V European Conference on Computational Fluid Dynamics
ECCOMAS CFD 2010
J. C. F. Pereira and A. Sequeira (Eds)
Lisbon, Portugal,14-17 June 2010

A NUMERICAL METHOD FOR MOVING-BOUNDARY PROBLEMS OF COMPRESSIBLE VISCOUS FLOW

Daniel Hartmann, Lennart Schneiders, Matthias Meinke, and Wolfgang Schröder

Institute of Aerodynamics, RWTH Aachen University
Wüllnerstr. 5a, 52062 Aachen, Germany
e-mail: office@aia.rwth-aachen.de

Key words: Moving-boundary problems, Level-set method, Cartesian cut-cell method, Compressible Flows

Abstract. *A strictly conservative Cartesian cut-cell method for two- and three-dimensional compressible viscous flows involving moving boundaries is presented. The method allows local quadtree/octree mesh refinement in the framework of a hierarchical cell-tree data structure. One of the challenges of finite-volume based immersed boundary methods such as the cut-cell method are cells of very small volume which are inevitably generated at the embedded boundary. These small cells typically result in numerical instability and must be removed. To remedy this problem, a new conservative multi-level treatment of small cells is proposed. The moving boundary is represented by the zero contour of a level-set function, which is initialized into a signed distance function, from which the cut-cell information can be efficiently generated. The motion of the embedded body is captured by solving the level-set equation governing the evolution of the level-set function. The developed numerical method is validated for two- and three-dimensional flows past fixed and moving bluff bodies such as the canonical problem of a transversely oscillating circular cylinder in freestream.*

1 INTRODUCTION

Fluid flow problems involving moving and deforming boundaries are of particular interest in a wide range of applications ranging from engineering problems such as the flow inside internal combustion engines and multiphase flows occuring in spray combustion to biomechanics problems such as the swimming of fish and bird flight. In the latter cases, the fluid flow and the motion of the boundary, i.e., the surface of the moving animal immersed in the fluid, are typically strongly coupled, increasing the complexity of the problem.

The numerical simulation of such fluid flows is particularly challenging. In recent years, numerical methods based on non-boundary-fitted grids to solve the Euler and Navier-Stokes equations in complex and possibly moving geometries have gained popularity.[1–5] One of the challenges of these methods is the representation of the embedded boundaries which do not coincide with grid lines. A decomposition of the existing methods for non-boundary-fitted grids into two categories is given by Mittal and Iaccarino.[6]

In general immersed boundary methods for compressible viscous flows are rather rare.[7,8] The method proposed by de Tullio et al.[8] uses a finite-volume method away from the embedded boundaries to allow for adaptively refined meshes. However, on the cells which are intersected by the embedded boundaries a direct forcing approach is adopted lacking discrete conservation. Similarly, the method of Ghias et al.[7] utilizes a finite-difference scheme making it not strictly conservative. Recently, a strictly conservative cut-cell based Cartesian grid method for the solution of the compressible Navier-Stokes equations on locally refined grids in two- and three-dimensional space was developed by Hartmann et al.[9,10] The methodology allows dynamic solution-adaptive mesh refinement as demonstrated in an earlier work of Hartmann et al.[11]

The objective of this article is to extend the developed cut-cell Cartesian grid method to general two- and three-dimensional moving-boundary problems of compressible viscous flow. The moving boundaries are represented using the level-set method,[12–15] based on which the cut-cells can be quickly regenerated after each time step. Such a representation of the moving boundary has been recently employed by several authors.[5,16,17] In particular for moving-boundary problems a major challenge of cut-cell methods are cells of very small volume, which emerge at the embedded boundaries and lead to numerical instabilities. A novel small-cell treatment is developed to avoid the spurious oscillations resulting from the previously employed cell-merging/cell-linking technique.[10] These oscillations occur due to the frequent merging and breaking of cell clusters near the moving boundary. Another challenge of three-dimensional implementations of cut-cell based Cartesian grid methods is the complexity to formulate a finite-volume discretization on cells which are arbitrarily intersected by the embedded moving boundaries of the computational domain. Following Hartmann et al.,[9,10] we use the least-squares method to compute the cell-center derivatives such that the proposed method is uniformly formulated and simple to implement. The application of these boundary conditions is simplified by introducing ghost cells which

can be arbitrarily positioned in space such that the proposed method is flexible in terms of shape and size of embedded moving boundaries.

The organization of this paper is as follows. In Section 2 the governing equations are given, before the developed numerical method is presented in Section 3. Finally, the results are presented in Section 4 and in Section 5 the findings of this paper are summarized.

2 MATHEMATICAL MODEL

We study the interaction of a viscous flow with a rigid solid body. In general, forces exerted by the fluid result in an acceleration and deformation of the body. The latter effect can be assumed to be negligibly small for the present investigation such that the problem can be fully described by the non-dimensional Navier-Stokes equations for compressible flows

$$\frac{\partial \mathbf{Q}}{\partial t} + \boldsymbol{\nabla}\overline{\mathbf{H}} = \mathbf{0}, \tag{1}$$

where $\mathbf{Q} = [\varrho, \varrho\mathbf{v}, \varrho E]^T$ denotes the vector of the conservative variables and $\overline{\mathbf{H}}$ is the flux vector containing an inviscid part $\overline{\mathbf{H}}^i$ and a viscous part $\overline{\mathbf{H}}^v$. Using the Reynolds number Re, the flux vector can be written

$$\overline{\mathbf{H}} = \overline{\mathbf{H}}^i - \overline{\mathbf{H}}^v = \begin{pmatrix} \varrho\mathbf{v} \\ \varrho\mathbf{v}\mathbf{v} + p \\ \mathbf{v}(\varrho E + p) \end{pmatrix} + \frac{1}{Re}\begin{pmatrix} 0 \\ \overline{\boldsymbol{\tau}} \\ \overline{\boldsymbol{\tau}}\mathbf{v} + \mathbf{q} \end{pmatrix}. \tag{2}$$

We assume a Newtonian fluid and zero bulk viscosity such that the second-rank stress tensor $\overline{\boldsymbol{\tau}}$ can be formulated

$$\overline{\boldsymbol{\tau}} = -2\mu\overline{\mathbf{S}} + \frac{2}{3}\mu(\boldsymbol{\nabla}\cdot\mathbf{v})\bar{\mathbf{I}}, \tag{3}$$

where $\bar{\mathbf{I}}$ is the unit tensor and $\overline{\mathbf{S}}$ is the rate-of-strain tensor

$$\overline{\mathbf{S}} = \frac{1}{2}\left(\boldsymbol{\nabla}\mathbf{v} + (\boldsymbol{\nabla}\mathbf{v})^T\right). \tag{4}$$

The vector of heat conduction $\mathbf{q}$ is expressed by Fourier's law

$$\mathbf{q} = -\frac{k}{Pr(\gamma-1)}\boldsymbol{\nabla}T, \tag{5}$$

where γ is the ratio of specific heats. The Prandtl number $Pr = \frac{\mu_\infty c_p}{k_\infty}$ contains the specific heat at constant pressure c_p. Finally, the thermal conductivity is evaluated from $k(T) = \mu(T)$, which holds for a constant Prandtl number.

