ELSEVIER

Mathematics and Computers in Simulation 61 (2003) 561–572

MATHEMATICS AND COMPUTERS IN SIMULATION

www.elsevier.com/locate/matcom

Developments in Cartesian cut cell methods

D.M. Ingram*, D.M. Causon, C.G. Mingham

Centre for Mathematical Modelling and Flow Analysis, Manchester Metropolitan University, Chester Street, Manchester M1 5GD, UK

Abstract

This paper describes the Cartesian cut cell method, which provides a flexible and efficient alternative to traditional boundary fitted grid methods. The Cartesian cut cell approach uses a background Cartesian grid for the majority of the flow domain with special treatments being applied to cells which are cut by solid bodies, thus retaining a boundary conforming grid. The development of the method is described with applications to problems involving both moving bodies and moving material interfaces.
© 2002 IMACS. Published by Elsevier Science B.V. All rights reserved.

Keywords: Cut cells; Finite volume solver; Cartesian grids; Inviscid flows

1. Introduction

One of the first tasks to be faced in computational fluid dynamics (CFD) is the generation of a suitable computational mesh. Although a variety of mesh generation techniques are available [1], the generation of a suitable mesh for complex, multi-element, geometries is still a complex and tedious task. The two, traditional, approaches are: the use of a structured body-fitted mesh utilising a multi-block structure, in which the blocks may overlap [2–5] and the use of a completely unstructured body-fitted mesh [6–8]. Both of these approaches require significant efforts in mesh generation to ensure that the generated mesh is of sufficient quality to both accurately represent the geometry and provide a high quality solution. Even in cases where a detailed description of the geometry is available from a CAD system, mesh generation can still be a complex task [9], requiring much more time to generate the grid than to simulate the fluid flow.

An alternative approach is the use of Cartesian cut cells. This conceptually simple approach "cuts" solid bodies out of a background Cartesian mesh. Although originally developed for potential flow, the method has been successfully applied to the Euler equations in two [10–12] and three [13,14] space dimensions, to the shallow water equations (SWE) [15] and extended to deal with low speed incompressible flows [16–18] and flows involving moving material interfaces [19–21].

* Corresponding author. Tel.: +44-161-247-3349; fax: +44-161-247-6337.
E-mail address: d.m.ingram@mmu.ac.uk (D.M. Ingram).

0378-4754/02/$ – see front matter © 2002 IMACS. Published by Elsevier Science B.V. All rights reserved.
PII: S0378-4754(02)00107-6

This paper describes the basic principals behind the Cartesian cut cell method and illustrates its application to problems involving both static and moving boundaries and moving material interfaces.

2. The Cartesian cut cell method

A Cartesian cut cell mesh is generated by "cutting" solid bodies out of a background Cartesian mesh. This results in the formation of fluid, solid and cut grid cells [12]. In order to generate the cut cells the body surfaces are represented using poly-lines, whose knots are defined in an anti-clockwise direction. Thus,

$$P_i = \{(x_0, y_0), \ldots, (x_j, y_j), \ldots, (x_n, y_n)\}$$

defines the ith solid region.

The intersection points of a particular line segment, defined by its start and end coordinates (x_s, y_s) and (x_e, y_e), are found as follows. The address (I_s, J_s) of the cell containing the start coordinate is computed:

$$I_s = \text{int}\left(\frac{x_s - x_0}{\Delta x}\right) + 1 \quad \text{and} \quad J_s = \text{int}\left(\frac{y_s - y_0}{\Delta y}\right) + 1 \tag{1}$$

where x_0 and y_0 are the coordinates of the bottom left corner of the computational domain. The address (I_e, J_e) of the end point is found in a similar way. For convenience, we also identify which of the four quadrants $(0°, 90°]$, $(90°, 180°]$, $(180°, 270°]$ or $(270°, 360°]$ the slope of the line lies in.

The required intersection points can now be found. Fig. 1 shows a line segment cutting a uniform background Cartesian mesh. Suppose that the intersection points of the line segment with cell (i, j) are to be found: clearly the point a at which the line segment enters the cell is already known because it is the exit point from the the previous cut cell. It thus remains necessary only to determine the exit point, b, for the cell (i, j). Since a is on the left side of the cell and $Q \in (0°, 90°]$, the exit point must lie above and to the right of a, locating the exit point on either the top or right hand side of the cell. The intersection points b, between the line segment and the line $y_{j+1} = y_0 + (j + 1)\Delta y$, and c, between the line segment and the line $x_{i+1} = x_0 + (i + 1)\Delta x$, are now found. Since $y_c > y_{j+1}$ the exit point must be (x_b, y_b). This

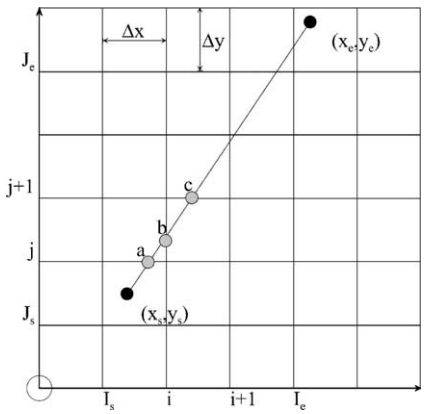

Fig. 1. Finding cut cells: locating the intersection points of a line segment.

process is repeated for all subsequent grid cells intersected by the line segment until the cell (I_e, J_e) is reached. The cases where Q lies in the other quadrants can be treated analogously.

