ABSTRACT

Title of dissertation:	AN EMBEDDED-BOUNDARY FORMULATION FOR LARGE-EDDY SIMULATION OF TURBULENT FLOWS INTERACTING WITH MOVING BOUNDARIES
	Jianming Yang, Doctor of Philosophy, 2005
Directed by:	Assistant Professor Elias Balaras Department of Mechanical Engineering

A non-boundary-conforming formulation for simulating transitional and turbulent flows with complex geometries and dynamically moving boundaries on fixed orthogonal grids is developed. The underlying finite-difference solver for the filtered incompressible Navier-Stokes equations in both Cartesian and cylindrical coordinates is based on a second-order fractional step method on staggered grid. To satisfy the boundary conditions on an arbitrary immersed interface, the velocity field at the grid points near the interface is reconstructed locally without smearing the sharp interface. The complications caused by the Eulerian grid points emerging from a moving solid body into the fluid phase are treated with a novel "field-extension" strategy. To treat the two-way interactions between the fluid and structure, a strong coupling scheme based on Hamming's fourth-order predictor-corrector method has been developed. The fluid and the structure are treated as elements of a single dynamical system, and all of the governing equations are integrated simultaneously, and iteratively in the time-domain.

A variety of two and three-dimensional fluid-structure interaction problems of increasing complexity have been considered to demonstrate the accuracy and the range of applicability of the method. In particular, forced vibrations of a rigid circular cylinder including the harmonic in-line vibrations in a quiescent fluid and the transverse vibrations in a free-stream, and the vortex-induced vibrations of an elastic cylinder with one and two degrees of freedom in a free-stream are presented and compared with reference simulations and experiments. Three-dimensional DNS and LES of fluid flows involving stationary complex geometries include the flow past a sphere at $Re = 50 \sim 1,000$, the transitional flow past an airfoil with a 10° attack angle at $Re = 10,000$. Then, the turbulent flow over a traveling wavy wall at $Re = 10,170$ are simulated are compared with the detailed DNS using body-fitted grid in the literature. Finally, the simulation of the transitional flow past a prosthetic mechanical heart valve with moving leaflets at $Re = 4,000$ has been performed. All results are in good agreement with the available reference data.

AN EMBEDDED BOUNDARY FORMULATION FOR LARGE-EDDY SIMULATION OF TURBULENT FLOWS INTERACTING WITH MOVING BOUNDARIES

by

Jianming Yang

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2005

Advisory Committee:

Assistant Professor Elias Balaras, Chair/Advisor
Professor Richard V. Calabrese
Professor Ugo Piomelli
Associate Professor Arnaud Trouvé
Professor James M. Wallace

Dedicated to my parents,

to my wife, Guangming,

and to my son, Reece

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor, Prof. Elias Balaras, for his great patience, frequent encouragement, and generous support throughout this work. Elias not only introduced me to the large-eddy simulation of turbulence, but also guided me into the depth and breadth of my research project. I have learned a great deal from his invaluable insight into many of the numerical and physical problems. This manuscript has also benefited from his countless corrections.

I am very grateful to the members of my advisory committee, Prof. Richard V. Calabrese, Prof. Ugo Piomelli, Prof. Arnaud Trouvé, and Prof. James M. Wallace, for reading this dissertation and offering many constructive comments and valuable suggestions. To Prof. Piomelli in particular, thanks are due for teaching me three courses including almost everything in Fluid mechanics and numerical simulations. His lectures were among the best ones I have ever attended. To Prof. Wallace, my gratitude extends to his very careful reading and many grammar and spelling corrections to this manuscript. I also would like to thank Prof. Trouvé and Prof. Piomelli for their constant input to my work in our weekly CFD group meeting.

My thanks are due to Prof. Sergio Preidikman for providing me the ODE solver for rigid body motion and many valuable discussions. I am also grateful to Mr. Antonio Cristallo for his help on the heart valve computation.

My life and study in Maryland has been enriched greatly by my friends and colleagues here. I would like to thank all of them for their help and support, especially, Dr. Ning Li for many useful discussions and helps during the course of this work.

During my PhD study, my family have been very supportive. Their unconditional support has given me the strength to finish my study. I am very grateful to them. I am also much indebted to my wife Guangming and my son Reece for all their love and support.

This work was supported in part by the University of Maryland, College Park, the National Institutes of Health under Grant No. R01-HL-07262, and the National Science Foundation under Grant No. CTS-0347011. These sponsors are gratefully acknowledged.

NOMENCLATURE

Roman Symbols

a, A, A_x, A_y	Oscillation amplitude.
a_i, b_i, c_i	Coefficient in quadratic polynomial.
am_i, al_j, an_k	Coefficients in the discretizated Poisson equation in cross-stream, spanwise, and streamwise directions, respectively.
$\mathbf{A}$	Coefficient matrix.
A_i	Operator in the ith momentum equation containing terms treated explicitly.
B_i	Operator in the ith momentum equation containing terms treated implicitly.
c	Damping coefficient; Chord length of an airfoil; Phase speed of wave.
C	Model coefficient in the Smagorinsky model.
$\mathbf{C}$	Damping matrix.
C_f	Skin friction coefficient.
C_p	Pressure coefficient.

C_D	Drag coefficient.
C_L	Lift coefficient.
C_S	Coefficient of side force.
C_x	Coefficient of force in x direction.
C_y	Coefficient of force in y direction.
D	Diameter of cylinder/sphere.
$\mathbf{e}$	Local truncation error.
f	Scalar function; Frequency.
$\mathbf{f}$	Momentum forcing vector; Hydrodynamic force vector.
f_0	Shedding frequency of stationary cylinder.
f_e	Excitation frequency.
f_i	$i = 1, 2, 3$ Momentum forcing in i direction.
f_n	Natural vibrating frequency of a structure.
F_i	$i = 1, 2, 3$ i direction force on a structure.
$\overline{G}$	Filtering operation using the grid-filter.
$\hat{G}$	Filtering operation using the test-filter.
$\mathbf{i}, \mathbf{j}, \mathbf{k}$	Unit normal vectors.
k	Time step index in time-advancement scheme; Spring constant; Wave number.
k'_l	Modified wave number in spanwise direction.

$\mathbf{K}$	Stiffness matrix.
KC	Keulegan-Carpenter number.
l	Wave number in spanwise direction.
L	Reference length scale.
L_i	Length of computational domain in i direction.
$\mathcal{L}_{ij}$	Resolved turbulent stress tensor.
m	Structural mass.
$\mathbf{M}$	Mass matrix.
n	Time step index in time-advancement scheme; Normal direction of the boundary; Unit normal vector; Degrees of freedom; Mass ratio.
N_x, N_y, N_z	Number of grid points in the transverse, spanwise, and streamwise directions, respectively.
N_r, N_θ, N_z	Number of grid points in the radial, azimuthal, and axial directions, respectively.
p	Pressure normalized by ρU^2.
$\mathbf{r}$	A spatial ray.
R	Radius of a pipe.
Q	The second invariant of velocity gradient tensor.
Re	Reynolds number.
s	Arclength.

$\overline{S}_{ij}$ Strain rate tensor based on the resolved velocity field, $\overline{u}_i$.

St Strouhal number.

t Time.

T_{ij} Subgrid scale stress tensor at the test-filter level.

$\overline{u}_i$ Resolved velocity components: velocity components filtered at the grid-filter level.

$\hat{u}_i$ Components of the predicted velocity field.

$\hat{\overline{u}}_{ij}$ Velocity components filtered at both the grid-filter and test-filter level.

u_τ Friction velocity.

u_i General notation for velocity components:

Cartesian coordinates: $u_1 = u, u_2 = v, u_3 = w$.

Cylindrical coordinates: $u_1 = u_r, u_2 = u_\theta, u_3 = u_z$

u, v, w Cartesian coordinates: cross-stream, spanwise, and streamwise velocity components, respectively.

Cylindrical coordinates: radial, azimuthal, and axial velocity components, respectively.

u_x, u_y, u_z Cartesian coordinates: cross-stream, spanwise, and streamwise velocity components, respectively.

u_r, u_θ, u_z Cylindrical coordinates: radial, azimuthal, and axial velocity components, respectively.

u_b, v_b, w_b Normal velocity components at the boundary of the computational domain.

u_t, v_t, w_t Tangential velocity components at the boundary of the computational domain.

U Reference velocity.

U_{red} Reduced velocity.

U_{max} Maximum velocity.

U_∞ Free-stream velocity.

U_{bulk} Bulk velocity.

U_{cl} Centerline velocity in a pipe.

U_{conv} Convective velocity used in the convective boundary condition.

W Weighting function.

x_i General notation for (non-dimensional) spatial coordinates:

Cartesian coordinates: $x_1 = x, x_2 = y, x_3 = z$.

Cylindrical coordinates: $x_1 = r, x_2 = \theta, x_3 = z$.

X_0, Y_0, Z_0 Structural displacement.

$\mathbf{x}$ Spatial position vector.

$\mathbf{X_i}$ Arclength coordinate.

x, y, z Cartesian coordinates: Spatial coordinates in the cross-stream, spanwise, and streamwise directions, respectively.

z Cylindrical coordinates: axial coordinate.

