AIAA JOURNAL
Vol. 43, No. 1, January 2005

# Cartesian Grid Method for Moderate-Reynolds-Number Flows Around Complex Moving Objects

Jo-Einar Emblemsvåg,* Ryuta Suzuki,† and Graham V. Candler‡
*University of Minnesota, Minneapolis, Minnesota 55455*

**A method is presented to simulate the fluid dynamics of low- to moderate-Reynolds-number flows around solid objects with complex shapes on a fixed Cartesian grid. On grid points occupied by a solid object, the method forces the fluid motion to be equal to the motion of the solid object, whereas the boundary conditions on the solid–fluid interface are enforced through a specific treatment of grid points close to the interface. The method is fairly easily implemented in both two and three dimensions, and a grid-refinement study shows that the method is globally close to second-order accurate. Steady and unsteady flow results over a cylinder and a sphere fixed in the grid show very good agreement with previous experimental and computational investigations. Computations of a cylinder moving in the grid with a surrounding fluid initially at rest produce results that compare very well with the fixed-cylinder results, thus demonstrating the validity of the method for moving geometry simulations. The method's applicability to investigations of the unsteady aerodynamics of microscale flapping flight is demonstrated through simulations of the fluid dynamics of a flapping cantilever beam and a free-falling two-dimensional "leaf."**

## I. Introduction

THE motivation for the work presented in this paper is that recent advances in the research and development of microelectromechanical system technology have encouraged research into the potential construction of micro–air vehicles (MAVs) at millimeter to centimeter scales. One of the many challenges associated with the design of MAVs is to develop a better understanding of aerodynamics at these small scales. MAVs operate at much smaller Reynolds numbers than traditional flying machines, which means that viscous effects are more dominant, forcing the development of new flight mechanisms for the efficient generation of lift and thrust. Nature has already mastered flight at these scales and gives a good indication of the challenges that lie ahead. Insects and birds achieve flight through a wide range of flapping wing motions that vary with animal size and specialization. This type of flight is characterized by a high degree of flow unsteadiness that is exploited for aerodynamic efficiency, thus making it very hard to analyze and mimic for the purpose of designing robotic MAVs.

It is clear that computational fluid dynamics can be a very powerful tool in the effort to analyze the flow physics of flapping flight, but the very nature of the unsteady aerodynamics introduces complex moving boundaries. Usually, body-fitted deforming grids are used to simulate flows involving moving boundaries because this allows for a simple and accurate enforcement of the boundary conditions. Here one has the choice of using structured or unstructured grids (or a combination of both), but these approaches have some significant disadvantages for our application. With structured grids, usually only the simplest of geometries can be represented without running into grid quality issues unless a multiblock approach is used. Hennighausen et al.[1] developed a method that successfully generated structured curvilinear grids around relatively complex shapes of prototype MAVs using a multiblock approach. Unfortunately, sufficiently large flapping motions cannot be modeled with this method because the grids are too severely distorted when the grid is deformed by the moving boundaries. Unstructured grid methods have been developed to avoid the aforementioned problems. Practically any geometrical shape can be represented with this approach and distorted grids can be avoided using remeshing strategies. The major disadvantage here is that the implementation of these methods is very complex and the remeshing algorithms are computationally expensive. Instead, if the boundary conditions can be imposed with sufficient accuracy on a fixed Cartesian grid, the complexities of body-conforming grid methods can be avoided. A Cartesian grid approach is also computationally less expensive because one avoids the extra computations involved with transforming the governing equations to a curvilinear coordinate system. There is also no need for expensive remeshing strategies, which makes this approach more flexible when adapted to different problems.

Cartesian grid methods have been used quite extensively for a number of different types of problems. Generally, the main challenge concerning the implementation of these methods is accurate treatment of irregular boundaries that do not conform to the underlying Cartesian mesh. Several approaches have been developed for a range of problems of different physical nature, but a distinction can be made between approaches that treat the boundary as a "diffuse" interface and approaches that treat it as a "sharp" interface.

For the diffuse interface methods, the boundary effects are typically represented by a discrete set of body or surface forces, and examples of such are the immersed boundary method, developed by Peskin[2] for simulations of blood flow in the heart, and the volume-of-fluid method originally developed by Nichols et al.[3] and Hirt and Nichols[4] for fluid flows with multiple free surfaces. The diffuse interface approach often results in a smearing of the boundary effects across a few grid points, which might work well in problems where the physical boundary is less sharp, such as for fluid–fluid interactions.

The sharp interface methods, on the other hand, model the discontinuity across the boundary directly by splitting grid cells located at the interface. On the resulting irregular cells at the solid–fluid boundary, the challenge is to maintain the accuracy of the discretization and conservation of the flow and material quantities. A major issue concerning the splitting of grid cells is that arbitrarily small volumes can be created, which means that the computations will be unstable unless the time step is reduced proportionally ("the small-cell problem"). There have been several approaches to address this issue

Received 24 February 2004; revision received 14 June 2004; accepted for publication 16 June 2004. 

*Graduate Research Assistant, Aerospace Engineering and Mechanics and Army High Performance Computing Research Center, 110 Union Street SE; emblem@aem.umn.edu.

†Undergraduate Research Assistant, Aerospace Engineering and Mechanics and Army High Performance Computing Research Center, 110 Union Street SE.

‡Professor, Aerospace Engineering and Mechanics and Army High Performance Computing Research Center, 110 Union Street SE. Associate Fellow AIAA.

such as the large-time-step approach developed by LeVeque[5,6] and Berger and LeVeque.[7] Berger and LeVeque[8] also studied approaches in which the small-cell problem is avoided by the use of a rotated difference scheme in the cells cut by the fluid–solid interface. These methods are globally second-order accurate, but less than second-order accurate at the interface, and have only been used for inviscid flows. Another approach to address the time-stepping issue is the merging of cells near the interface to eliminate small cells. A cell-merging technique was used together with a finite volume method by Ye et al.[9] to simulate two-dimensional viscous flows over immersed stationary solid boundaries. Udaykumar et al.[10] extended the method to successfully simulate viscous flow problems with moving solid boundaries similar to the ones we are interested in here. This method demonstrates second-order-accurate treatment of the boundaries, but unfortunately the complex logic involved with cell merging makes it very hard to extend the method to three dimensions.

Recently, another class of Cartesian grid methods has been developed to simulate flows in and around complex solid structures. These methods use an extra term in the governing equations to impose an external force field on the grid points occupied by a solid object to make the fluid act like a solid. Interpolation schemes or weighting functions are used on grid points close to solid–fluid interfaces to determine the appropriate strength of the forcing function necessary to correctly represent boundaries that are not grid aligned. Examples of such methods can be found in the papers by Mohd-Yusof,[11] Fadlun et al.,[12] Kim et al.,[13] and Balaras.[14]

The goal of the present work was to develop a robust and inexpensive method that could accurately simulate the three-dimensional fluid dynamics of biomimetic flight mechanisms. First, a proven and efficient finite difference scheme that is second-order accurate in both time and space was chosen to solve the incompressible Navier–Stokes equations on a staggered grid. Second, an approach was devised that accounts for immersed solid objects in a fairly uncomplicated manner while maintaining adequate accuracy. The resulting method imposes a fluid velocity equal to the solid velocity at grid points occupied by a solid object through a forcing function similar to that of other force-field methods.[11–14] At solid–fluid interfaces, however, grid points inside the solid boundary are treated as ghost points to impose the boundary conditions. The small-cell problem that is usually encountered with split cells is avoided through a "softened" boundary condition while still maintaining a sharp interface. Some accuracy is compromised, however, because continuity is enforced locally everywhere including at solid–fluid interfaces. This results in some smearing of the boundary, but because the elliptic equation remains untouched, it exhibits very good convergence properties and any "off-the-shelf" Poisson solver can be used. In a way, the current method can be considered a mix between sharp and diffuse interface methods and, as such, has similarities to some of the methods discussed earlier. Although the boundary is tracked as a sharp interface and boundary conditions are imposed according to its presence, the inclusion of the boundary effects in the source term of an elliptic equation for a continuous flowfield resembles that of diffuse interface methods.

## II. Computational Method

### A. Flow Solver

Two conservation equations must be satisfied for the incompressible Navier–Stokes equations.

