \bar{\mathbf{U}}^n_{k+1,l})$ and $\mathbf{F}^n_{k+1,l} = \mathcal{F}(\mathbf{U}^n_{k-1,l}, \mathbf{U}^n_{k,l}, \bar{\mathbf{U}}^n_{k+1,l}, \bar{\mathbf{U}}^n_{k+2,l})$ and $\mathbf{U}^n_{kl}$ and $\mathbf{U}^n_{k-1,l}$ can be updated. As in one dimension the interpolation (33) is linearity-preserving because it is obtained by overlapping the regular Cartesian-grid cells with equal-sized reflected Cartesian-grid cells.

Remark 2. Equation (33) involves the evaluation of intersections of arbitrarily oriented polygons, which is complicated to program. Reasonable and computationally simpler ghost cell values $\bar{\mathbf{U}}^n_{k+1,l}$ and $\bar{\mathbf{U}}^n_{k+2,l}$ can be obtained by evaluating a piece-wise linear reconstruction of the solution at the center of the reflected cells (cf. Remark 1),

$$\begin{aligned}\bar{\mathbf{U}}^n_{k+1,l} &= \mathbf{R}_{\alpha_{kl}} \mathbf{U}^{**}\left(\mathbf{r}_{z_{kl},\alpha_{kl}}\left(x_{k+1} + \frac{1}{2}h, y_l + \frac{1}{2}\right), t_n\right),\\ \bar{\mathbf{U}}^n_{k+2,l} &= \mathbf{R}_{\alpha_{kl}} \mathbf{U}^{**}\left(\mathbf{r}_{z_{kl},\alpha_{kl}}\left(x_{k+2} + \frac{1}{2}h, y_l + \frac{1}{2}\right), t_n\right),\end{aligned} \tag{34}$$

where $\mathbf{U}^{**}(x, y, t_n)$ is a piece-wise linear reconstruction through the cell averages $\mathbf{U}^n_{ij}$ at the nearest three Cartesian-grid cell centers. But (33) and (34) are not equivalent anymore as they were in one dimension (cf. previous section), because the orientation of a reflected cell is defined not only by its center but also by an angle. We prefer (34), however, since it is simpler than using (33)—especially in three dimensions—and the results are qualitatively the same, as well as with respect to the order of accuracy.

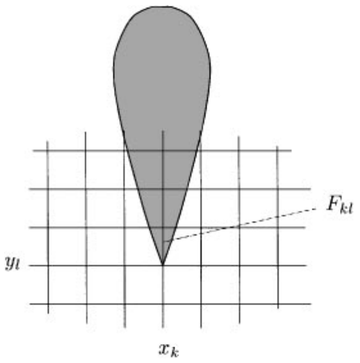

FIG. 10. Discretization of an airfoil.

Remark 3. Some difficulties in the implementation can arise with thin geometry. If the cells $C_{k+1,l}$ and $C_{k+2,l}$ are empty cells next to a boundary cell C_{kl}, for some cases one cannot use their memory location to store the ghost cell values $\bar{\mathbf{U}}^n_{k+1,l}$ and $\bar{\mathbf{U}}^n_{k+2,l}$. For example, if the cells $C_{k-1,l}$ and C_{kl} are lying on the trailing edge of an airfoil as in Fig. 10. At the cell interface between these two cells, two different **F**-fluxes $\mathbf{F}_{kl}$ have to be calculated, $\mathbf{F}^1_{kl}$ for the $C_{k-1,l}$ update and $\mathbf{F}^2_{kl}$ for the C_{kl} update. Using the 4-point stencil (30),

$$\mathbf{F}^1_{kl} = \mathcal{F}\big(\mathbf{U}^n_{k-2,l}, \mathbf{U}^n_{k-1,l}, \bar{\mathbf{U}}^n_{kl}, \bar{\mathbf{U}}^n_{k+1,l}\big), \tag{35}$$

$$\mathbf{F}^2_{kl} = \mathcal{F}\big(\bar{\mathbf{U}}^n_{k-2,l}, \bar{\mathbf{U}}^n_{k-1,l}, \mathbf{U}^n_{kl}, \mathbf{U}^n_{k+1,l}\big), \tag{36}$$