A similar approach can be used in three space dimensions [14,13] but the body surface is defined using a conformal surface triangulation. In this surface, triangulation has only to be of sufficient quality to give an accurate body representation as it is not used directly to discretise the flow solution, representations obtained from CAD systems are thus normally appropriate.

For a finite volume method, the direction of the outward pointing normal must be known for each solid face and, the area of each uncut cell face and the uncut volume of the cell must be determined. In the original implementations, 16 cut cell types [11,12] were used to compute this geometric information. In the three-dimensional solvers vector algebra has been used [13,14]. The fluid area of the cell sides can be calculated based on the intersection points at the cell edges. If the solid face is now approximated by a non-planar quadrilateral, its normal vector and area can be computed using

$$|\boldsymbol{S}| = \sqrt{(S_x^{\mathrm{L}} - S_x^{\mathrm{R}})^2 + (S_y^{\mathrm{L}} - S_y^{\mathrm{R}})^2 + (S_z^{\mathrm{L}} - S_z^{\mathrm{R}})^2} \tag{2}$$

$$\boldsymbol{n} = \frac{1}{|\boldsymbol{S}|} \begin{pmatrix} S_x^{\mathrm{R}} - S_x^{\mathrm{L}} \\ S_y^{\mathrm{R}} - S_y^{\mathrm{L}} \\ S_z^{\mathrm{R}} - S_z^{\mathrm{L}} \end{pmatrix} \tag{3}$$

$$V = \frac{1}{3} \sum_{i=1}^{M} \boldsymbol{P}_i \cdot \boldsymbol{n}_i S_i \tag{4}$$

where S_{xyz}^{LR} is the left or right face of the Cartesian grid cell when viewed from the x, y or z direction, $\boldsymbol{P}_i$, $\boldsymbol{n}_i$ and S_i are the centroid, normal vector and area of the ith face of the cell. In Cieslak et al.'s [14] implementation, an arbitrary number of cuts to one cell is permitted.

Within the Cartesian cut cell method it is necessary to use a finite volume, flow solver which is based on conducting a flux balance around the edges of the cell. This approach allows the unusual number of edges which occur in cut cells to be considered. In the present implementation, the solver is based on the second-order MUSCL-Hancock [22] solver, though this is not the only possible solver. LeVeque and Shyue [19] used a large time-step version of Roe's solver while Tucker and Pan [17] use the CONDIF scheme [23]. In all approaches, however, the vast majority of the grid is treated using the simplest possible implementation of the chosen solver, i.e. the Cartesian form.

2.1. Cell merging

In practice, cut cells may be arbitrarily small and some technique must be employed to overcome time step stability restrictions. In the present work, a cell merging technique [24,25] has been implemented where small cells (i.e. cells with a fluid area $A_{ij} < A_{\min}$) are combined, so that the waves can travel freely in the super-cell without reducing the global time step. The cell to merge with is selected by finding the neighbouring cell in the direction of the normal vector to the solid face. The choice of $A_{\min}$ is based on a trade-off between time step and geometric resolution, in all the present calculations, $A_{\min} = 0.5\,\Delta x\,\Delta y$. An alternative approach is to use a method which is stable for large time steps [19].

Table 1
The Euler and shallow water equations: coefficients of the general hyperbolic system $U_t + F(U)_x + G(U)_y = 0$

Equation set	Conserved quantities ($\boldsymbol{U}$)	Flux ($\boldsymbol{F}$)
Euler	$(\rho,\ \rho u,\ \rho v,\ e)^{\mathrm{T}}$	$(\rho u,\ \rho u^2 + p,\ \rho uv,\ (e+p)u)^{\mathrm{T}}$
SWE	$(\phi,\ \phi u,\ \phi v)^{\mathrm{T}}$	$(\phi u,\ \phi u^2 + 1/2\phi^2,\ \phi uv)^{\mathrm{T}}$

ρ is the density, ϕ the geopotential, and u and v are the velocity components.