Continuity:

$$\nabla \cdot \boldsymbol{u} = 0 \tag{1}$$

Momentum:

$$\rho\left(\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u}\right) = -\nabla p + \nabla\cdot\tau \tag{2}$$

where $\boldsymbol{u}$ is the velocity vector, $\rho$ is the constant for density, $p$ is pressure, and $\tau$ is the stress tensor. For the numerical scheme used, a simplification is made to the momentum equation by first using the vector identity $(\boldsymbol{u}\cdot\nabla)\boldsymbol{u} = \frac{1}{2}\nabla|\boldsymbol{u}|^2 - \boldsymbol{u}\times\omega$ to get

$$\frac{\partial \boldsymbol{u}}{\partial t} - \boldsymbol{u}\times\omega + \nabla\left(\frac{p}{\rho} + \frac{1}{2}|\boldsymbol{u}|^2\right) = \frac{1}{\rho}\nabla\cdot\tau \tag{3}$$

It is recognized that the expression in parentheses is just total pressure divided by the density, and a substitution is made:

$$P = p/\rho + \tfrac{1}{2}|\boldsymbol{u}|^2 \tag{4}$$

Now the elliptic partial differential equation is obtained by taking the divergence of the momentum equation:

$$\nabla^2 P = -\frac{\partial(\nabla\cdot\boldsymbol{u})}{\partial t} - \nabla\cdot\boldsymbol{F}, \qquad \boldsymbol{F} = -\boldsymbol{u}\times\omega - \frac{1}{\rho}\nabla\cdot\tau \tag{5}$$

Flow variables are updated in time with an explicit second-order-accurate predictor–corrector scheme in the following procedure:

1) A Poisson equation is solved for the pressure:

$$\nabla^2 P^n = (\nabla\cdot\boldsymbol{u})^n/\Delta t - \nabla\cdot\boldsymbol{F}^n \tag{6}$$

Then the velocity is estimated at the next time step:

$$\boldsymbol{u}^{(n+1)_e} = \boldsymbol{u}^n - \Delta t\,(\boldsymbol{F}^n + \nabla P^n) \tag{7}$$

2) The pressure is recomputed using estimated quantities:

$$\nabla^2 P^{(n+1)_e} = \frac{(\nabla\cdot\boldsymbol{u})^{(n+1)_e} + (\nabla\cdot\boldsymbol{u})^n}{\Delta t} - \nabla\cdot\boldsymbol{F}^{(n+1)_e} \tag{8}$$

The velocity is then corrected:

$$\boldsymbol{u}^{n+1} = \tfrac{1}{2}\left[\boldsymbol{u}^n + \boldsymbol{u}^{(n+1)_e} - \Delta t\left(\boldsymbol{F}^{(n+1)_e} + \nabla P^{(n+1)_e}\right)\right] \tag{9}$$

Spatial derivatives of $\boldsymbol{F}$ (which contains the convective and diffusive terms of the momentum equation) are approximated with second-order-accurate central differences on a staggered rectilinear grid. Scalar quantities are assigned in the center of each grid cell, while vector quantities are assigned at forward cell faces. A detailed description of the discretization can be found in the literature.[15]

### B. Immersed Solid Objects

The Navier–Stokes equations are solved in the entire computational domain, including parts that are occupied by an object in the flow. To enforce a no-flux condition through a solid object, a fluid velocity equal to the solid velocity has to be enforced at grid points occupied or "masked" by the object (Fig. 1). To achieve this, a forced-flux condition originally developed specifically for the current numerical scheme by McGrattan et al.[15] has been adopted and

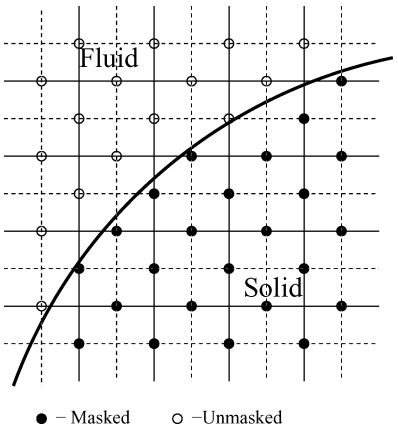


**Fig. 1 Illustration of masked grid points located at cell faces represented by solid grid lines.**

slightly modified for our purposes. In this approach a velocity can be enforced at masked points by specifying

$$F_n = -\frac{\partial P}{\partial n} + \beta \frac{(u_{nf} - u_n)}{\Delta t} \tag{10}$$

The preceding equation can be recognized as the momentum equation, where $F_n$ is the normal component of $\boldsymbol{F}$ at masked cell faces. On the right-hand side, the first term is the normal derivative of the total pressure across the cell face from the previous estimate of pressure, since pressure at the current time step is not known before the Poisson equation is solved. The temporal change in the velocity field is specified in the second term, where $u_{nf}$ is the normal component of velocity to be enforced on the next time step and $u_n$ is the solution from the current time step. Because the evaluation of $F_n$ relies on the previous estimate of pressure, the temporal change in velocity is only approximate. However, because pressure changes little from one time step to the next, the approximation is very good. The empirical relaxation factor $\beta$ should be set to 1 (no relaxation) in cases when the motion of the solid object is continuous, but if a flow is started impulsively a good estimate of pressure may not be available for evaluation of the forcing term. A bad estimate of pressure will lead to oscillations in the forced solid velocity, but choosing $\beta < 1$ serves to dampen and stabilize the oscillations in a period of transition before the relaxation factor can be reset to 1. Therefore, implemented as part of the predictor–corrector updating scheme, the forced-flux condition works very well as long as the motion pattern of the solid object is continuous, but it cannot accurately capture impulsively started flows.

### C. Boundaries

At nonmasked grid points close to the boundary, the stencil required to compute $\boldsymbol{F}$ may include masked grid points. Instead of altering the stencil in this case, we let the masked points serve the purpose of ghost points to impose the no-slip wall boundary conditions at the solid–fluid interface. The value given to the ghost point is purely fictional for the purpose of computing $\boldsymbol{F}$ with the desired boundary conditions and is completely independent of its treatment as a masked grid point. Traditionally, when ghost points (or cells) are used with body-fitted grids, the ghost-point value is chosen such that an interpolated value on the boundary matches the desired boundary condition. However, when a ghost point is treated this way with an arbitrary boundary on a Cartesian grid, a problem arises that is related to the small-cell problem discussed earlier. Figure 2 illustrates how the interpolated value might become very large if the boundary cuts the grid very close to a "fluid" point. This will severely limit the size of the stable convective time step, which is inversely proportional to velocity:

$$\Delta t < \min(\Delta x/u, \Delta y/v, \Delta z/w) \tag{11}$$

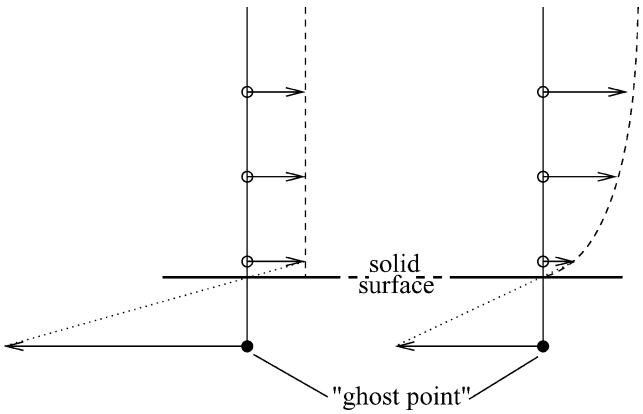


**Fig. 2 One-dimensional illustration of the interpolation of the fictional value of a ghost point: discontinuous (initial) flowfield on the left and continuous (developed) flowfield on the right.**

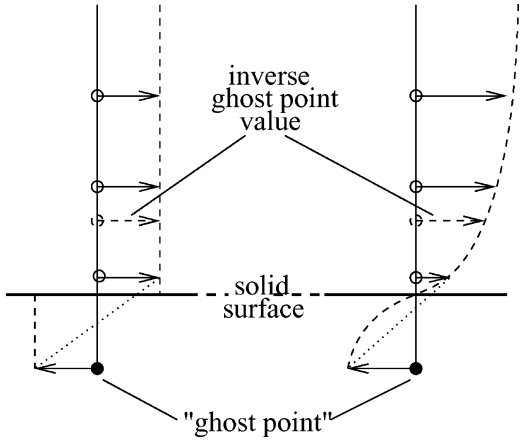


**Fig. 3 One-dimensional illustration of the interpolation of the ghost point value by treating the boundary line segment as a symmetry line: discontinuous (initial) flowfield on the left and continuous (developed) flowfield on the right.**

When the time step decreases, the computations become very time-consuming, and in theory the time step may approach zero, which is of course unacceptable.

To avoid the limitations on the time step, our approach instead considers the boundary as a symmetry line. (An analogy to this can be found in potential flow theory in the representation of wall boundaries with "the method of images.") More accurately, because the no-slip boundary condition is to be imposed, the ghost points must reflect a symmetric flowfield for the normal component of velocity and an inverse symmetric flowfield for the tangential component. Because the boundary line can be curved, however, the symmetry must be imposed for small line segments of the boundary separately. Figure 3 illustrates how this treatment of the boundary conditions affects the value of the ghost points for the tangential part of the velocity. If the illustration in Fig. 2 is the exact way of imposing the boundary conditions, then the symmetry approach will in certain situations underestimate the value of the ghost point. However, as illustrated in Fig. 3, underestimating the ghost-point value has the same effect as mislocating the boundary, and the error will always be less than half the local grid spacing. It is also clear that the largest error occurs when the flowfield is discontinuous at the boundary (as can occur with an unphysical initial flowfield), but, with a developed viscous flowfield where the velocity stagnates at the wall, the boundary condition is more accurate. Therefore, if a boundary layer is adequately resolved, the error will most likely be very small.