where $\mathbf{U}^n_{k-2,l}, \ldots, \mathbf{U}^n_{k+1,l}$ are the values of the regular or boundary cells and $\bar{\mathbf{U}}^n_{k-2,l}, \ldots, \bar{\mathbf{U}}^n_{k+1,l}$ are the ghost cell values. Thus, $\bar{\mathbf{U}}^n_{k-2,l}, \ldots, \bar{\mathbf{U}}^n_{k+1,l}$ need separate memory locations or $C_{k-1,l}$ and C_{kl} need to change the values between updating $\mathbf{U}^n_{k-1,l}$ and $\mathbf{U}^n_{kl}$. Analogously, if the trailing edge of the airfoil divides a cell into two parts, the so-called "split-cell" problem, two different values of the solution corresponding to a weighted mean on each side of the airfoil have to be maintained.

3.2. *Numerical Results*

For the following numerical examples the CLAWPACK package of LeVeque [5] is used as the underlying method for the regular cells. At the domain boundaries standard super- and subsonic in- and outflow techniques are used. For all calculations except the one involving the cylinder, ghost cell values are obtained by method (34).

In the first and second examples the geometry is given by a circular cylinder. First an incident shock wave reflects off a cylinder. For this example the mass nonconservation is analyzed. The second example is a subcritical Mach 0.38 inflow presented at a GAMM workshop [3]. The next example is a flow over a ramp, an example introduced by Woodward and Colella resulting in a double Mach reflection [14]. Next, a transonic flow around a NACA0012 airfoil described also in [3] is calculated. However, for this example Cartesian grids are not appropriate because of the smallness of the airfoil compared to the size of the computational domain necessary to obtain reliable results. The last example is a calculation of a Prandtl–Meyer expansion. As in [1, 9], with this example the order of accuracy of the method can be measured numerically.

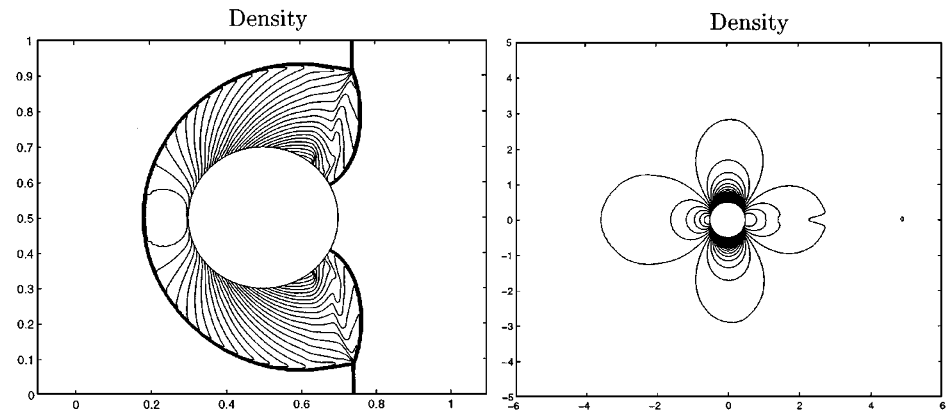

FIG. 11. Shock wave reflecting off a cylinder at time $t_n = 0.15$ (left), subcritical flow around a cylinder with Mach 0.38 (right).

In the first example the reflection off a cylinder of an incident shock wave traveling at a relative Mach number 3.0 is studied. 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. The Courant number ($cfl = 0.9$. Figure 11 on the left shows density contours for a calculation with a 300×300 grid at time $t_n = 0.15$. As in one dimension, we make an analysis of the nonphysical mass production or loss. For time $t_n = 0.15$ such that the shock wave is still in the computational domain, the total mass generation is

$$\Delta\text{mass} = \sum_{ij} \left|C_{ij}^1\right| \left(\rho_{ij}^n - \rho_{ij}^0\right) - v_S(\rho_l - \rho_r)t_n, \tag{37}$$

where it is summed 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 wave. Table 3 shows the relative mass production for different grids. 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. A subcritical Mach 0.38 inflow is studied on a 400×400 grid.