3 NUMERICAL METHOD

3.1 Discretization of the Navier-Stokes equations

The Navier-Stokes equations (1) are integrated in time using a 5-stage second-order accurate Runge-Kutta scheme optimized for stability,

$$\begin{cases} \mathbf{Q}^{(0)} = \mathbf{Q}^{w}, \\ \mathbf{Q}^{(k)} = \mathbf{Q}^{(0)} - \alpha_k \Delta t L(\mathbf{Q}^{(k-1)}), \\ \mathbf{Q}^{w+1} = \mathbf{Q}^{(N)}, \end{cases} \tag{6}$$

where $N = 5$ and the coefficients $\boldsymbol{\alpha} = (\frac{1}{4}, \frac{1}{6}, \frac{3}{8}, \frac{1}{2}, 1)$ are used. The superscript k denotes the Runge-Kutta stage, while the superscript w represents the time step Δt computed globally by

$$\Delta t = \min_{n \in \mathcal{D}} \left\{ \min_{j \in \{1,2,3\}} \left(\frac{Ch^n}{|v_j^n| + a^n} \right) \right\}, \tag{7}$$

where $a^n = \sqrt{\gamma \frac{p^n}{\varrho^n}}$ is the local speed of sound and C is the CFL number. Finally, the operator $L(\mathbf{Q})$ denotes the numerical approximation of the term $\boldsymbol{\nabla}\overline{\mathbf{H}}$ in Eq. (1).

The inviscid surface flux $\overline{\mathbf{H}}^i(\mathbf{x}_s^m)$ is computed based on the left and right interpolated states L and R in the surface centroid, which are computed at second-order accuracy using a monotone upstream centered schemes for conservation laws (MUSCL) approach for the primitive variables,[18] which for an unlimited reconstruction reads

$$\mathbf{P}(\mathbf{x}_s^m)^{L/R} = \mathbf{P}(\mathbf{x}^{L/R}) + \boldsymbol{\nabla}\mathbf{P}|^{L/R} \left(\mathbf{x}_s^m - \mathbf{x}^{L/R} \right). \tag{8}$$

This flux is formulated using a modified version[10] of the advection upstream splitting method (AUSM) originally proposed by Liou and Steffen.[19] In this method, the inviscid flux is split into a convective component and a pressure term. The cell-center gradient $\boldsymbol{\nabla}\mathbf{P}|^n$, $n = \{L, R\}$, is computed using an unweighted linear least-squares reconstruction scheme which can be written

$$\overline{\mathcal{H}}^n \, \boldsymbol{\nabla}\mathbf{P}|^n = \Delta\mathbf{P}^n, \tag{9}$$

where $\overline{\mathcal{H}}^n$ is a matrix of dimension $s \times d$ and $\Delta\mathbf{P}^n$ is a vector of dimension s,

$$\begin{cases} \overline{\mathcal{H}}^n = \left[\Delta\mathbf{x}^{n,\mathcal{S}_1^n}, \ldots, \Delta\mathbf{x}^{n,\mathcal{S}_s^n} \right]^T, \\ \Delta\mathbf{P}^n = \left[\Delta\mathbf{P}^{n,\mathcal{S}_1^n}, \ldots, \Delta\mathbf{P}^{n,\mathcal{S}_s^n} \right]^T, \end{cases}$$

with $\Delta\phi^{a,b} = \phi^b - \phi^a$, $\phi \in \{\Delta\mathbf{x}, \Delta\mathbf{P}\}$. The quantity $\mathcal{S}^n$ denotes the least-squares stencil of s cells which are clustered around the cell n and $\mathcal{S}_k^n$ is the kth element of $\mathcal{S}^n$. For details, we refer to Hartmann et al.[10]

The least-squares system of equations (9) is rearranged to obtain

$$\boldsymbol{\nabla}\mathbf{P}|^n = \overline{\mathbf{C}}^n \Delta\mathbf{P}^n, \tag{10}$$

where the matrix $\overline{\mathbf{C}}^n = \left(\left(\overline{\mathcal{H}}^n\right)^T \overline{\mathcal{H}}^n\right)^{-1} \left(\overline{\mathcal{H}}^n\right)^T$ has dimension $d \times s$ and contains geometric constants, which can be computed and stored in a preprocessing stage. In structured grid areas the least-squares formulation reduces to standard finite-difference operators. As discussed in Hartmann et al.,[10] the additional constraint which is usually imposed on the least-squares system to ensure the reconstructed linear in-cell distributions conserve the cell averages represented by the cell center values $\mathbf{P}^n$ is automatically fulfilled by this reconstruction.

Finally, the viscous flux $\overline{\mathbf{H}}^v$ is computed using the cell-center gradients available from computing $\overline{\mathbf{H}}^i$.[10]

3.2 Representation of the moving boundary

The moving boundary is described using a level-set representation. A highly accurate level-set method has been developed in a series of papers[13–15,20] and coupled to the hierarchical cell tree data structure of the flow solver in the context of a dual-mesh approach for premixed combustion,[21,22] in which the level-set method is used to describe the motion of the flame front. Let the moving boundary Γ, which is generally a hypersurface in the considered n-dimensional space, be represented by the contour

$$\phi_0 = \{(\mathbf{x}, t) : \phi(\mathbf{x}, t) = 0\},\ \mathbf{x} \in R^n,\ t \in R^+, \tag{11}$$

where $\phi(\mathbf{x}, t)$ is the scalar level-set function and t denotes the time. For $n = 3$, let the components of the coordinate vector be denoted by $\mathbf{x} = (x, y, z)^T$. As illustrated in Fig. 1, the level-set function ϕ is specified as a signed distance function with respect to ϕ_0 with the properties

$$\begin{cases} \phi > 0 & \text{for } \mathbf{x} \in \Omega^f, \\ \phi = 0 & \text{for } \mathbf{x} \in \phi_0 \equiv \Gamma, \\ \phi < 0 & \text{for } \mathbf{x} \in \Omega^s, \end{cases} \tag{12}$$

where the computational domain Ω has been decomposed into a fluid region Ω^f and a solid region Ω^s, $\Omega = \{\Omega^f, \Omega^s, \phi_0\}$, with $\Omega^f \cap \Omega^s = \emptyset$ and $\phi_0 \notin \{\Omega^f, \Omega^s\}$. The level-set equation governing the evolution of ϕ in Ω can be formulated

$$\frac{\partial \phi}{\partial t} + \mathbf{f} \cdot \boldsymbol{\nabla}\phi = 0, \tag{13}$$

where $\mathbf{f} = \mathbf{f}(t)$ is the extension velocity vector describing the motion of the embedded boundary. The velocity vector $\mathbf{f}$ is spatially constant since all points on the surface of a rigid body move at the same speed. Local geometric properties of the moving boundary

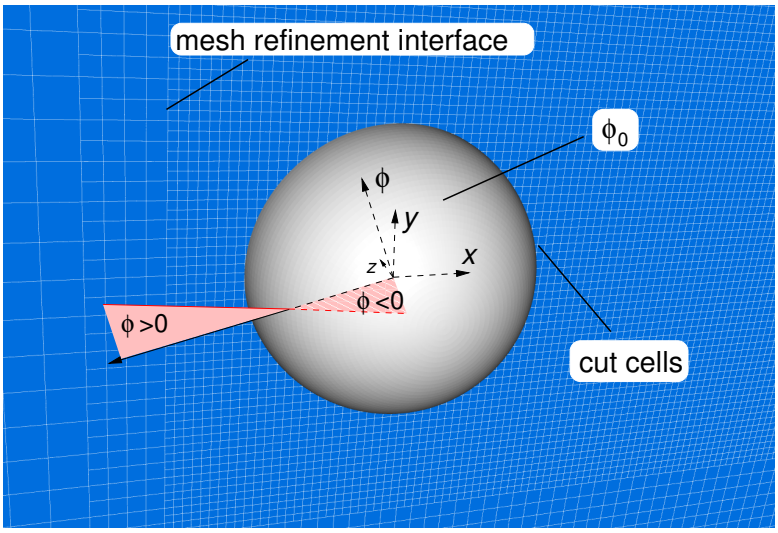

Figure 1: Embedded-body representation by the zero contour ϕ_0 of the level-set function ϕ

can be obtained by differentiating the scalar level-set function. For example, the normal vector $\mathbf{n}$ pointing into Ω^f is computed by

$$\mathbf{n} = \frac{\boldsymbol{\nabla}\phi}{|\boldsymbol{\nabla}\phi|}. \tag{14}$$