A major advantage with the current approach is that the logic involved with enforcing the no-slip boundary condition is fairly easy to implement. The procedure is to locate the exact opposite location of a ghost point across a line segment and then interpolate the value from four neighboring grid points. In the case that one or more of the neighboring points are inside the boundary (masked point), the intersection point of the boundary line segment with the interpolation direction becomes an interpolation point instead. Figures 4a and 4b illustrate the procedure for two sample situations, where in Fig. 4a the horizontal component of the boundary line segment is larger than the vertical component, whereas in Fig. 4b the vertical component is the largest. First the value at $e$ is interpolated from $a$ and $b$, and $f$ is interpolated from $c$ and $d$. Then the value at the symmetry point is interpolated from $e$ and $f$. The value of the ghost point becomes the negative of the interpolated value to give $\boldsymbol{u} = -\tilde{\boldsymbol{u}}$, where $\tilde{\boldsymbol{u}}$ is the interpolated value. This means that the components of $\boldsymbol{u}$ can be handled separately, which is very convenient for the staggered grid approach, where the components are stored at different locations. In the case of a moving boundary, the value of the ghost point becomes $\boldsymbol{u} = 2\boldsymbol{u}_s - \tilde{\boldsymbol{u}}$, where $\boldsymbol{u}_s$ is the velocity of the boundary line segment. There are no significant complications in extending the procedure to three dimensions.

It is worth mentioning that to preserve the second-order accuracy of the numerical scheme at the boundaries, quadratic interpolation

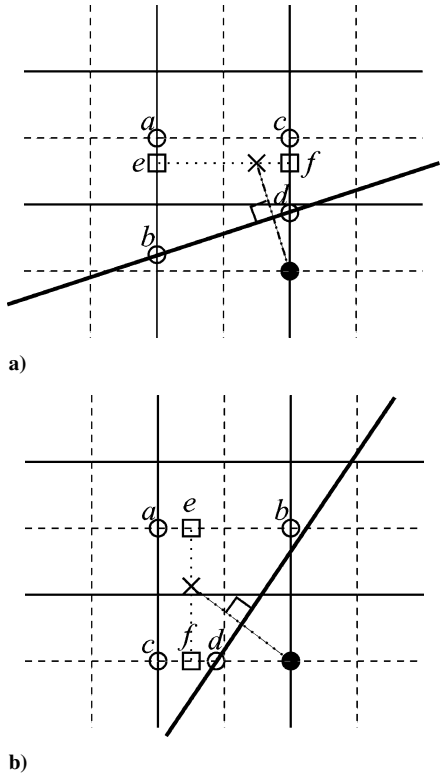


**Fig. 4 Two-dimensional illustration of the interpolation of the inverse value of a ghost point (●, ghost point; ○, interpolation stencil points; and ×, location of the symmetry point).**

of ghost points would seem more preferable instead of the simple linear interpolation explained earlier. However, there are reasons why this has not been implemented as part of the current method. First of all, quadratic interpolation would require a 9-point stencil for two-dimensional computations and a 27-point stencil for three-dimensional computations, which would severely complicate implementation logic. Second, quadratic interpolation is much more sensitive to the topology of the interpolation stencil and could actually worsen an unphysical disturbance where linear interpolation would serve as a damper.

### D. Elliptic Equation and Boundary Effects

Because the overall numerical method developed accounts for immersed solid objects without altering the underlying numerical scheme, the treatment of immersed solid objects can be adopted with minor modifications to other numerical schemes as well. Also, because the Poisson equation is untouched, it allows for easy implementation of efficient algorithms for solving this most expensive part of the computations. However, because split grid cells are not accounted for directly in the discretization of the Poisson equation, the flexibility in the choice of Poisson solvers comes at a cost. The elliptic equation enforces continuity on every grid cell, including cells that are split by the solid–fluid interface. In the split cells, the solid boundary is represented by the source term and continuity holds within the entire cell. This results in at most a one-cell smearing of the solid–fluid interface. However, as will be demonstrated in the results section, this smearing of the boundary effects has relatively little influence on accuracy for low to moderate Reynolds numbers. For higher Reynolds-number flows with thinner boundary layers, the one-cell smoothing of the boundary may affect the solution more significantly, but that is not investigated here.

## III. Results

First, a grid-convergence study will be presented to establish the influence of the treatment of immersed moving boundaries on the overall accuracy of the numerical method. Then the computed results of two-dimensional flows over a cylinder and three-dimensional flows over a sphere are compared with other experimental and computational results, before a few sample applications of the method are demonstrated for a flapping cantilever beam and a falling "leaf."

### A. Grid Convergence

In the absence of a suitable problem with an available exact solution, this grid-convergence study took the approach of comparing the solution of an oscillating cylinder on a very fine grid vs the solutions on coarser grids (similar to a study by Udaykumar et al.[10]). The cylinder was placed in the middle of a closed chamber with no-slip walls and a width and height of four cylinder diameters. A sinusoidal vertical oscillation pattern with an amplitude of one-half the cylinder diameter was imposed, giving a peak Reynolds number of 50. Figure 5a shows a vorticity contour plot for the solution after one oscillation cycle on the highest grid resolution of $520 \times 520$. This solution was compared vs grid resolutions of $40 \times 40$, $120 \times 120$, $200 \times 200$, and $360 \times 360$. Both the $L_1$ and $L_2$ norms of the error

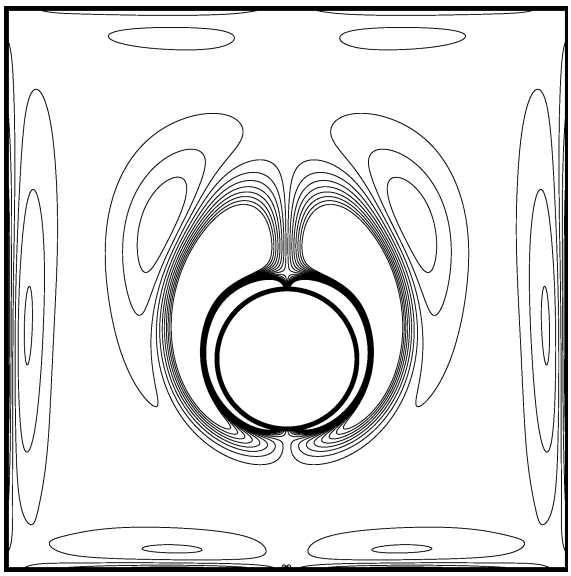

**Fig. 5a Vorticity contour plot of the oscillating cylinder in the closed chamber after one cycle.**

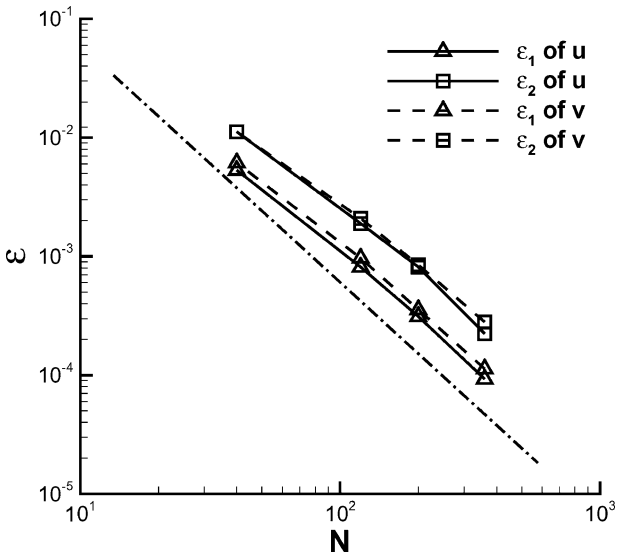


**Fig. 5b Behavior of the error vs grid resolution plotted together with a reference line of slope −2 indicating second-order convergence behavior.**

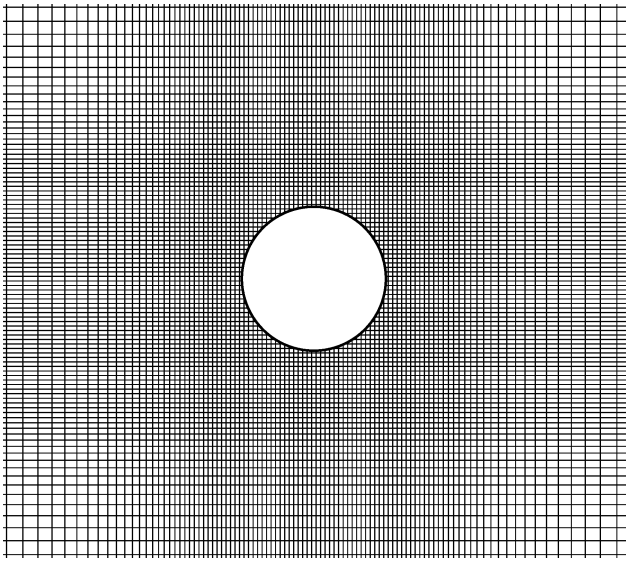

**Fig. 6 Grid topology for cylinder flow computations.**

were computed in the following manner:

$$\varepsilon_1 = \frac{1}{N}\sum_{j=1}^{N}\left|\phi_j^N - \phi_j^{520}\right|, \qquad \varepsilon_2 = \left[\frac{1}{N}\sum_{j=1}^{N}\left(\phi_j^N - \phi_j^{520}\right)^2\right]^{\frac{1}{2}} \tag{12}$$

where $N$ is the number of grid points and $\phi$ is the flow quantity chosen for comparison. For our analysis the error norms were computed for the $x$ and $y$ components of velocity and the result is plotted vs log scales in Fig. 5b. Included in the plot is a reference line of slope $-2$ that corresponds to second-order convergence, and it is clear that globally the numerical method exhibits almost second-order convergence. Therefore, the treatment of the immersed boundaries appears to reduce accuracy slightly, which is not unexpected, but the performance of the method is still essentially second-order and is therefore acceptable.