TABLE 3
Nonphysical Relative Mass Production for a Shock Reflection Off a Cylinder at Time t_n = 0.15

Grid	$\frac{\Delta\text{mass}}{\text{mass}}$
100×100	0.0216
200×200	0.0140
400×400	0.0088
800×800	0.0045

The grid cells have a length of $1/40 = 0.025$ such that the results can be compared with methods using a 128×32 O-mesh, where the cells of the first layer around the cylinder have a length of $\pi/128 = 0.0245$. The initial condition is a constant flow and the steady state is reached after a sufficiently large time. Figure 11 on the right shows the density contours at steady state. The maximum values for the entropy deviation, $\Sigma = (p/p_\infty)/(\rho/\rho_\infty)^\gamma - 1$, is $\Sigma_{\max} = 0.0038$ and for the Mach number is $M_{\max} = 0.9094$, which compares well with results using O-meshes, for example $\Sigma_{\max} = 0.0048$ in [8]. For a 200×200 grid a value $\Sigma_{\max} = 0.0098$ is obtained, which demonstrates the higher-order accuracy of the boundary treatment, since the entropy deviation here is a boundary effect.

The next example was introduced in [14] to compare different numerical schemes. It is a reflection of a Mach 10 shock wave at a ramp. The resulting configuration of a double Mach reflection contains several fluid dynamical features, such as a moving shock wave, a quasi-steady-state shock wave, a contact discontinuity, and two reflected shock waves. This example with tilting the reflecting wall needs a boundary treatment to model the reflecting boundary oblique to the grid. In [14] the incident shock wave is tilted, instead. In the computation $\Delta x = \Delta y = 1/120$, the angle of the ramp is 30° and for the gas at rest $p = 1.0$, $\rho = 1.4$ as in [14] to compare the results. The Mach 10 shock wave 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 Fig. 12 the results are shown using a 300×240 grid. The results compare well

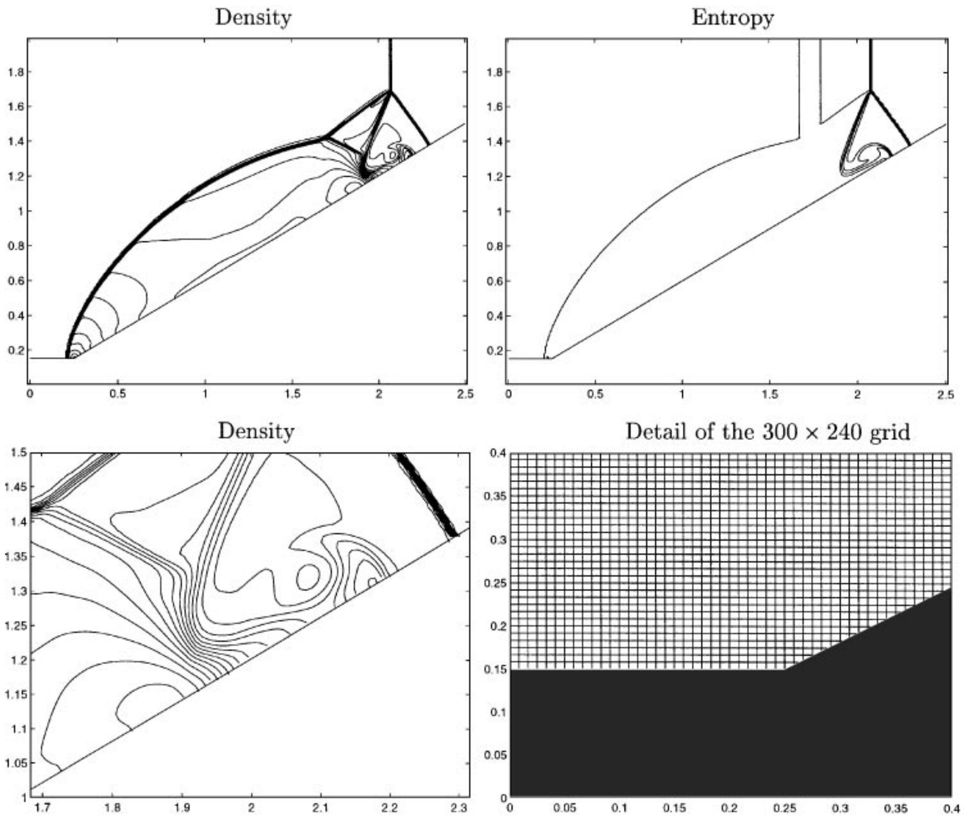

FIG. 12. Reflection of a Mach 10 shock wave at a 30° angle at time $t = 0.2$ using a grid with $\Delta x = \Delta y = 1/120$.

