

# A pressure-correction method and its applications on an unstructured Chimera grid

Xing Zhang<sup>\*</sup>, Saizhen Ni, Guowei He

*The State Key Laboratory of Nonlinear Mechanics (LNM), Institute of Mechanics, Chinese Academy of Sciences, Beijing 100080, PR China*

Received 19 January 2007; received in revised form 16 May 2007; accepted 25 July 2007

Available online 1 November 2007

## Abstract

In this paper, an unstructured Chimera mesh method is used to compute incompressible flow around a rotating body. To implement the pressure correction algorithm on unstructured overlapping sub-grids, a novel interpolation scheme for pressure correction is proposed. This indirect interpolation scheme can ensure a tight coupling of pressure between sub-domains. A moving-mesh finite volume approach is used to treat the rotating sub-domain and the governing equations are formulated in an inertial reference frame. Since the mesh that surrounds the rotating body undergoes only solid body rotation and the background mesh remains stationary, no mesh deformation is encountered in the computation. As a benefit from the utilization of an inertial frame, tensorial transformation for velocity is not needed. Three numerical simulations are successfully performed. They include flow over a fixed circular cylinder, flow over a rotating circular cylinder and flow over a rotating elliptic cylinder. These numerical examples demonstrate the capability of the current scheme in handling moving boundaries. The numerical results are in good agreement with experimental and computational data in literature.

© 2007 Elsevier Ltd. All rights reserved.

## 1. Introduction

Flow around moving boundaries is ubiquitous in engineering applications, such as rotating machinery, flapping wing micro-air vehicle (MAV) and particulate flows in chemical process. Efficient algorithm to handle moving boundaries is still one major challenge in computational fluid dynamics (CFD).

The methodology to tackle this problem can be categorized as three types. The first type is based on body conformal mesh and arbitrary-Lagrangian–Eulerian (ALE) algorithm. In this approach, the Navier–Stokes (NS) equation is solved on a mesh conforming to the moving boundary throughout the computation. At the moving boundaries, velocities of the mesh points are determined by the dynamics of the object, whereas at other computa-

tional (fixed) boundaries they are set to zero. Elsewhere in between, the mesh velocity is arbitrarily chosen so as to absorb the mesh deformation properly. Unstructured mesh has gained its popularity since the early 1990s because of its simplicity in mesh generation around complex objects. An extra strength of the unstructured grid lies in the implementation of the moving-mesh and/or mesh-adaptive schemes. In conjunction with ALE method, smoothing techniques have been developed for unstructured mesh to avoid re-meshing or at least prevent it from occurring too frequently when the motion of the moving boundary becomes large. Nevertheless, re-meshing is unavoidable when dealing with certain kind of problem such as the rotation of blades. Under such circumstance, mesh-related operations are very computational intensive and sometime also very tedious to implement, especially in a three-dimensional situation. A comprehensive review on the development of unstructured techniques is presented in [1]. More detailed discussions on the mathematical formulation and implementation issues can be found in [2]. There are also some recent works on high-order

<sup>\*</sup> Corresponding author. Tel.: +86 10 62545533x3185; fax: +86 10 62579511.

E-mail addresses: [zhangx@lnm.imech.ac.cn](mailto:zhangx@lnm.imech.ac.cn) (X. Zhang), [nisaizhen@lnm.imech.ac.cn](mailto:nisaizhen@lnm.imech.ac.cn) (S. Ni), [hgw@lnm.imech.ac.cn](mailto:hgw@lnm.imech.ac.cn) (G. He).

unstructured method, e.g. [3] by Sun et al. For dynamic unstructured mesh method, please refer to [4,5]. For mesh smoothing techniques, please refer to [6,7].

The second type of method is based on regular Cartesian mesh. When an object of arbitrary shape is immersed in a rectangular computational domain, in order to impose a non-slip boundary condition, one can either modify the governing equation to include a ‘forcing’ term or modify the grid cell when it intersects with the boundary of the object. The former sub-type is termed as ‘immersed boundary’ technique, whereas the latter one is known as ‘cut-cell Cartesian’ method. Comprehensive reviews on the Cartesian grid method are given by Peskin [8] and Mittal and Iaccarino [9].

The third type is ‘Chimera’ grid, or overset grid method. It is based on the concept of domain decomposition by Steger [10] and Berriz et al. [11]. The central idea behind it is the use of several meshes instead of only one mesh in the first two types of methods. In the method, sub-domains are meshed independently and governing equations are also solved separately on them. Actually, it can extend the capability of an existing solver to treat moving boundaries with very little code-development effort. The complexity of this type method lies in the data interchange between different sub-domains. The method proposed in the paper falls into this category.

It should be emphasized that the three methods aforementioned can be used to treat arbitrary motion of boundaries. If only some particular types of motion are considered, one can resort to some techniques that are non-generalizable. For example, if only one moving part is considered, a non-inertial reference frame which moves with the solid body can be used in the formulation and subsequent solution of the Navier–Stokes equation. Another method exclusively designed for rotation is the ‘shear-slip mesh update’ method proposed by Behr and Tezduyar [12]. Actually, this method is very similar to the overset grid method in handling moving boundaries. The only difference is that it uses a simple ‘*shear-slip layer*’ (which includes very few elements) to connect the two independent meshes in lieu of the overlapping region in the overset method. In this method, a local re-meshing in the shear-slip layer is needed. Although these methods are effective in dealing with certain problems, their limitations are also obvious. The non-inertial frame approach cannot handle problem that involves multiple parts with relative motions; whereas the ‘shear-slip mesh update’ method fails in the presence of irregular boundary motions.

The Chimera grid method was originally used only on structured (curvilinear) meshes. Abundant literatures can be found regarding the application and improvement of this method, e.g. [13,14]. It seems that the idea of combining unstructured grid with Chimera grid is not very intuitive. A first impression is that this combination at least cannot incorporate the best aspects of both methods. On an unstructured grid, it is easier to implement mesh-smoothing algorithm but this feature is not needed in the

Chimera grid method. Furthermore, unstructured mesh makes it difficult to use high-order schemes in solving the Navier–Stokes equation. However, after further investigation, one may find that this impression is not necessarily correct. Unstructured Chimera grid method is not only possible but also of great potential in tackling practical problems. In a problem which involves both moving boundary and complex geometry, since a structured solver relies on multi-block technique, the number of domain required in a traditional (structured) Chimera method becomes fairly large. Thus the time required in the interior boundary locating, link-building and data exchanging also increases. The use of unstructured Chimera grid can reduce the time consumption significantly by the reduction of domain number. Generally speaking however, the use of Chimera grid in the framework of unstructured grid is relatively unexplored and only a few literatures can be found. In [15,16], a realistic CFD simulation is performed of an insect in flight. The unstructured grid allows a precise representation of an insect including antennas, legs and a sting. The Chimera grid method is used to treat the flapping wing motion including translation and rotation. In these papers, the authors solve a *compressible* NS equation by finite volume method (FVM). Another example is presented by Maruoka [17], in which *incompressible* NS equation is solved to compute flow around a rotating body using unstructured Chimera grid. His computation is based on *finite element method* (FEM) and the moving part is solved in a non-inertial frame. Similar studies are performed by Houzeaux and Codian [18,19], both of them are based on FEM and a non-inertial reference frame.

In the CFD field, *finite volume* based *pressure-correction* method is widely used in simulating *incompressible* flows. The core of many famous packages (both commercial and in-house) that target at industrial application is an *unstructured* solver that makes use of this scheme. However, the implementing of Chimera grid method on such kind of solver is rarely reported in academic papers at least to the knowledge of the authors of this paper.

In this paper, a well-known pressure-correction scheme – SIMPLEC is modified and implemented on unstructured Chimera mesh. The governing equations are solved in an *inertial frame* with *moving control volume* approach. This technique is then used to simulate flow over a rotating body. Although other methods may be equally effective in solving this type of problem, the method proposed in this paper can be extended to tackle more challenge cases such as free-falling objects and heavy particles in fluid. In our future plan, this method will be applied to study such complex problems.

This paper is organized into four sections. In the second section, the numerical discretization procedure will be presented. It includes the pressure-correction scheme, the moving-mesh algorithm and the interior boundary treatment. In the third section, flow over a stationary circular cylinder is first presented as a numerical validation. Then other two examples of flow over rotating body are fol-

lowed. They include flow over a rotating circular cylinder and a rotating elliptic cylinder. A summary is presented in the last section.

## 2. Numerical methodologies

### 2.1. Navier–Stokes equations in an integral form

We consider a two-dimensional unsteady laminar flow in this study. The continuity and momentum equations used in the simulation can be written in an integral form as

$$\frac{\partial}{\partial t} \int_V \rho dV + \int_{\partial S} \rho(\mathbf{v} - \mathbf{v}_g) \cdot \mathbf{n} dS = 0 \quad (1)$$

$$\begin{aligned} \frac{\partial}{\partial t} \int_V \rho \mathbf{v} dV + \int_{\partial S} \left[ \rho \mathbf{v}(\mathbf{v} - \mathbf{v}_g) \cdot \mathbf{n} - \mu \left( \frac{\partial \mathbf{v}}{\partial n} \right) \right] dS \\ = - \int_{\partial S} p \mathbf{n} dS \end{aligned} \quad (2)$$

where  $t$  is time,  $\rho$  the density of the fluid,  $\mathbf{v}$  the velocity vector,  $p$  the pressure and  $\mu$  the dynamic viscosity.  $\mathbf{v}_g$  is the velocity vector of a moving-mesh.  $dS$  and  $dV$  are the surface area and volume of a control volume, respectively.  $\mathbf{n}$  denotes the unit out-normal vector on the control surfaces. Since the problem involves moving control surface, the space conservation law (SCL) also has to be satisfied, i.e.

$$\frac{\partial}{\partial t} \int_V dV - \int_{\partial S} \mathbf{v}_g \cdot \mathbf{n} dS = 0 \quad (3)$$

Variables in (1)–(3) can be non-dimensionalized by a characteristic length  $L$ , a characteristic velocity  $U_\infty$  and a reference density  $\rho_\infty$ , i.e.

$$\begin{aligned} \tilde{\mathbf{x}} = \mathbf{x}/L; \quad \tilde{t} = tU_\infty/L; \quad \tilde{\mathbf{v}} = \mathbf{v}/U_\infty; \quad \tilde{\mathbf{v}}_g = \mathbf{v}_g/U_\infty; \\ \tilde{p} = p/(\rho_\infty U_\infty^2); \quad \tilde{\rho} = \rho/\rho_\infty \end{aligned} \quad (4)$$

The governing equations for the dimensionless variables are in the same form as (1)–(3) with  $\mu$  replaced by  $1/Re$ , where  $Re$  is the Reynolds number which is defined as

$$Re = \frac{\rho_\infty U_\infty L}{\mu} \quad (5)$$

Since only incompressible fluid is considered in this paper, the dimensionless density  $\tilde{\rho}$  equals to unity.

### 2.2. Temporal and spatial discretization

To solve the NS equation numerically, the method described in [20,21] is followed with minor modifications. This method was originally implemented on a block-structured mesh and later generalized to unstructured mesh. The discretization procedure will only be summarized briefly here, for the details, please refer to those papers. In the current implementation, a SIMPLEC algorithm is used to couple the pressure with the velocity. A second-order upwind scheme is used for the discretization of convective term and the Crank–Nicholson scheme (second order in time) is used for the temporal advancement.