3. Equations of motion

The Cartesian cut cell technique described here can be employed to solve any hyperbolic system of conservation laws. In the present work, both the Euler equations, of gas dynamics, and the inviscid non-linear SWE, of hydrodynamics, are employed. These can be written in integral form as:

$$\frac{\partial}{\partial t}\int_{\omega_i} \boldsymbol{U}\,\mathrm{d}\omega_i + \oint_{s_i} \boldsymbol{f}(\boldsymbol{U})\cdot\boldsymbol{n}\,\mathrm{d}s = 0 \tag{5}$$

where $\boldsymbol{U}$ is the vector of conserved quantities, $\boldsymbol{F}$ is the flux tensor (see Table 1), s_i is the boundary of an arbritrary control volume, ω_i and $\boldsymbol{n}$ are the outward pointing normal. The solution space, Ω, is partitioned so that $\Omega = \{\omega_1, \ldots, \omega_n\}$. The semi-discrete form of (5),

$$\frac{\partial}{\partial t}\int_{\omega_i} \boldsymbol{U}\,\mathrm{d}\omega_i + \sum_{k=1}^{m} \boldsymbol{f}_k(\boldsymbol{U})\cdot\boldsymbol{n}_k = 0 \tag{6}$$

is obtained by using an m sided region for ω_i and a finite volume solver can now be obtained by choosing an appropriate time discretisation.

This approach has several advantages. First, all the conserved quantities are stored at the centre of the control volume. Second, the geometric terms are represented via the normal vectors, $\boldsymbol{n}_k$, and the cell area, thus eliminating the need for a transformation Jacobian. Finally, the grid is boundary conforming.

4. The numerical integration scheme

The MUSCL-Hancock finite volume scheme [22] with appropriate modifications for Cartesian cut cell meshes is applied which is a two-step, high resolution, upwind scheme of the Godunov type. The non-conservative predictor defines an intermediate cell-centre value at $\Delta t/2$:

$$(A\boldsymbol{U})_{ij}^{n+(1/2)} = (A\boldsymbol{U})_{ij}^{n} - \frac{\Delta t}{2}\sum_{k=1}^{m} \boldsymbol{H}(\boldsymbol{U}_k)\cdot\boldsymbol{S}_k^n \tag{7}$$

where A is the cell area, $\boldsymbol{S}_k^n$ the cell side vector and m is the maximum number of cell sides. The flux vector function $\boldsymbol{H}(\boldsymbol{U}_k)$ is evaluated at each cell side k following a linear reconstruction of the flow solution within each cell, via, $\boldsymbol{U}_k = \boldsymbol{U}_{ij}^n + \boldsymbol{r}_k\cdot\nabla\boldsymbol{U}_{ij}^n$, where $\boldsymbol{r}_k$ is the normal distance vector from the cell centroid to side k and $\nabla\boldsymbol{U}_{ij}^n$ is a slope limited gradient vector.

The corrector step is fully conservative and uses upwind cell interface fluxes based on the solution at the half time level:

$$(A\boldsymbol{U})_{ij}^{n+1} = (A\boldsymbol{U})_{ij}^{n} - \Delta t \sum_{k=1}^{m} \boldsymbol{H}(\boldsymbol{U}_k^{\mathrm{L}}, \boldsymbol{U}_k^{\mathrm{R}}) \cdot \boldsymbol{S}_k^n \tag{8}$$

The upwind flux, $\boldsymbol{H}(\boldsymbol{U}_k^{\mathrm{L}}, \boldsymbol{U}_k^{\mathrm{R}})$, is obtained by solving a local Riemann problem normal to cell interface k. To solve the Riemann problem, the approximate Riemann solver of Harten, Lax and van Leer (HLL) [26] is used. Other Riemann solvers (either approximate or exact) can be employed but the HLL solver has proved to be robust and sufficiently accurate in practice.

The time step employed is:

$$\Delta t = \nu \min(\Delta t_x, \Delta t_y), \quad \text{where } \Delta t_x = \min_i \frac{A_{ij}}{|\boldsymbol{q}_{ij} \cdot \boldsymbol{S}_{i+(1/2)j}| + a_{ij}|\boldsymbol{S}_{i+(1/2)j}|} \tag{9}$$

where $a_{ij} = \sqrt{\phi_{ij}}$ for the SWE and $a_{ij} = \sqrt{\gamma p_{ij}/\rho_{ij}}$ for the Euler equations. An analogous definition is used for Δt_y. The Courant number ν was taken in our calculations to be 0.45 which is close to the stability bound.

The HLL Riemann solver provides the a Riemann solution at each cell interface based on the assumptions that: (i) the star region between the waves is simplified to a single intermediate state and (ii) the only waves present are the left and right acoustic waves which depend on a priori wave speed estimates s_L and s_R. Suitable estimates for the wave speeds are available for both the Euler [27] and SWEs [15,28].

4.1. Calculation of gradients and reconstruction technique

In order to achieve second-order spatial accuracy, reconstruction of the flow data within each cell is necessary. The gradient calculation is straightforward on flow cells away from a solid boundary employing a central difference gradient of conserved variables and a slope limiter function to prevent spurious oscillations. For cells near a solid boundary, however, a different gradient calculation is needed. In [7], a linear reconstruction technique has been proposed for different types of cut cell, based on cells in the neighbourhood of the cut cell interface and a suitable path integral. However, since we wish to accommodate moving boundaries, a gradient calculation is appropriate.