Using this geometrical information as well as the location of the moving boundary, which can be obtained by interpolation of the scalar level-set function, the shape of the cut-cells, i.e., the cells which are intersected by ϕ_0, and the surfaces enclosing these cells, which are needed for the flux computation, can be readily computed.[16] A significant advantage of using an implicit level-set representation of the moving boundary is given by the fact that its sign inherently carries the information on which cells fall into the fluid region and which cells belong to the solid region. When an explicit geometric description of the moving boundary such as a surface triangulation is used instead, this information has to be regenerated each time the solid body moves, which can be of significant computational cost.

Under the evolution of Eq. (13), the level-set function, however, loses the signed distance property and needs to be frequently reinitialized.[13,15,23] This can be efficiently achieved by solving a partial differential equation, which transforms the perturbed scalar field $\tilde{\phi}$ into a level-set field satisfying the signed distance property $|\boldsymbol{\nabla}\phi| = 1$ with respect to the zero level set ϕ_0. A family of constrained reinitialization schemes has been developed by Hartmann et al.[13–15] These schemes avoid the unphysical displacement of the zero level set ϕ_0 during the reinitialization, which standard schemes are known to suffer from.[13] In the present work, we solve the constrained reinitialization equation[14]

$$\partial_\tau \phi + S(\tilde{\phi})(|\boldsymbol{\nabla}\phi| - 1) = \beta F, \tag{15}$$

where $S(\tilde{\phi})$ is a smoothed sign function[13,14,23] and β is a weighting factor which is set $\beta = 0.5$. The quantity F is a forcing term, which corrects the displacement of the ϕ_0 contour.[14]

3.2.1 Discretization of the equations used by the level-set method

The level-set equation (13) is solved using the level-set method[12] on a uniform grid using a fifth-order spatial and third-order temporal discretization. Furthermore, the high-order constrained reinitialization scheme HCR-2 is used in combination with a fifth-order WENO discretization[24] of the constrained reinitialization equation (15) to frequently regularize the level-set field.[14]

3.3 Cut-cell generation

In Cartesian cut-cell methods, the boundary cells are reshaped such that embedded bodies are cut out of the computational domain Ω_λ.[9,10] In the final grid, the embedded boundaries are then represented by the piecewise linear boundary surfaces. This process has to be repeated every time the embedded boundary moves. Following Hartmann et al.,[9,10] boundary conditions are imposed by prescribing the primitive variables on ghost cells one of which is created for each boundary cell. The location of the ghost cells $\widetilde{\mathbf{x}^n}$ is not restricted by the Cartesian grid lines.[10,11] It is computed by

$$\widetilde{\mathbf{x}^n} = \hat{\mathbf{x}}^n + 2\left(\mathbf{x}_s^{\Gamma^n} - \hat{\mathbf{x}}^n\right). \tag{16}$$

That is, the centroid $\mathbf{x}_s^{\Gamma^n}$ of the boundary surface segment Γ^n is located on the line connecting the boundary cell center $\hat{\mathbf{x}}^n$ and the corresponding ghost cell center. The application of suitable Dirichlet and Neumann boundary conditions and the computation of the wall shear stress at the embedded boundary is discussed in detail in Hartmann et al.[9,10]

3.3.1 Small-cell treatment

One of the main challenges in the development of cut-cell methods is the so-called small-cell problem. This problem arises since the volume of a cut cell can become arbitrarily small and hence, very small time steps are necessary to satisfy the CFL stability constraint when explicit time integrators are used. Several approaches have been proposed to resolve this problem.[2,11,25–27]

A fully conservative method is the cell-merging method,[10,26,27] in which small cells are lumped together with adjacent cells. While such approaches have been successfully used for problems involving fixed boundaries[10,11] they are, however, prone to numerical oscillations when moving geometries are considered. These oscillations can be traced back to the frequent changes of the discrete operators in the vicinity of the moving boundary, the motion of which locally changes the grid structure. Another issue of this methodology

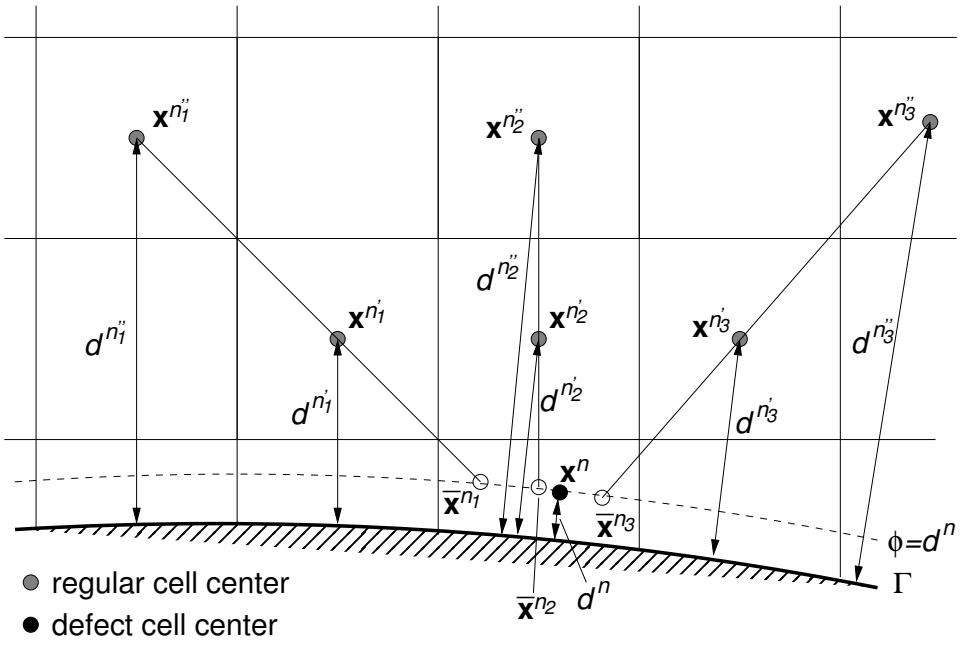

Figure 2: Illustration of the interpolation procedure

when used for moving-boundary problems is the separation of previously merged cells while conserving mass, momentum, and energy.

In the present work, we develop a conservative small-cell treatment which is based on a flux redistribution technique.[28,29] A cell n at refinement level λ is declared a small cell if its cell volume V^n is less than half of the original grid cell volume V^λ. In the first step, the primitive variables in the small cells are computed by interpolation rather than by solving Eq. (6). Hence, the aforementioned stability problem concerning the small cells is avoided in the first place. However, this procedure obviously violates mass, momentum, and energy conservation in the small cells, which we will in the following refer to as defect volumes. To correct the conservation error, the flux redistribution is needed in a second step as it will be outlined below.