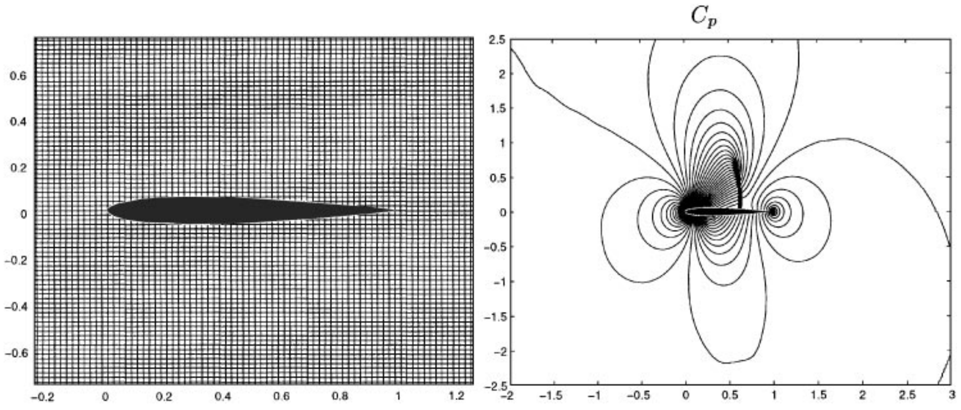

FIG. 13. Grid and pressure contours for a calculation of a Mach 0.8 flow around a NACA0012 airfoil with a 1.25° angle of attack, $\Delta x = \Delta y = 1/50$.

with [14]. It is very encouraging that the first reflected shock wave does not appear smeared as it hits the boundary.

Our next example is of more practical importance, namely, a transonic steady-state flow around a NACA0012 airfoil. The geometry is described in [3]. As asymptotic conditions, the free-stream Mach number $M_\infty = 0.8$ and the angle of attack $\alpha_\infty = 1.25°$ are chosen. For the in- and outflow condition we use a far-field correction following the ideas of Thomas and Salas [12].

The 400×400 Cartesian grid has a side length of 8 chords and the airfoil is placed at a distance of 3 chords from the inflow boundary. Our Cartesian grid has $\Delta x = \Delta y = 1/50$, resulting in about 100 grid cells surrounding the airfoil. No refinement of the grid at the leading edge is used. Thus, we cannot expect that our results can compete with [12]. Figure 13 shows details of the grid 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 along the airfoil surface are shown in Fig. 14, where cell-centered boundary values are plotted. The forces on the airfoil are given in the nondimensionalized drag and lift coefficients,

$$c_d = \frac{2}{\rho_\infty u_\infty^2 L} \oint (\rho u \mathbf{u}^T \mathbf{n} + p n_x)\, dS, \tag{38}$$

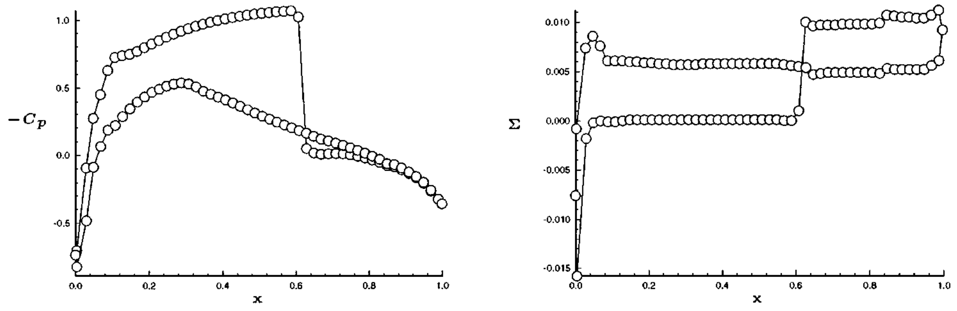

FIG. 14. Distribution of pressure and entropy deviation around the surface of the airfoil.