Greek Symbols

α_i	Factor in RK time-advancement scheme.
δ_{ij}	Kronecker's delta.
δ	Discrete form of the partial operator.
Δ	Filter width.
$\overline{\Delta}$	Filter width at the grid-filter level.
$\overline{\hat{\Delta}}$	Filter width at the test-filter level.
$\Delta x, \Delta y, \Delta z$	Cartesian coordinates. Grid-spacing in the cross-stream, spanwise, and streamwise directions, respectively.
$\Delta r, \Delta \theta, \Delta z$	Cylindrical coordinates. Grid-spacing in the radial, azimuthal, and axial directions, respectively.
Δt	Time step.
γ_i	Factor in RK time-advancement scheme.
λ	Wavelength.
μ	Dynamic viscosity of a fluid.
ν	Total viscosity: $\nu = 1/Re + \nu_t$.
ν_m	Kinematic viscosity of a fluid.
ν_t	Turbulent eddy viscosity.

ω	Angular frequency oscillation; Vorticity.
φ	Scalar function.
ϕ	Scalar to project the predicted velocity field into a divergence-free space.
$\mathit{\Psi}$	Immersed interface.
ρ	Density of a fluid.
ρ_i	Factor in RK time-advancement scheme.
τ	Time
τ_{ij}	Subgrid scale stress tensor at the grid-filter level.
τ_w	Shear stress at the wall.
θ	Cylindrical coordinates: Azimuthal coordinate.
Θ	Structural rotation.
ξ, η, ζ	Cartesian coordinates: spatial coordinates in the computational space in the cross-stream, spanwise, and streamwise directions, respectively. Cylindrical coordinates: spatial coordinates in the computational space in the the radial, azimuthal, and axial directions, respectively.
ζ	Damping ratio.

Ohter Symbols

$\dot{()}$	First derivative.
$\ddot{()}$	Second derivative.
$<>$	Averaging operator.
$()^+$	Variable given in wall units: $()^+ = \rho()u_\tau/\mu$.

Abbreviations

AB	Adams-Bashforth.
CFL	Courant-Friedrichs-Levy.
CN	Crank-Nicholson.
CPU	central processing unit.
DNS	direct numerical simulation.
FFT	fast Fourier transform.
LES	large-eddy simulation.
MPI	message passing interface.
ODE	ordinary differential equation.
RHS	right-hand side.
RK	Runge-Kutta.
SGS	subgrid scale.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Background

Numerical simulations of turbulent and transitional flows with dynamically moving boundaries are amongst the most challenging problems in computational mechanics. The main difficulty comes from the fact that the spatial domain occupied by the fluid changes with time, and the location of the boundary is usually an unknown itself that depends on the fluid flow and the solution of additional equations describing the motion/deformation of the body. There is only a limited number of special cases where established boundary-conforming formulations can be directly applied to such problems with a relatively small overhead. The use of moving reference frames [31], or coordinate transformations [40, 54] are characteristic examples. In more complex configurations formulations utilizing moving and/or deforming grids that continuously adapt to the changing location of the body have to be adopted (see for example [53, 24]). For problems that involve multiple bodies undergoing large motions and/or deformations, these algorithms are fairly complicated and have an adverse impact on the accuracy and efficiency of the fluid solvers.

Therefore, although a variety of fluid-structure interaction algorithms has been developed over the years, relatively few applications in turbulent and transitional flows have been reported. In most cases, this is due to prohibitively high computational cost, or dissipative discretizations that limit the applicability of such methods to classical turbulence modeling strategies. Further advancements in this field can be achieved by coupling state-of-the-art tools to model turbulence and transition (i.e. large-eddy simulations (LES) or hybrid formulations) with cost-effective numerical methods applicable to problems with large boundary motions and deformations.

An alternative class of methods that has the potential to overcome some of the above limitations are non-boundary-conforming formulations. In such methods the requirement that the grid conforms to a solid boundary is relaxed, and the effect of a complex object on the flow is introduced through proper treatment of the solution variables at the grid cells in the vicinity the body. The basic advantage of these formulations is the simplification of grid generation, especially in cases of moving boundaries where the need for regeneration or deformation of the grid is eliminated. In addition, efficient, Cartesian solvers can be directly applied to complex flow problems. Both the above features are particularly attractive for Direct Numerical Simulations (DNS) or LES of turbulent and transitional flows, where the use of highly efficient, energy conserving solvers is imperative for accurate computations. It is therefore conceivable that successful integration of non-boundary-conforming strategies with robust Cartesian or cylindrical coordinate solvers developed for DNS/LES will open a wide new area of applications for these tools. Example applications of such a tool include a variety of low and moderate Reynolds number turbulent flow problems from engineering, biology, and medicine, where fluid-structure interactions are central to the dynamics of the flow.

The objective of this work is the development of an embedded-boundary method for DNS and LES of turbulent and transitional flows interacting with dynamically moving boundaries. The target application fields are the a variety of low and moderate Reynolds number turbulent flows with complex geometries and moving boundaries in engineering, biology, and medicine.

1.2 Literature Survey

Over the past decades a variety of non-boundary-conforming methods with various degrees of accuracy and complexity have been proposed. The so-called immersed-boundary formulation pioneered by Peskin [43] represents a family of methods where a set of body forces is used to represent the effect of an object to the flow. Initially the method was used to study fluid-structure interaction problems in the cardiovascular circulation [44, 45]. In these computations the vascular boundary was modeled as a set of elements linked by springs. As a result the forces required to enforce boundary conditions could be evaluated in a straightforward manner (i.e. Hooke's law). In the case of rigid immersed bodies, however, the corresponding forces are not known *a priori* and must be calculated by some feedback algorithm. Lai and Peskin [30] suggested a formulation where the body is allowed to move a little -rather than being fixed- by connecting it to a very stiff spring. They tested this approach for the flow around a cylinder with satisfactory results, although the specification of the stiffness constant is somehow *ad hoc*. Goldstein *et al.* [19] introduced an alternative approach where a feedback-forcing scheme is used to asymptotically enforce the desired boundary conditions on a solid boundary. Application of the method to three-dimensional computations of turbulent flow in plane and ribbed channels yielded results in good agreement with reference data [20, 21]. An advantage of immersed-boundary formulations is their straightforward implementation in existing solvers (modifications are confined to the RHS of the equations of motion). On the other hand, the need for a smooth transition between the fluid and the solid body spreads the forcing function over several grid cells and introduces some blurring between the two regions. This feature can decrease the order of accuracy of the scheme near the body or increase the resolution requirements making their use in turbulent flows problematic.

Another class of methods which does not suffer from the "blurring" mentioned above are the so-called Cartesian or cut-cell formulations. In this case the solid boundary is tracked as a sharp interface and the grid cells at the body interface are modified according to their intersections with the underlying Cartesian grid. The discrete operators at these cells are then modified to reflect the desired boundary conditions. Successful applications of such methods in two dimensional flows with stationary boundaries can be found in [73, 7, 52, 64, 71]. However, due the large number of possible intersections between the grid and the boundary a variety of interface-cells is generated leading to an equally large number of special treatments. Also in complex configurations the unavoidable generation of irregularly shaped cells with very small size can have an adverse impact of the conservation and stability properties of the solver. Recently Ye *et al.* [71] suggested a cell merging scheme to address this problem. This formulation was also extended to treat moving boundaries with good results for a variety of two-dimensional problems [63]. The extension of the methodology in complex three-dimensional configurations remains to be investigated.

Recently, Fadlun *et al.* [15] introduced an embedded-boundary method, which shares a number of features with both approaches discussed above. As in immersed-boundary formulations the method still utilizes a force-field to enforce boundary conditions. In this case, however, the forces are not specified in the continuous space by means of some physical arguments, but rather in the discrete space by directly requiring the solution to respect the desired boundary conditions. This process is equivalent to a local reconstruction of the solution near the interface and enforces the desired boundary conditions “exactly”, as in cut-cell formulations. The encouraging results reported in [65, 15] together with the straightforward implementation of the method in existing Navier-Stokes solvers, motivated a number of recent studies, where alternative embedded-boundary formulations based on the principals outlined above have been proposed. The main difference between them is the way the solution is reconstructed near the interface. In Fadlun *et al.* [15] and Balaras [4], for example, the solution is reconstructed at the fluid nodes closest to the immersed boundary (fluid points with at least one neighbor in the solid phase). In the former study a one-dimensional interpolation scheme along an arbitrary grid line is used for this purpose, while in the latter the reconstruction is performed along the well defined line normal to the interface. In Kim *et al.* [29], Majumdar *et al.* [35], or Tseng and Ferziger [61] on the other hand, the solution is reconstructed at 'ghost-cells', which are points inside the solid phase with at least one neighbor in the fluid phase. Despite the different strategies, however, both approaches have the velocity boundary conditions 'implicitly' build into the reconstruction stencil, which results in a sharply defined interface.

A related issue, which also has implications on the overall accuracy of the approach, is the way the direct-forcing is incorporated into the time advancement scheme (fractional-step methods are almost exclusively used in the all the above studies). In [15], for example, the use of a one-dimensional interpolation strategy facilitates the imposition of boundary conditions to the predicted velocity by directly modifying the coefficients of the standard linear system. In [4] where a multidimensional reconstruction is used in the framework of an explicit time advancement scheme the same result can be simply achieved by directly modifying the predicted velocity field. As a result in both cases the actual value of the forcing function is not explicitly computed and the boundary conditions on the immersed interface are satisfied by a direct modification of the discrete operators. When a semi-implicit scheme (i.e. third-order Runge-Kutta scheme for advective terms, and Crank-Nikolson for the viscous terms) is used, however, the multi-dimensional reconstruction stencil involves values from the yet to be determined predicted velocity. To implicitly satisfy the boundary conditions, as in [15, 4], one needs to modify the standard linear system of equations in the predictor step, which is now going to become a sparse system with a costly solution. To alleviate this problem, Kim *et al.* [29], proposed an approach where the forcing is estimated by taking a preliminary fully explicit 'predictor' step (a forward Euler step for the viscous terms in their case). The estimated forcing function is then incorporated into the RHS of the regular linear system to enforce the desired boundary conditions. Balaras and Yang

[6] used a similar approach in the framework of a semi-implicit, finite-difference, fractional step method in cylindrical coordinates with good results.