To interpolate the primitive variables into a defect cell n located at $\mathbf{x}^n$, we identify M^n internal cell pairs n_i' and n_i'', $i = 1, \ldots, M^n$ in the vicinity of n. This is illustrated in Fig. 2, where $M^n = 3$. Let furthermore the wall-normal distance of the cell $\mathbf{x}^n$ be denoted by d^n such that the contour $\phi = d^n$ passes through $\mathbf{x}^n$. Next, M^n points $\overline{\mathbf{x}}^{n_i}$ on the $\phi = d^n$ contour are identified along the line connecting $\mathbf{x}^{n_i'}$ and $\mathbf{x}^{n_i''}$, Fig. 2. Using the velocity $\mathbf{v}^\Gamma$ on the boundary, the velocities at these points are computed via quadratic interpolation, while pressure and density are linearly extrapolated from $\mathbf{x}^{n_i'}$ and $\mathbf{x}^{n_i''}$. These intermediate values are interpolated into $\mathbf{x}^n$ along the arc or surface defined by $\phi = d^n$.

Summarizing, the primitive variables in the cell n are computed by

$$\begin{cases} \mathbf{v}^n = \sum_{i=1}^{M^n} \omega^{n_i} \left[\mathbf{v}^\Gamma + C^{n_i'} \left(\mathbf{v}^{n_i'} - \mathbf{v}^\Gamma \right) + C^{n_i''} \left(\mathbf{v}^{n_i''} - \mathbf{v}^\Gamma \right) \right], \\ p^n = \sum_{i=1}^{M^n} \omega^{n_i} \left[p^{n_i'} + \kappa^{n_i} (p^{n_i'} - p^{n_i''}) \right], \\ \rho^n = \sum_{i=1}^{M^n} \omega^{n_i} \left[\rho^{n_i'} + \kappa^{n_i} (\rho^{n_i'} - \rho^{n_i''}) \right], \end{cases} \tag{17}$$

where the interpolation constants are computed by

$$\begin{aligned} \kappa^{n_i} &= \frac{d^{n_i'} - d^n}{d^{n_i''} - d^{n_i'}}, \\ C^{n_i'} &= \frac{d^n (d^{n_i''} - d^n)}{d^{n_i'} (d^{n_i''} - d^{n_i'})}, \\ C^{n_i''} &= \frac{d^n (d^n - d^{n_i'})}{d^{n_i''} (d^{n_i''} - d^{n_i'})}, \\ e^{n_i} &= ||\mathbf{x}^{n_i'} + \kappa^{n_i} (\mathbf{x}^{n_i'} - \mathbf{x}^{n_i''}) - \mathbf{x}^n||_2, \\ \omega^{n_i} &= \frac{1/e^{n_i}}{\sum_{j=1}^{M^n} 1/e^{n_j}}. \end{aligned}$$

The normal distance d is readily obtained by $d = (\mathbf{x} - \mathbf{x}^\Gamma) \cdot \mathbf{n}$, where the normal vector $\mathbf{n}$ can be locally computed from the scalar level-set field using Eq. (14). The above interpolation scheme is also used to initialize the primitive variables in emerging cells at the boundary, i.e., cells which at a time level w belong to Ω^s and at the time level $w+1$ belong to Ω^f.

To correct the conservation error, a multi-level redistribution technique is used. Consider the solution $\mathbf{Q}^{(k-1)}$ at Runge-Kutta stage $k-1$. Let $\mathbf{Q}^{\star(k)}$ denote the solution after the interpolation step, i.e., after a Runge-Kutta substep has been performed on all regular cells and the small cells have been updated according to Eq. (17). Then, a defect vector $\mathbf{D}$ can be computed for each small cell by

$$\mathbf{D} = V \left(\mathbf{Q}^{(0)} - \alpha_k \Delta t \; L(\mathbf{Q}^{(k-1)}) - \mathbf{Q}^{\star(k)} \right), \tag{18}$$

which contains the local mass, momentum, and energy loss or gain in this cell due to the interpolation. As described in the following, the defect is finally distributed over a certain range of cells adjacent to the considered small cell. This is achieved by utilizing the hierarchical cell-tree data structure,[11] by which underlying parent and grandparent cells of the small cells can be quickly identified. Hence, we move from the small cell at the computation level λ to its (grand)parent, which is at the level $\lambda - \delta^C$, Fig. 3, and

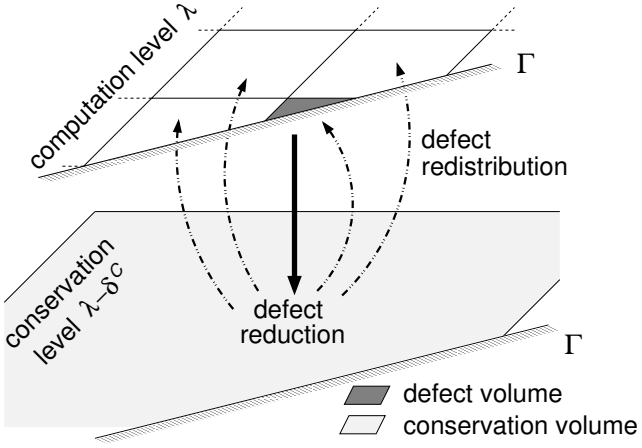

Figure 3: Schematic sketch of the multi-level small-cell treatment

which is denoted the conservation cell. In case the conservation cell is a small cell itself, it is merged with an adjacent neighbor to form a conservation patch. Next, all cells at level λ underlying the conservation cell or patch $\mathcal{C}$ are identified and collected in the set $S^{\mathcal{C}}$. The defect $\mathbf{D}^{\mathcal{C}}$ in $\mathcal{C}$ is the sum of the local defects of all small cells in $S^{\mathcal{C}}$, i.e.,

$$\mathbf{D}^{\mathcal{C}} = \sum_{n \in S^{\mathcal{C}}} \mathbf{D}^n, \quad \frac{V^n}{V^\lambda} < 0.5. \tag{19}$$

Finally, the redistribution step yields the final solution $\mathbf{Q}^{(k)}$ at Runge-Kutta stage k

$$\mathbf{Q}^{n,(k)} = \mathbf{Q}^{\star n,(k)} + \frac{\mathbf{D}^{\mathcal{C}}}{\sum_{n' \in \mathcal{C}} V^{n'}}, \quad \forall n \in \mathcal{C} \tag{20}$$

on the cells underlying a conservation cell or patch. On all other cells, we simply set $\mathbf{Q}^{n,(k)} = \mathbf{Q}^{\star n,(k)}$.

Obviously, for the redistribution step to be stable we need to require $V^n \ll \sum_{n' \in \mathcal{C}} V^{n'}$ for each small cell n, i.e., the conservation cell or patch must be significantly larger than the small cell. To be more precise, the total volume of the conservation cell or patch is controlled by the level difference δ^C between the computation level and the conservation level, as illustrated in Fig. 3. Preliminary numerical experiments suggested a value of $\delta^C = 3$, which has been used for the computations presented in the following.