In order to obtain a discrete solution of Eqs. (1) and (2), the computational domain is discretized into finite number of contiguous control volumes (CVs) or cells. Cells with arbitrary shapes are permitted in this method. Hybrid mesh that consists of triangular and quadrilateral cells will be used in this study. All dependent variables are stored at the centroid of cells, i.e. a collocated arrangement is adopted. To explain the discretization procedure of the momentum equation (2) term by term, we first re-formulate it into a transport equation for an arbitrary variable  $\phi$ :

$$\frac{\partial}{\partial t} \int_V \phi dV + \int_S [\phi(\mathbf{v} - \mathbf{v}_g) - \Gamma_\phi \nabla \phi] \cdot \mathbf{n} dS = \int_S Q_f dS \quad (6)$$

Transient term    Convective term    Diffusive term    Surface source

where  $\phi$  represents the velocity components  $v_i$  ( $i = 1, 2$ ) and  $\Gamma_\phi$  is the coefficient of diffusivity ( $1/Re$ ).

This time-dependent equation is discretized by the Crank–Nicholson scheme (second order in time):

$$\begin{aligned} \frac{1}{\Delta t} [(\phi V)^{n+1,k} - (\phi V)^n] = \frac{1}{2} [D^{n+1,k}(\phi) + D^n(\phi)] + \frac{1}{2} \\ \times [C^{n+1,k-1}(\phi) + C^n(\phi)] \\ + \left( \int_S Q_f dS \right)^n \end{aligned} \quad (7)$$

where ‘ $n$ ’ and ‘ $n + 1$ ’ are the time step counters;  $C$  and  $D$  denote the convective and diffusive terms, respectively. The source term is related to the pressure gradient at time step ‘ $n$ ’. ‘ $k$ ’ is the counter of the inner iteration regarding pressure–velocity coupling (this so called ‘SIMPLEC’ loop will be discussed later).

The convective term is discretized as

$$\int_S \phi(\mathbf{v} - \mathbf{v}_g) \cdot \mathbf{n} dS \approx \sum_j \dot{m}_j \tilde{\phi}_j \quad (8)$$

where  $\dot{m}_j$  is the volume flux across face  $j$ . It is computed by  $\dot{m}_j = A_j(\tilde{\mathbf{v}}_j - \mathbf{v}_g) \cdot \mathbf{n}_j$  (9)

where  $A_j$  is the area of face  $j$ ;  $\tilde{\mathbf{v}}_j$  is the face velocity;  $\tilde{\phi}_j$  in (8) is the variable interpolated to face  $j$  by a blended scheme

$$\tilde{\phi}_j = \phi_j^{(1)} + \gamma_\phi(\phi_j^{(2)} - \phi_j^{(1)}) \quad (10)$$

where the superscript ‘ $(1)$ ’ and ‘ $(2)$ ’ denote first-order and second-order interpolation, respectively. The first-order interpolation is just a simple ‘upwind’ scheme. In a second-order scheme, we first use the gradient of variable  $\phi$  and Taylor expansion to evaluate the value of  $\phi$  on the face centers from either side, then the one from the ‘upwind’ direction is chosen as the value for that particular face. The gradient of  $\phi$  is constructed using a linear least-square approach.  $\gamma_\phi$  in (10) is a blending factor which is set to 1.0 in this paper. The first term on the right-hand side of (10) is treated implicitly and the second term is treated explicitly as a source term, i.e. a *second-order upwind* scheme is implemented through a ‘deferred correction’ in this paper.

The diffusive term is discretized as

$$\int_S -\Gamma_\phi \nabla \phi \cdot \mathbf{n} dS \approx \sum_j -\Gamma \phi_j \frac{A_j}{L_j} ((\phi_{P_j} - \phi_{P_0}) + [(\nabla \phi)_{P_j} \cdot \boldsymbol{\tau}_1 - (\nabla \phi)_{P_0} \cdot \boldsymbol{\tau}_2]) \quad (11)$$

where  $L_j$  is the distance from the center of cell  $P_0$  to that of cell  $P_j$  projected to the normal direction of face  $j$ .  $\boldsymbol{\tau}_1$  and  $\boldsymbol{\tau}_2$  are two vectors in the tangential direction of face  $j$  (see Fig. 1). The first term on the right-hand side of (11) is the ‘normal diffusion’ and the second term is the ‘cross diffusion’ which is a correction for non-orthogonal meshes.

The pressure gradient term is treated as a surface source

$$\int_S Q_\tau dS \approx \sum_j -p_\tau A_\tau n_i \quad (12)$$

A linear system is obtained as a result of the discretization of the momentum equation:

$$a_{P_0}^C \phi_{P_0} = \sum_j^{nb\_cells} a_{P_0}^j \phi_{P_0}^j + b_{P_0} \quad (13)$$

where superscript ‘C’ denotes the diagonal elements of the coefficient matrix (related to cell  $P_0$ ) and ‘j’ denotes the non-diagonal elements of the matrix (related to the neighboring cells of  $P_0$ ). ‘nb\_cells’ denotes that the summation is done on all neighboring cells of  $P_0$ . The contributions to the coefficient matrix are the mass matrix, ‘upwind’ part of the convective term and the ‘normal diffusion’. The source term  $b_{P_0}$  in (13) has three contributions, the pressure gradient, the ‘cross diffusion’ and the departures of the convective flux from the upwind differencing (deferred correction).

In the SIMPLEC algorithm, the pressure-correction equation is derived from the continuity Eq. (1)

$$\sum_j^{cell\_faces} \left( \frac{1}{a_{P_0}^j} \right)_j (\nabla p')_j \cdot A_j \mathbf{n}_j = \sum_j^{cell\_faces} \dot{m}_j \quad (14)$$

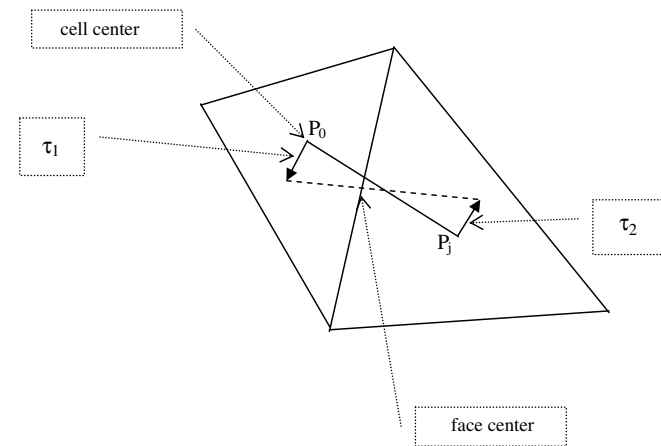


Fig. 1. Stencil used in the discretization. The dashed line is perpendicular to the cell face. It is not necessarily parallel to the line connecting cell  $P_0$  and cell  $P_1$ .

where  $p'$  is the pressure correction; ‘.’ stands for the arithmetic averaging from cell to face. ‘cell\_faces’ indicates that the summation is done on all faces that belong to cell  $P_0$ . The coefficient on the left-hand side of (14) is defined as

$$a_{P_0}^C = a_{P_0}^C - \sum_j^{nb\_cells} a_{P_0}^j \quad (15)$$

where the summation is done on all neighboring cells. A discretized equation of (14) can be written in a generic form very similar to (13). The Laplacian operator on the left-hand side of (14) is also treated similarly as the diffusive term in the momentum equation. Correction for the mesh non-orthogonality also has to be considered.

After the pressure correction  $p'$  is obtained from (14), the pressure and velocity are corrected by

$$p^m = p^{m-1} + p'^m$$

$$\mathbf{v}^m = \mathbf{v}^{m-1} - \frac{1}{a_{P_0}^C} \sum_j^{cell\_faces} p_j^m A_j \mathbf{n}_j \quad (16)$$

where  $m = 1, 2, \dots$  is the pressure-correction loop counter at time step  $n$ .  $P^0$  is equivalent to the pressure value at time step  $n$ . When it converges, the value at time step  $n + 1$  is obtained for both the pressure and the velocity.

After the correction of (16), the coefficient matrix and source term in (13) are computed using the updated value of  $p$  and  $\mathbf{v}$ . A new velocity is then obtained by solving (13) again. This new velocity is then substituted into (14) to compute a new pressure correction. The pressure-correction iteration is repeated until the convergence criterion is satisfied.

For the momentum equation and pressure-correction equation which can be expressed in a matrix form as  $\mathbf{Ax} = b$ , it is required that the residuals to be reduced by two orders each time the iterative solver is called. Two convergence criteria have to be met before the SIMPLEC loop ends. The first one is for the momentum equation, i.e.

$$\frac{\|b - \mathbf{Ax}\|}{\max(\|v_1\|, \|v_2\|)} \leq 1.0 \times 10^{-8} \quad (17)$$

where  $v_1$  and  $v_2$  stand for the two components of the velocity vector, respectively. The second one is for the divergence-free condition, i.e.

$$\left( \frac{1}{V_{cell}} \left| \sum_j^{cell\_faces} \dot{m}_j \right| \right)_{max} \leq 1.0 \times 10^{-4} \quad (18)$$

The selection of these criteria is based on experiences and parameters found in published literatures.

Attentions should be paid to the face velocity  $\bar{v}_j$  in (9) that is used to calculate the volume flux. This velocity cannot be approximated by an arithmetic average of the values in the neighboring cells. Instead, a Rhie–Chow interpolation, which introduces some dependency on the pressure, is used. More details of this interpolation can be found in [20].

Iterative methods are implemented to solve the algebraic equations. For the momentum equation, a generalized

conjugate residue (GCR) solver is used whereas for the pressure-correction equation, a conjugate gradient (CG) solver is used. A Jacobi pre-conditioner is applied to both solvers to speed up the convergence.

The ‘order of accuracy’ in the aforementioned discretization needs to be addressed here. The so called ‘second order’ in space is only valid on a uniformed mesh (e.g. equilateral triangles), which is not possible in the numerical simulation of this paper. On a mesh which is neither homogeneous nor orthogonal, the effective order of accuracy is lower than its idealistic value. The *overall* order of accuracy for this numerical scheme is more complicated since the split error in the pressure-correction method also has to be taken into account. The authors of this paper believe that (but with no rigorous proof) for the velocity field, the spatial accuracy is between first and second order and the temporal accuracy is of second order. As to the conservation property, finite volume method is built to conserve mass and momentum. However, in reality, mass and momentum are only conserved *approximately* (and *not* exactly). The reason behind this is that for the pressure-correction method such as SIMPLEC, it is very hard to satisfy the divergence-free condition exactly (to machine precision).

### 2.3. Moving-mesh and space conservation law

As it is stated in the introductory section, either inertial or non-inertial reference frame can be used to treat the problems with moving parts. In the implementation of Chimera method in [17–19], both frames are used, inertial frame for the stationary domain and non-inertial frame for the moving domain. Thus in these simulations, a tensorial transformation has to be performed before the velocities are interpolated on the interior boundaries. In this paper, we pursued an alternative strategy which uses only the inertial frame in conjunction with moving control volumes. The advantages of the current method are twofold. First, by the introduction of mesh velocity, the conservative formulation of the Navier–Stokes equations is preserved without any additional source term; second, the tensorial transformation is not needed. In the moving control volume approach, the space conservation law (SCL) in (3) has to be satisfied throughout the computation; otherwise non-physical mass source will be introduced to the system. The discretized form of (3) can be written as

$$\frac{V_c^{n+1} - V_c^n}{\Delta t} = \sum_f^{\text{cell\_face}} (\mathbf{v}_g \cdot \mathbf{n}) A_f \quad (19)$$

However, this equation is not solved in the computation but is a constraint on the mesh velocity and should be satisfied exactly. In order to be compatible with the order of accuracy in the spatial and temporal discretization, the approach suggested in [4] is used to calculate the grid velocity. Please note that on each face of a 2D cell (triangular or quadrilateral), the face normal vector and area (length)

vary with time. The mesh velocity is defined as a *constant* within the time interval of  $\Delta t$  but varies linearly in space (see Fig. 2). The product of the face normal and face area are assumed to vary linearly in time. Thus, on each face, the mass (volume) flux due to the mesh movement can be approximated by

$$(\mathbf{v}_g \cdot \mathbf{n}) A_f \approx \frac{1}{2} (\mathbf{v}_{gN_1} + \mathbf{v}_{gN_2}) \cdot \frac{1}{2} (A_f^{n+1} \mathbf{n}_f^{n+1} + A_f^n \mathbf{n}_f^n) \quad (20)$$

The resultant flux  $\dot{m}_f$  on each face in Eq. (9) is computed using the grid mesh velocity obtained from (20). Based on the mesh size and time step used in this paper, the mesh velocity (flux) calculated from Eq. (20) satisfies the conservation constraint (19) *almost* to machine precision. This conclusion is drawn from numerical tests.