The boundary motion/deformation over a fixed grid is usually the source of additional complications in most of the non-boundary-conforming strategies that were discussed in the previous paragraphs. In classical immersed-boundary formulations [43, 30], such problems are usually minimal due to the smooth transition from the solid to the fluid. In cut-cell [73, 7, 52, 64, 71] or embedded-boundary formulations [15, 29, 61, 4], however, complications are encountered due the fact that the role of the Eulerian grid points near the interface changes from timestep to timestep (for example a point in the solid body can became a point in the fluid at the next timestep and vise-versa), as the body moves through the fixed grid. As a result the velocity and pressure for some points in the flow will get non-physical values due to their previous association with the solid phase. In the framework of cut-cell formulations Udaykumar *et al.* [63] proposed a cell merging scheme to handle the cells that emerge from the body which they call 'freshly-cleared' cells. In embedded-boundary formulations on the other hand, no systematic study that identifies and addresses problems associated with large boundary motions has been reported. Due to the explicit dependence of such methods on the details of the adopted numerical method it is difficult to formulate a general approach. In the case of 'ghost-cell' methods, for example, as the body moves through the fixed grid some of the 'ghost-cells' will emerge into the fluid and will became fluid cells. Since they were previously in the solid they have no history in the fluid phase and no physically realizable value for the velocity and pressure at the previous timestep(s).

As a consequence their treatment to some degree would have to be ad-hoc. In formulations where the solution is reconstructed at the fluid nodes closest to the boundary, (i.e. [15]), the points that emerge from the solid become the boundary points that are central to the reconstruction procedure, and therefore their history in the fluid phase is irrelevant. The points that require special treatment in this case are the boundary points that move further into the fluid.

The later approach is the one that will be discussed in detail in the present study, where a variance of the embedded-boundary formulation proposed in [4] will be extended to moving boundary problems. To address the problems due to boundary motion we will propose a field-extension strategy that practically extends the pressure and velocity fields into the solid body to implicitly treat problematic grid cells in a robust manner. We will also discuss a straightforward method to compute the local force distribution on a complex body that is based on the same reconstruction stencil used for the velocity field. As it will be demonstrated in the results section the reconstructed local forces are in good agreement with reference results obtained with boundary-conforming formulations.

1.3 Present Contributions

The major contributions of this work are summarized as follows:

- Extension of the Cartesian, finite-difference, fractional-step, Navier-Stokes solver [3] in cylindrical coordinates, including dynamic SGS models for LES.
- Development of a generalized local reconstruction scheme for immersed bound-

ary treatment in the embedded-boundary formulation.

- Development of a novel field-extension strategy for treating grid points emerging from the solid into the fluid phase.

- Development of a strong coupling scheme for the simulation of Fluid-Structure Interaction problems.

- First comprehensive validation of non-boundary-conforming methods on surface force distribution for moving boundary problems.

- First simulation of transitional flow past prosthetic mechanical heart valve with moving leaflets in a realistic geometry.

1.4 Outline

The outline of this dissertation is as follows:

- In Chap. 2, the basic fluid solver is described in detail. The governing equations, the subgrid-scale stress model, spatial discretization and time advancement schemes in both Cartesian and cylindrical coordinates are all documented there. The discussion of boundary conditions is included. A validation of the basic fluid solver is performed for the LES of fully developed turbulent pipe flow.

- Chap. 3 presents the embedded-boundary formulation, the strong coupling scheme for fluid-structure interaction problems, and the parallelization of the whole algorithm. For the embedded-boundary formulation, the establishment of the interface-grid relation, the treatment of stationary immersed boundaries and moving immersed boundaries are given in detail. A novel surface force calculation procedure, which computes the hydrodynamic loads to the structure, is also covered.

- A series of two-dimensional cases with increasing complexity are presented in Chap. 4. First, the two dimensional flow around one element of a moving cylinder-array in a planar channel is computed to demonstrate the formal accuracy of the method. Then, laminar flow problems involving prescribed motions of two-dimensional bodies are simulated: the flow induced by the harmonic in-line oscillation of a circular cylinder in a quiescent fluid and the flow from a transversely oscillating cylinder in a free-stream. Finally, the vortex-induced vibrations of circular cylinders are used in the present study to evaluate the accuracy and efficiency of the proposed fluid-structure interaction scheme. Two different configurations are considered: one degree-of-freedom oscillations in the cross-stream direction and two degree-of-freedom oscillations in both the streamwise and cross-stream directions.

- In Chap. 5, four three-dimensional large-scale simulations are presented. The first case is the flow past a sphere at Reynolds numbers from 50, to 300, and 1000. The transitional flow past an airfoil at Reynold number $10,000$ is also simulated. Then, LES of flow over a traveling wave are presented in comparison to reference DNS results. To demonstrate the robustness and applicability of the method in highly unsteady turbulent flows that involve multiple moving complex boundaries, the flow around a bileaflet prosthetic heart valve is also presented.

- Finally, in Chap. 6 conclusions and future directions are given.

Chapter 2

Basic Navier-Stokes Solver

The aim of this study is to develop a computational tool for DNS and LES of turbulent flows interacting with complex moving boundaries. Rigid bodies undergoing motion that is either prescribed or governed by additional ODEs that have to be solved as a coupled system together with the Navier-Stokes equations are considered. The overall solver will be given in two parts. In this chapter the basic solver in Cartesian and cylindrical coordinates is discussed in detail. A brief validation for fully developed turbulent pipe flow is also included. The treatment of immersed boundary and fluid-structure coupling scheme will be discussed in the next chapter.

2.1 Formulation in Cartesian Coordinates

2.1.1 Governing Equations

The governing equations for unsteady, incompressible, viscous flow of a Newtonian fluid with constant density can be written in Cartesian coordinates as follows:

$$\frac{\partial u_i}{\partial t} + \frac{\partial (u_i u_j)}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{1}{Re}\frac{\partial^2 u_i}{\partial x_j \partial x_j}, \tag{2.1}$$

$$\frac{\partial u_i}{\partial x_i} = 0, \tag{2.2}$$

where x_i and x_j $(i, j = 1, 2, 3)$ are the Cartesian coordinates, u_i and u_j are the velocity components in the corresponding directions, normalized by a reference velocity U, t is the time normalized by L/U with L the reference length scale, p is the pressure normalized by ρU^2 with ρ the density of the fluid, and Re is the Reynolds number defined as $Re = \rho U L/\mu$ with μ the dynamic viscosity of the fluid.

For the laminar flow cases or DNS of turbulent flow cases considered in this study, the above equations are directly integrated in space and time without introducing any model. In the LES approach, however, a spatial filtering operation is applied and it separates the large, energy carrying eddies, which are directly resolved, from the small scales, which are modeled. In the present finite-difference implementation a top-hat filter is implicitly applied by the discrete operators. The resulting equations governing the evolution of the large scales have the following form:

$$\frac{\partial \overline{u}_i}{\partial t} + \frac{\partial (\overline{u}_i \overline{u}_j)}{\partial x_j} = -\frac{\partial \overline{p}}{\partial x_i} + \frac{1}{Re} \frac{\partial^2 \overline{u}_i}{\partial x_j \partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j}, \tag{2.3}$$

$$\frac{\partial \overline{u}_i}{\partial x_i} = 0, \tag{2.4}$$

where the overbar denotes a filtered variable and

$$\tau_{ij} = \overline{u_i u_j} - \overline{u}_i \overline{u}_j, \tag{2.5}$$

are the subgrid scale (SGS) stresses.

2.1.2 Subgrid Scale Stress Models

The spatial filtering operation on the Navier-Stokes equations produces the subgrid scale stresses, which have to be parametrized in order to close the system of equations (2.3) and (2.4).

The most widely used model for the subgrid scale stresses is the Smagorinsky eddy viscosity model [56]. In this model, the subgrid scale stresses are related to the resolved strain rate tensor as follows:

$$\tau_{ij} - \frac{1}{3}\delta_{ij}\tau_{kk} = -2\nu_t\overline{S}_{ij} = -2C\overline{\Delta}^2|\overline{S}|\overline{S}_{ij} \tag{2.6}$$

where δ_{ij} is Kronecker's delta, $\overline{S}_{ij}$ is the resolved strain rate tensor,

$$\overline{S}_{ij} = \frac{1}{2}\left(\frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{u}_j}{\partial x_i}\right). \tag{2.7}$$

with magnitude:

$$|\overline{S}| = \sqrt{2\overline{S}_{ij}\overline{S}_{ij}}, \tag{2.8}$$

ν_t is the turbulent eddy viscosity, $\overline{\Delta}$ is the filter size, defined as $(\Delta x \Delta y \Delta z)^{1/3}$ with Δx, Δy, and Δz the local grid size in x, y, and z coordinate direction, respectively.

In the model C is a user-specified constant which usually varies between $0.1 \sim 0.2$ depending on the flow [32, 12]. This, together with the fact that the model does not vanish in the laminar flow regions and does not have the proper limiting behavior near solid boundaries make its use in complex flows problematic.

The dynamic modeling procedure [17] addresses most of the above deficiencies. The basic idea behind the family of dynamic models is to use information available at the resolved scales to predict the unresolved subgrid scales by the resolved scales.