4 RESULTS

To validate the developed numerical method, results of the laminar flow past non-moving and moving cylinders and a fixed sphere are discussed. All computations were performed on locally refined grids at a low Mach number of $M = 0.1$ to allow for a comparison with published data from simulations of incompressible flow. In all flow

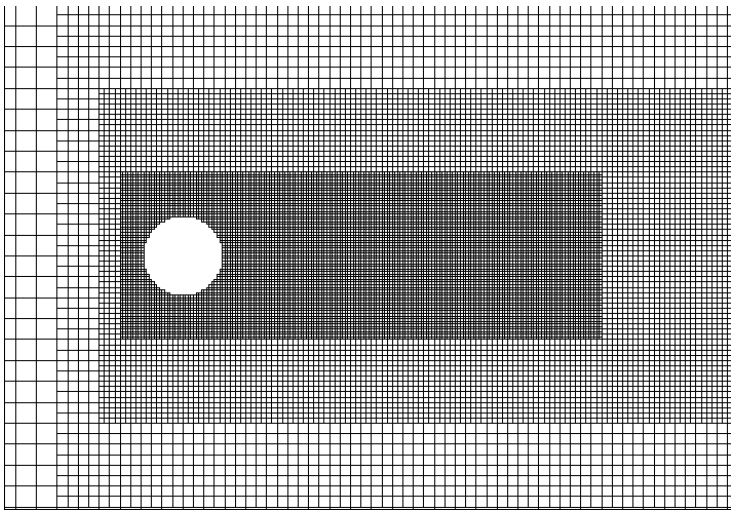

Figure 4: Topology of the refined Cartesian grid used for the simulations of the flow past a circular cylinder

simulations, a sponge layer formulation for the pressure and the density as proposed in[11] is used to drive the solution variables on the far-field boundaries towards the freestream values as well as to avoid spurious waves in the vicinity of these boundaries.

4.1 Uniform flow past a circular cylinder

We begin the discussion of our results with the flow past a circular cylinder, which is an appropriate validation test case since the flow field exhibits different flow structures depending on the Reynolds number and the curved boundary is challenging for Cartesian grid methods. Furthermore, it has been studied in numerous publications such that excellent reference data are available.[30–34] With D being the diameter of the circular cylinder the Reynolds number based on the freestream velocity is defined $Re_D = \frac{\varrho_\infty v_\infty D}{\mu_\infty}$. At very low Reynolds numbers $Re_D \ll 1$ the viscous terms are dominant and creeping flow is observed. When the Reynolds number is increased the flow separates on the leeward side of the cylinder and forms a symmetric pair of vortices, which are attached to the cylinder. At further increase of the Reynolds number these vortices are stretched and at a critical Reynolds number of about $Re_{D,crit} \approx 49$ the recirculation region becomes incipiently unstable.[34] This point of transition is called Bénard-von Kármán instability.[31] Above the critical Reynolds number the cylinder wake begins to oscillate and exhibits a wavy structure known as the von Kármán vortex street. Vortices are alternately shed from either side of the cylinder and convected downstream.[31] This essentially two-dimensional laminar flow regime appears for Reynolds numbers up to $140 - 194$,[34] above which three-dimensional flow phenomena emerge.

We performed two-dimensional simulations in a Reynolds number range of $20 - 100$. A patchwise refined grid is used to discretize the computational domain $\Omega : [-15D, 49D] \times$

Contribution	$Re_D = 20$		$Re_D = 40$		$Re_D = 100$	
	C_d	L_r/D	C_d	L_r/D	$\overline{C_d}$	St
Dennis and Chang[30]	2.045	-	1.522	-	-	-
Ye et al.[2]	2.03	0.92	1.52	2.27	-	-
Marella et al.[17]	-	-	1.52	2.30	-	-
Tseng and Ferziger[35]	-	-	-	-	1.42	0.164
Kim et al.[36]	-	-	-	-	1.33	0.165
De Palma et al.[37]	-	-	-	-	1.32	0.163
Present	2.046	0.95	1.530	2.35	1.341	0.165

Table 1: Computed parameters for the two-dimensional laminar flow past a circular cylinder

Contribution	$Re_D = 25$		$Re_D = 100$	
	C_d	L_r/D	C_d	L_r/D
Johnson and Patel[38]	2.36	0.08	1.08	0.88
Marella et al.[17]	-	-	1.06	0.88
Present	2.33	0.07	1.08	0.88

Table 2: Computed parameters for the three-dimensional laminar flow past a sphere

$[-15D, 15D]$, where the cylinder axis is located at $(x, y) = (0, 0)$ and the x direction is the streamwise direction. Using the refined meshes enables a fine resolution of the cylinder boundary layer and the near wake at a mesh width $h_\lambda \approx 0.03$, while a coarse mesh can be used in the far-field regions. The topology of the grid is shown in Fig. 4. The results are summarized in Table 1 in terms of the drag coefficient C_d, the length of the recirculation zone L_r, and the Strouhal number $St = \frac{Df_0}{v_\infty}$ characterizing the natural vortex shedding frequency f_0. The results are in good agreement with the reference data from the literature.

4.2 Uniform flow past a sphere

The numerical simulation of the flow past a sphere is a canonical and challenging test case for Cartesian grid methods since all angles of intersection between the geometry and the computational grid appear. Similar to the circular cylinder, the laminar flow past a sphere exhibits several regimes and the transition between those regimes depends on the Reynolds number of the freestream. We shall investigate the flow at Reynolds numbers $Re_D = 25$ and $Re_D = 100$, where D is the diameter of the sphere. In this range of Reynolds numbers the flow is steady and axisymmetric.

A three-dimensional computational domain $\Omega : [-15, 49] \times [-15, 15] \times [-15, 15]$ is used for the simulations and the center of the sphere located in the origin of the frame of reference. The patchwise refined grid shown in Fig. 5 with roughly 1.85 million cells at refinement levels 7 through 11 is used. It is refined axisymmetrically about the streamwise central axis in the x direction. As for the cylinder flow simulations, the grid is clustered

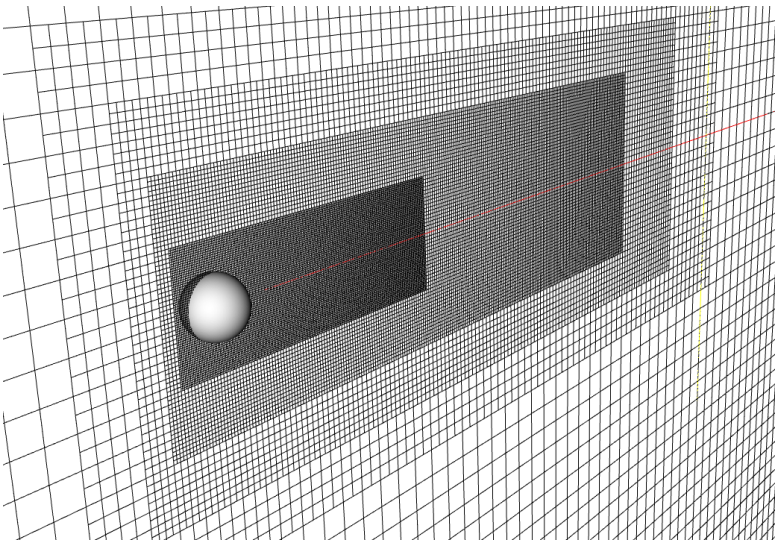

Figure 5: Topology of the refined Cartesian grid used for the simulations of the flow past a sphere

near the embedded sphere with a mesh spacing of $\Delta_x \approx 0.03$ such that the cell centers of the boundary cells are located at a distance $\Delta < 0.015$ from the sphere. The CFL number used in all simulations is 1.0 indicating the small cell treatment to be very effective. In fact, simulations at even larger CFL numbers were successful. The results which are summarized in Table 2 show a satisfactory agreement with the data from the literature.