### B. Flow over a Cylinder

Flow over a cylinder was chosen as a case study for the validation of the method because a number of studies by others serve well for comparisons. Also, because the cylinder boundary cuts grid cells at arbitrary angles, producing good results is a stringent test of the method. For all computations, the cylinder is placed in the middle of a rectangular domain with a mesh that is uniform in the area occupied by the cylinder and nonuniform elsewhere, as shown in Fig. 6. One of the domain boundaries is chosen as an inflow boundary and the other boundaries are open boundaries where the flowfield can either exit or enter. On the inflow boundary a uniform far-field velocity is prescribed together with a Neumann boundary condition for the pressure. On the open boundaries, pressure is set equal to the far-field pressure and Neumann boundary conditions are prescribed for velocity.

The drag coefficient for the cylinder was computed for Reynolds numbers between 20 and 500. The Reynolds number is computed as $Re_d = U_\infty d/\nu$, where $U_\infty$ is the far-field flow velocity, $d$ is the cylinder diameter, and $\nu$ is kinematic viscosity. All of these computations were performed on a $30d \times 30d$ domain to place the domain boundaries far enough away to minimize their influence on the solution. Up to a Reynolds number of 80, the total number of grid points is 160 in both the $x$ and $y$ directions, and the diameter of the cylinder is resolved with 32 grid points. In the Reynolds number range of 100 to 300, the numbers of grid points in the $x$ and $y$ directions are 220 and 180, respectively, and the cylinder diameter is resolved with 50 grid points. At the Reynolds number of 500, the cylinder was resolved with 64 grid points and the numbers of grid points were 540 in the $x$ direction and 280 in the $y$ direction. The drag coefficient is computed as $c_d = F_d/\frac{1}{2}\rho U_\infty^2 d$, where $F_d$ is the cylinder drag force and $\rho$ is the density. The results are plotted in Fig. 7 (Refs. 9, 16, and 17) together with other experimental and computational results.

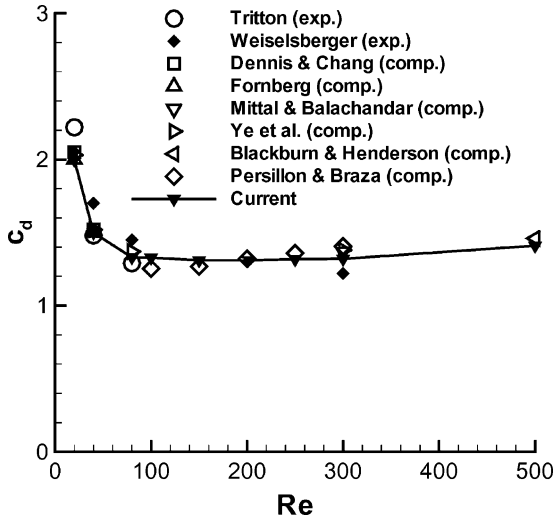


**Fig. 7 Comparison of drag coefficient on a cylinder with other experimental and computational data. The data have mainly been extracted from validation studies in papers by Ye et al.,[9] Blackburn and Henderson,[17] and Persillon and Braza.[16]**

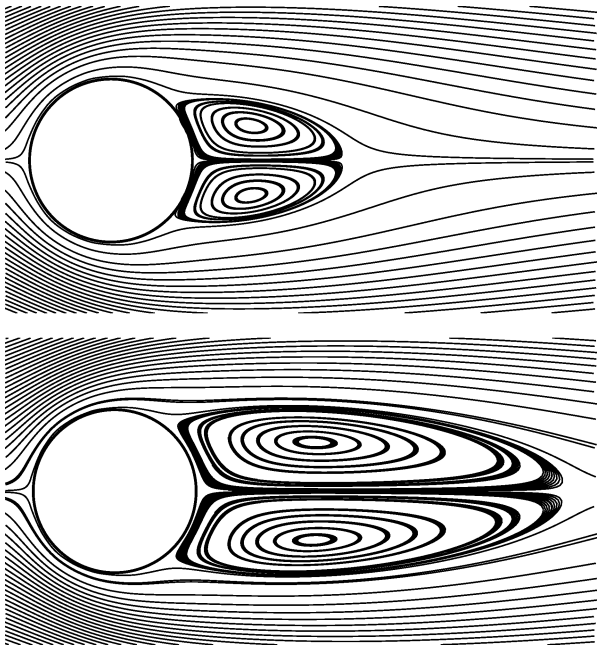

**Fig. 8 Streamlines of the converged flow over a cylinder (top, $Re = 20$ and bottom, $Re = 40$).**

For the Reynolds numbers of 20 and 40, the flow is steady with two separated vortices in the wake of the cylinder, as can be seen in Fig. 8. The computed drag coefficient in these two cases agrees very well with computations by Ye et al.,[9] Fornberg,[18] and Dennis and Chang,[19] and also agrees reasonably well with the experimental measurements of Weiselsberger[20] and Tritton.[21] The wake separation length, measured from the rear of the cylinder to the tail end of the wake, is a feature of the flow that is also useful to compare for validation. Table 1 shows that the current results achieve good agreement with computations by others.

For Reynolds numbers of 80 and above, the flow is unsteady and Fig. 7 shows that the computed average drag coefficient continues to agree well with other computations. The experimental value from Weiselsberger at a Reynolds number of 300 is a bit lower than the computed values, but this is to be expected because three-dimensional effects become important at higher Reynolds

**Table 1 Comparison of measured wake length $L_w/d$ for Reynolds numbers of 20 and 40 with other computational data[a]**

| | $L_w/d$ | |
|---|---|---|
| $Re_d \rightarrow$ | 20 | 40 |
| Dennis and Chang[19] | 0.94 | 2.35 |
| Fornberg[18] | 0.91 | 2.24 |
| Ye et al.[9] | 0.92 | 2.27 |
| Current | 0.93 | 2.28 |

[a]Data are extracted from a validation study in the paper by Ye et al.[9]

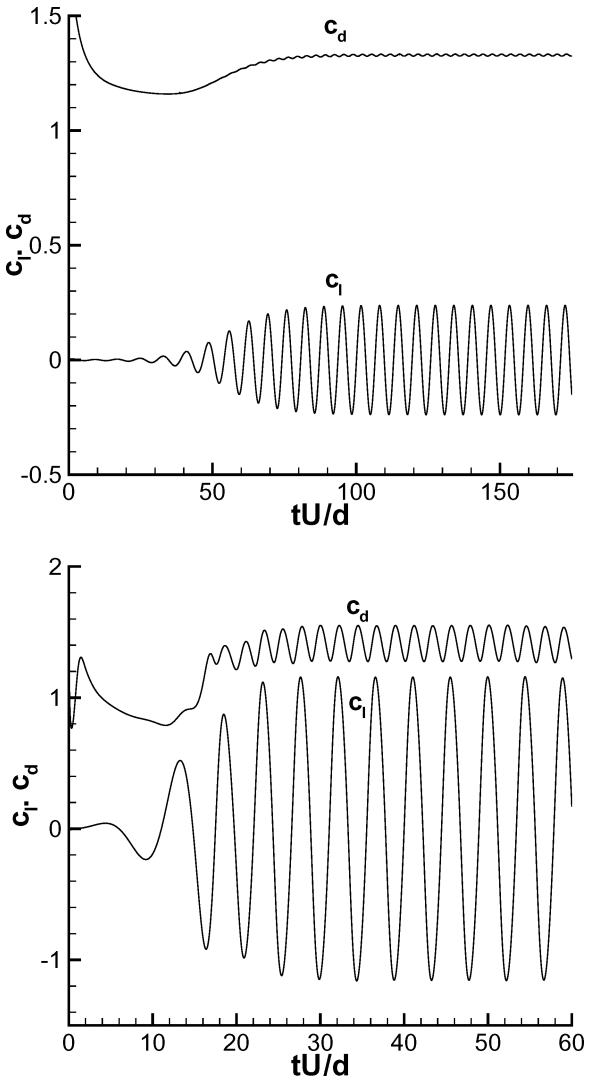


**Fig. 9 Time evolution of lift and drag coefficient on flow over a cylinder (top, $Re = 80$ and bottom, $Re = 500$).**

numbers, and two-dimensional computations tend to overestimate the drag. This phenomenon is studied and described by Mittal and Balachandar[22] as well as Persillon and Braza.[16] The unsteadiness at this Reynolds-number range consists of periodic shedding of vortices, and the Strouhal number defined as $Sr = fd/U_\infty$, where $f$ is the shedding frequency, is commonly used to characterize this shedding process. The shedding frequency can easily be measured from the oscillation frequency of the lift coefficient (Fig. 9). Good agreement with other computations is achieved also with the Strouhal number as illustrated in the plot in Fig. 10.