### 2.4. Boundary conditions

To solve the Navier–Stokes equation, boundary and initial conditions have to be provided. Initial condition is easy to specify. For incompressible fluid, divergence-free condition is the only constraint. In this paper, a constant velocity and pressure field is used as the initial condition. Boundary condition is more complicated, and sometimes can be the key factor to an accurate numerical simulation. Some commonly used boundary conditions are summarized as follows. Here we only discuss physical and far-field boundaries (those truncated for the purpose of computation), whereas interior boundaries that are introduced in the Chimera grid method will be discussed in the next section.

#### (a) The boundary where velocity is prescribed

On this type of boundary, the normal velocity component is specified. As to the tangential component of velocity, one can either specify it (such as on a non-slip wall), or its normal gradient (such as on a slip wall). The normal gradient of pressure correction is set to zero on this type of boundary, i.e.

$$\frac{\partial p'}{\partial n} = 0 \quad (21)$$

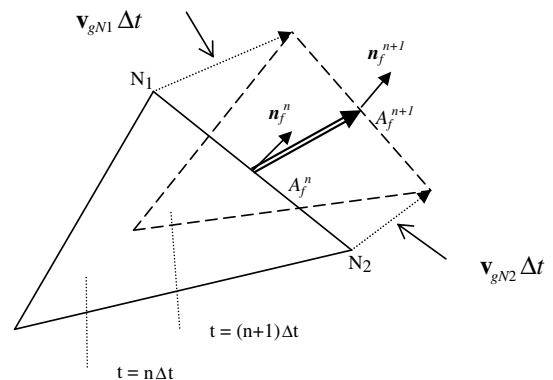


Fig. 2. Moving control volume and the mesh velocity that satisfies SCL.

In some references, it is claimed that the boundary condition (21) is not needed at all. When the velocity is prescribed, the given velocity can be directly substituted into the pressure-correction equation. The authors of this paper believe that these two viewpoints are consistent with each other at least in a discrete sense. The direct substitution of a given velocity on some faces is equivalent to the *exclusion* of the contribution from these faces when discretizing the Laplacian operator in the pressure-correction equation. Apparently, this is equivalent to applying a Neumann boundary condition on these faces.

(b) *The boundary where pressure is prescribed*

On this type of boundary, pressure is set to a constant (usually zero). In order to satisfy the continuity constraint, the velocity is corrected using the gradient of pressure correction  $p'$  in such a manner similar to (16). The implementation of this boundary condition is addressed in detail by Ferziger and Peric [22]. It is appropriate to apply such condition on an outlet which is sufficiently far from the inlet. Since the pressure is never corrected on this boundary, the following condition is specified for the pressure correction:

$$p' = 0 \quad (22)$$

(c) *The convective boundary*

This type of condition is exclusively designed for outlets to suppress reflections. The main purpose of this boundary condition is to minimize the distortion of vortices when passing through such boundaries. Usually, velocity components are obtained by a *simplified* one-dimensional NS equation. Pressure gradient is also set to zero on this type of boundary, same as that of boundary type (a). Since global mass conservation is not necessarily guaranteed by applying such condition, usually the velocities are scaled to satisfy the mass conservation globally. The convective boundary condition can be expressed in a mathematical formula as

$$\begin{aligned} \frac{\partial \phi}{\partial t} + U_{\text{conv}} \frac{\partial \phi}{\partial n} &= 0 \\ \frac{\partial p'}{\partial n} &= 0 \end{aligned} \quad (23)$$

where  $\phi$  represents the velocity components  $v_i$  ( $i = 1, 2$ ) and  $U_{\text{conv}}$  is the convective velocity at the outlet. In the case of uniform inflow,  $U_{\text{conv}}$  is equivalent to the inlet velocity. The details regarding the implementation of this boundary condition can be found in [22]. Throughout this paper, this type of condition is applied to the outlet.

sub-domains. To exchange data on the interfaces of these domains, inter-grid links have to be built after the grids are generated independently. The ‘link-building’ procedure is to identify the donor cell for each ghost cell (on which boundary condition is imposed) of the interior boundaries (see Fig. 3). Since no *irregular* mesh motion is involved in this paper, a repetitive ‘hole-cutting’ is not needed. In this paper, a ‘neighbor to neighbor’ searching scheme is used to build the links. This algorithm is very efficient compared with a complete search. For the details of this method, please refer to [15].

2.5.2. *Data transfer on the interior boundaries*

A key step in the pressure-correction scheme on a Chimera grid is to impose correct boundary condition on the interior boundaries. These boundaries are neither physical ones nor the ones artificially truncated for computational purpose, thus the boundary condition on them is quite different from those described in Section 2.4. A Schwarz alternating procedure is followed to couple the solutions from individual component grids. The Schwarz method states that the data between sub-domains that overlapping each other are exchanged in the form of Dirichlet condition on the interior boundaries. Here the data on these boundaries are obtained through an interpolation from the solution of the donor cells.

For one single PDE, the implementation of Schwarz alternating procedure is straightforward. However, the incompressible Navier–Stokes is a system of PDEs that consists of the momentum and the continuity equations. Moreover, the pressure-correction algorithm described in Section 2.2 has introduced more auxiliary variables besides the primary ones (velocity and pressure), such as the pressure correction  $p'$  and mass flux  $\dot{m}_f$ , etc. This makes the situation more complicated. In order to do it in a consistent

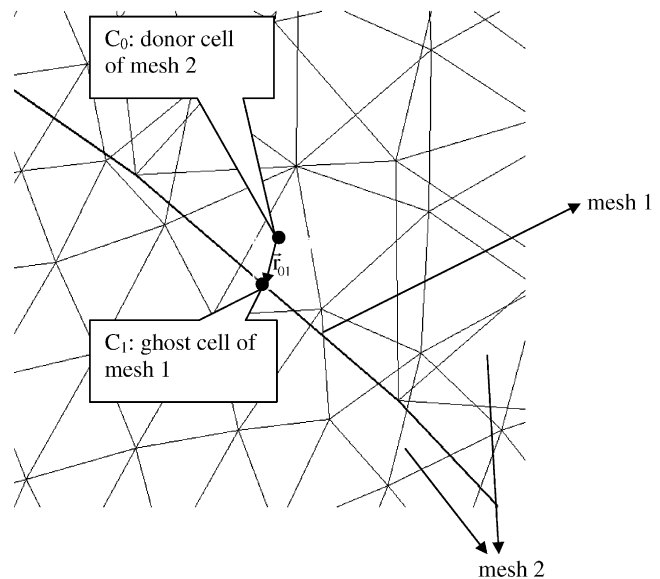


Fig. 3. The interior boundary where data is transferred.  $C_0$ : donor cell of mesh 2;  $C_1$ : ghost cell of mesh 1.

2.5. Implementation issues related to the Chimera grid

2.5.1. Construction of inter-grid links

In order to implement the pressure-correction scheme on a Chimera grid, information transfer is needed between

manner, one has to be very careful when interpolating the auxiliary variables. Auxiliary variables should be interpolated and exchanged when necessary, but these exchanges of information could lead to the decoupling of primary variables (velocity and pressure).

Since the data transfer between sub-domains are done iteratively within one time step. This iterative procedure can be incorporated into the SIMPLEC loop where the velocity–pressure coupling is achieved. The major steps in this algorithm are summarized as follows:

- (1) Read mesh connectivity, compute geometry and build inter-grid links for later use.
- (2) Start the computing from an initial guess of pressure and velocity.
- (3) Solve the momentum equation based on the guessed pressure field. The velocity values on the interior boundary (ghost cells) are obtained by interpolating from their donor cells. This interpolation can be expressed in a mathematical formula as (see Fig. 3)

$$\phi_i^k|_{C_1} = \phi_i^{k-1}|_{C_0} + (\nabla\phi_i)^{k-1}|_{C_0} \cdot \vec{r}_{01} \quad (24)$$

where  $i = 1, 2$ ;  $\phi_i$  denotes the two velocity components and  $k$  is the inner iteration counter.

- (4) Solve the pressure-correction equation. The pressure-correction values on the interior boundary (ghost cells) are determined by the following two-steps:
  - a. A temporary pressure values on these cells are obtained by interpolating from their donor cells.
  - b. Subtract the values of current pressure from the temporary pressure and the result is treated as a Dirichlet condition prescribed on the ghost cells for the pressure-correction equation.

This interpolation can be expressed in a mathematical formula as

$$\begin{aligned} \tilde{p}^k|_{C_1} &= p^{k-1}|_{C_0} + (\nabla p)^{k-1}|_{C_0} \cdot \vec{r}_{01} \\ p^k|_{C_1} &= \tilde{p}^k|_{C_1} - p^{k-1}|_{C_1} \end{aligned} \quad (25)$$

- (5) Correct velocity and pressure, respectively.
- (6) Check if the convergence criteria are met. If not, replace the initial values of pressure and velocity with the current one and go to (3); if yes, then exit from this iteration and advance to the next time step.
- (7) Repeat (3)–(6) until the terminating time is reached.

In step (3), the transfer of velocity  $\mathbf{v}$  between sub-domains is done by using a direct interpolation. However, the situation is more complicated in step (4) regarding the transfer of *pressure correction*  $p'$ . Through numerical test, it is found that a direct interpolation of  $p'$  only works well for *steady* flows (such as a lid-driven cavity). It will lead to a decoupling of pressure in the overlapping region when unsteady flow (such as the vortex shedding) is computed. The authors believe that the failure of the direct interpolation approach is caused by the accumulation of error in the solution of  $p'$ . The remedy to this is proposed in step (4)

and is verified by the numerical examples in the next section. The philosophy behind this method is to ‘glue’ the pressures of different sub-domains to each other more tightly. A pressure-correction value is assigned to the interior boundary (ghost cells) such that after this correction, the pressure on that boundary will reach the value obtained by interpolation (using the latest pressure and pressure gradient) from its donor cell which belongs to another sub-domain. A discussion on this phenomenon and our remedy is presented in [23] by the first author.

The process of information exchange between sub-domains is summarized as follows. Eqs. (24) and (25) are applied to all interior boundaries (such as  $\Gamma_{12}$  and  $\Gamma_{21}$  in Fig. 4). The solution of sub-domain  $\Omega_1$  is obtained based on the boundary information interpolated from sub-domain  $\Omega_2$  and vice versa. The data exchange is done through a Schwarz alternating procedure.