An additional spatial filter, or *test* filter, is applied to the Navier-Stokes equations for this purpose, which are filtered by the *grid* filter. The filter width of the test-filter, $\widehat{\overline{\Delta}}$, is larger (usually double size) than that of the grid-filter, $\overline{\Delta}$. Similar to the subgrid scale stresses, the subtest scale stresses will appear in the resulting equations:

$$T_{ij} = \widehat{\overline{u}_i\overline{u}_j} - \widehat{\overline{u}}_i\widehat{\overline{u}}_j. \tag{2.9}$$

Assume the subtest scale stresses can be modeled using the same model as the subgrid scale stresses in Eq. (2.6):

$$T_{ij} - \frac{1}{3}\delta_{ij}T_{kk} = -2C\widehat{\overline{\Delta}}^2|\widehat{\overline{S}}|\widehat{\overline{S}}_{ij}, \tag{2.10}$$

where the filtered strain rate tensor is

$$\widehat{\overline{S}}_{ij} = \frac{1}{2}\left(\frac{\partial\widehat{\overline{u}}_i}{\partial x_j} + \frac{\partial\widehat{\overline{u}}_j}{\partial x_i}\right), \quad |\widehat{\overline{S}}| = \sqrt{2\widehat{\overline{S}}_{ij}\widehat{\overline{S}}_{ij}}. \tag{2.11}$$

The subgrid scale stresses and the subtest scale stresses can be related to the smallest resolved turbulent stresses, $\mathcal{L}_{ij}$, and the *Germano* identity [18] as follows

$$\mathcal{L}_{ij} = T_{ij} - \widehat{\tau}_{ij} = \widehat{\overline{u}_i\overline{u}_j} - \widehat{\overline{u}}_i\widehat{\overline{u}}_j. \tag{2.12}$$

with the substitution of Eqs. (2.6) and (2.10) into Eq. (2.12), an equation with the only unknown parameter C is obtained

$$\mathcal{L}_{ij} - \frac{1}{3}\delta_{ij}\mathcal{L}_{kk} = -2\overline{\Delta}^2\left[(\widehat{\overline{\Delta}}/\overline{\Delta})^2C|\widehat{\overline{S}}|\widehat{\overline{S}}_{ij} - \widehat{C|\overline{S}|\overline{S}_{ij}}\right]. \tag{2.13}$$

In this study, the trapezoidal rule is used to evaluate the test-filtering operation (On a uniform grid, the trapezoidal rule gives a sequence of weights $(W_{-1}, W_0, W_1) =$

$(1/4, 1/2, 1/4)$), and consistently, the ratio between the test filter and grid filter $\widehat{\overline{\Delta}}/\overline{\Delta}$ is chosen to be $\sqrt{6}$ [34].

Various approaches were used to solve the above equation. In general, C is assumed to be a weak function in space and could be taken outside the test-filter operation. One commonly used approach is the least-square method proposed by Lilly [33]:

$$C(\mathbf{x}, t) = -\frac{1}{2}\frac{\mathcal{L}_{ij} M_{ij}}{M_{ij} M_{ij}}, \tag{2.14}$$

where M_{ij} is:

$$M_{ij} = \widehat{\overline{\Delta}}^2 |\widehat{\overline{S}}| \widehat{\overline{S}}_{ij} - \overline{\Delta}^2 \widehat{|\overline{S}| \overline{S}_{ij}}. \tag{2.15}$$

The model parameter C given by (2.14) may become locally negative, resulting in negative eddy viscosity that are known to cause numerical instabilities. Averaging C in the homogeneous directions can help alleviate this problem, but in the present study the target applications are biological flows with complex geometries and moving boundaries, in which a homogeneous direction is rarely available in practical problems. For this reason, the Lagrangian averaging procedure proposed by Meneveau *et al.* [36] is adopted. In this approach, the averaging is done over the fluid particle pathlines rather than the homogeneous directions and the Lagrangian averaging operator is defined as:

$$< \varphi >= \int_{-\infty}^{t} \varphi(t') W(t - t') dt', \tag{2.16}$$

where $W(t)$ is an exponential weighting function chosen to give more weight to recent times in flow history. Details of the Lagrangian dynamic model can be found in [36].

If the SGS stresses in Eq. (2.3) are substituted by the model in Eq. (2.6) and $\overline{p}$ is replaced by $\overline{p}+\frac{1}{3}\tau_{kk}$, then the governing equations for LES can be written as:

$$\frac{\partial \overline{u}_i}{\partial t} = -\frac{\partial(\overline{u}_i\overline{u}_j)}{\partial x_j} - \frac{\partial \overline{p}}{\partial x_i} + \frac{\partial}{\partial x_j}\left(\nu\frac{\partial \overline{u}_i}{\partial x_j}\right) + \frac{\partial}{\partial x_j}\left(\nu_t\frac{\partial \overline{u}_j}{\partial x_i}\right), \tag{2.17}$$

$$\frac{\partial \overline{u}_i}{\partial x_i} = 0, \tag{2.18}$$

where $\nu = \nu_t + 1/Re$ is the total viscosity. Note that in the cross diffusive terms, the constant molecular viscosity $1/Re$ is removed due to the continuity equation.

2.1.3 Spatial Discretization

A standard second-order central-difference scheme on a staggered grid is used in the present study. A typical grid cell and the staggered variable arrangement is shown in Fig. 2.1, with the pressure and scalar variables located at the center of the grid cell and velocity components located at the cell face centers. Below the half-cell nomenclature (such as $i \pm \frac{1}{2}$, $j \pm \frac{1}{2}$, $k \pm \frac{1}{2}$) for velocity components is discarded to be consistent with the numerical implementation in the code.

The finite difference schemes can be either constructed in physical space or computational space. Usually, the construction is more convenient when the non-uniform grids in physical space are converted to the uniform grids in computational space. Here, the mapping between the physical space and computational space is

$$x = x(\xi), \quad y = y(\eta), \quad z = z(\zeta), \tag{2.19}$$

and

$$\xi = \xi(x), \quad \eta = \eta(y), \quad \zeta = \zeta(z), \tag{2.20}$$

where ξ, η, and ζ are the coordinates in the computational space, which correspond to x, y, and z in physical space, respectively. Therefore, the derivatives in physical space are transformed to computational space as follows:

$$\frac{\partial}{\partial x} = \frac{\partial}{\partial \xi}\xi_x, \quad \frac{\partial}{\partial y} = \frac{\partial}{\partial \eta}\eta_y, \quad \frac{\partial}{\partial z} = \frac{\partial}{\partial \zeta}\zeta_z, \tag{2.21}$$

with

$$\xi_x = \frac{2\Delta\xi}{x_{i+1} - x_{i-1}}, \quad \eta_y = \frac{2\Delta\eta}{y_{j+1} - y_{j-1}}, \quad \zeta_z = \frac{2\Delta\zeta}{z_{k+1} - z_{k-1}}, \tag{2.22}$$

where x_{i-1}, x_{i+1}, y_{j-1}, y_{j+1}, z_{k-1}, and z_{k+1} are discrete coordinates of the Cartesian grid in physical space. A sketch of two-dimensional uniform grid and variable arrangement in $x - z$ coordinates is shown in Fig. 2.2, in which the definition of x, z grid lines and relative imposition of variables are given. In the computational space, the cell sizes are set to unity, i.e., $\Delta\xi = \Delta\eta = \Delta\zeta = 1$. By introducing this

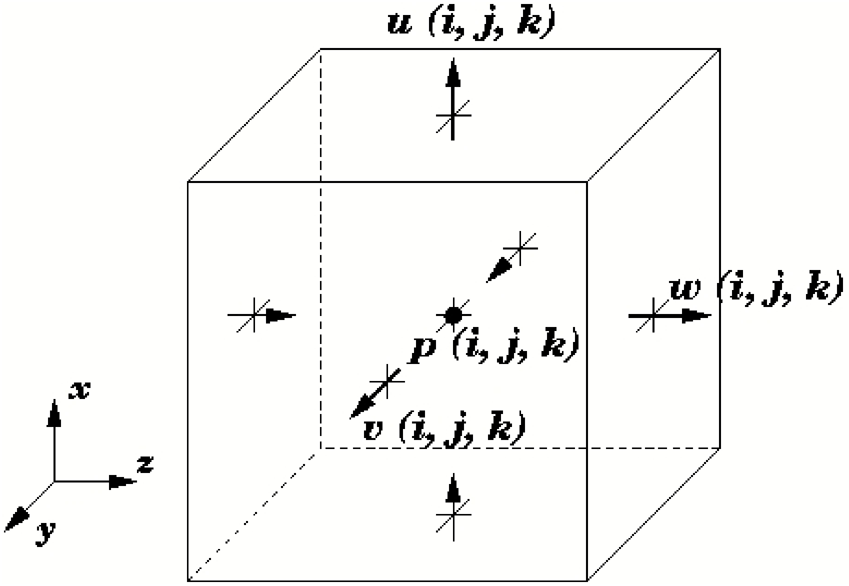


Figure 2.1: A sketch of grid cell and the variable arrangement in Cartesian coordinates.