In the results presented so far, the cylinder has been fixed in the grid; however, the method has been developed to allow an immersed object to move within the grid. To validate this capability, computations were carried out for a cylinder that translates through the grid

**Table 2 Comparison of computed results of flow over a cylinder when the cylinder is moving in a surrounding fluid at rest and a stationary cylinder in a freestream**

| Case | $Re_d$ | $c_d$ | $L_w/d$ | $Sr$ |
|---|---|---|---|---|
| Moving | 20 | 2.07 | 0.95 | —— |
| Stationary | 20 | 2.04 | 0.94 | —— |
| Moving | 40 | 1.55 | 2.36 | —— |
| Stationary | 40 | 1.52 | 2.36 | —— |
| Moving | 80 | 1.36 | —— | 1.54 |
| Stationary | 80 | 1.35 | —— | 1.54 |

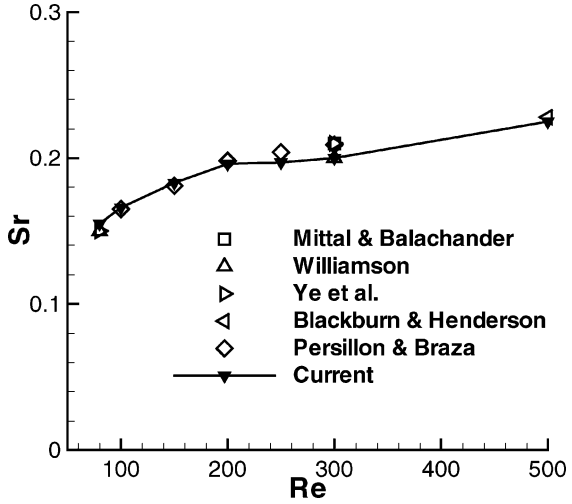


**Fig. 10 Comparison of the vortex shedding Strouhal number with other computational data.**

with the surrounding fluid initially at rest. The entire computational domain was allowed to move in space together with the cylinder such that the cylinder remained in the center of the grid away from the domain boundaries. To allow for the domain translation, the grid in the entire domain was uniform such that data could be copied from one grid point to the next when the domain translated a distance equal to the grid spacing. The domain boundary conditions remained the same as explained earlier for the stationary cylinder, but the uniform zero freestream velocity had to be reimposed on the freed grid points on the front end of the grid for every translational step.

Because of the uniform grid, the domain size of the previous nonuniform grid simulations could not be obtained without the computations becoming too expensive. Instead the moving cylinder computations were performed on a smaller $16d \times 16d$ domain with a total number of 512 points in both the $x$ and $y$ directions, and the cylinder diameter was resolved with 32 grid points. The moving-cylinder results were performed for Reynolds numbers of 20, 40, and 80 and compared with simulations on a stationary cylinder in the same domain with a freestream velocity imposed. The numerical parameters were exactly the same in both cases including the time step, which was chosen to be as large as possible with Courant and von Neumann numbers of 0.8. The results are shown in Table 2 and the numbers agree very well in all respects. One thing that should be mentioned is that the forces computed on the moving cylinder have a small amount of noise, but the averaged values agree very well with the stationary cylinder. If one compares the pressure contours on the stationary and moving cylinder they also agree very well, as shown in Fig. 11.

### C. Flow over a Sphere

Flow around a sphere was chosen for a validation study of the method's performance in three dimensions. In a paper by Johnson and Patel,[23] the dynamics of the flow over a sphere are studied computationally for Reynolds numbers up to 300. It is found that up to a Reynolds number of 200, the flow is steady and axisymmetric.

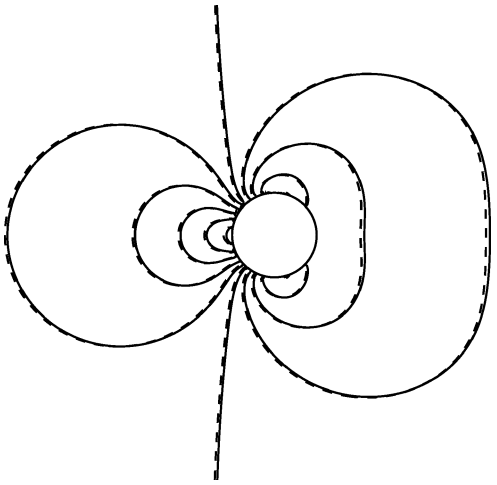

**Fig. 11 Comparison of the pressure contours on the cylinder for a Reynolds number of 20 (——, stationary cylinder and – – –, moving cylinder).**

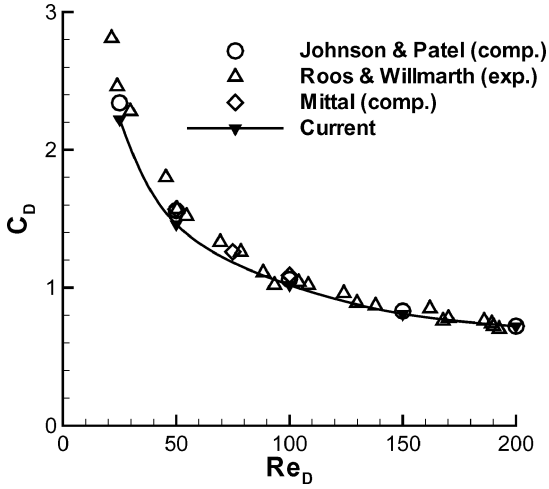


**Fig. 12 Comparison of drag coefficient on the flow over a sphere with data extracted from papers by Johnson and Patel[23] and Mittal.[24]**

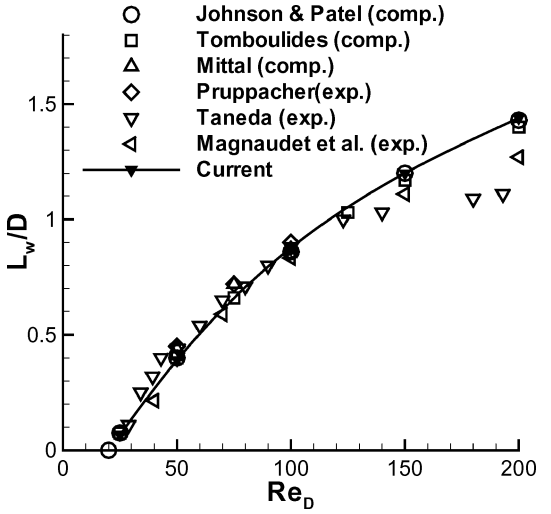


**Fig. 13 Comparison of wake recirculation length on the flow over a sphere with data extracted from papers by Johnson and Patel[23] and Mittal.[24]**

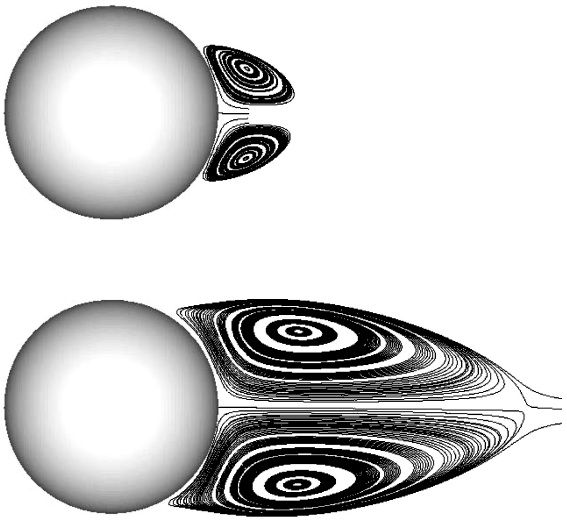

**Fig. 14 Streamline plots of the recirculation region on a cross section of the flow over a sphere (top, *Re* = 50 and bottom, *Re* = 200).**

In the Reynolds number range of 210 to 270 the flow is steady and nonaxisymmetric, and above a Reynolds number of 270 the flow is unsteady. These results agree well with other studies, as documented in their paper, and reproducing these results would therefore serve to validate our method. Numbers extracted from a validation study in a paper by Mittal[24] have also been used for comparisons.

For the computations up to a Reynolds number of 270, the sphere was placed in the middle of a $(16D)^3$ domain, where $D$ is the sphere diameter. A computation for a Reynolds number of 25 on a $(30D)^3$ domain showed only a minor difference in result with that for the $(16D)^3$ domain. The blockage ratio for the three-dimensional sphere is smaller than the blockage ratio for the two-dimensional cylinder for the same diameter vs domain width ratio. The total number of points was 128 in all directions, and the sphere diameter was resolved with 32 grid points, which proved adequate. At a Reynolds number of 300 a big $(30D)^3$ domain was used with a total number of 250 points in the streamwise direction and 160 points in the other two directions.