For Chimera grid method, global conservation is an important issue that should be addressed. For incompressible flows, global mass conservation is a necessary condition for the existence and uniqueness of a smooth pressure field. Thus it is crucial for any numerical scheme that solves the incompressible NS equation to conserve mass. Since the treatment of the interior boundary condition aforementioned is non-conservative in nature, it seems that some corrections are needed. However, after further investigation, it is found that such correction is not necessary in the modified pressure-correction algorithm proposed in this paper. These interior boundaries are very similar to those where pressure is prescribed (see category b of Section 2.3). The only difference is that the pressure correction is set to some *non-zero* value (which varies with the inner iteration number  $m$ ) to ‘glue’ the pressure field of different sub-mains together on the internal boundaries. On these boundaries, the *mass flux*  $\dot{m}_f$  is corrected iteratively until the convergence criterion is met. It is seen that in this algorithm, mass conservation is satisfied approximately both locally and globally everywhere in computational domain (if the convergence criterion is stringent enough). The only remaining issue is that whether the *velocity vector* computed by interpolation is compatible with this mass

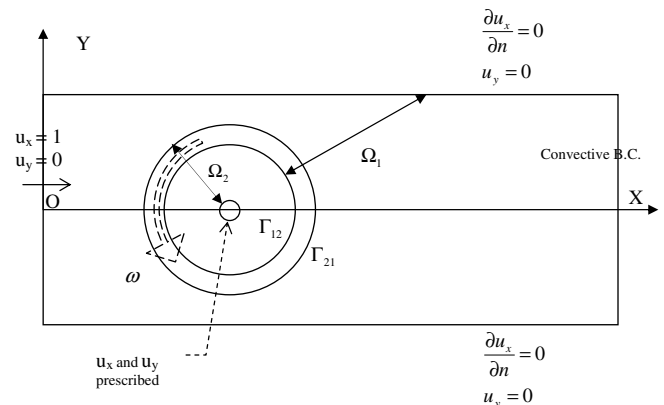


Fig. 4. Domain decomposition and boundary conditions for the simulation of the flow over a circular cylinder.

flux. From our experiences on the numerical test conducted in this paper, the answer is yes. Non-physical ‘boundary layer’ of velocity is not observed near these interior boundaries.

The order of accuracy in the interpolation is another important issue for numerical methods on Chimera grid. In the traditional (structured) Chimera methods (such as [13,14]), it is very common that the order of accuracy in the interpolation is higher than that in the discretization of the Navier–Stokes equations. In this paper, however, these two are of the same order (second order). There are several reasons for our choice. First, the cost of quadratic reconstruction is much higher in terms of CPU time and memory. In the framework of unstructured finite volume method, it requires the connectivity information of the ‘second-level’ neighbors. Thus the matrix which is inverted on each cell has a larger size (3 by 3 for linear and 5 by 5 for quadratic reconstruction for 2D problems). Second, for unstructured mesh, it is very easy to refine the mesh locally near the overlapping region with only a minor increase in the total cell number. Third, through numerical tests, it is shown that no significant improvement on the solution quality in the overlapping region is observed by simply replacing the linear reconstruction with a quadratic reconstruction in the interpolation of velocity and pressure correction (Eqs. (24) and (25)).

The issue of overlapping length also needs to be addressed here. It is reported in some references (such as [24]) that the convergence of Schwarz alternating method *does* depend on the overlapping length. In all simulations conducted in this paper, the width of the ‘ring-shaped’ overlapping region is approximately 0.45 (which is about the five grid size locally). Our major concern regarding the overlapping length is the stencil which is used in the interpolation. For a least-square reconstruction, a layer of three grid points is enough; for a quadratic reconstruction it is safe to have a layer of five grid points. We did not find any problem with the current length of five points in all subsequent simulations. As to the influence of the overlapping length on the convergence, more investigations are still needed.

### 3. Numerical experiments

#### 3.1. Flow over a stationary circular cylinder

The first example is laminar flow over a stationary circular cylinder. The Reynolds number of this simulation is based on the diameter of the cylinder  $D$  and the inlet velocity. This problem is inherently unsteady when the Reynolds number is larger than a critical value of approximately 50. For simulations at a  $Re$  number higher than the critical value, unsteady vortex shedding can be triggered by machine error alone even if the boundary conditions are perfectly symmetric and no artificial perturbation is added to the initial condition. Simulations are performed at

$Re = 100$  and  $200$ . Vortices that shed periodically from the cylinder have to pass through the interior (interpolation) boundaries of the Chimera grids. This makes it a non-trivial test to our numerical scheme.

The computational domain consists of two overlapping sub-domains are shown in Fig. 4. The larger domain  $\Omega_1$  is a  $30D \times 20D$  rectangle subtracting a circular ‘hole’ of radius  $2D$ . The smaller one  $\Omega_2$  is a circular domain of radius  $2.45D$  subtracting the circular cylinder of radius  $0.5D$ . The three circles aforementioned share a common center at the origin. The size of this computational domain is chosen so as to minimize the blocking effect but with minimal number of computational cells. After comparing with domain sizes used in other literatures (such as [25–29]), we believe that the current size is a moderate one.

Sub-domain  $\Omega_1$  is meshed with 12,586 triangular cells. Sub-domain  $\Omega_2$  is meshed with 1260 rectangular cells and 4830 triangular cells. The reason for using such a hybrid mesh is to resolve the boundary layer more accurately. The mesh size on the cylinder surface is 0.025 (with 125 grid points on the circumference). In the normal direction, the distance between the surface and the first grid point is 0.01. The sub-meshes of the two domains and the composite mesh are presented in Fig. 5. For the purpose of comparison, a *single* mesh that consists of 1260 rectangular cells and 18,722 triangular cells is also generated. The resolution of this mesh is comparable with the composite (Chimera) mesh of Fig. 5 and the same vortex shedding problem is computed on this mesh. These highly inhomogeneous and anisotropic meshes are pre-processed through a spring-analogy smoothing before they are actually used in the code. Mesh quality has been improved to some extent after such treatment.

In the current simulation, the mesh velocities of both sub-domains are zero. The time-step is taken to be 0.002 and the resultant maximum CFL number is approximately 0.4. The results obtained using the method proposed in this paper are summarized as follows.

##### 3.1.1. Pressure, drag and lift coefficients and the flow field of the non-rotating case

An instantaneous pressure field (which is recorded after the periodic pattern has been established) for  $Re = 100$  is plotted in Fig. 6. It is seen that the solution in the overlapping region is of satisfactory quality. This result is obtained using the modified interpolation scheme of (25). For an unsteady problem like this, a direct interpolation could lead to a slightly decoupled pressure field in the overlapping region. For further discussions on the reason behind this phenomenon, please refer to [23]. In all subsequent simulations conducted in this paper, the pressure correction on the interior boundaries is computed using (25).

The drag and lift forces are computed by the integration of the forces along the surface of the cylinder. The drag and lift coefficients are then computed by



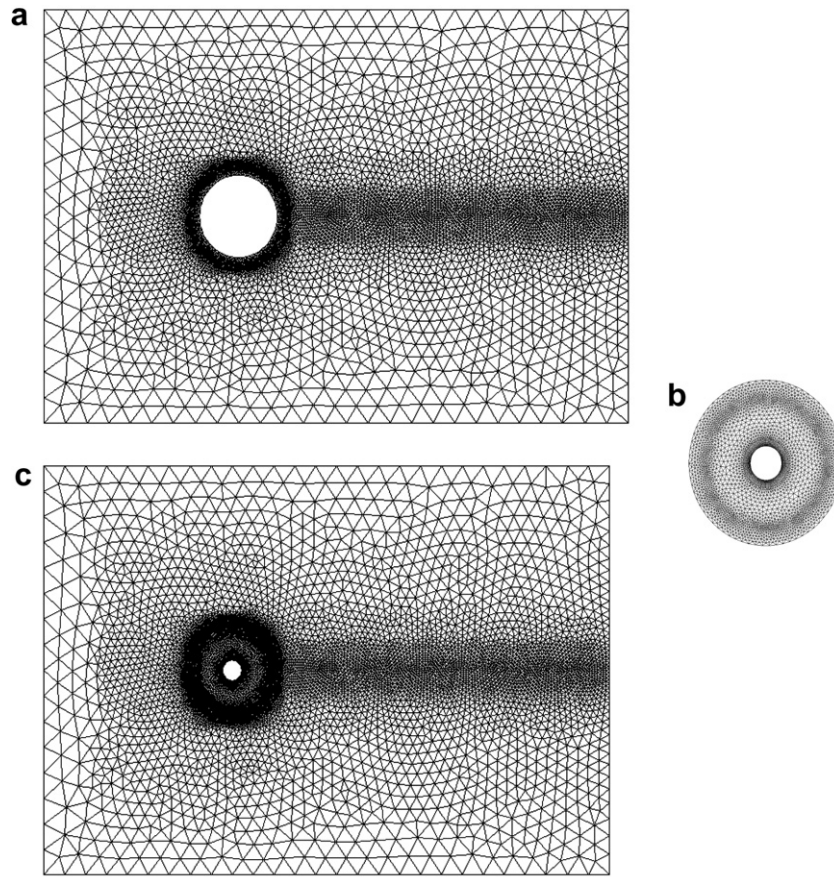


Fig. 5. Sub-grids and the composite grid for the computation of flow over a circular cylinder; (a) background grid; (b) grid around the circular cylinder; (c) composite (Chimera) grid.

$$C_d = \frac{F_x}{\frac{1}{2} \rho_0 U_\infty^2} \tag{26}$$

$$C_l = \frac{F_y}{\frac{1}{2} \rho_0 U_\infty^2}$$

where  $F_x$  and  $F_y$  are the stream-wise and transverse components of the force exerted on the cylinder. The time history of  $C_d$  and  $C_l$  for  $Re = 100$  and  $200$  are shown in Fig. 7. The frequency of vortex shedding is determined by taking the FFT of the lift coefficient. The Strouhal number is defined as

$$St = \frac{fD}{U_\infty} \tag{27}$$

where  $f$  is the shedding frequency and  $D$  the diameter of the cylinder.

The drag and lift coefficients and Strouhal number are listed in Table 1. For the purpose of comparison, data from some other references are also included in the table. From this table, it is clearly seen that our result using a Chimera grid matches well with both the solution on a single grid and those from other literatures.

Fig. 8 shows the time-averaged pressure coefficient  $C_p = 2(p - p_\infty) / \rho_0 U_\infty^2$  along the cylinder surface at  $Re = 100$ . A good agreement with the well-established data of Park et al. [30] is clearly seen from this figure.

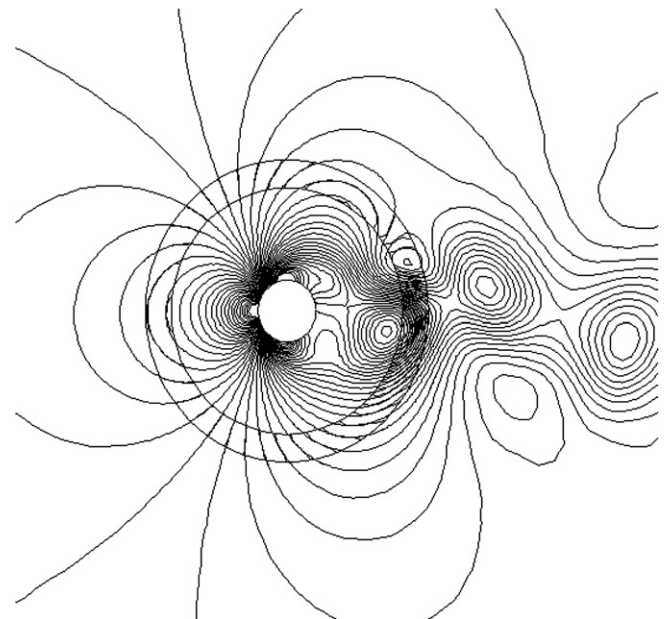


Fig. 6. Instantaneous pressure field of the flow over a fixed cylinder at  $Re = 100$ .