transformation, the Eqs. (2.17) and (2.18) now can be written as

$$\begin{aligned}
\frac{\partial u_x}{\partial t} &= - \left[\xi_x \frac{\partial (u_x u_x)}{\partial \xi} + \eta_y \frac{\partial (u_y u_x)}{\partial \eta} + \zeta_z \frac{\partial (u_z u_x)}{\partial \zeta}\right] - \xi_x \frac{\partial p}{\partial \xi} \\
&+ \left[\xi_x \frac{\partial}{\partial \xi}\left(\nu \xi_x \frac{\partial u_x}{\partial \xi}\right) + \eta_y \frac{\partial}{\partial \eta}\left(\nu \eta_y \frac{\partial u_x}{\partial \eta}\right) + \zeta_z \frac{\partial}{\partial \zeta}\left(\nu \zeta_z \frac{\partial u_x}{\partial \zeta}\right)\right] \\
&+ \left[\xi_x \frac{\partial}{\partial \xi}\left(\nu_t \xi_x \frac{\partial u_x}{\partial \xi}\right) + \eta_y \frac{\partial}{\partial \eta}\left(\nu_t \xi_x \frac{\partial u_y}{\partial \xi}\right) + \zeta_z \frac{\partial}{\partial \zeta}\left(\nu_t \xi_x \frac{\partial u_z}{\partial \xi}\right)\right], \qquad (2.23)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial u_y}{\partial t} &= - \left[\xi_x \frac{\partial (u_x u_y)}{\partial \xi} + \eta_y \frac{\partial (u_y u_y)}{\partial \eta} + \zeta_z \frac{\partial (u_z u_y)}{\partial \zeta}\right] - \eta_y \frac{\partial p}{\partial \eta} \\
&+ \left[\xi_x \frac{\partial}{\partial \xi}\left(\nu \xi_x \frac{\partial u_y}{\partial \xi}\right) + \eta_y \frac{\partial}{\partial \eta}\left(\nu \eta_y \frac{\partial u_y}{\partial \eta}\right) + \zeta_z \frac{\partial}{\partial \zeta}\left(\nu \zeta_z \frac{\partial u_y}{\partial \zeta}\right)\right] \\
&+ \left[\xi_x \frac{\partial}{\partial \xi}\left(\nu_t \eta_y \frac{\partial u_x}{\partial \eta}\right) + \eta_y \frac{\partial}{\partial \eta}\left(\nu_t \eta_y \frac{\partial u_y}{\partial \eta}\right) + \zeta_z \frac{\partial}{\partial \zeta}\left(\nu_t \eta_y \frac{\partial u_z}{\partial \eta}\right)\right], \qquad (2.24)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial u_z}{\partial t} &= - \left[\xi_x \frac{\partial (u_x u_z)}{\partial \xi} + \eta_y \frac{\partial (u_y u_z)}{\partial \eta} + \zeta_z \frac{\partial (u_z u_z)}{\partial \zeta}\right] - \zeta_z \frac{\partial p}{\partial \zeta} \\
&+ \left[\xi_x \frac{\partial}{\partial \xi}\left(\nu \xi_x \frac{\partial u_z}{\partial \xi}\right) + \eta_y \frac{\partial}{\partial \eta}\left(\nu \eta_y \frac{\partial u_z}{\partial \eta}\right) + \zeta_z \frac{\partial}{\partial \zeta}\left(\nu \zeta_z \frac{\partial u_z}{\partial \zeta}\right)\right] \\
&+ \left[\xi_x \frac{\partial}{\partial \xi}\left(\nu_t \zeta_z \frac{\partial u_x}{\partial \zeta}\right) + \eta_y \frac{\partial}{\partial \eta}\left(\nu_t \zeta_z \frac{\partial u_y}{\partial \zeta}\right) + \zeta_z \frac{\partial}{\partial \zeta}\left(\nu_t \zeta_z \frac{\partial u_z}{\partial \zeta}\right)\right] \qquad (2.25)
\end{aligned}$$

$$\xi_x \frac{\partial u_x}{\partial \xi} + \eta_y \frac{\partial u_y}{\partial \eta} + \zeta_z \frac{\partial u_z}{\partial \zeta} = 0. \tag{2.26}$$

The discretization of above equations on a staggered grid in computational space is described given in detail in [3]. Here the discretization of the diagonal convective term in the w-momentum equation is given as an example. Note that an arithmetic average is used to obtain variables at grid locations between the points where the variables are defined (see Fig. 2.2):

$$\begin{aligned}
\left.\frac{\partial (ww)}{\partial z}\right|_{i,j,k}^{w} &\approx \left.\zeta_z\right|_k^w \left.\frac{\delta (ww)}{\delta \zeta}\right|_{i,j,k}^{w} \\
&= \left.\zeta_z\right|_k^w \frac{1}{\Delta \zeta}\left[\frac{w_{i,j,k} + w_{i,j,k+1}}{2} - \frac{w_{i,j,k-1} + w_{i,j,k}}{2}\right]. \qquad (2.27)
\end{aligned}$$

The diagonal diffusive term can be evaluated as

$$
\begin{aligned}
\frac{\partial}{\partial z}\left(\nu\frac{\partial w}{\partial z}\right)\Big|_{i,j,k}^{w} &\approx \zeta_z\big|_k^w \frac{\partial}{\partial \zeta}\left(\nu\zeta_z\frac{\partial w}{\partial \zeta}\right)\Big|_{i,j,k}^{w} \\
&= \zeta_z\big|_k^w \frac{1}{\Delta\zeta}\left[\left(\nu\zeta_z\frac{\delta w}{\delta\zeta}\right)\Big|_{i,j,k+1}^{p} - \left(\nu\zeta_z\frac{\delta w}{\delta\zeta}\right)\Big|_{i,j,k}^{p}\right] \qquad (2.28)
\end{aligned}
$$

where,

$$
\begin{aligned}
\left(\nu\zeta_z\frac{\delta w}{\delta\zeta}\right)\Big|_{i,j,k+1}^{p} &= \frac{\nu|_{i,j,k}+\nu|_{i,j,k+1}}{2}\zeta_z\big|_{k+1}^{p}\frac{w|_{i,j,k+1}-w|_{i,j,k}}{\Delta\zeta}, \\
\left(\nu\zeta_z\frac{\delta w}{\delta\zeta}\right)\Big|_{i,j,k}^{p} &= \frac{\nu|_{i,j,k-1}+\nu|_{i,j,k}}{2}\zeta_z\big|_{k}^{p}\;\frac{w|_{i,j,k}-w|_{i,j,k-1}}{\Delta\zeta}. \qquad (2.29)
\end{aligned}
$$

In above equations, the symbols $|_{i,j,k}^{p}$ and $|_{i,j,k}^{w}$ refer to quantities evaluated at p points and w points, respectively. Hereinafter, $u_x \leftrightarrow u \leftrightarrow u_r$, $u_y \leftrightarrow v \leftrightarrow u_\theta$, and $u_z \leftrightarrow w \leftrightarrow u_z$ will be used interchangeably for both Cartesian and cylindrical coordinates.

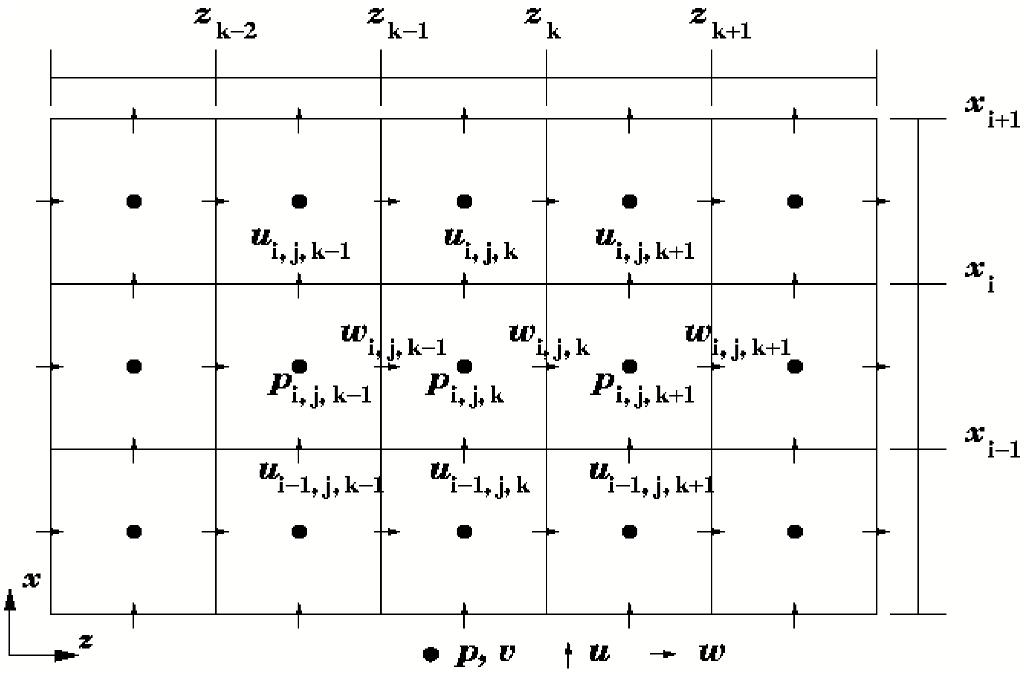


Figure 2.2: Staggered grid and variable arrangement in $x - z$ coordinates.