4.3 Uniform flow past a transversely oscillating circular cylinder

Having validated the developed methodology for steady and unsteady flow problems in two and three space dimensions we move on to demonstrate its accuracy for moving-boundary problems. A canonical test case in this class of problems is the uniform flow past a circular cylinder which is forced to oscillate transversely with respect to the oncoming flow. When the Reynolds number of the flow is in a range where periodic vortex shedding occurs at the cylinder and the frequency of the transverse oscillation of the cylinder is near the natural shedding frequency of the vortex wake a phenomenon called "locking-on" occurs.[39,40] To be more precise, the vortex shedding frequency in this case changes to match the frequency of the forced oscillation. This effect depends primarily on the Reynolds number of the oncoming flow and the amplitude of the transverse oscillation of the cylinder.[40]

The computational set-up of the test case corresponds to that of the simulations performed for the flow past a fixed cylinder. The refinement patches on the grid were enlarged in the transverse y direction to account for the vertical motion of the cylinder, which is prescribed as a harmonic oscillation

$$y(t) = A_e \sin(2\pi f_e t), \tag{21}$$

where A_e and f_e denote the amplitude and the frequency of the excitation, respectively.

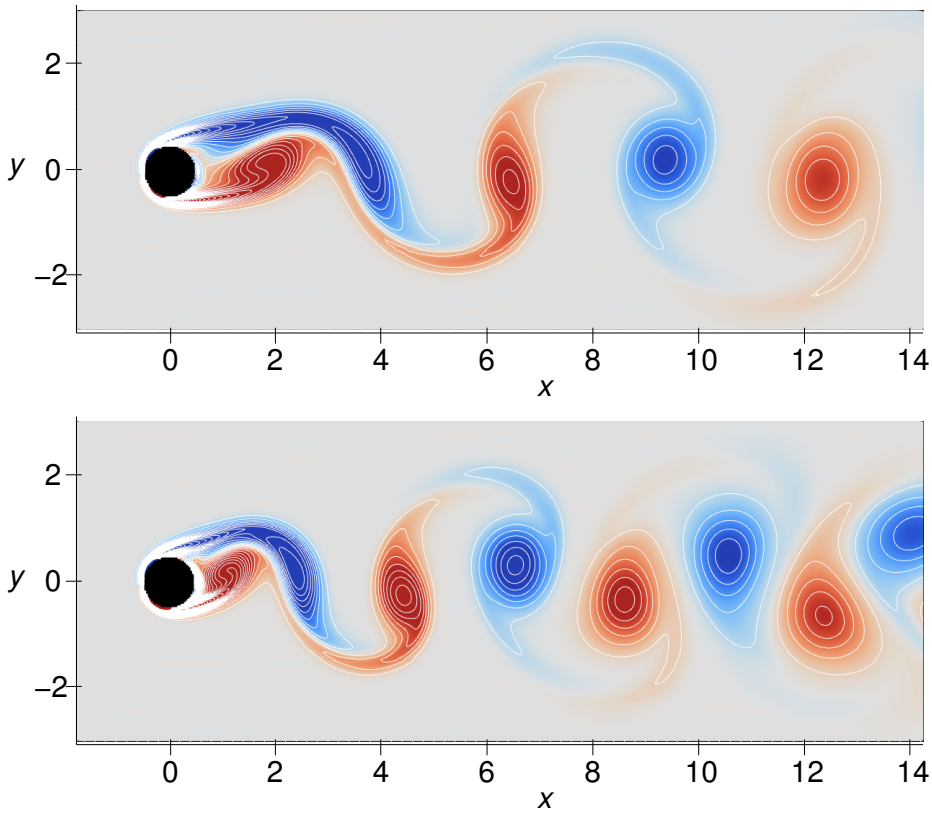

Figure 6: Flow past a transversely oscillating cylinder at $Re_D = 185$. Contours of the vorticity component ω_z in the range $-1 \leq \omega_z \leq 1$ for the cylinder position $(x, y) = (0, 0)$ and the velocity $v_{cyl} = -2\pi f_e A_e$: (a) $f_e/f_0 = 0.8$; (b) $f_e/f_0 = 1.0$.

We investigate a parametric space that has been used in previous experimental work[41] and numerical studies.[42–46] Three simulations are performed at a Reynolds number of $Re_D = 185$, an excitation amplitude $A_e = 0.2D$, and excitation frequencies $f_e/f_0 = 0.8$, 1.0, and 1.2. The natural vortex shedding frequency at $Re_D = 185$ was determined first in a simulation with a fixed cylinder, which yielded $St = 0.194$.

In agreement with the findings of Young et al.,[46] locking-on occurs for the cases $f_e/f_0 = 0.8$ and $f_e/f_0 = 1.0$. This is illustrated in Fig. 6, which shows contours of the out-of-plane vorticity component ω_z for these cases. It can be clearly observed that the vortex shedding occurs at a higher frequency for $f_e/f_0 = 1.0$ than for $f_e/f_0 = 0.8$. This is corroborated by the Mach number contours shown in Fig. 7. To obtain quantitative data, the shedding frequency f_s is measured by taking an FFT of the v velocity history recorded at the position $(x, y) = (1, 0)$, which is shown in Fig. 8. At $f_e/f_0 = 0.8$ and $f_e/f_0 = 1.0$ we obtain $f_s = f_e$, which evidences the occurence of the locking phenomenon. That is, the vortex shedding frequency matches the excitation frequency in these cases. As reported by Young et al.[46] subharmonics of the excitation become visible in the v velocity plots for $f_e/f_0 = 1.2$, see Fig. 8(c). Furthermore, the von Kármán vortex street does not lock on

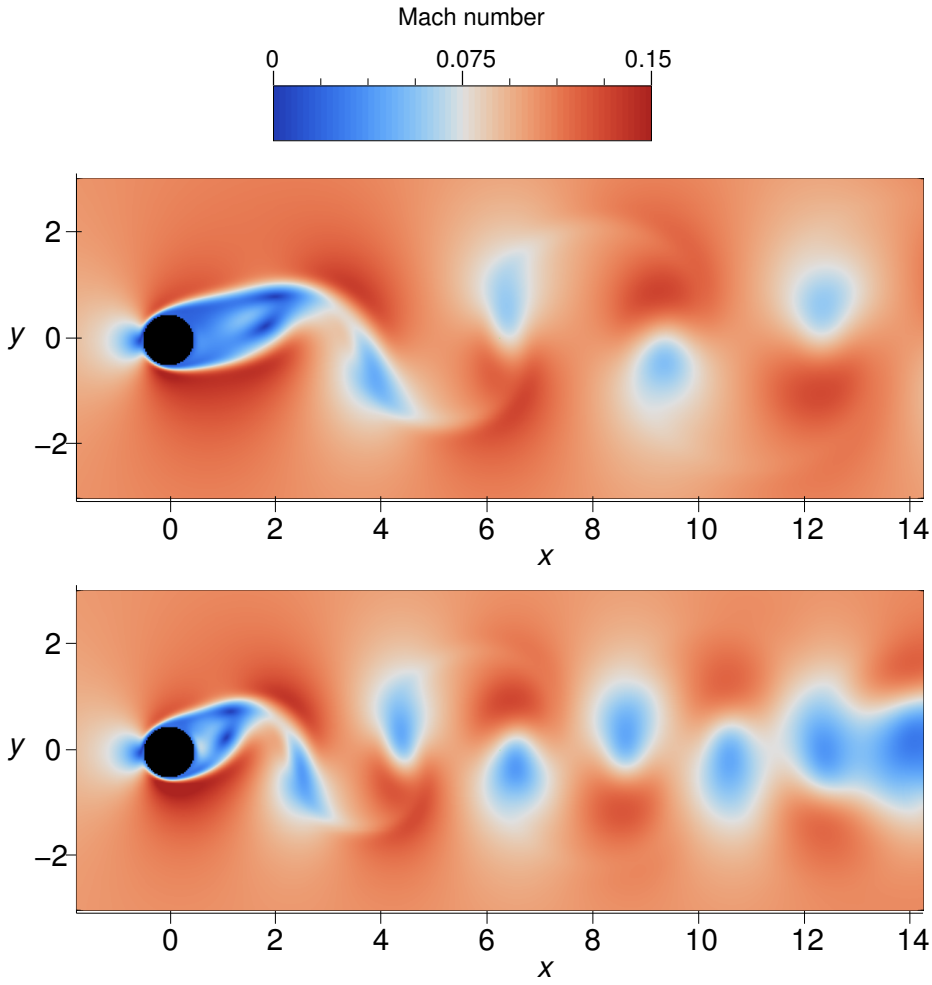

Figure 7: Flow past a transversely oscillating cylinder at $Re_D = 185$. Mach number contours for the cylinder position $(x, y) = (0, 0)$ and the velocity $v_{cyl} = -2\pi f_e A_e$: (a) $f_e/f_0 = 0.8$; (b) $f_e/f_0 = 1.0$.