Instantaneous streamlines and vorticity contours at  $Re = 100$  are plotted in Figs. 9 and 10, respectively. Together with the evidences shown in Table 1, it is

confirmed that vortices pass the two interior boundaries with only minor distortion. This example demonstrates the excellent performance of the present numerical scheme in handling unsteady vortex shedding.

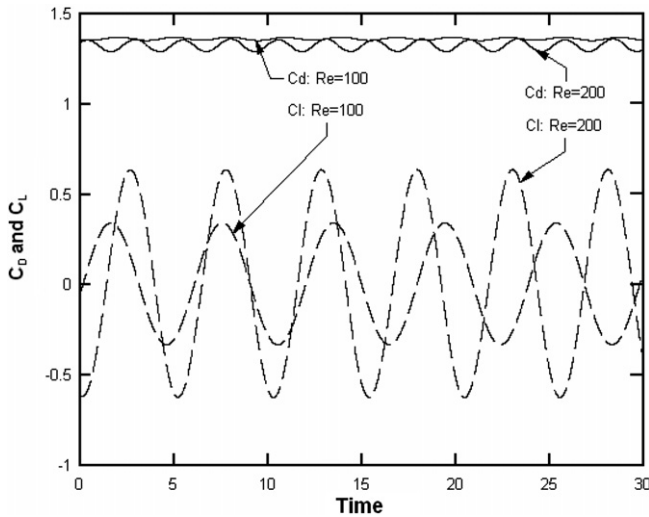


Fig. 7. Time history of drag and lift coefficients for the flow over a fixed cylinder at  $Re = 100$  and  $Re = 200$ .

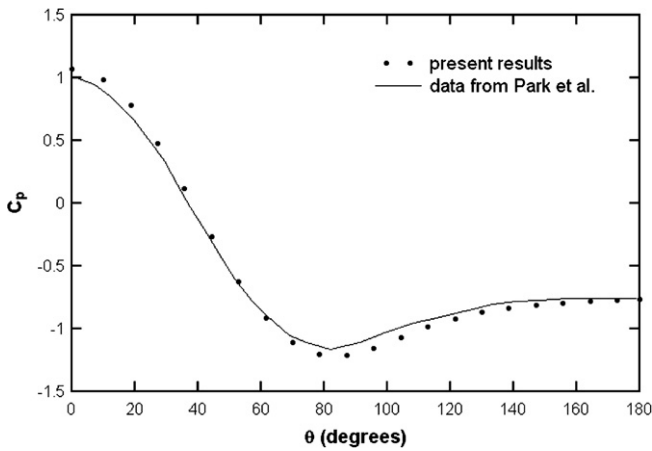


Fig. 8. Averaged surface pressure coefficient  $C_p$  as a function of the angle  $\theta$  ( $\theta = 0^\circ$  corresponds to the stagnation point,  $\theta = 180^\circ$  corresponds to the base point.) for the flow over a fixed cylinder at  $Re = 100$ . Black dots denote present results and solid line denotes the data from Park et al.

To check the quality of prediction in the overlapping region, the transverse velocity, pressure and vorticity along the horizontal centerline are extracted at an instant when the lift coefficient  $C_l$  is at the peak value. They are plotted as a function of the  $X$ -coordinate in Figs. 11–13, respectively. It is found that the largest difference for the transverse velocity and pressure from domain  $\Omega_1$  and domain  $\Omega_2$  occurs in the overlapping area *behind* the cylinder (labeled as ‘overlapping 2’ in these figures) and close to the interior boundary  $\Gamma_{21}$ . The largest difference is 7% and 5% for the transverse velocity and pressure, respectively. Away from the boundary  $\Gamma_{21}$  within the overlapping area, the difference is much smaller ( $<2\%$ ). From Fig. 13, it is seen that the difference in vorticity is reasonably small within the overlapping area. The largest difference in vorticity occurs near the boundary  $\Gamma_{12}$  (about 14%). Since the vorticity magnitude is quite small in this region, the relative difference is magnified by division.

### 3.1.2. Mesh-independent tests

Mesh-independent test is carried out by doing the simulation at Reynolds number of 200 on three different meshes. The mesh which is used to obtain the hydrodynamic force in the previous section is denoted as the *med-*

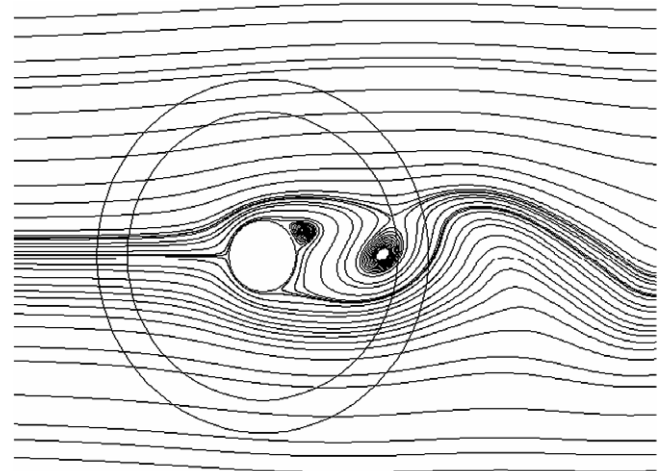


Fig. 9. Instantaneous streamlines of the flow over a fixed cylinder at  $Re = 100$ .

Table 1  
Drag, lift coefficients and Strouhal number of the flow over a fixed circular cylinder at Reynolds number 100 and 200

	$Re = 100$			$Re = 200$		
	$C_d$	$C_l$	$St$	$C_d$	$C_l$	$St$
Present: Chimera grid	$1.36 \pm 0.01$	$\pm 0.34$	0.168	$1.34 \pm 0.03$	$\pm 0.66$	0.197
Present: Single grid	$1.36 \pm 0.01$	$\pm 0.33$	0.168	$1.33 \pm 0.04$	$\pm 0.65$	0.198
Braza et al. [25]	$1.3 \pm 0.015$	$\pm 0.25$	–	–	–	–
Calhoun [26]	$1.33 \pm 0.01$	$\pm 0.30$	0.175	–	–	–
Liu et al. [27]	$1.36 \pm 0.01$	$\pm 0.34$	0.164	–	–	–
Pan and Damodaran [13]	–	–	–	$1.37 \pm 0.04$	$\pm 0.63$	0.192
Rogers and Kwak [28]	–	–	–	$1.23 \pm 0.05$	$\pm 0.65$	0.185
Rosenfeld et al. [29]	–	–	–	$1.46 \pm 0.05$	$\pm 0.69$	0.211



Fig. 10. Instantaneous vorticity contours of the flow over a fixed cylinder at  $Re = 100$ ; solid and dashed lines denote positive and negative contours, respectively.

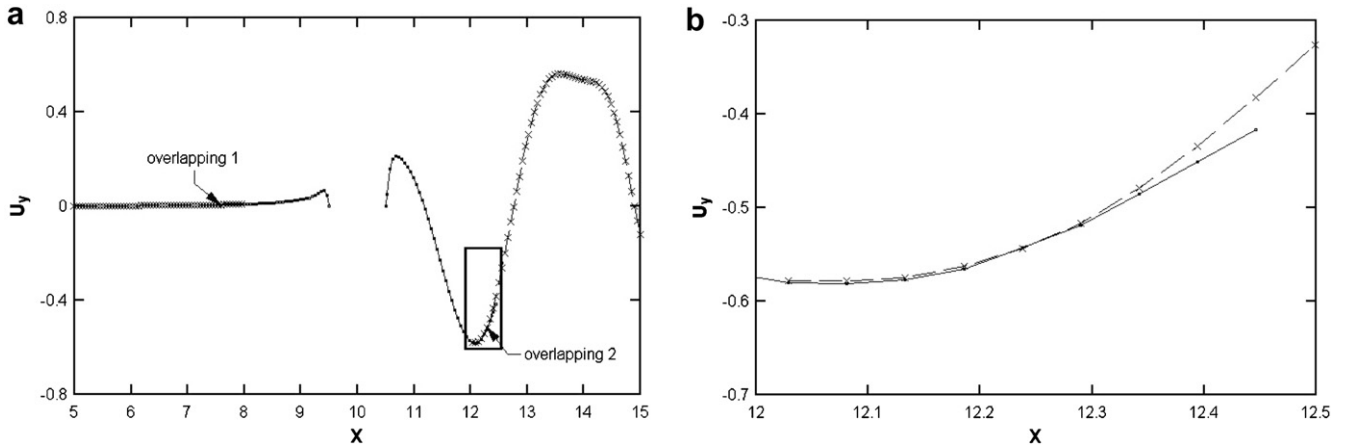


Fig. 11. Transverse velocity distribution along the horizontal centerline when  $C_1$  is at peak value for the flow over a fixed cylinder at  $Re = 100$ ; (a) transverse velocity as a function of  $X$ ; (b) a close-up near the overlapping area behind the cylinder.

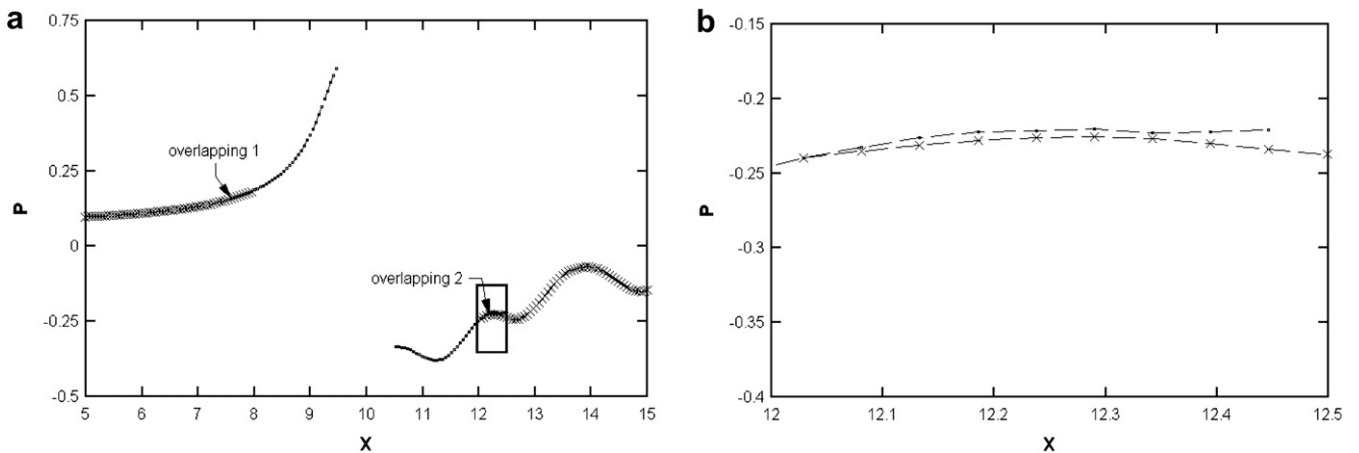


Fig. 12. Pressure distribution along the horizontal centerline when  $C_1$  is at peak value for the flow over a fixed cylinder at  $Re = 100$ ; (a) pressure as a function of  $X$ ; (b) a close-up near the overlapping area behind the cylinder.

*ium* mesh. This mesh also serves as the *standard* mesh for the generation of other two meshes. First, a *coarser* mesh is generated by using only 1/2 number of partitions on the surface and a double wall-distance of the first off-wall grid point. Then, a *finer* mesh is generated by using twice number of surface partitions and 1/2 the wall-distance. The total number of cells in the coarser, medium and finer mesh is 12,844, 18,676 and 30,852, respectively.