Using reflection boundary conditions at solid a boundary, the variables in the ghost cell R (see Fig. 2) can be obtained by

$$\phi_R = \phi_{ij}, \qquad \boldsymbol{v}_R = \boldsymbol{v}_{ij} - 2(\boldsymbol{v}_{ij} \cdot \boldsymbol{n})\boldsymbol{n} \tag{10}$$

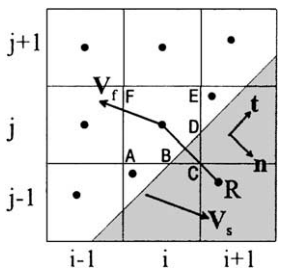

Fig. 2. Boundary treatment applied to a cut cell.

The gradients on cut cell (i, j) may be of two types: fluid and solid. We calculate the fluid gradients, $\boldsymbol{U}_x^{\mathrm{f}}$ and $\boldsymbol{U}_y^{\mathrm{f}}$, and the solid gradients, $\boldsymbol{U}_x^{\mathrm{s}}$ and $\boldsymbol{U}_y^{\mathrm{s}}$ separately, as follows:

$$\boldsymbol{U}_x^{\mathrm{f}} = G\left(\frac{\boldsymbol{U}_{i+1,j} - \boldsymbol{U}_{i,j}}{\Delta x_{i+(1/2),j}}, \frac{\boldsymbol{U}_{i,j} - \boldsymbol{U}_{i-1,j}}{\Delta x_{i-(1/2),j}}\right) \tag{11}$$

$$\boldsymbol{U}_x^{\mathrm{s}} = G\left(\frac{\boldsymbol{U}_R - \boldsymbol{U}_{i,j}}{\Delta x_{i,R}}, \frac{\boldsymbol{U}_{i,j} - \boldsymbol{U}_{i-1,j}}{\Delta x_{i-(1/2),j}}\right) \tag{12}$$

with analogous definitions for $\boldsymbol{U}_y^{\mathrm{f}}$ and $\boldsymbol{U}_y^{\mathrm{s}}$. Here, $\Delta x_{i,R} = x_R - x_{i,j}$, $\Delta y_{j,R} = y_{i,j} - y_R$ and G is a slope limiter function, e.g. the van Leer limiter

$$G(a, b) = \frac{a|b| + |a|b}{|a| + |b|} \tag{13}$$

Once the two types of gradients are calculated, a length average technique is used to obtain unique gradients in the cut cell,

$$\boldsymbol{U}_x = \frac{\Delta y_{\mathrm{s}} \boldsymbol{U}_x^{\mathrm{s}} + \Delta y_{\mathrm{f}} \boldsymbol{U}_x^{\mathrm{f}}}{\Delta y}, \qquad \boldsymbol{U}_y = \frac{\Delta x_{\mathrm{s}} \boldsymbol{U}_y^{\mathrm{s}} + \Delta x_{\mathrm{f}} \boldsymbol{U}_y^{\mathrm{f}}}{\Delta x} \tag{14}$$

where $\Delta x_{\mathrm{f}} = |AB|$, $\Delta x_{\mathrm{s}} = |BC|$, $\Delta y_{\mathrm{s}} = |CD|$ and $\Delta y_{\mathrm{f}} = |DE|$; Δx and Δy are the uncut cell side lengths in the x and y directions, respectively. Given the gradient vector $\nabla \boldsymbol{U}_{ij} = iU_x + jU_y$, a reconstructed solution vector $\boldsymbol{U}(x, y)$ can be found anywhere within the cut cell using $\boldsymbol{U}(x, y) = U_{ij} + \boldsymbol{r} \cdot \nabla \boldsymbol{U}_{ij}$, where $\boldsymbol{r}$ is the normal distance vector from the cell centroid to any specific interface or solid boundary.

4.1.1. The Riemann solution at a moving boundary

The standard Riemann solver employed at cell interfaces must be modified to deal with moving boundaries. Consider a cut cell with a face coincident with a moving boundary, as shown in Fig. 2, where $\boldsymbol{v}_{\mathrm{f}}$ is the reconstructed flow velocity at the solid boundary and $\boldsymbol{v}_{\mathrm{s}}$ is the solid boundary velocity. Projecting both $\boldsymbol{v}_{\mathrm{f}}$ and $\boldsymbol{v}_{\mathrm{s}}$ in directions normal to and tangential to the solid boundary with unit normal and unit tangential vectors $\boldsymbol{n}, \boldsymbol{t}$, we have $u_{\mathrm{f}n} = \boldsymbol{v}_{\mathrm{f}} \cdot \boldsymbol{n}$, $u_{\mathrm{f}t} = \boldsymbol{v}_{\mathrm{f}} \cdot \boldsymbol{t}$, and $u_{\mathrm{s}n} = \boldsymbol{v}_{\mathrm{s}} \cdot \boldsymbol{n}$, $u_{\mathrm{s}t} = \boldsymbol{v}_{\mathrm{s}} \cdot \boldsymbol{t}$. In the tangential direction, any difference between $u_{\mathrm{f}t}$ and $u_{\mathrm{s}t}$ is treated as a shear wave superimposed on a contact surface. In the normal direction, the Riemann solution for a moving piston is incorporated; hence, the solution for $u_n^* = u_{\mathrm{s}n}$ is trivial. The solution for P_n^* (Euler) or ϕ_n^* (SWE) can be obtained by comparing the two velocities $u_{\mathrm{f}n}$ and $u_{\mathrm{s}n}$. If $u_{\mathrm{f}n} < u_{\mathrm{s}n}$, both left and right moving waves are rarefactions and ϕ_n^*, for example, can be obtained from the rarefaction relations [29],

$$\phi_n^* = \tfrac{1}{4}\left(u_{\mathrm{f}n} - u_n^* + 2\sqrt{\phi_{\mathrm{f}}}\right)^2 \tag{15}$$