to the forced cylinder movement at $f_e/f_0 = 1.2$ and the simulations yield $f_s/f_0 = 0.93$.

5 CONCLUDING REMARKS

A cut-cell Cartesian grid method for general two- and three-dimensional moving-boundary problems of compressible viscous flow was presented. In this method, the moving boundary is represented using the level-set method and boundary conditions are applied using ghost cells, which can be freely positioned in space, making the present method flexible in terms of the geometry of the embedded boundaries. A novel small-cell treatment is developed to avoid the numerical instability caused by cells of very small volume emerging near the moving boundaries. Unlike the cell-merging technique, this small-cell treatment proved to be robust also for moving boundaries.

The developed method was validated in several test cases. Fully compressible simu-

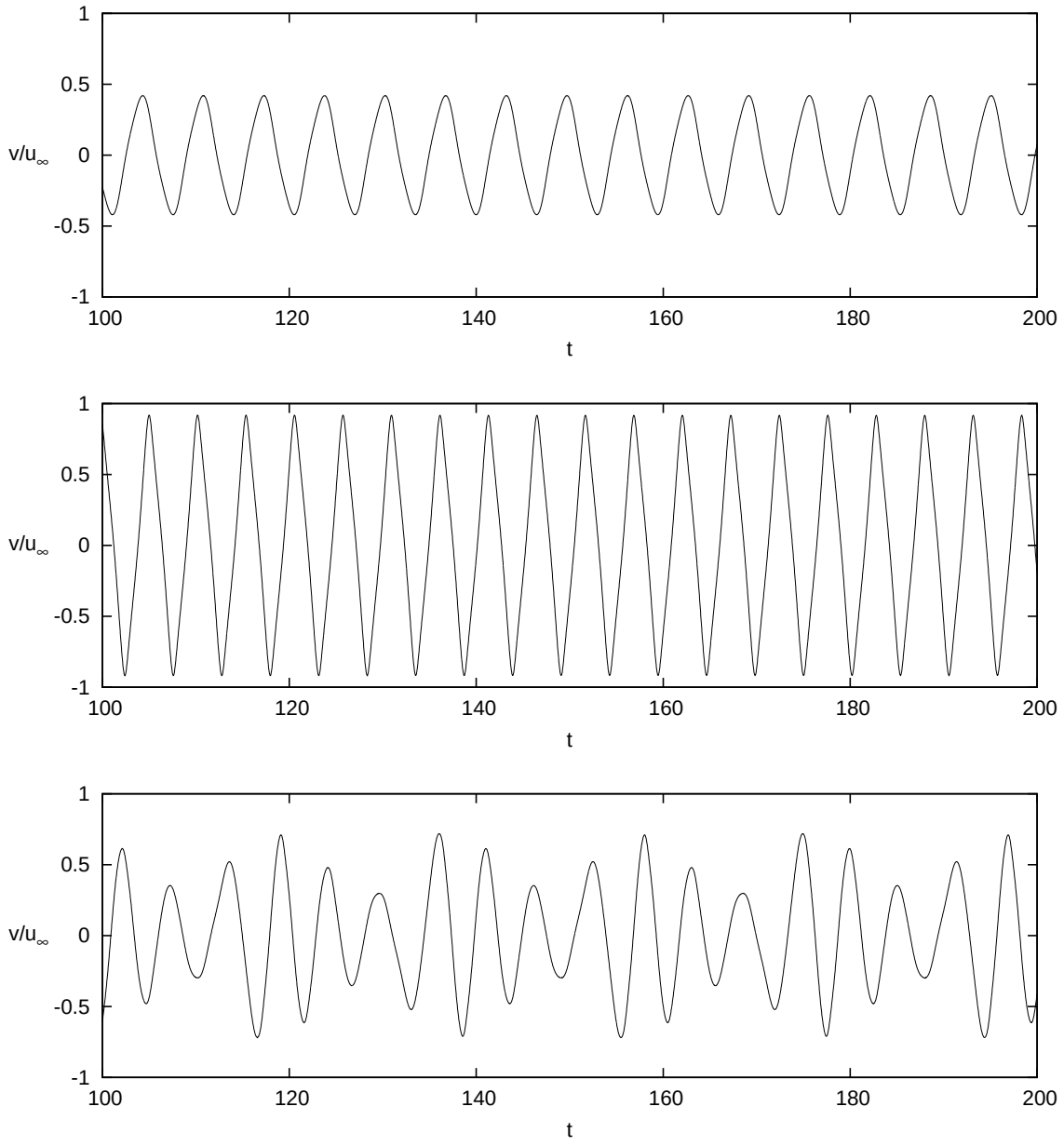

Figure 8: Time history of the v velocity at $(x, y) = (1, 0)$ for non-dimensional values of the excitation frequency of: (a) $f_e/f_0 = 0.8$; (b) $f_e/f_0 = 1.0$; (b) $f_e/f_0 = 1.2$.

lations of the uniform flow past non-moving and transversely oscillating cylinders and a sphere were discussed to demonstrate the accuracy of the cut-cell method for two- and three-dimensional problems. Three-dimensional simulations of moving-boundary problems are under way. Furthermore, an extension of the method to problems involving fluid-structure interaction is envisaged.

ACKNOWLEDGMENTS

This research was performed as part of the collaborative research center SFB 686, which is funded by the German Research Association (Deutsche Forschungsgemeinschaft (DFG)), and the Cluster of Excellence "Tailor-Made Fuels from Biomass", which is funded by the German Excellence Initiative. The support is gratefully acknowledged.

REFERENCES

[1] C. Peskin, Flow patterns around heart valves: a numerical method, *J. Comput. Phys.*, **10**, 252–271 (1972) .

[2] 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**, 209–240 (1999) .

[3] R. Mittal, H. Dong, M. Bozkurttas, F. Najjar, A. Vargas, A. von Loebbecke, A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries, *J. Comput. Phys.*, **227**, 4825–4852 (2008) .

[4] M. Barad, P. Colella, S. Schladow, An adaptive cut-cell method for environmental fluid mechanics, *Int. J. Numer. Meth. Fluids*, **60**, 473–514 (2009) .

[5] R. Yang, F. Stern, Sharp interface immersed-boundary/level-set method for wave-body interactions, *J. Comput. Phys.*, **228**, 6590–6616 (2009) .

[6] R. Mittal, G. Iaccarino, Immersed boundary methods, *Ann. Rev. Fluid Mech.*, **37**, 239–261 (2005) .