The computed lift and drag coefficients on these meshes are listed in Table 2. It is seen that the maximum difference in Strouhal number is less than 1.5% among the three results. When comparing the data from the *medium* mesh and those from the *finer* mesh, it is found that the difference in lift and drag are 3% and 7%, respectively. Although the result is not mesh-independent by a rigorous standard, since the difference between these two meshes is reasonably

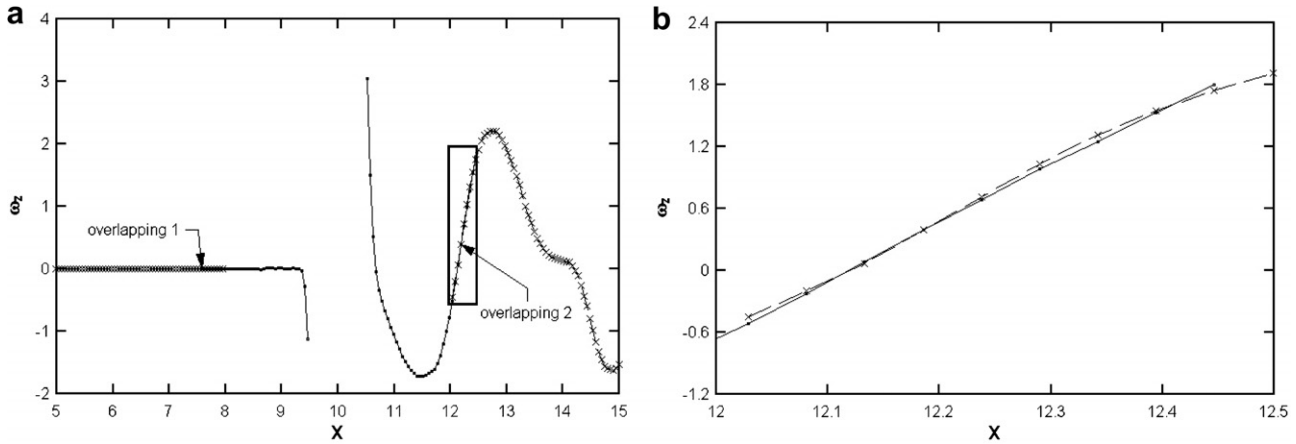


Fig. 13. Vorticity distribution along the horizontal centerline when  $C_l$  is at peak value for the flow over a fixed cylinder at  $Re = 100$ ; (a) vorticity as a function of  $X$ ; (b) a close-up near the overlapping area behind the cylinder.

small, the rotating cylinder simulation is conducted on the *medium* mesh to save computational time.

### 3.2. Flow over a rotating circular cylinder

Our next examples are flow over a cylinder which rotates counterclockwise with an angular velocity  $\omega = \pi$  and  $\omega = 4.0$ . For these two angular velocities, the tip velocity ratio  $q = D\omega/2U_\infty$  is 1.57 and 2, respectively. The  $Re$  number based on the inflow velocity is also 200, as same as that of the non-rotating case.

The simulation is performed on the *medium* mesh which is used for the stationary cylinder case. The mesh of sub-domain  $\Omega_1$  remains stationary, thus the mesh velocity  $\mathbf{v}_g$  is zero. For sub-domain  $\Omega_2$ , the mesh undergoes a solid-body rotation. Thus in this sub-domain, the velocity for a solid-body rotation is prescribed on each grid point. On the surface of the rotating cylinder, fluid velocity is set to the tip velocity of the cylinder. This velocity is also equivalent to the local mesh velocity. The time step used in this simulation is 0.002, which is also the same as that in the stationary case.

An overview of the characteristics in the flow field at various values of  $q$  can be found in [31]. A short summary is presented here. When the tip velocity ratio is low, this flow is unsteady. The unsteadiness is caused by the usual Karman vortex street similar to that in the stationary case. At a higher value of  $q$  ( $1.91 \leq q \leq 4.35$ ), a steady solution

can be achieved. At even higher values of  $q$  ( $4.34 \leq q \leq 4.75$ ), a second region of instability appears. Beyond  $q \geq 4.75$ , the flow become steady again but multiple solutions are observed. For the detailed discussions, please refer to [31]. In our current study, we only conduct numerical simulations at  $q$  equals to 1.57 and 2. These two values fall into the first unsteady and steady region, respectively. The numerical results are summarized as follows.

#### 3.2.1. Tip velocity ratio $q = 1.57$

The time history of drag and lift coefficients are shown in Fig. 14. It is well known that for a stationary cylinder, the frequency of drag is twice as that of the lift. However, it is seen from this figure that the frequencies of drag and lift are the same in this rotating case. This interesting phenomenon is also reported in [17]. The Strouhal number in the rotating case is slightly lower compared with the non-rotating case at the same  $Re$  number. For a *counterclockwise* rotation, a negative lift is obtained due to the Magnus effect. The averaged drag in the rotating case is significantly smaller than that in the non-rotating case. The utilization of a spinning cylinder in flow control and drag reduction is based on this feature. Some discussions regarding this application can be found in [32]. The predicted drag and lift coefficients and Strouhal number are listed in Table 3. For the purpose of comparison, data from [17] are also included in the table. Good agreement between our prediction and that in [17] is clearly seen in this table. The averaged drag in this paper is 8% lower and lift is 3% higher than the results of [17]. The Strouhal number in the current study is 5% lower.

An instantaneous pressure field is shown in Fig. 15. Comparing with Fig. 6 of the non-rotating case, it is observed that the stagnation point is shifted *clockwise* due to rotation. The instantaneous streamlines are presented in Fig. 16. It is seen that the vortex pair is shifted *counterclockwise* and the two vortices are compacted slightly (compared with Fig. 10). Instantaneous vorticity contours are shown in Fig. 17. It is noted that the Karman

Table 2  
Drag, lift coefficients and Strouhal number computed using the coarser, the medium and the finer mesh for stationary cylinder at  $Re = 200$

	$Re = 200$ ; stationary			
	$C_{d\_avg}$	$C_{l\_avg}$	$C_{l\_max}$	$St$
Mesh 1 (coarser)	1.36	0.002	0.72	0.197
Mesh 2 (medium)	1.34	0.007	0.66	0.198
Mesh 3 (finer)	1.37	$4.6e-5$	0.71	0.20

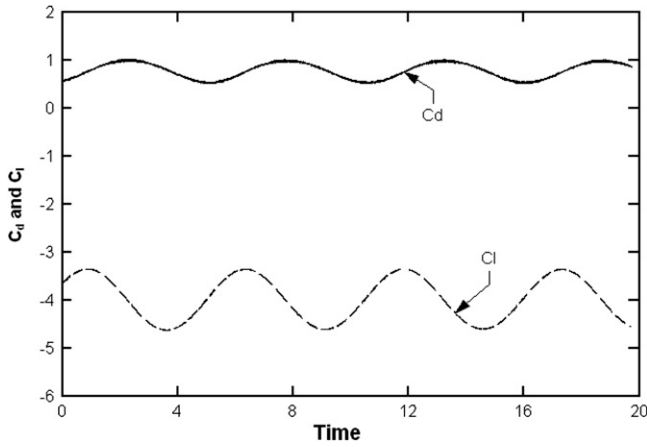


Fig. 14. Time history of drag and lift coefficients for the flow over a rotating circular cylinder at  $q = 1.57$ .

vortex street tilts upward and the vortices pass through the outlet at an oblique angle.

Table 3  
Drag, lift coefficients and Strouhal number of flow over a circular cylinder rotating counterclockwise at  $Re = 200$  with  $\omega = \pi$  and  $\omega = 4.0$

	$Re = 200; \omega = \pi$			$Re = 200; \omega = 4.0$	
	$C_{d\_avg}$	$C_{l\_avg}$	$St$	$C_d$	$C_l$
Present	0.77	-3.98	0.191	0.39	-5.26
Maruoka [17]	0.84	-4.11	0.191	-	-
Mittal and Kumar [31]	-	-	-	0.31	-5.29

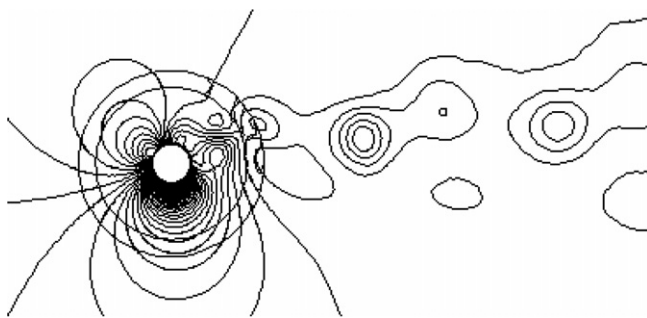


Fig. 15. Instantaneous pressure contours of the flow over a rotating circular cylinder at  $q = 1.57$ .

### 3.2.2. Tip velocity ratio $q = 2.0$

According to the simulation in [31], in this case, the vortex shedding should cease and the solution is steady. However, the authors in that paper also pointed out that it could take a long time for the flow to develop to the final state. They believe that this is the reason why some researchers in the past have associated their simulation with vortex shedding. The time history of drag and lift coefficients of our simulation is shown in Fig. 18. As it is seen in this figure, at non-dimensional time of 300, some trace of oscillation still exists. But since the amplitude of oscillation for both drag and lift are sufficiently small ( $<0.02$ ) and the trend of decline is obvious. We believe that a steady solution could be achieved eventually giving a long enough simulation time although we did not try that in this study. Our predicted lift coefficient (averaged value at  $t = 300$ ) is only 1% lower than the value from [31]. The discrepancy in the drag is larger (off by 26%). The drag coefficient computed in this paper is 0.39 whereas the value is 0.31 in [31]. Since the drag coefficient is quite small, the relative error is magnified. At first, we suspected that the reason for the discrepancy is due to the difference in domain size. According to the analysis in [31], in the case of medium and large value of  $q$ , lift is fairly insensitive to the change in domain size and the drag coefficient shows strong

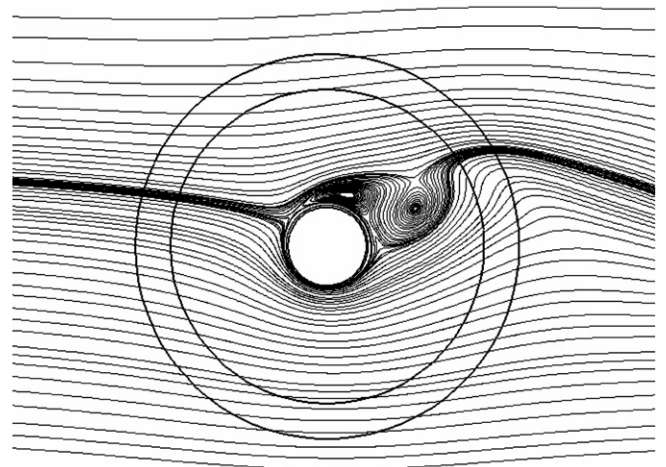


Fig. 16. Instantaneous streamlines of the flow over a rotating circular cylinder at  $q = 1.57$ .