In the steady axisymmetric regime, computations were carried out for Reynolds numbers of 25, 50, 100, 150, and 200. A comparison of the computed drag coefficient and wake length are presented in Figs. 12 and 13, respectively, whereas streamlines of the converged flows at the Reynolds numbers of 50 and 200 are plotted in Fig. 14. The drag coefficient is computed as $C_D = F_D/(\frac{1}{2}\rho U_\infty^2 \pi D^2/4)$, where $F_D$ is the drag force in the streamwise direction, and the wake recirculation length $(L_W/D)$ is computed as explained previously with the cylinder. The drag coefficient plot in Fig. 12 shows that the current method underpredicts the drag by about 5% at the lower end of the regime compared with the other experimental and computational data, whereas it comes more in agreement and within 1% of the computed results of Johnson and Patel in the higher end of the regime. At the same time, the measured wake recirculation lengths agree very well with the results of Johnson and Patel and also reasonably well with Tomboulides et al.[25] and Mittal.

One computation at a Reynolds number of 250 was carried out for the steady and nonaxisymmetric region. The nature of the nonaxisymmetry is explained well in the paper by Johnson and Patel and it will not be explained in detail here. Figure 15 shows swirl surface plots in the wake region to illustrate the structure of the wake. The swirl detects an isolated vortex structure by considering the velocity gradient tensor, $\nabla \boldsymbol{u}$, and its characteristic equation:

$$\lambda^3 + P\lambda^2 + Q\lambda + R = 0$$

where $\lambda$ are the eigenvalues of $\nabla \boldsymbol{u}$. The discriminant of the equation is

$$\Delta = \tfrac{1}{2}\tilde{R}^2 + \tfrac{1}{3}\tilde{Q}^3$$

where $\tilde{Q} = Q - 1/3P^2$ and $\tilde{R} = R + 2/27P^3 - 1/3PQ$. When $\Delta > 0$, a complex pair of eigenvalues exists, which coincides with

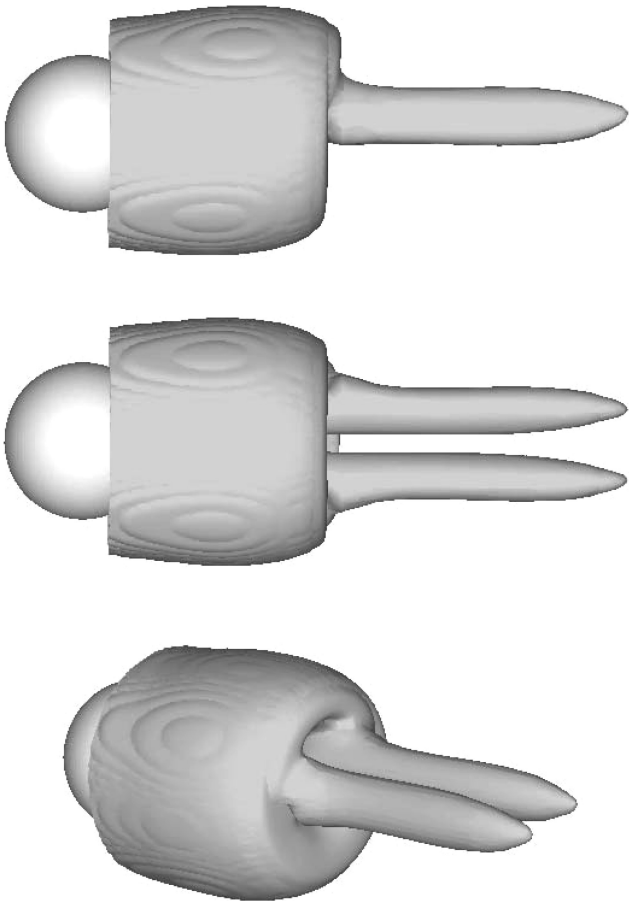

**Fig. 15 Surface plots of the swirl on the converged flow over a sphere at a Reynolds number of 250. The swirl is calculated with the method described by Chong et al.[26] The surface is cut close to the sphere to make the sphere visible.**

a region of swirl, and the method is explained in detail in the paper by Chong et al.[26] In the figures, the plotted value of $\Delta$ has been selected from visual judgment of what best illustrates the flow structure. From Fig. 15 it is clear that there is only one plane of symmetry and there are two tails extending from the separation region. Unlike the axisymmetric flows at lower Reynolds numbers, fluid enters and exits the vortex region at a steady rate, and the two tails are where fluid exits the separation region. The orientation of the plane of symmetry is arbitrary, but in our simulations the plane of symmetry has oriented itself consistently within a 45-deg region, which can be attributed to the topology of the grid. Due to the nonaxisymmetric wake, there is a lift component on the sphere normal to a plane perpendicular to the symmetry plane, which is computed to be $C_L = -0.062$ by Johnson and Patel, and is in good agreement with a lift coefficient of $C_L = -0.059$ computed with our Cartesian grid method.

A Reynolds number of 300 was chosen for the unsteady flow regime. The unsteadiness is characterized by a nonaxisymmetric periodic shedding of hairpin vortices with one plane of symmetry. Isosurface plots of the swirl are shown in Fig. 16 to illustrate the structure of the wake flow, which is very similar to the results of Johnson and Patel. A plot of the time evolution of the drag coefficient ($C_D$), the lift coefficient ($C_L$), and the side force coefficient ($C_S$) after a periodic state is obtained and is shown in Fig. 17. The side force coefficient is the force normal to the symmetry plane and the lift force coefficient is the force normal to a plane perpendicular to the symmetry plane. From the side force coefficient history, it is obvious that the plane of symmetry remains fixed in time as is also apparent from the wake structure in Fig. 15. In fact, a slight rotation was observed, but not enough to show up in the force coefficient plot. The time-averaged drag coefficient and lift coefficient were found

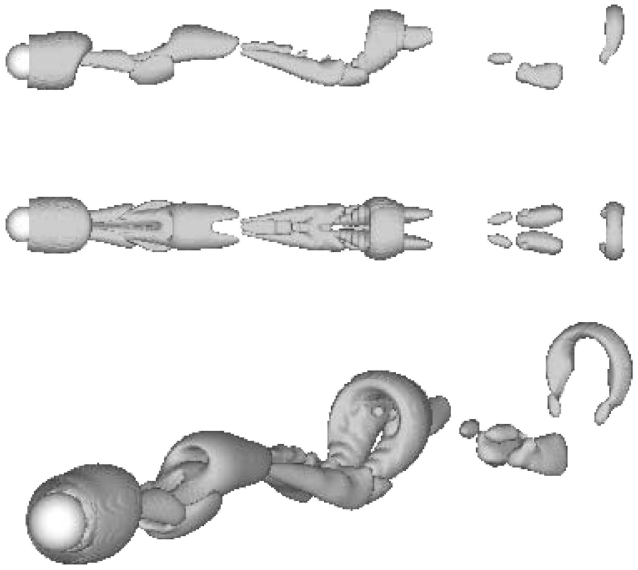

**Fig. 16 Surface plots of the swirl at one instant in time for the flow over a sphere at a Reynolds number of 300. The surface is cut close to the sphere to make the sphere visible.**

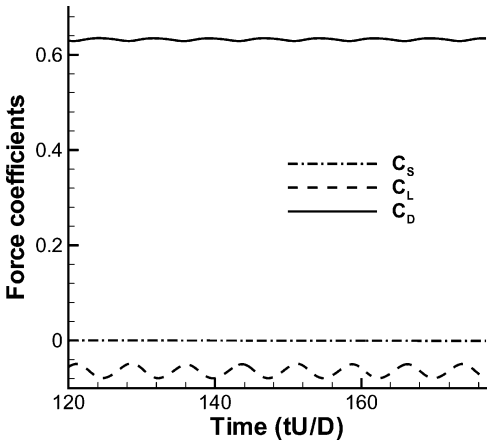


**Fig. 17 Force coefficient history of the flow over a sphere at a Reynolds number of 300.**

to be 0.638 and $-0.065$, respectively, which compare well with the values of 0.656 and $-0.069$ calculated by Johnson and Patel. Computations by Tomboulides et al. found a drag coefficient of 0.671 and experimental results of Roos and Willmarth[27] give an interpolated value of 0.629. The respective oscillation amplitudes of $C_D$ and $C_L$ are found to be $3.5 \times 10^{-3}$ and $1.6 \times 10^{-2}$, and the current simulations give the values $3.0 \times 10^{-3}$ and $1.5 \times 10^{-2}$. Tomboulides et al. found an oscillation amplitude of about $2.8 \times 10^{-3}$ for $C_D$. The vortex shedding Strouhal number $Sr = fD/U_\infty$ was computed to be 0.137 by Johnson and Patel, whereas the current simulations give 0.133. Tomboulides et al. found a Strouhal number of 0.136.

Validation of the method on the flow over a sphere has shown that it can produce results that compare well with other computations and experiments, and that the method performs equally well in three dimensions as in two dimensions. Simulations of the sphere moving through the grid were not performed because a uniform grid in the entire three-dimensional domain were computationally too expensive. However, because there is no difference in the treatment of immersed solid objects in three dimensions vs two dimensions, it is expected to perform comparably.