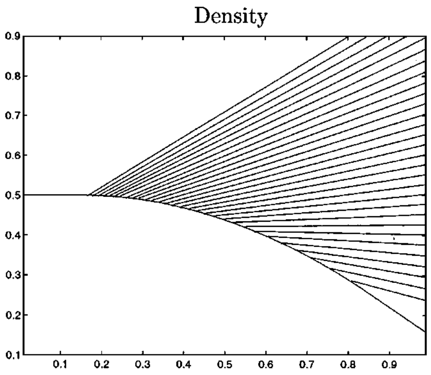

FIG. 15. Expansion of a Mach 1.83 flow over a 35° bend, $h = \Delta x = \Delta y = 1/200$.

$$c_l = \frac{2}{\rho_\infty u_\infty^2 L} \oint (\rho v \mathbf{u}^T \mathbf{n} + p n_y)\, dS, \tag{39}$$

where L is the chord length of the airfoil and the integration goes along an arbitrary closed contour S enclosing the airfoil with outward-facing normal vector $\mathbf{n} = (nx, ny)^T$. Our results yield $c_d = 0.0205$ and $c_l = 0.3118$, compared with $c_d = 0.0226$ and $c_l = 0.358$ of [12], where a 256×64 O-mesh and a domain size of 200 chords is used. As can be seen in Fig. 14 on the right, the numerical entropy production at the leading edge of the airfoil is of the order of the physical entropy production across the shock wave on the upper surface in our calculation. Thus, to get better results for this example the grid should be refined there.

The last example is a steady-state calculation of a Prandtl–Meyer expansion for a Mach 1.83 flow over a 35° bend. The exact solution is a smooth flow with constant entropy $S = p/\rho^\gamma$ and constant stagnation enthalpy $H = (p + \rho e)/\rho$. Figure 15 shows a density contour plot of a calculation with a 200×200 grid. As introduced by Berger and LeVeque, this example can be used to analyze the order of accuracy of the method by taking the L_1-norm of the error in entropy and stagnation enthalpy in the whole flow area or only along the boundary (cf. [1, 9]):

$$\text{error}_{\text{total}} = \frac{\sum^1 \left|u_{ij} - u_{ij}^c\right|}{\sum^1 1}, \tag{40}$$

$$\text{error}_{\text{boundary}} = \frac{\sum^2 \left|u_{ij} - u_{ij}^c\right|}{\sum^2 1}, \tag{41}$$

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 grid cells, and $\sum^2$ is a summation over the boundary cells only. Under the assumption that the error can be expressed as error $= Ch^p$, the order p of the method can be obtained numerically in the whole area or only along the boundary. Table 4 shows two convergence histories. For the first one, Table 4 (top), a first-order boundary treatment is used. Our first-order boundary method is given by the new boundary treatment without the second-order correction terms in the flux calculation along the interface of boundary cells and ghost cells. The second one, Table 4 (bottom), uses the higher-order boundary treatment. The results for entropy suggest that the method is of second-order accuracy in the whole area for both the first- and the second-order boundary treatment;

TABLE 4
Convergence History for the Prandtl–Meyer Expansion, First-Order Boundary Treatment (Top) and Higher-Order Boundary Treatment (Bottom)

	Entropy		Stagnation enthalpy	
h	$\text{Error}_{\text{total}}$	Order	$\text{Error}_{\text{total}}$	Order
$\frac{1}{200}$	0.0013671	—	0.019595	—
$\frac{1}{400}$	0.0003556	1.94	0.008849	1.15
$\frac{1}{700}$	0.00011875	1.96	0.0049026	1.06
	$\text{Error}_{\text{boundary}}$		$\text{Error}_{\text{boundary}}$	
$\frac{1}{200}$	0.020652	—	0.201197	—
$\frac{1}{400}$	0.008769	1.24	0.164443	0.29
$\frac{1}{700}$	0.004372	1.24	0.148709	0.18
	$\text{Error}_{\text{total}}$		$\text{Error}_{\text{total}}$	
$\frac{1}{200}$	0.0007390	—	0.004332	—
$\frac{1}{400}$	0.0001849	2.00	0.001758	1.30
$\frac{1}{700}$	0.00005895	2.04	0.0008545	1.29
	$\text{Error}_{\text{boundary}}$		$\text{Error}_{\text{boundary}}$	
$\frac{1}{200}$	0.009998	—	0.033900	—
$\frac{1}{400}$	0.003786	1.40	0.018621	0.87
$\frac{1}{700}$	0.001688	1.48	0.016400	0.23

i.e., the order in the whole domain is not affected by a lower order boundary treatment (cf. [13]). But the order of accuracy along the boundary is better using the higher-order boundary treatment, and the value of $p = 1.48$ for the order in entropy along the boundary is an improvement, compared with other Cartesian-grid methods, e.g., [9], where $p = 1.2$ is obtained, or [2], where $p = 1.4$ is obtained, using a code for the steady-state Euler equations. The results for stagnation enthalpy are less satisfying, but neither in this paper nor in [9] can an explanation for this can be given.