If, however, $u_{\mathrm{f}n} \geq u_{\mathrm{s}n}$, both left and right moving waves are shocks and ϕ_n^* must be obtained from the Rankine–Hugoniot conditions (written here for the SWE),

$$(\phi_n^* - \phi_{\mathrm{f}})\sqrt{\frac{\phi_n^* + \phi_{\mathrm{f}}}{2\phi_n^*\phi_{\mathrm{f}}}} + u_n^* - u_{\mathrm{f}n} = 0 \tag{16}$$

There is no closed form solution for (16), but ϕ_n^* can be obtained to the required accuracy by iteration. However, an approximate solution to (16) can be obtained [30].

5. Moving material interfaces

The algorithm as described for moving boundaries can also be applied to problems involving heterogeneous mixtures. It is well established [31] that conservative methods produce erroneous pressure perturbations around the material interface, which contaminate the other flow variables. The use of a cut cell representation of the interface avoids this problem. The interface is represented in exactly the same way as if it were a solid boundary but with two important differences: (1) the flow solution is found on both sides of the interface, and (2) the movement of the interfaces depends on local conditions in the flow field. Udaykumar et al. [20] used a mixed Eulerian–Lagrangian method to locate the interface, while LeVeque and Shyue [19] used the volume of fluid (VoF) method.

Ivings [21], however, used an interface tracking approach. At the end of each time step, the locations of the mid and end points of each line segment move a distance $u^* \Delta t$ normal to the segment and the interface is reconstructed using quadratic Bezier curves, which provide a C1 continuous representation of the interface (Fig. 3). In this case, u^* must be determined by solving the Riemann problem for two heterogeneous fluids [32]. As the interface moves, different merging configurations will be used. This means that the solution values and volumes may change for every cell. Consider the example in Fig. 4,

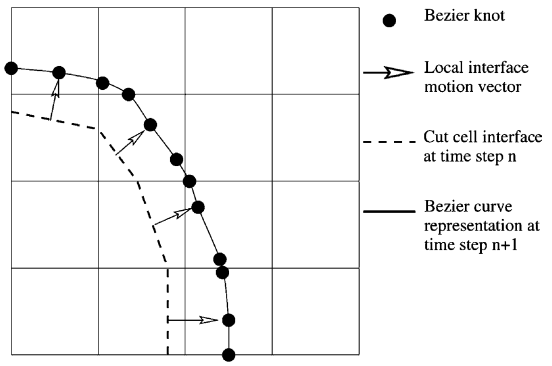

Fig. 3. Bezier curve reconstruction of a moving material interface.

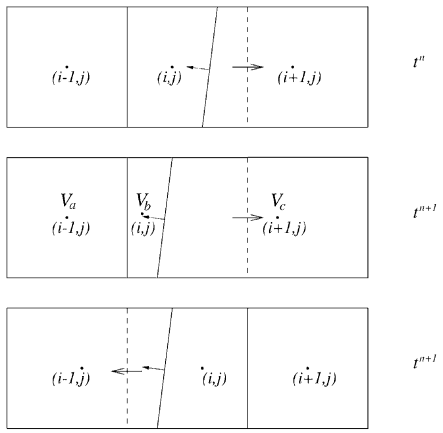

Fig. 4. Cell merging around a moving material interface.

where the interface is moving from right to left. The following solution values are then assigned to these three cells:

$$\boldsymbol{U}_{i-1}^{n+1} = \frac{V_a \boldsymbol{U}_{i-1}^{n+1} + V_b \boldsymbol{U}_i^{n+1}}{V_a + V_b}, \qquad \boldsymbol{U}_i^{n+1} = \boldsymbol{U}_{i+1}^{n+1}, \qquad \boldsymbol{U}_{i+1}^{n+1} = \boldsymbol{U}_{i+1}^{n+1} \tag{17}$$

6. Results

The Cartesian cut cell method has been successfully applied to a number of quite complex problems requiring the solution of either the Euler or SWEs in two and three dimensions [12,13,15,21,30] and has been shown to be accurate, efficient and robust. In this paper, three examples are presented to demonstrate the application of the method to hydrodynamic flows, aeronautical flows and a problem involving a moving material interface.