### D. Flapping Cantilever Beam

Maybe the simplest flapping mechanism that can be used to generate propulsion for flight is that of a flapping cantilever beam. This mechanism has attracted attention because it is also the simplest

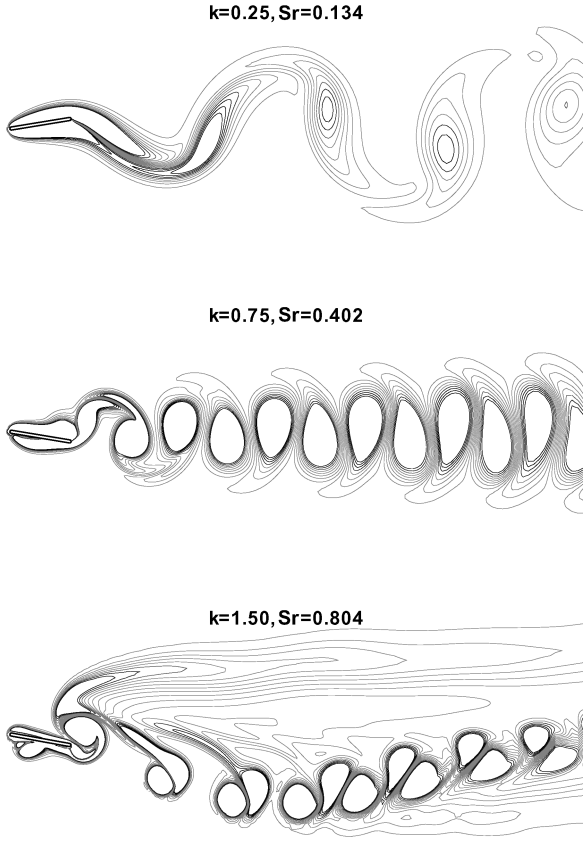


**Fig. 18 Two-dimensional flapping cantilever beam for three different flapping frequencies.**

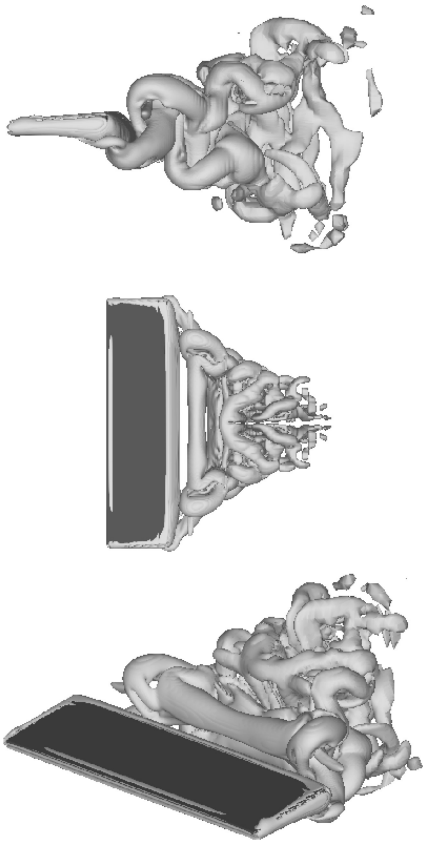

**Fig. 19 Isosurfaces of vortex strength for the three-dimensional flapping cantilever beam.**

mechanism to construct with technologies developed from research on active materials for MAV applications. More information about the materials and kinematics of such a device can be found in the literature.[28,29] Here we present some simulation results from a larger study of the propulsive potential of the cantilever only to demonstrate the flexibility of the current method and to show that it is a very useful tool for its intended purposes. The entire study will be presented in more detail in a separate paper at a later stage.

For two-dimensional simulations, the cantilever has been modeled as a flat-plate airfoil, with rounded ends, flapping about its leading edge. A Reynolds number of 200 was chosen for investigations because it is representative of the Reynolds numbers that we are interested in and it was high enough to produce a persistent wake for the exploration of wake dynamics. Figure 18 shows vorticity contour plots of the vortex wake for low, moderate, and high flapping frequencies and flapping amplitudes of 15 deg. We define a nondimensional frequency as $k = fc/U_\infty$, where $f$ is the flapping frequency and $c$ is the cantilever chord. The low flapping frequency ($k = 0.25$) produces a periodic wake flowfield that is similar to a von Kármán vortex street. This wake structure produces a time-averaged drag force and the cantilever does not produce thrust in this case. At the moderate flapping frequency ($k = 0.75$), the vortices are trailing on a straight line and the cantilever does not produce a significant amount of drag or thrust. For the high-frequency case ($k = 1.50$), the wake takes the form of a deflected inverse von Kármán vortex street, and in this case the cantilever produces thrust. The only thing asymmetric in the simulation is the initial flap direction and, interestingly, changing the initial flap direction will deflect the wake the other way. The deflected wake also results in lift in the direction of the vertical component of the wake direction, and the wake behavior is also observed experimentally and computationally for flapping airfoils at higher Reynolds numbers in studies by Jones et al.[30]

Three-dimensional simulations of a cantilever with finite width show different wake structures than observed for the two-dimensional simulations. Figure 19 shows isosurfaces of vortex structures in a simulation result of a cantilever with an aspect ratio of 4 flapping at a frequency that would give the deflected wake in the two-dimensional case. Instead we can see that vortex structures generated at the edges of the cantilever cause the wake to collapse on itself in the horizontal direction, while it stretches out in the vertical direction in a symmetrical fashion. In other words, the wing-tip effects are very important and a very high aspect ratio is needed to achieve the two-dimensional effects for the three-dimensional cantilever. The simulations indicate that the propulsive efficiency of the cantilever increases with flapping frequency, amplitude, and aspect ratio, but a more detailed study will be left for a separate paper, as mentioned previously.

As a curiosity it can be mentioned that the three-dimensional simulation shown in Fig. 19 is computed with a parallel version of the code on a Linux cluster machine. The computation consisted of 12 million points and was at times computed on 108 Pentium Xeon processors finishing the simulation within two to three days. For efficient solution of the Poisson equation, the code was linked up with the PETSc library,[31,§] which provides a number of different routines for solving partial differential equations in parallel, and in this case a preconditioned GMRES routine was used.

### E. Tumbling Two-Dimensional Leaf

Finally we will present a simulation of a falling and tumbling leaf, modeled as a two-dimensional thin cambered airfoil, purely as another demonstration of the method's flexibility and potential. The simulation was performed on a 2 GHz Pentium Xeon desktop computer and finished in about two hours. A MUDPACK library[32] of very efficient multigrid routines was implemented with the code for the solution of the Poisson equation on a uniform Cartesian grid. Because the leaf was allowed to fall freely, the path could not be predicted, but the $512 \times 512$ domain was allowed to move with the airfoil in a similar fashion that is described earlier for the simulation

[§]Data available online at http://www.mcs.anl.gov/petsc [cited 10 Sept. 2004].

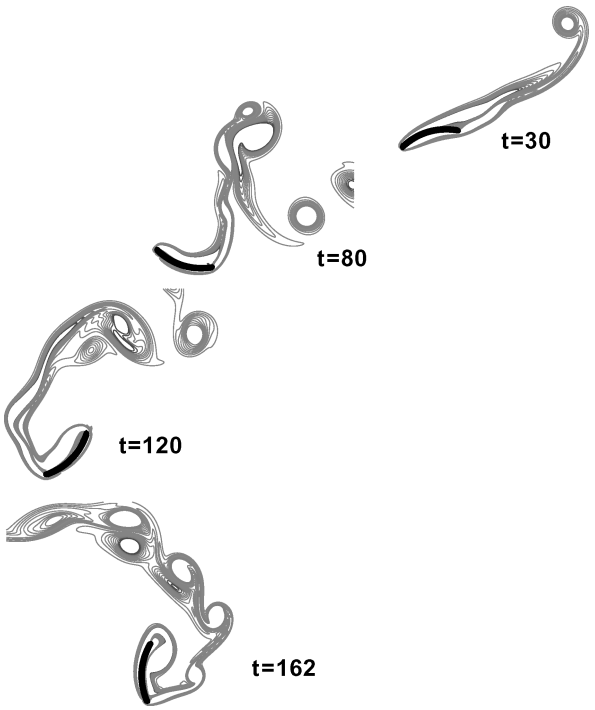


**Fig. 20 Two-dimensional leaf in tumbling free fall.**

of the moving cylinder. The Reynolds number for the simulation is about 1000 and snapshots of the tumbling leaf at different times are shown as vorticity contour plots in Fig. 20.

## IV. Conclusions

A method has been developed for the simulation of moderate-Reynolds-number flows around complex moving solid objects on a uniform Cartesian grid. A very efficient solution algorithm is used for solving the incompressible Navier–Stokes equations with second-order accuracy on a staggered grid.[15] Immersed solid objects are accounted for without altering the computational stencil of the numerical scheme close to the boundaries, which preserves the attractive qualities of a staggered grid approach in the entire computational domain. Grid points occupied by the solid object are given a fluid velocity equal to the solid object but also serve the purpose as ghost points for solving the fluid equations close to the boundary. Because the numerical stencil is not altered to accommodate irregular boundaries on the uniform mesh, the boundary effects only enter through the source term of the elliptic equation, which allows for the easy implementation of fast Poisson solvers for the most efficient solution of the governing equations. Although this approach causes some smearing of the boundary effects, a mesh-refinement study of a cylinder oscillating in a closed chamber shows that the boundary treatment does not significantly degrade the overall accuracy of the numerical scheme.