[7] R. Ghias, R. Mittal, H. Dong, A sharp interface immersed boundary method for compressible viscous flows, *J. Comput. Phys.*, **225**, 528–553 (2007) .

[8] M. de Tullio, P. De Palma, G. Iaccarino, G. Pascazio, M. Napolitano, An immersed boundary method for compressible flows using local grid refinement, *J. Comput. Phys.*, **225**, 2098–2117 (2007) .

[9] D. Hartmann, M. Meinke, W. Schröder, A general formulation of boundary conditions on Cartesian cut cells for compressible viscous flow, AIAA Paper 2009-3878 (2009).

[10] D. Hartmann, M. Meinke, W. Schröder, A strictly conservative Cartesian cut-cell method for compressible viscous flows on adaptive grids, *Comput. Meth. Appl. Mech. Eng.*, **(submitted)**.

[11] D. Hartmann, M. Meinke, W. Schröder, An adaptive multilevel multigrid formulation for Cartesian hierarchical grid methods, *Comput. Fluids*, **37**, 1103–1125 (2008) .

[12] S. Osher, J. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations, *J. Comput. Phys.*, **79**, 12–49 (1988) .

[13] D. Hartmann, M. Meinke, W. Schröder, Differential equation based constrained reinitialization for level set methods, *J. Comput. Phys.*, **227**, 6821–6845 (2008) .

[14] D. Hartmann, M. Meinke, W. Schröder, The constrained reinitialization equation for level set methods, *J. Comp. Phys.*, **229**, 1514–1535 (2010) .

[15] D. Hartmann, M. Meinke, W. Schröder, On accuracy and efficiency of constrained reinitialization, *Int. J. Numer. Meth. Fluids*, DOI:10.1002/fld.2135 (2009) .

[16] Y. Cheny, O. Botella, The LS-STAG method: A new immersed boundary/level-set method for the computation of incompressible viscous flows in complex moving geometries with good conservation properties, *J. Comput. Phys.*, **229**, 1043–1076 (2010) .

[17] S. Marella, S. Krishnan, H. Liu, H. Udaykumar, Sharp interface Cartesian grid method I: an easily implemented technique for 3D moving boundary computations, *J. Comput. Phys.*, **210**, 1–31 (2005) .

[18] B. van Leer, Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method, *J. Comput. Phys.*, **32**, 101–136 (1979) .

[19] M.-S. Liou, C. J. Steffen Jr., A new flux splitting scheme, *J. Comput. Phys.*, **107**, 23–39 (1993) .

[20] D. Hartmann, M. Meinke, W. Schröder, Erratum to "Differential equation based constrained reinitialization for level set methods" [J. Comp. Phys. 227 (2008) 6821-6845], *J. Comput. Phys.*, **227**, 9696 (2008) .

[21] D. Hartmann, M. Meinke, W. Schröder, An adaptive dual-mesh method for premixed combustion using the level set approach, AIAA Paper 2010-1068 (2010).

[22] D. Hartmann, M. Meinke, W. Schröder, A level-set based adaptive-grid method for premixed combustion, *Combust. Flame*, **(submitted)**.

[23] M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flow, *J. Comput. Phys.*, **114**, 146–159 (1994) .

[24] G.-S. Jiang, D. Peng, Weighted ENO schemes for Hamilton-Jacobi equations, *SIAM J. Sci. Comput.*, **21**, 2126–2143 (2000) .

[25] M. Berger, R. LeVeque, Stable boundary conditions for Cartesian grid calculations, *Comp. Sys. Eng.*, **1**, 305–311 (1990) .

[26] J. Quirk, An alternative to unstructured grids for computing gas dynamic flows around arbitrarily complex two-dimensional bodies, *Comput. Fluids*, **23**, 125–142 (1994) .

[27] M. Kirkpatrick, S. Armfield, J. Kent, A representation of curved boundaries for the solution of the Navier-Stokes equations on a staggered three-dimensional Cartesian grid, *J. Comput. Phys.*, **184**, 1–36 (2003) .

[28] R. Pember, J. Bell, P. Colella, W. Crutchfield, M. Welcome, An adaptive Cartesian grid method for unsteady compressible flow in irregular regions, *J. Comput. Phys.*, **120**, 278–304 (1995) .

[29] P. Colella, D. Graves, B. Keen, D. Modiano, A Cartesian grid embedded boundary method for hyperbolic conservation laws, *J. Comput. Phys.*, **211**, 347–366 (2006) .

[30] S. Dennis, G.-Z. Chang, Numerical solutions for steady flow past a circular cylinder at Reynolds numbers up to 100, *J. Fluid Mech.*, **42**, 471–489 (1970) .

[31] M. Provansal, C. Mathis, L. Boyer, Bénard-von Kármán instability: Transient and forced regimes, *J. Fluid Mech.*, **182**, 1–22 (1987) .

[32] C. Williamson, Oblique and parallel modes of vortex shedding in the wake of a circular cylinder at low Reynolds numbers, *J. Fluid Mech.*, **206**, 579–627 (1989) .

[33] H.-Q. Zhang, U. Fey, B. Noack, M. König, H. Eckelmann, On the transition of the cylinder wake, *Phys. Fluids*, **7** (4), 779–794 (1995) .

[34] C. Williamson, Vortex dynamics in the cylinder wake, *Ann. Rev. Fluid Mech.*, **28**, 477–539 (1996) .

[35] Y.-H. Tseng, J. Ferziger, A ghost-cell immersed boundary method for flow in complex geometry, *J. Comput. Phys.*, **192**, 593–623 (2003) .

[36] J. Kim, D. Kim, H. Choi, An immersed-boundary finite-volume method for simulations of flow in complex geometries, *J. Comput. Phys.*, **171**, 132–150 (2001) .

[37] P. De Palma, M. de Tullio, G. Pascazio, M. Napolitano, An immersed-boundary method for compressible viscous flows, *Comput. Fluids*, **35**, 693–702 (2006) .

[38] T. Johnson, V. Patel, Flow past a sphere up to a Reynolds number of 300, *J. Fluid Mech.*, **378**, 19–70 (1999) .

[39] G. Koopmann, The vortex wakes of vibrating cylinders at low reynolds numbers, *J. Fluid Mech.*, **28**, 501–512 (1967) .

[40] P. Stansby, The locking-on of vortex shedding due to the cross-stream vibration of circular cylinders in uniform and shear flows, *J. Fluid Mech.*, **74**, 641–665 (1976) .

[41] W. Gu, C. Chyu, D. Rockwell, Timing of vortex formation from an oscillating cylinder, *Phys. Fluids*, **6**, 3677–3682 (1994) .

[42] E. Guilmineau, P. Queutey, A numerical simulation of vortex shedding from an oscillating circular cylinder, *J. Fluids Struct.*, **16**, 773–794 (2002) .

[43] J. Yang, E. Balaras, An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries, *J. Comput. Phys.*, **215**, 12–40 (2006) .

[44] J. Yang, S. Preidikman, E. Balaras, A strongly coupled embedded-boundary method for fluid-structure interactions of elastically mounted rigid bodies, *J. Fluids Struct.*, **24**, 167–182 (2008) .

[45] M. Vanella, E. Balaras, A moving-least-squares reconstruction for embedded-boundary formulations, *J. Comput. Phys.*, **228**, 6617–6628 (2009) .

[46] D. Young, Y. Jan, C. Chiu, A novel immersed boundary procedure for flow and heat simulations with moving boundary, *Comput. Fluids*, **38**, 1145–1159 (2009) .