6.1. Bore wave diffraction in a contraction–expansion channel

An incident bore wave at Froude number 3 travelling through a symmetric contraction–expansion channel has been simulated using the SWEs. The channel is symmetrical about the centre line and the describing equations for the left hand (or upper) bank are:

$$y = \begin{cases} 1.0, & -2 \le x \le -1 \\ -0.375\cos(\pi x) + 0.625, & -1 < x \le 0 \\ -0.625\cos(\pi/2x) + 0.875, & 0 < x \le 2 \\ 1.5, & 2 < x \le 4 \end{cases} \tag{18}$$

The incident bore wave was positioned initially at the entrance to the curved portion of the channel ($x = -1$) and a Cartesian mesh with 360×200 cells was used on the computational domain. Fig. 5 shows a sequence of height contours of bore diffraction in the channel. Initially, the incident bore is reflected at the walls of the contraction section of the channel where regular reflection occurs. As the incident bore reaches the throat section of the channel, transition to Mach reflection occurs. Mach stems are formed in the throat region which subsequently diffract to form a secondary bore wave in the expansion section. Meanwhile, the reflected bore waves are reflected repeatedly at the channel walls. At a later time, the leading bore wave continues to evolve with the secondary bore formed by the interaction of the two Mach discs following behind while the reflected bore waves develop into a planar bore wave travelling upstream. The computed results compare well with the calculations performed by Yang and Hsu [33] and Mingham et al. [34] using curviliner grids.

6.2. Axisymmetric muzzle brake

Widhopf et al. [35] presented both experimental and numerical results for a projectile emerging from a tube fitted with a muzzle break. This demanding test problem requires the projectile to move freely across the computational domain and is an ideal candidate for the present methods. The calculations were performed using the Euler equations on a 240×96 Cartesian grid on a 360 mm $\times$ 144 mm domain. The projectile is driven by the high pressure gas in the tube which can be seen expanding into the atmosphere

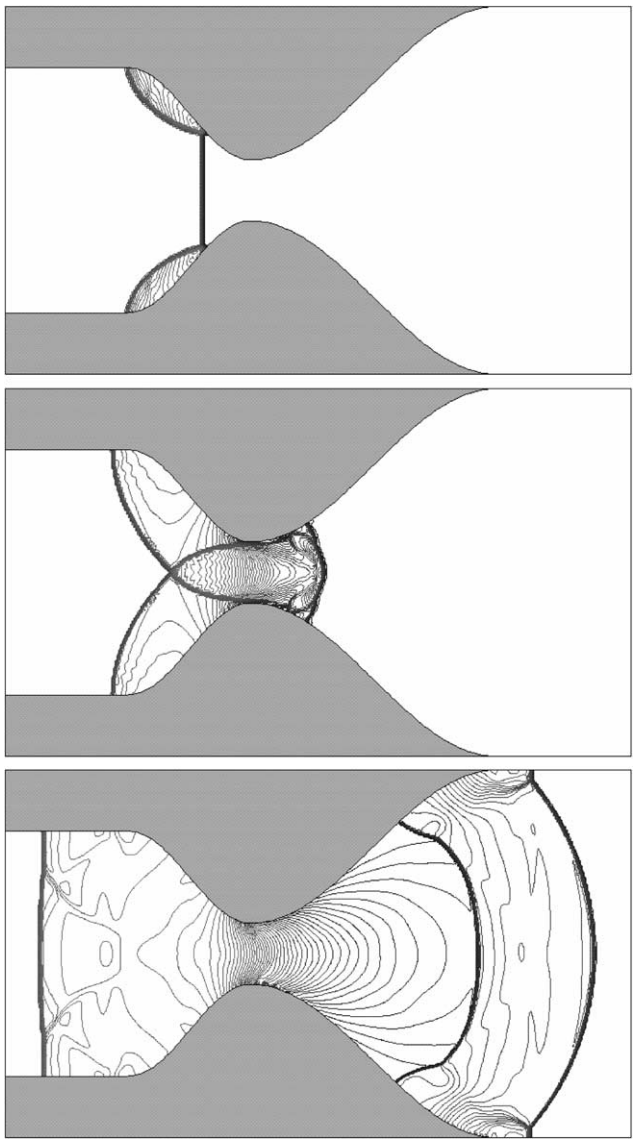

Fig. 5. Bore propagation in a contraction–expansion chanel: height contours at $t = 0.067,\ 0.15$ and 0.4 s.

in Fig. 6. Fig. 7 shows a comparison between the measured peak and steady state overpressures on the first baffle which are in good agreement with the experiment.

6.3. Shock wave interaction with a bubble

The interaction of a shock wave, $M_s = 1.22$, with a cylindrical bubble of R22 gas in air was described experimentally [36]. As the R22 gas has a higher density than the air, and therefore a lower speed of

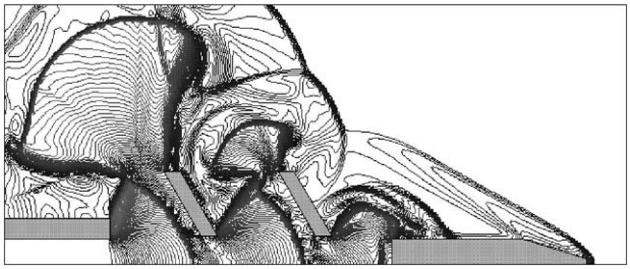

Fig. 6. Axisymmetric muzzle break: iso-Mach lines at $t = 0.37$ ms.