Fig. 17. Instantaneous vorticity contours of the flow over a rotating circular cylinder at  $q = 1.57$ ; solid and dashed lines denote positive and negative contours, respectively.

dependence on the distance  $L$  between the cylinder and the outlet. The drag coefficient becomes independence of  $L$  only when  $L > 75D$ . The computational result in [31] is based on a mesh of  $L = 100D$  and our simulation is conducted on a mesh of  $L = 20D$ . To test this explanation, we did another simulation with comparable mesh resolution in the near field and a larger domain of  $L = 50D$ . However, when the solution is almost steady, we found that the drag coefficient is unchanged. Our second conjecture is that this discrepancy is due to the insufficient mesh

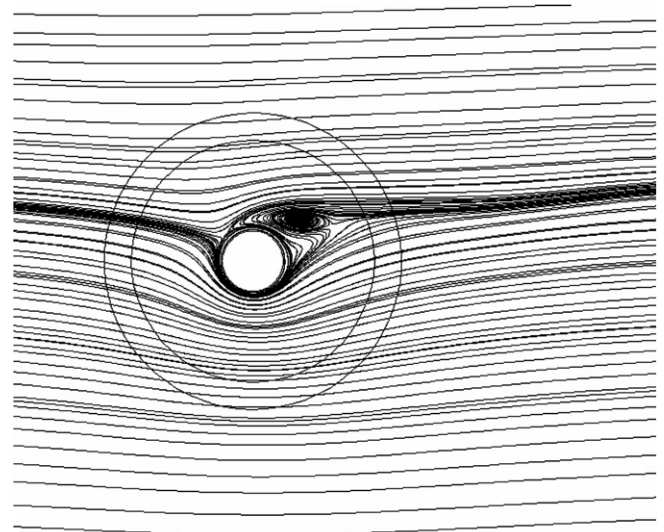


Fig. 21. Streamlines of the flow over a rotating circular cylinder at  $q = 2.0$ .

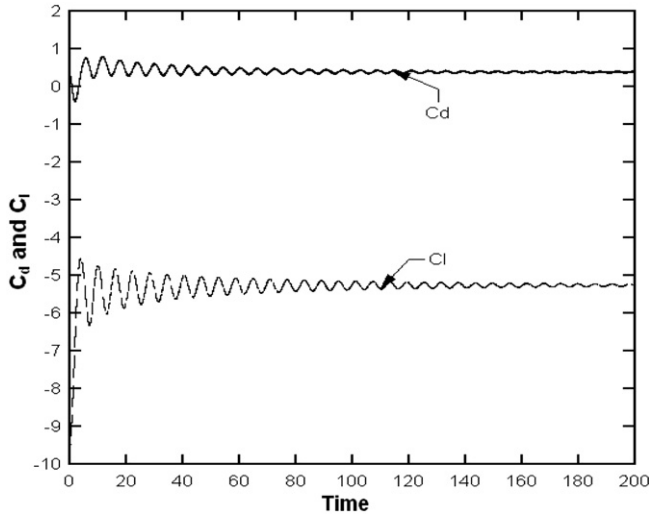


Fig. 18. Time history of drag and lift coefficients for the flow over a rotating circular cylinder at  $q = 2.0$ .

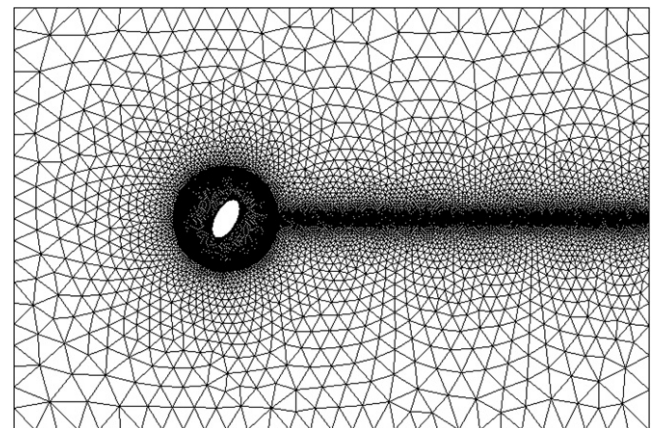


Fig. 22. The composite grid for the computation of flow over a rotating elliptic cylinder.

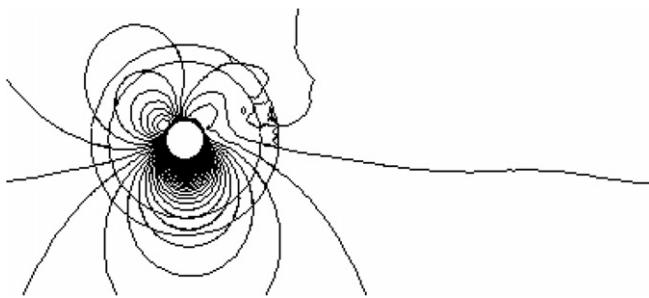


Fig. 19. Pressure contours of the flow over a rotating circular cylinder at  $q = 2.0$ .



Fig. 20. Vorticity contours of the flow over a rotating circular cylinder at  $q = 2.0$ ; solid and dashed lines denote positive and negative contours, respectively.

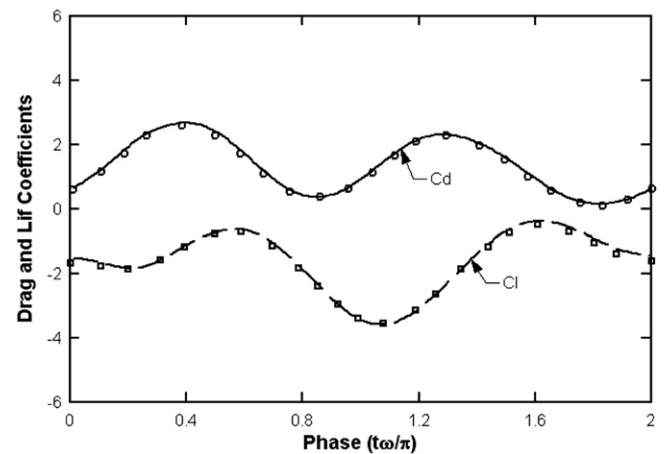


Fig. 23. Time history of drag and lift coefficients for the flow over a rotating elliptic cylinder during one rotation. Lines denote results of the current simulation and symbols denote the data from Maruoka.

resolution near the shear layer which extend to the far field. Since our mesh in Fig. 4 is generated based on the picture of Karman vortex street behind a fixed cylinder, the distri-

bution of mesh points is not appropriate for the simulation of a steady flow at high value of  $q$ . To test this conjecture, another mesh which is adaptive to the current solution of

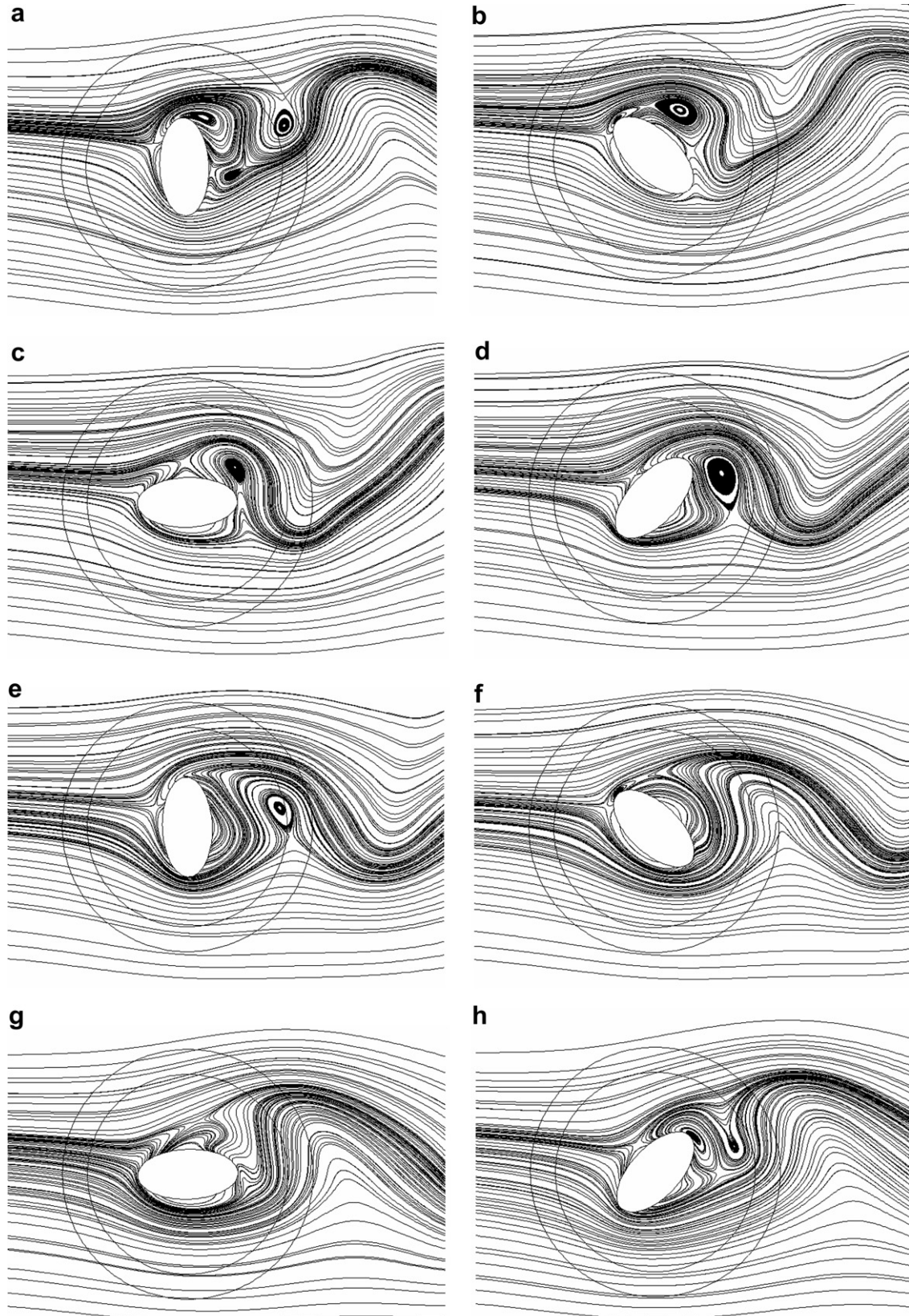


Fig. 24. Eight snapshots of instantaneous streamlines of the flow over a rotating elliptic cylinder during one rotation.

$q = 2.0$  is generated. In this new mesh, points are densely concentrated in the region where the absolute value of vorticity is high. But the solution on this new mesh again fails to ‘correct’ the drag coefficient. The true reason behind the discrepancy in drag is still not very clear. Similar phenomenon is also reported in [33] when the authors in that paper try to compare their predicted drag with [31] for  $Re = 200$  and  $q = 3$ . The authors in [33] claim that the drag prediction at higher value of  $q$  is very sensitive to *mesh shape* and *numerical schemes*.

The pressure field at  $t = 300$  is shown in Fig. 19. When comparing with the case of  $q = 1.57$  (see Fig. 15), the position of the stagnation point and the orientation of the contours near this point are very similar. However, the fluctuation in the wake disappears due to the suppression of vortex shedding. The vorticity contours are shown in Fig. 20. It is seen that two narrow-banded regions with large positive and negative vorticity extend to the far field behind the cylinder. The streamlines are presented in Fig. 21. It is well known that for the steady solution at low  $Re$  number (4–40) of the non-rotating case, two symmetric separation bubbles are formed at the rear. However, in this rotating case, only one separation bubble is observed. It is also noticed that this separation bubble tilts upwards and sits on the shoulder of the cylinder. Similar asymmetrical streamline pattern due to rotation are also reported in [34,35]. A picture with vanishing lower bubble is given in [35] at the Reynolds number of 60 and  $q = 1$ .

### 3.3. Flow over a rotating elliptic cylinder

Our final example is flow over an elliptic cylinder with thickness ratio (minor axis to major axis) of 0.5 which rotates counterclockwise at an angular velocity  $\omega = 0.5\pi$ . The  $Re$  number based on the major axis is 200 and the tip velocity ratio is 0.785.

For this simulation, the whole computational domain is meshed with 8996 moving cells (including 776 rectangular cells) and 16,132 stationary cells in the two sub-domains, respectively. The mesh resolution is comparable with the *medium* mesh for the circular cylinder case (see Fig. 22). The time step is also 0.002. The computed results are summarized as follows.