A validation study of flows over a cylinder and sphere shows that the method accurately computes both steady and unsteady flow dynamics of low- to moderate-Reynolds-number flows. The flow results of a cylinder moving through the grid in a surrounding fluid and a stationary cylinder in a freestream were compared to show that the method performs equally well for each cases. In addition, some simulation results of the flow dynamics of a flapping cantilever propulsion device and a free-falling two-dimensional leaf were presented to show that the method can be a very powerful tool for the investigation of low- to moderate-Reynolds-number unsteady flight aerodynamics. It is therefore demonstrated that our method is well suited for flapping MAV applications.

## Acknowledgments

This work is funded by the Air Force Office of Scientific Research through Multidisciplinary University Research Initiative Grant F49620-98-1-0433 to the University of Minnesota. This work is also sponsored by the Army High Performance Computing Research Center under the auspices of the Department of the Army, Army Research Laboratory cooperative agreement DAAD191-01-2-0014, the content of which does not necessarily reflect the position or the policy of the government, and no official endorsement should be inferred. Computer time was provided by the University of Minnesota Supercomputing Institute.

## References

[1]Hennighausen, K., Candler, G. V., Einarsson, G. O., and Boyd, I. D., "Numerical Simulation of Three Dimensional Flow About a Micro Air Vehicle," AIAA 2000-2615, June 2000.

[2]Peskin, C. S., "Numerical Analysis of Blood Flow in the Heart," *Journal of Computational Physics*, Vol. 25, No. 3, 1977, pp. 220–252.

[3]Nichols, B. D., Hirth, C. W., and Hotchkiss, R. S., "SOLA-VOF: A Solution Algorithm for Transient Fluid Flow with Multiple Free Boundaries," Los Alamos National Lab., Rept. LA-8355, Los Alamos, NM, Aug. 1980.

[4]Hirt, C. W., and Nichols, B. D., "Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries," *Journal of Computational Physics*, Vol. 39, No. 1, 1981, pp. 201–225.

[5]LeVeque, R., "A Large Time Step Generalization of Godunov's Method for Systems of Conservation Laws," *SIAM Journal on Numerical Analysis*, Vol. 22, No. 6, 1985, pp. 1051–1073.

[6]LeVeque, R., "Cartesian Grid Methods for Flow in Irregular Regions," *Numerical Methods in Fluid Dynamics*, edited by K. Morton and M. Baines, Vol. 3, Oxford Univ. Press, Oxford, 1988, pp. 1051–1073.

[7]Berger, M., and LeVeque, R., "An Adaptive Cartesian Mesh Algorithm for the Euler Equations in Arbitrary Geometries," AIAA Paper 89-1930, June 1989.

[8]Berger, M., and LeVeque, R., "A Rotated Difference Scheme for Cartesian Grids in Complex Geometries," AIAA Paper 91-1602, June 1991.

[9]Ye, T., Mittal, R., Udaykumar, H. S., and Shyy, W., "An Accurate Cartesian Grid Method for Viscous Incompressible Flows with Complex Immersed Boundaries," *Journal of Computational Physics*, Vol. 156, No. 2, 1999, pp. 209–240.

[10]Udaykumar, H. S., Mittal, R., Rampunggoon, P., and Khanna, A., "A Sharp Interface Cartesian Grid Method for Simulating Flows with Complex Moving Boundaries," *Journal of Computational Physics*, Vol. 174, No. 1, 2001, pp. 345–380.

[11]Mohd-Yusof, J., "Combined Immersed Boundaries/B-splines Methods for Simulations of Flows in Complex Geometries," *CTR Annual Research Briefs*, NASA Ames, Center for Turbulence Research, Stanford, CA, 1997, pp. 317–327.

[12]Fadlun, E. A., Versicco, R., Orlandi, P., and Mohd-Yusof, J., "Combined Immersed-Boundary Finite-Difference Methods for Three-Dimensional Complex Flow Simulations," *Journal of Computational Physics*, Vol. 161, No. 1, 2000, pp. 35–60.

[13]Kim, J., Kim, D., and Choi, H., "An Immersed-Boundary Finite-Volume Method for Simulations of Flow in Complex Geometries," *Journal of Computational Physics*, Vol. 171, No. 1, 2001, pp. 132–150.

[14]Balaras, E., "Modeling Complex Boundaries Using an External Force Field on Fixed Cartesian Grids in Large-Eddy Simulations," *Computers and Fluids*, Vol. 33, No. 3, 2004, pp. 375–404.

[15]McGrattan, K. B., Baum, H. R., Rehm, R. G., Hamins, A., and Forney, G. P., "Fire Dynamics Simulator—Technical Reference Guide," National Inst. of Standards and Technology, NISTIR 6467, Jan. 2000.

[16]Persillon, H., and Braza, M., "Physical Analysis of the Transition to Turbulence in the Wake of a Circular Cylinder by Three-Dimensional Navier–Stokes Simulation," *Journal of Fluid Mechanics*, Vol. 365, 1998, pp. 23–88.

[17]Blackburn, H. M., and Henderson, R. D., "A Study of Two-Dimensional Flow past an Oscillating Cylinder," *Journal of Fluid Mechanics*, Vol. 385, 1999, pp. 255–286.

[18]Fornberg, B., "A Numerical Study of Steady Viscous Flow past a Circular Cylinder," *Journal of Fluid Dynamics*, Vol. 98, No. 4, 1980, pp. 19–55.

[19]Dennis, S. C. R., and Chang, G.-Z., "Numerical Solution for Steady Flow past a Circular Cylinder at Reynolds Numbers up to 100," *Journal of Fluid Mechanics*, Vol. 42, 1970, p. 471.

[20]Weiselsberger, C., "New Data on the Laws of Fluid Resistance," NACA TN 84, 1922.

[21]Tritton, D. J., "Experiments on the Flow past a Circular Cylinder at Low Reynolds Number, *Journal of Fluid Mechanics*, Vol. 6, 1959, p. 547.

[22]Mittal, R., and Balachandar, S., "Effect of Intrinsic Three-Dimensionality on the Lift and Drag of Nominally Two-Dimensional Cylinders," *Journal of Computational Physics*, Vol. 124, No. 8, 1996, p. 351.

[23]Johnson, T. A., and Patel, V. C., "Flow past a Sphere up to a Reynolds Number of 300," *Journal of Fluid Mechanics*, Vol. 378, 1999, pp. 19–70.

[24]Mittal, R., "A Fourier–Chebyshev Spectral Collocation Method for Simulating Flow past Spheres and Spheroids," *International Journal for Numerical Methods in Fluids*, Vol. 30, No. 7, 1999, pp. 921–937.

[25]Tomboulides, A. G., Orszag, S. A., and Karniadakis, G. E., "Direct and Large-Eddy Simulation of Axisymmetric Wakes," AIAA Paper 93-0546, 1993.

[26]Chong, M. S., Perry, A. E., and Cantwell, B. J., "A General Classification of Three-Dimensional Flow Fields," *Physics of Fluids*, Vol. 2, No. 5, 1990, pp. 765–777.

[27]Roos, F. W., and Willmarth, W. W., "Some Experimental Results on Sphere and Disk Drag," *AIAA Journal*, Vol. 9, No. 2, 1971, pp. 285–291.

[28]James, R. D., and Bhattacharaya, K., "A Theory of Thin Films of Martensitic Materials with Applications to Microactuators," *Journal of the Mechanics and Physics of Solids*, Vol. 47, No. 3, 1999, pp. 531–576.

[29]James, R. D., "Configurational Forces in Magnetism with Application to the Dynamics of a Small-Scale Ferromagnetic Shape Memory Cantilever," *Continuum Mechanics and Thermodynamics*, Vol. 14, No. 1, 2002, pp. 55–86.

[30]Jones, K. D., Lund, T. C., and Platzer, M. F., "Experimental and Computational Investigation of Flapping Wing Propulsion for Micro Air Vehicles," *Fixed and Flapping Wing Aerodynamics for Micro Air Vehicle Applications*, edited by T. J. Mueller, Vol. 195, Progress in Astronautics and Aeronautics, AIAA, Reston, VA, 2001, pp. 307–339.

[31]Balay, S., Buschelman, K., Gropp, W. D., Kaushik, D., Knepley, M., McInnes, L. C., Smith, B. F., and Zhang, H., "PETSc Users Manual," Argonne National Lab., ANL-95/11, rev. 2.1.5, Argonne, IL, 2002.

[32]Adams, J., "MUDPACK: Multigrid Software for Elliptic Partial Differential Equations," URL: http://www.scd.ucar.edu/css/software/mudpack/ [cited 20 Sept. 2004].

C. Kaplan
*Associate Editor*