4. CONCLUSIONS

For Cartesian-grid methods for the Euler equations, a boundary treatment is given based on reflecting locally the flow field at a straight boundary line. The method is higher-order accurate, simple, and applicable to any finite-volume method.

Numerical results in one dimension show that second-order accuracy is obtained for smooth flows also at the boundary. Despite the violation of the conservation at the boundary, correct shock reflections off the boundary in one dimension are obtained within first-order accuracy.

In two dimensions we have 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 is 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 is shown to be negligible for a shock wave traveling along the boundary oblique to the grid 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 is limited, due to the high curvature of the boundary at the leading edge of the airfoil. By means of a smooth steady-state flow calculation, the order of accuracy of the method in the whole domain and along the boundary is measured analytically. In the whole domain the method is second order and along the boundary we obtain an order of 1.48 which is an improvement compared with other boundary treatments.

In future research the accuracy of the two-dimensional method may be further improved by taking into account the curvature of the wall boundary or by improving the way ghost cell values are obtained. Also for some examples the loss of conservation could be a problem, where we would have to look for techniques to fix the conservation.

ACKNOWLEDGMENTS

We thank Professor R. LeVeque, Dr. N. Botta, and Professor M. Berger for their suggestions and for helpful discussions during the course of this research.

REFERENCES

1. M. J. Berger and R. J. LeVeque, *A Rotated Difference Scheme for Cartesian Grids in Complex Geometries*, AIAA Paper CP-91-1602, 1991.
2. W. J. Coirier and K. G. Powell, An accuracy assessment of Cartesian-mesh approaches for the Euler equations, *J. Comput. Phys.* **117**, 121 (1995).
3. A. Dervieux and B. van Leer, *GAMM workshop on Numerical Solution of the Euler Equation*, Notes on Numerical Fluid Mechanics, Vol. 26 (Vieweg, Wiesbaden, 1989).
4. H. Forrer, *Boundary Treatment for a Cartesian Grid Method*, ETH Research-Report No. 96-04, available via `netscape` in `http://www.sam.math.ethz.ch/Reports/1996-04.html`, 1996.
5. R. J. LeVeque, CLAWPACK software, available from `http://www.amath.washington.edu/~rjl/clawpack.html`.
6. R. J. LeVeque, *Numerical Methods for Conservation Laws* (Birkhäuser, Zürich, 1992).
7. R. J. LeVeque, Simplified multi-dimensional flux limiter methods, *Numer. Methods Fluid Dyn.* **4**, 175 (1993).
8. R. Jeltsch and N. Botta, A numerical method for unsteady flows, *Appl. Math.* **40**(3), 175 (1995).
9. R. B. Pember, J. B. Bell, P. Colella, W. Y. Crutchfield, and M. L. Welcome, An adaptive Cartesian grid method for unsteady compressible flow in irregular regions, *J. Comput. Phys.* **120**, 278 (1995).
10. J. J. Quirk, An alternative to unstructered grids for computing gas dynamic flows around arbitrarily complex two-dimensional bodies, *Comput. Fluids* **23**(1), 125 (1994).
11. P. L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, *J. Comput. Phys.* **43**, 357 (1981).
12. J. L. Thomas and M. D. Salas, Far-field boundary conditions for transonic lifting solutions to the Euler equations, *AIAA Journal* **24**(7), (1986).
13. B. Wendroff and A. White, *On Irregular Grids*, Proc. 2nd Intl. Conf. Nonlinear Hyperbolic Problems, Notes on Num. Fluid Mech., Vol. 24 (1988).
14. P. Woodward and P. Colella, The numerical simulation of 2D fluid flow with strong shocks, *J. Comp. Phys.* **54**, 115 (1984).