2.1.4 Time Advancement Scheme

The fractional step method is used to integrate the governing equations in time. In the current work, there are several choices of time advancement schemes. For solving the equations in Cartesian coordinates, only explicit schemes are used in all simulations, although the implicit scheme, which is discussed in the next section, can also be adopted. In particular, the second-order Adams-Bashforth (AB2) scheme or the low-storage third-order Runge-Kutta (RK3) scheme is used:

$$\begin{aligned}
\frac{\hat{u}_i^k - u_i^{k-1}}{\Delta t} &= \gamma_k A\left(u_i^{k-1}\right) + \rho_k A\left(u_i^{k-2}\right) - \alpha_k \frac{\partial p^{k-1}}{\partial x_i}, && (2.30)\\
\frac{\partial^2 \phi^k}{\partial x_i \partial x_i} &= \frac{1}{\alpha_k \Delta t}\frac{\partial \hat{u}_i^k}{\partial x_i}, && (2.31)\\
u_i^k &= \hat{u}_i^k - \alpha_k \Delta t \frac{\partial \phi^k}{\partial x_i}, && (2.32)\\
p^k &= p^{k-1} + \phi^k, && (2.33)
\end{aligned}$$

where k is the substep index, which ranges from 1 to 3 for Runge-Kutta scheme and equals 1 for Adams-Bashforth scheme, $\hat{u}_i^k$ is the intermediate velocity and ϕ is the scalar used to project $\hat{u}_i^k$ into a divergence-free space, A is the spatial operator containing the convective, viscous and SGS terms, Δt is the time step, the RK3 coefficients are

$$\begin{array}{llllll}
\alpha_1 = & 8/15, & \gamma_1 = & 8/15, & \rho_1 = & 0; \\
\alpha_2 = & 2/15, & \gamma_2 = & 5/12, & \rho_2 = & -17/60; \\
\alpha_3 = & 1/3, & \gamma_3 = & 3/4, & \rho_3 = & -5/12,
\end{array} \qquad (2.34)$$

with

$$\sum_{k=1}^{3} \alpha_k = \sum_{k=1}^{3} (\gamma_k + \rho_k) = 1, \qquad (2.35)$$

and the AB2 coefficients are

$$\alpha_1 = \ 1, \quad \gamma_1 = \ 3/2, \quad \rho_1 = \ -1/2. \tag{2.36}$$

with

$$\alpha_1 = \gamma_1 + \rho_1 = 1. \tag{2.37}$$

The following stability criterion (or the generalized CFL number including the time step constraint from the viscous terms) is adopted [1]:

$$CFL = \Delta t \left[\frac{|u|}{\Delta x} + \frac{|v|}{\Delta y} + \frac{|w|}{\Delta z} + 2\nu \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2} \right) \right]. \tag{2.38}$$

The theoretical stability limit for the third-order Runge-Kutta scheme is $\sqrt{3}$. Since the cross-terms are not included in above equation, the actual CFL number in the simulations is lower. $CFL = 1.2$ is used in most of the simulations in this work. On the other hand, the theoretical stability limit for the second-order Adams-Bashforth scheme is $CFL < 1$. However, the actual CFL number in the simulations has to be much lower, e.g., $CFL = 0.6$ to prevent the code from blowing up. In this study, the Adams-Bashforth scheme is used together with the fourth-order Hamming's predictor-corrector for fluid-structure interaction problems, in which a constant timestep is used to maintain the accuracy of the schemes.

2.2 Transformation in Cylindrical Coordinates

Cylindrical coordinates are convenient for the description of many fluid flows of practical interest. Also, they provide an easy way of grid adaptation for a variety of external flows, e.g., insect and bird flight, fish swimming, etc., when combined with immersed boundary formulations.

2.2.1 Governing Equations

The corresponding governing equations in cylindrical coordinates can be written as follows:

$$\begin{aligned}
&\frac{\partial u_r}{\partial t} + \frac{1}{r}\frac{\partial (r u_r u_r)}{\partial r} + \frac{1}{r}\frac{\partial (u_\theta u_r)}{\partial \theta} + \frac{\partial (u_z u_r)}{\partial z} - \frac{u_\theta^2}{r} \\
&= \quad -\frac{\partial p}{\partial r} + \frac{1}{Re}\left[\frac{\partial}{\partial r}\left(\frac{1}{r}\frac{\partial}{\partial r}(r u_r)\right) + \frac{1}{r^2}\frac{\partial^2 u_r}{\partial \theta^2} + \frac{\partial^2 u_r}{\partial z^2} - \frac{2}{r^2}\frac{\partial u_\theta}{\partial \theta}\right], \qquad (2.39) \\
&\frac{\partial u_\theta}{\partial t} + \frac{1}{r}\frac{\partial (r u_r u_\theta)}{\partial r} + \frac{1}{r}\frac{\partial (u_\theta u_\theta)}{\partial \theta} + \frac{\partial (u_z u_\theta)}{\partial z} + \frac{u_r u_\theta}{r} \\
&= \quad -\frac{1}{r}\frac{\partial p}{\partial \theta} + \frac{1}{Re}\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u_\theta}{\partial r}\right) - \frac{u_\theta}{r^2} + \frac{1}{r^2}\frac{\partial^2 u_\theta}{\partial \theta^2} + \frac{\partial^2 u_\theta}{\partial z^2} + \frac{2}{r^2}\frac{\partial u_r}{\partial \theta}\right], \qquad (2.40) \\
&\frac{\partial u_z}{\partial t} + \frac{1}{r}\frac{\partial (r u_r u_z)}{\partial r} + \frac{1}{r}\frac{\partial (u_\theta u_z)}{\partial \theta} + \frac{\partial (u_z u_z)}{\partial z} \\
&= \quad -\frac{\partial p}{\partial z} + \frac{1}{Re}\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u_z}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 u_z}{\partial \theta^2} + \frac{\partial^2 u_z}{\partial z^2}\right], \qquad (2.41)
\end{aligned}$$

$$\frac{1}{r}\frac{\partial (r u_r)}{\partial r} + \frac{1}{r}\frac{\partial u_\theta}{\partial \theta} + \frac{\partial u_z}{\partial z} = 0, \qquad (2.42)$$

where the subscript r represents the radial direction, θ represents the azimuthal direction, and z represents the axial direction (or streamwise direction in this study).

As with the case of Cartesian coordinates a filtering operation can be applied to the above equations to define the resolved and unresolved scales. The resulting equations governing the evolution of the large scales for incompressible flow in cylindrical coordinates can be written as follows:

$$\begin{aligned}
\frac{\partial \overline{u}_r}{\partial t} = & - \left[\frac{1}{r}\frac{\partial (r\overline{u}_r\overline{u}_r)}{\partial r} + \frac{1}{r}\frac{\partial (\overline{u}_\theta\overline{u}_r)}{\partial \theta} + \frac{\partial (\overline{u}_z\overline{u}_r)}{\partial z}\right] - \frac{\partial \overline{p}}{\partial r} \\
& + \frac{\partial}{\partial r}\left(\nu\frac{1}{r}\frac{\partial}{\partial r}(r\overline{u}_r)\right) + \frac{1}{r^2}\frac{\partial}{\partial \theta}\left(\nu\frac{\partial \overline{u}_r}{\partial \theta}\right) + \frac{\partial}{\partial z}\left(\nu\frac{\partial \overline{u}_r}{\partial z}\right) \\
& + \frac{\partial}{\partial r}\left(\nu_t\frac{1}{r}\frac{\partial}{\partial r}(r\overline{u}_r)\right) + \frac{1}{r}\frac{\partial}{\partial \theta}\left(\nu_t\frac{\partial \overline{u}_\theta}{\partial r}\right) + \frac{\partial}{\partial z}\left(\nu_t\frac{\partial \overline{u}_z}{\partial r}\right) \\
& + \frac{\overline{u}_\theta^2}{r} \\
& - \nu\frac{2}{r^2}\frac{\partial \overline{u}_\theta}{\partial \theta} - \nu_t\frac{1}{r^2}\frac{\partial \overline{u}_\theta}{\partial \theta} - \frac{\overline{u}_\theta}{r^2}\frac{\partial \nu_t}{\partial \theta} - \frac{2\overline{u}_r}{r}\frac{\partial \nu_t}{\partial r},
\end{aligned} \tag{2.43}$$

$$\begin{aligned}
\frac{\partial \overline{u}_\theta}{\partial t} = & - \left[\frac{1}{r}\frac{\partial (r\overline{u}_r\overline{u}_\theta)}{\partial r} + \frac{1}{r}\frac{\partial (\overline{u}_\theta\overline{u}_\theta)}{\partial \theta} + \frac{\partial (\overline{u}_z\overline{u}_\theta)}{\partial z}\right] - \frac{1}{r}\frac{\partial \overline{p}}{\partial \theta} \\
& + \frac{1}{r}\frac{\partial}{\partial r}\left(\nu r\frac{\partial \overline{u}_\theta}{\partial r}\right) + \frac{1}{r^2}\frac{\partial}{\partial \theta}\left(\nu\frac{\partial \overline{u}_\theta}{\partial \theta}\right) + \frac{\partial}{\partial z}\left(\nu\frac{\partial \overline{u}_\theta}{\partial z}\right) \\
& + \frac{1}{r}\frac{\partial}{\partial r}\left(\nu_t\frac{\partial \overline{u}_r}{\partial \theta}\right) + \frac{1}{r^2}\frac{\partial}{\partial \theta}\left(\nu_t\frac{\partial \overline{u}_\theta}{\partial \theta}\right) + \frac{1}{r}\frac{\partial}{\partial z}\left(\nu_t\frac{\partial \overline{u}_z}{\partial \theta}\right) \\
& - \frac{\overline{u}_r\overline{u}_\theta}{r} \\
& + \nu\frac{2}{r^2}\frac{\partial \overline{u}_r}{\partial \theta} + \nu_t\frac{1}{r^2}\frac{\partial \overline{u}_r}{\partial \theta} - \nu\frac{\overline{u}_\theta}{r^2} + \frac{2\overline{u}_r}{r^2}\frac{\partial \nu_t}{\partial \theta} - \frac{\overline{u}_\theta}{r}\frac{\partial \nu_t}{\partial r},
\end{aligned} \tag{2.44}$$