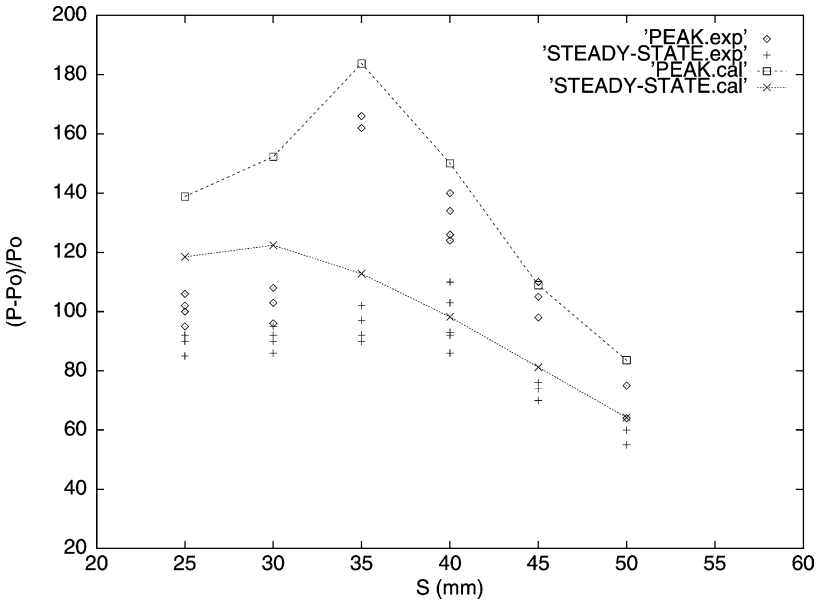

Fig. 7. Axisymmetric muzzle break: peak/steady-state overpressures on the first baffle.

sound, the bubble acts as a convergent lens on interaction with the shock. The initial states of the gas for this experiment are given in Table 2, where the shock is initiated using standard shock relations. The radius of the bubble is 25 mm and the width of the shock tube is 89 mm. The centre of the bubble is located 88 mm from the left hand end of the tube and the length of the tube is 178 mm. Fig. 8 shows computational

Table 2
Initial conditions for Haas and Sturtevant's shock interaction with an R22 bubble problem

	ρ	u	v	p (bar)	γ
Pre-shock	1.376363	124.824	0	1.56908	1.4
Ambient	1	0	0	1	1.4
R22 bubble	3.15385	0	0	1	1.249

Fig. 8. Shock interaction with an R22 bubble in air: computer generated schlieren images, $t = 115$ and 247 μs.

schlieren images from the simulation, these show that the bubble interface is captured sharply and that the results are in good agreement with the experimental results.

7. Conclusions

The Cartesian cut cell method provides and efficient and flexible alternative to traditional body fitted grid techniques. Once provided with a suitable description of the body surface the grid generation step is replaced by the computation of the locations at which the body surface cuts the Cartesian background grid. The approach requires the use of a finite-volume solver to provide the boundary conforming treatment of the cut cells. In this paper, the MUSCL-Hancock scheme has been used together with the HLL Riemann solver and applied to simulate flows involving the shallow water and Euler equations and to a heterogeneous multi-fluid problem. The results obtained give confidence in the Cartesian cut cell approach.

Acknowledgements

The authors wish to thank Dr. Matthew Ivings for his simulations of the Haas and Sturtevant test problem.

References

[1] J. Thompson, Z. Warsi, C. Wayne-Mastin, Numerical Grid Generation: Foundation and Applications, Elsevier, Amsterdam, 1985.
[2] N. Weatherill, C. Forsey, Grid Generation and Flow Calculations for Complex Aircraft Geometries using a Multi-Block Scheme, AIAA paper 84-1665, 1984.
[3] J. Steger, F. Dougherty, J. Benek, A chimera grid scheme, in: Advances in Grid Generation, ASME FED-5, 1983, pp. 59–69.
[4] C.M. Albone, Embedded Meshes of Controllable Quality Synthesised from Elementary Geometric Features, AIAA paper 92-0662, 1992.
[5] J. Rantakokko, Partitioning strategies for structured multiblock grids, Parallel Comput. 26 (12) (2000) 1661–1680.
[6] J. Peraire, M. Vahdati, K. Morgan, O.C. Zienkiewicz, Adaptive remeshing for compressible flow computations, J. Comput. Phys. 72 (1987) 449–466.
[7] D. De Zeeuw, K. Powell, An adaptively refined Cartesian mesh solver for the Euler equations, J. Comput. Phys. 104 (1) (1993) 56–68.
[8] R. Lohner, A parallel advancing front grid generation scheme, Int. J. Numer. Methods Eng. 51 (6) (2001) 663–678.