Numerical results show that the drag and lift coefficients are perfect periodic solutions. The variation of drag and lift coefficients versus phase ( $\omega t/\pi$ ) within one period are plotted in Fig. 23. The computational data from [17] is also plotted on top for comparison. It is seen that the agreement between these two sets of data is quite satisfactory. From this figure, it is also seen that the phases of maximum drag does not coincide with the phases when the major axis of the ellipse is perpendicular to the inflow direction (i.e.  $0.5\pi$  and  $1.5\pi$ ). They are slightly shifted. This interesting phenomenon is also reported in [17].

The streamlines at eight different phases are presented in Fig. 24. From Fig. 24(b)–(g), it is seen that the upper bubble grows larger and then sheds from the surface. This procedure is just the same as that of a stationary circular cylinder. However, from (h) to (a), the evolution of vortex structure is quite complicated. To explore this more carefully, three snapshots in between Fig. 24(h) and (a) are shown in Fig. 25. The formation of a triple-vortex system can be seen very clearly from these pictures. In Fig. 25(a), vortex *A* (with negative vorticity) is generated at the trailing edge. In Fig. 25(b), vortex *A* grows larger and at the same time attaches to surface and rotates with it. The movement of vortex *A* somehow induces the emergence of another vortex *B* (with positive vorticity). In Fig. 25(c), vortex *C* (with positive vorticity) comes into sight at the back of the ellipse. This newly created vortex will then merge with vortex *B* before being convected downstream. In Fig. 24(b) and (f), a tiny separation bubble is also observed near the leading edge. However, this bub-

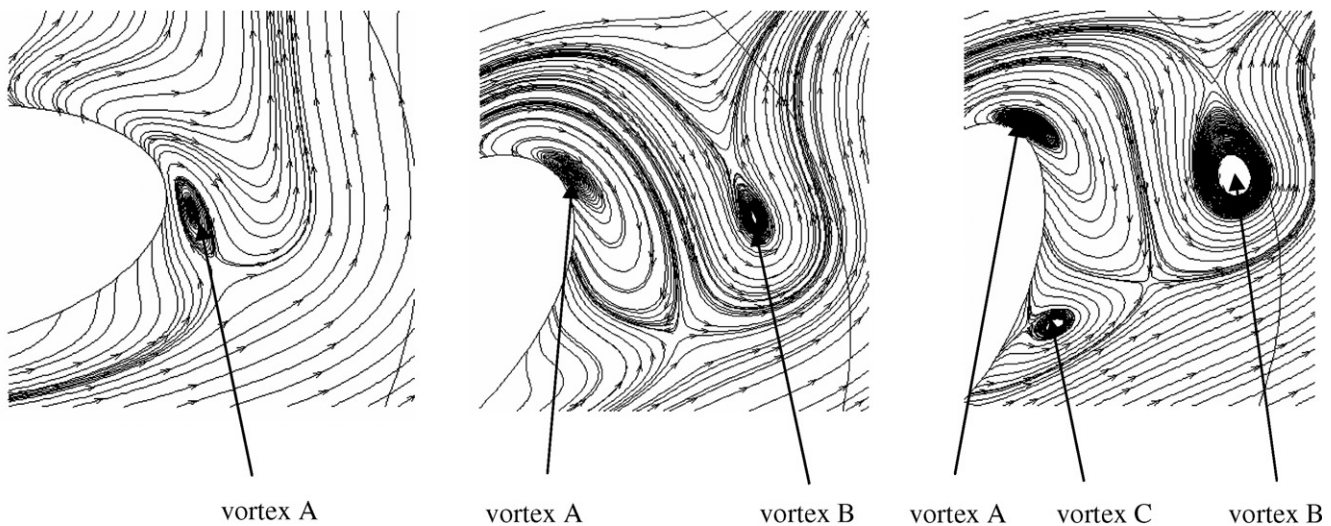


Fig. 25. Three snapshots in between Fig. 24(g) and (a).



ble does not exist for a long time and disappears immediately due to pressure rise. Similar vortex structures are also reported in [36] where flows over a *stationary* elliptic cylinder at different orientations are simulated.

#### 4. Summary

In the paper, a numerical simulation of flow over rotating body is presented. The numerical method used in the simulation is a modified pressure-correction scheme on an unstructured Chimera grid. To handle the sub-domain which undergoes rotation, a moving-mesh control volume scheme is implemented in an inertial reference frame. A new interpolation is proposed to transfer *pressure correction* on the interior boundaries. Such an interpolation can ensure a tight coupling of pressure between overlapping sub-domains.

In the simulation of laminar vortex shedding over stationary and rotating objects, the hydrodynamic force computed by this approach agrees well with experimental and computational results in literatures. These numerical examples demonstrate the credibility of this method in computing unsteady flows with and without rotating parts.

In our future plan, the unstructured Chimera method will be used to treat more challenge cases where irregular motions are encountered (such as the flapping wing) or where the motion is coupled with the dynamics (such as free-falling objects).

#### Acknowledgements

This work is partially sponsored by the National Natural Science Foundation of China under the project No. 10325211 and No. 10628206 and also the Chinese Academy of Sciences under the innovative project “Multi-scale modeling and simulation in complex system” (KJ CX-SW-L08). The author X.Z. also expresses his appreciation to the support and funds from “Project-sponsored by SRF for ROCS, SEM”. The authors also would like to thank the referees for their suggestions and comments.

#### References

- [1] Mavriplis DJ. Unstructured grid techniques. *Annu Rev Fluid Mech* 1997;29:473–514.
- [2] Barth TJ. Aspects of unstructured grids and finite-volume solvers for the Euler and Navier–Stokes equations. Lecture notes presented of the VKI lecture series 1994–2005. Belgium: Von Karman Institute for Fluid Dynamics; 1994.
- [3] Sun Y, Wang ZJ, Liu Y. High-order multi-domain spectral difference method for the Navier–Stokes equations on unstructured hexahedral grids. *Commun Comput Phys* 2007;2:310–33.
- [4] Perot JB, Nallapati R. A moving unstructured staggered mesh method for the simulation of incompressible free-surface flows. *J Comput Phys* 2003;184:192–214.
- [5] Slone KA, Pericleous K, Bailey C, Cross M. Dynamic fluid structure interaction using finite volume unstructured mesh procedures. *Comput Struct* 2002;80:371–90.
- [6] Farhat C, Degand C, Koobus B, Lesoinne M. Torsional springs for two-dimensional dynamic unstructured fluid mesh. *Comput Methods Appl Mech Eng* 1998;163:231–45.
- [7] Degand C, Farhat C. A three-dimensional torsional spring analogy method for unstructured dynamic meshes. *Comput Struct* 2002;80:305–16.
- [8] Peskin CS. The immersed boundary method. *Acta Numer* 2002;11(2):479–517.
- [9] Mittal R, Iaccarino G. Immersed boundary methods. *Ann Rev Fluid Mech* 2005;37:239–61.
- [10] Steger JL. The Chimera method of flow simulation. Workshop on applied CFD, Univ. of Tennessee Space Institute, 1991.
- [11] Brezzi F, Lions JL, Pironneau O. Analysis of Chimera method. *CR Acad Sci Paris* 2001;T.332(Serie I):655–60.
- [12] Behr M, Tezduyar TE. Shear-slip mesh update method. *Comput Methods Appl Mech Eng* 1999;174:261–74.
- [13] Pan H, Damodaran M. Parallel computation of viscous incompressible flows using Godunov-projection method on overlapping grids. *Int J Numer Methods Fluids* 2002;39:441–63.
- [14] Tang HS, Casey JS, Fotis S. An overset-grid method for 3D unsteady incompressible flows. *J Comput Phys* 2003;191:567–600.
- [15] Nakahashi K, Togashi F, Sharov D. Inter-grid boundary definition methods for overset unstructured grid approach. *AIAA J* 2000;38(11):2077–84.
- [16] Togashi F, Ito Y, Murayana M, Nakahashi K, Kato T. Flow simulation of flapping wings of an insect using overset unstructured grid. *AIAA J* 2001:261.
- [17] Maruoka A. Finite element analysis for flow around a rotating body using Chimera method. *Int J Comput Fluid Dyn* 2003;17(4):289–97.
- [18] Houzeaux G, Codina R. A Chimera method based on a Dirichlet/Neumann(Robin) coupling for Navier–Stokes equations. *Comput Meth Appl Mech Eng* 2003;192:3343–77.
- [19] Houzeaux G, Codina R. A Dirichlet/Neumann domain decomposition method for incompressible turbulent flows on overlapping sub-domains. *Comput Fluids* 2004;33:771–82.
- [20] Demirdzic I, Muzaferija S. Numerical method for coupled fluid flow, heat transfer and stress analysis using unstructured moving meshes with cells of arbitrary topology. *Comput Methods Appl Mech Eng* 1995;125:235–55.
- [21] Demirdzic I, Muzaferija S, Peric M. Advances in computation of heat transfer, fluid flow, and solid body deformation using finite volume approaches. In: Minkowycz WJ, Sparow EM, editors. *Advances in numerical heat transfer, vol. 1*. New York: Taylor & Francis; 1997 [Chapter 2].
- [22] Ferziger JH, Peric M. *Computational methods for fluid dynamics*. Berlin and Heidelberg: Springer-Verlag; 1996. pp.196 & 264.
- [23] Zhang X. Computation of viscous incompressible flow using pressure correction method on unstructured Chimera grid. *Int J Comput Fluid Dyn* 2006;20(9):637–50.
- [24] Liou PL. On the Schwarz alternating method. II. In: Chan T, Glowinski R, Periaux J, Widlund O, editors. *Domain decomposition methods*. Philadelphia: SIAM; 1989. p. 47–70.
- [25] Braza M, Chassaing P, Minh HH. Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder. *J Fluid Mech* 1986;165:79–130.
- [26] Calhoun D. A Cartesian grid method for solving the two-dimensional stream function-vorticity equations in irregular regions. *J Comput Phys* 2002;196:231–75.
- [27] Liu Z, Zheng X, Sung CH. Preconditioned multi-grid methods for unsteady incompressible flows. *J Comput Phys* 2000;160(1): 151–78.
- [28] Rogers SE, Kwak D. Upwinding differencing scheme for the time-accurate incompressible Navier–Stokes equations. *AIAA J* 1990;28(2):253–62.
- [29] Rosenfeld M, Kwak D, Vinokur M. A fractional-step solution method for the unsteady and incompressible Navier–Stokes equations in generalized co-ordinate systems. *J Comput Phys* 1991;94(1):102–37.

- [30] Park J, Kwon K, Choi H. Numerical solution of flow past a circular cylinder at Reynolds numbers up to 160. *KSME Int J* 1998;12(6): 81200–5.
- [31] Mittal S, Kumar B. Flow past a rotating cylinder. *J Fluid Mech* 2003;476:303–34.
- [32] Modi VJ. Moving surface boundary layer control: a review. *J Fluids Struct* 1997;11:627–63.
- [33] Padrino JC, Joseph DD. Numerical study of the steady-state uniform flow past a rotating cylinder. *J Fluid Mech* 2006;557:191–223.
- [34] Ingham DB, Tang T. A Numerical Investigation into the steady flow past a rotating circular cylinder at low and intermediate Reynolds numbers. *J Comput Phys* 1990;87(1):91–107.
- [35] Tang T, Ingham DB. On steady flow past a rotating circular cylinder at Reynolds numbers 60 and 100. *Comput Fluids* 1991;19(2):217–30.
- [36] Badr HM, Dennis SCR, Kocabiyik S. Numerical simulation of the unsteady flow over an elliptic cylinder at different orientations. *Int J Numer Methods Fluids* 2001;37:905–31.