$$\begin{aligned}
\frac{\partial \overline{u}_z}{\partial t} = & - \left[\frac{1}{r}\frac{\partial (r\overline{u}_r\overline{u}_z)}{\partial r} + \frac{1}{r}\frac{\partial (\overline{u}_\theta\overline{u}_z)}{\partial \theta} + \frac{\partial (\overline{u}_z\overline{u}_z)}{\partial z}\right] - \frac{\partial \overline{p}}{\partial z} \\
& + \frac{1}{r}\frac{\partial}{\partial r}\left(\nu r\frac{\partial \overline{u}_z}{\partial r}\right) + \frac{1}{r^2}\frac{\partial}{\partial \theta}\left(\nu\frac{\partial \overline{u}_z}{\partial \theta}\right) + \frac{\partial}{\partial z}\left(\nu\frac{\partial \overline{u}_z}{\partial z}\right) \\
& + \frac{1}{r}\frac{\partial}{\partial r}\left(\nu_t r\frac{\partial \overline{u}_r}{\partial z}\right) + \frac{1}{r}\frac{\partial}{\partial \theta}\left(\nu_t\frac{\partial \overline{u}_\theta}{\partial z}\right) + \frac{\partial}{\partial z}\left(\nu_t\frac{\partial \overline{u}_z}{\partial z}\right),
\end{aligned} \tag{2.45}$$

$$\frac{1}{r}\frac{\partial (r\overline{u}_r)}{\partial r} + \frac{1}{r}\frac{\partial \overline{u}_\theta}{\partial \theta} + \frac{\partial \overline{u}_z}{\partial z} = 0, \tag{2.46}$$

In the above equations ν_t is the SGS eddy viscosity as in the Cartesian formulation.

In the Smagorinsky model for example:

$$\tau_{ij} - \frac{1}{3}\delta_{ij}\tau_{kk} = -2\nu_t\overline{S}_{ij} = -2C\overline{\Delta}^2|\overline{S}|\overline{S}_{ij} \tag{2.47}$$

with $i, j, k = r, \theta, z$ and $\overline{\Delta} = (r\Delta r \Delta\theta \Delta z)^{1/3}$. Δr, $\Delta\theta$, and Δz the local grid size in r, θ, and z coordinate direction, respectively.

The strain rate tensor in cylindrical coordinates can be written as follows:

$$
\begin{aligned}
\overline{S}_{rr} &= \frac{\partial \overline{u}_r}{\partial r}, \\
\overline{S}_{r\theta} &= \frac{1}{2}\left(\frac{\partial \overline{u}_\theta}{\partial r} + \frac{\partial \overline{u}_r}{\partial \theta} - \frac{\overline{u}_\theta}{r}\right), \\
\overline{S}_{\theta\theta} &= \left(\frac{1}{r}\frac{\partial \overline{u}_\theta}{\partial \theta} + \frac{\overline{u}_r}{r}\right), \\
\overline{S}_{\theta z} &= \frac{1}{2}\left(\frac{\partial \overline{u}_\theta}{\partial z} + \frac{1}{r}\frac{\partial \overline{u}_z}{\partial \theta}\right), \\
\overline{S}_{zr} &= \frac{1}{2}\left(\frac{\partial \overline{u}_r}{\partial z} + \frac{\partial \overline{u}_z}{\partial r}\right), \\
\overline{S}_{zz} &= \frac{\partial \overline{u}_z}{\partial z}.
\end{aligned}
\tag{2.48}
$$

Note that the equations above in cylindrical coordinates and the equations in Cartesian coordinates have many common terms. This fact was utilized to formulate a generalized code for both cylindrical and Cartesian coordinates. The major differences between the equations in two coordinates systems are: a) r and $1/r$ terms in cylindrical coordinates; and b) additional terms due to coordinates curvature in cylindrical coordinates. In the current study, a flag is used in the input file to identify the reference frames. When this flag is switched on then all terms are calculated and r is used as defined in cylindrical coordinates. However, when this flag is switched off, then r is set to constant 1 and the additional terms (lines 4 and 5 in Eqs. (2.43) and (2.44)) are ignored in the calculation.

2.2.2 Spatial Discretization

A staggered grid is also used in the case of cylindrical coordinates. The variable arrangement for a typical grid cell and grid cell near the axis is shown in Fig. 2.3. Fig. 2.4 shows the staggered grid in $r - \theta$ plane. It is evident that only the radial velocity component is located at the centerline. The grid is uniform in the azimuthal direction, but non-uniform in the radial and axial (streamwise) directions. Here, the mapping between the physical space and computational space is derived as:

$$r = r(\xi), \quad \theta = \theta(\eta), \quad z = z(\zeta), \tag{2.49}$$

and

$$\xi = \xi(r), \quad \eta = \eta(\theta), \quad \zeta = \zeta(z), \tag{2.50}$$

where ξ, η, and ζ are the coordinates in the computational space, which correspond to r, θ, and z in physical space, respectively. Therefore, the derivatives in physical space are transformed to computational space as follows:

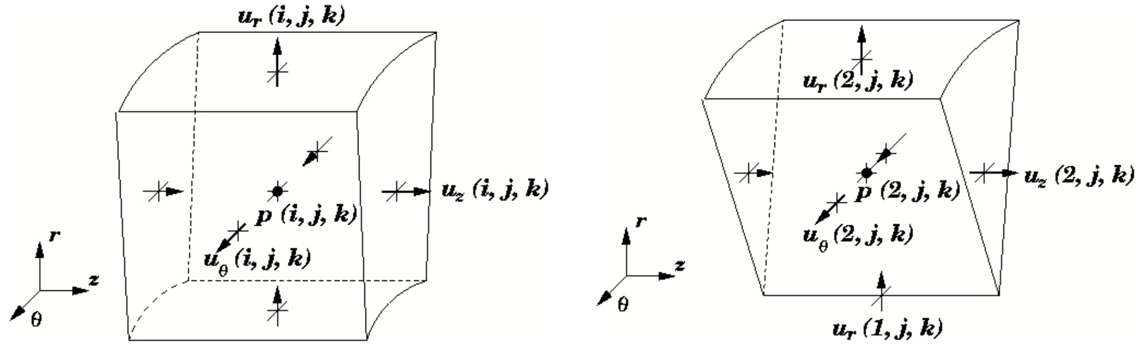
$$\frac{\partial}{\partial r} = \frac{\partial}{\partial \xi}\xi_r, \quad \frac{\partial}{\partial \theta} = \frac{\partial}{\partial \eta}\eta_\theta, \quad \frac{\partial}{\partial z} = \frac{\partial}{\partial \zeta}\zeta_z, \tag{2.51}$$

where

$$\xi_r = \frac{2\Delta\xi}{r_{i+1} - r_{i-1}}, \quad \eta_\theta = \frac{2\Delta\eta}{\theta_{j+1} - \theta_{j-1}}, \quad \zeta_z = \frac{2\Delta\zeta}{z_{k+1} - z_{k-1}}. \tag{2.52}$$

The cell sizes in the computational space are set to unity, i.e., $\Delta\xi = \Delta\eta = \Delta\zeta = 1$.

By introducing this transformation, Eqs. (2.17) and (2.18) now can be written



(a) (b)

Figure 2.3: A sketch of grid cell and the variable arrangement in cylindrical coordinates. a) A typical grid cell; b) a grid cell near the centerline.

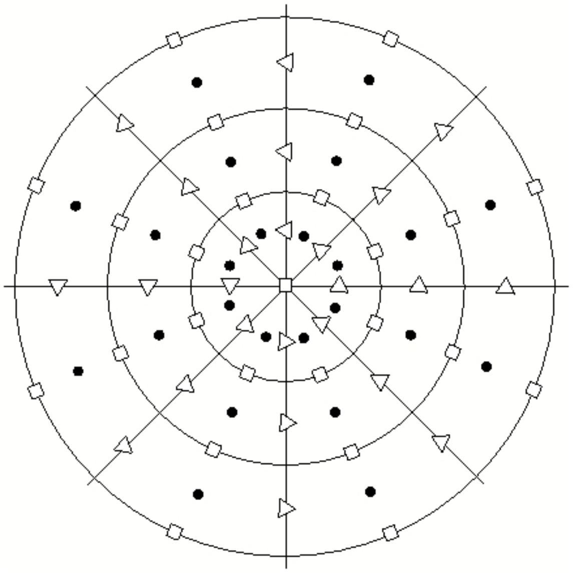


Figure 2.4: Staggered grid and the variable arrangement in $r - \theta$ plane. ● p, u_z; □ u_r; △ u_θ.