[9] M. Delanaye, A. Patel, B. Leonard, C. Hirsch, Automatic unstructured hexahedral grid generation and flow solutions, in: ECCOMAS CFD 2001, Institution of Mathematics and its Application, Southend-on-Sea, UK, 2001.
[10] M. Berger, R. LeVeque, An Adaptive Cartesian Mesh Algorithm for the Euler Equations in Arbitrary Geometries, AIAA paper 89-1930-CP, 1989.
[11] J. Quirk, An alternative to unstructured grids for computing gas dynamic flows around arbitrarily complex two-dimensional bodies, Comput. Fluids 23 (1) (1994) 125–142.
[12] G. Yang, D. Causon, D. Ingram, R. Saunders, P. Batten, A Cartesian cut cell method for compressible flows. Part A. Static body problems, Aeronaut. J. 101 (1001) (1997) 47–56.
[13] G. Yang, D. Causon, D. Ingram, Calculation of compressible flows about complex moving geometries using a 3D Cartesian cut cell method, Int. J. Numer. Methods Fluids 33 (2000) 1121–1151.
[14] S. Cieslak, S. Ben Khelil, I. Choqiet, A. Merlen, Cut cell stategy for 3D blast wave numerical simulation, Shock Waves 10 (6) (2001) 421–429.
[15] D. Causon, D. Ingram, C. Mingham, G. Yang, R. Pearson, Calculation of shallow water flows using a Cartesian cut cell approach, Adv. Water Resour. 23 (2000) 545–562.
[16] L. Qian, D. Causon, D. Ingram, C. Mingham, A Cartesian cut cell method for incompressible viscous flows, in: ECCOMAS CFD 2001, Institute of Mathematics and its Applications, Southend-on-Sea, UK, 2001.
[17] P. Tucker, Z. Pan, A Cartesian cut cell method for incompressible viscous flow, Appl. Math. Model. 24 (2000) 591–606.
[18] T. Ye, R. Mittal, H. Udaykumar, W. Shyy, An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries, J. Comput. Phys. 156 (1999) 209–240.
[19] R. LeVeque, K. Shyue, Two dimensional front tracking based on high resolution wave propagation methods, J. Comput. Phys. 123 (1996) 354–368.
[20] H. Udaykumar, H. Kan, W. Shyy, R. Tran-Son-Tay, Multiphase dynamics in arbritary geometries on fixed Cartesian grids, J. Comput. Phys. 137 (1997) 366–405.
[21] M. Ivings, Wave propagation through gases and liquids, Ph.D. thesis, Department of Computing and Mathematuics, Manchester Metropolitan University, 1997.
[22] B. van Leer, On the relation between the upwind-differencing schemes of Godunov, Engquist-Osher and Roe, SIAM J. Sci. Stat. Comput. 5 (1) (1984) 1–20.
[23] A. Runchal, Condif: a modified central-difference scheme for convective flows, Int. J. Numer. Methods Eng. 24 (1987) 1593–1608.
[24] D. Clarke, M. Salas, H. Hassan, Euler calculations for multielement airfoils using Cartesian grids, AIAA J. 24 (3) (1986) 353–358.
[25] Y. Chiang, B. van Leer, K.G. Powell, Simulation of Unsteady Inviscid Flow on an Adaptively Refined Cartesian Grid, AIAA paper 92-0443, 1992.
[26] A. Harten, P. Lax, B. van Leer, On upstream differencing and Godunov-type schemes for hyperbolic conservation laws, SIAM Rev. 25 (1) (1983) 35–61.
[27] S. Davis, Simplified second-order Godunov-type methods, SIAM J. Sci. Stat. Comput. 9 (3) (1988) 445–473.
[28] L. Fraccarollo, E. Toro, Experimental and numerical assessment of the shallow water model for two dimensional dam-break type problems, J. Hydraulic Res. 33 (6) (1995) 843–864.
[29] J. Stoker, Water Waves: The Mathematical Theory with Applications, Wiley Classics Library Edition, Wiley, New York, 1992.
[30] D.M. Causon, D.M. Ingram, C.G. Mingham, A Cartesian cut cell method for shallow water equations with moving boundaries, Adv. Water Resour. 24 (2001) 889–911.
[31] S. Karni, Viscous shock profiles and primitive formulations, SIAM J. Numer. Anal. 29 (1992) 1592–1609.
[32] M. Ivings, D. Causon, E. Toro, On Riemann solvers for compressible liquids, Int. J. Numer. Methods Fluids 28 (1998) 395–418.
[33] J. Yang, C. Hsu, Computation of free surface flows, J. Hydraulic Res. 31 (3) (1993) 403–413.
[34] C. Mingham, D. Causon, D. Ingram, A TVD MacCormack scheme for transcritical flow, Water Mar. Eng. 148 (3) (2001) 167–175.
[35] G. Widhopf, J. Buell, E. Schmidt, Time-Dependent Near Field Muzzle Brake Flow Simulations, AIAA paper 82-0973, 1982.
[36] J.-F. Haas, B. Sturtevant, Interaction of weak shock waves with cylindrical and spherical gas inhomogeneities, J. Fluid Mech. 181 (1987) 41–76.