as:

$$
\begin{aligned}
\frac{\partial u_r}{\partial t} = & - \left[\frac{1}{r}\xi_r\frac{\partial (r u_r u_r)}{\partial \xi} + \frac{1}{r}\eta_\theta\frac{\partial (u_\theta u_r)}{\partial \eta} + \zeta_z\frac{\partial (u_z u_r)}{\partial \zeta}\right] - \xi_r\frac{\partial p}{\partial \xi} \\
& + \left[\xi_r\frac{\partial}{\partial \xi}\left(\frac{1}{r}\nu\xi_r\frac{\partial (r u_r)}{\partial \xi}\right) + \frac{1}{r^2}\eta_\theta\frac{\partial}{\partial \eta}\left(\nu\eta_\theta\frac{\partial u_r}{\partial \eta}\right) + \zeta_z\frac{\partial}{\partial \zeta}\left(\nu\zeta_z\frac{\partial u_r}{\partial \zeta}\right)\right] \\
& + \left[\xi_r\frac{\partial}{\partial \xi}\left(\frac{1}{r}\nu_t\xi_r\frac{\partial (r u_r)}{\partial \xi}\right) + \frac{1}{r}\eta_\theta\frac{\partial}{\partial \eta}\left(\nu_t\xi_r\frac{\partial u_\theta}{\partial \xi}\right) + \zeta_z\frac{\partial}{\partial \zeta}\left(\nu_t\xi_r\frac{\partial u_z}{\partial \xi}\right)\right] \\
& + \frac{u_\theta^2}{r} \\
& - (2\nu + \nu_t)\frac{1}{r^2}\eta_\theta\frac{\partial u_\theta}{\partial \eta} - \frac{u_\theta}{r^2}\eta_\theta\frac{\partial \nu_t}{\partial \eta} - \frac{2u_r}{r}\xi_r\frac{\partial \nu_t}{\partial \xi},
\end{aligned}
\tag{2.53}
$$

$$
\begin{aligned}
\frac{\partial u_\theta}{\partial t} = & - \left[\frac{1}{r}\xi_r\frac{\partial (r u_r u_\theta)}{\partial \xi} + \frac{1}{r}\eta_\theta\frac{\partial (u_\theta u_\theta)}{\partial \eta} + \zeta_z\frac{\partial (u_z u_\theta)}{\partial \zeta}\right] - \frac{1}{r}\eta_\theta\frac{\partial p}{\partial \eta} \\
& + \left[\frac{1}{r}\xi_r\frac{\partial}{\partial \xi}\left(\nu r\xi_r\frac{\partial u_\theta}{\partial \xi}\right) + \frac{1}{r^2}\eta_\theta\frac{\partial}{\partial \eta}\left(\nu\eta_\theta\frac{\partial u_\theta}{\partial \eta}\right) + \zeta_z\frac{\partial}{\partial \zeta}\left(\nu\zeta_z\frac{\partial u_\theta}{\partial \zeta}\right)\right] \\
& + \left[\frac{1}{r}\xi_r\frac{\partial}{\partial \xi}\left(\nu_t\eta_\theta\frac{\partial u_r}{\partial \eta}\right) + \frac{1}{r^2}\eta_\theta\frac{\partial}{\partial \eta}\left(\nu_t\eta_\theta\frac{\partial u_\theta}{\partial \eta}\right) + \frac{1}{r}\zeta_z\frac{\partial}{\partial \zeta}\left(\nu_t\eta_\theta\frac{\partial u_z}{\partial \eta}\right)\right] \\
& - \frac{u_r u_\theta}{r} \\
& + (2\nu + \nu_t)\frac{1}{r^2}\eta_\theta\frac{\partial u_r}{\partial \eta} - \nu\frac{u_\theta}{r^2} + \frac{2u_r}{r^2}\eta_\theta\frac{\partial \nu_t}{\partial \eta} - \frac{u_\theta}{r}\xi_r\frac{\partial \nu_t}{\partial \xi},
\end{aligned}
\tag{2.54}
$$

$$
\begin{aligned}
\frac{\partial u_z}{\partial t} = & - \left[\frac{1}{r}\xi_r\frac{\partial (r u_r u_z)}{\partial \xi} + \frac{1}{r}\eta_\theta\frac{\partial (u_\theta u_z)}{\partial \eta} + \zeta_z\frac{\partial (u_z u_z)}{\partial \zeta}\right] - \zeta_z\frac{\partial p}{\partial \zeta} \\
& + \left[\frac{1}{r}\xi_r\frac{\partial}{\partial \xi}\left(\nu r\xi_r\frac{\partial u_z}{\partial \xi}\right) + \frac{1}{r^2}\eta_\theta\frac{\partial}{\partial \eta}\left(\nu\eta_\theta\frac{\partial u_z}{\partial \eta}\right) + \zeta_z\frac{\partial}{\partial \zeta}\left(\nu\zeta_z\frac{\partial u_z}{\partial \zeta}\right)\right] \\
& + \left[\frac{1}{r}\xi_r\frac{\partial}{\partial \xi}\left(\nu_t\zeta_z\frac{\partial u_r}{\partial \zeta}\right) + \frac{1}{r}\eta_\theta\frac{\partial}{\partial \eta}\left(\nu_t\zeta_z\frac{\partial u_\theta}{\partial \zeta}\right) + \zeta_z\frac{\partial}{\partial \zeta}\left(\nu_t\zeta_z\frac{\partial u_z}{\partial \zeta}\right)\right]
\end{aligned}
\tag{2.55}
$$

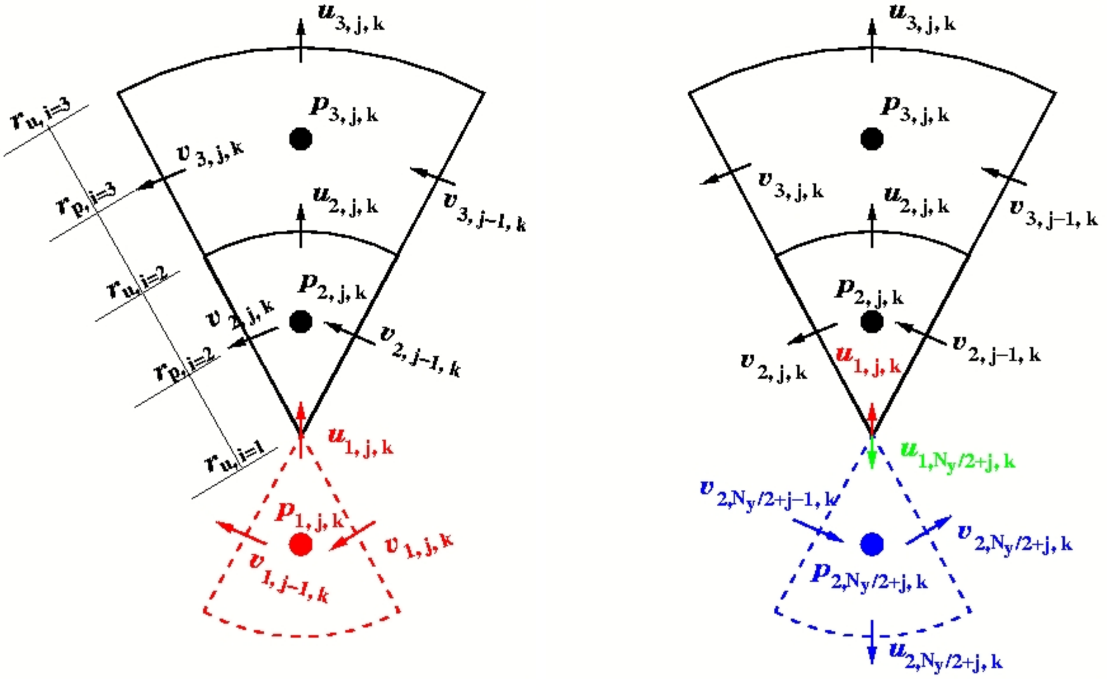
$$
\frac{1}{r}\xi_r\frac{\partial (r u_r)}{\partial \xi} + \frac{1}{r}\eta_\theta\frac{\partial u_\theta}{\partial \eta} + \zeta_z\frac{\partial u_z}{\partial \zeta} = 0. \tag{2.56}
$$

The discretization of all derivatives on the staggered grid in the computational space is similar to that in Cartesian coordinates that has described in th previous section. Also in this case, the arithmetic average is used to obtain variables at grid locations between the points where the variables are defined.

A major complication of the cylindrical coordinate formulation is the mathematical singularity at the centerline, which is not present in Cartesian coordinates.

Due the singularity at the axis, both the radial and the azimuthal velocity components cannot be defined at the axis even when the flow field itself does not have a singularity problem at the center line. Over the past years ways of addressing the singularity issue at the centerline with various degrees of complexity have been proposed [14, 1, 66, 38]. Here the approach proposed in [46] is adopted. Basically, the radial and the azimuthal velocity components are assumed to have multiple values at the axis and a linear averaging of two symmetrical points over the axis provides the value at the center line. Notice that for the radial velocity component this is an average over two grid cells and has a decreased order of accuracy. Therefore, finer grid spacing near the axis is required to obtain good resolution of the flow field near the axis.

Fig. 2.5 shows the current centerline treatment. In particular, ghost cells are used to implement the centerline boundary conditions. In Fig. 2.5a the required boundary conditions on the ghost cell for solving the Navier-Stokes equations are given. It is obvious from Fig. 2.5b, which shows the variable collocation on the grid cell across the centerline, the variables on the ghost cell in Fig. 2.5a can be obtained



(a) (b)

Figure 2.5: Centerline treatment of the cylindrical coordinates. a) The required boundary conditions and variable collacation on the ghost cell; b) Variables defined across the centerline on $r-\theta$ plane.

from the grid cell across the centerline in Fig. 2.5b:

$$\begin{aligned}
u_\theta(-\Delta r/2,\theta,z) &= -u_\theta(\Delta r/2,\theta+\pi,z) \quad \text{or} \quad v_{1,j,k} = -v_{2,N_y/2+j,k}, \\
u_z(-\Delta r/2,\theta,z) &= u_z(\Delta r/2,\theta+\pi,z) \quad \text{or} \quad w_{1,j,k} = w_{2,N_y/2+j,k}, \\
p(-\Delta r/2,\theta,z) &= p(\Delta r/2,\theta+\pi,z) \quad \text{or} \quad p_{1,j,k} = p_{2,N_y/2+j,k}, \\
\nu_t(-\Delta r/2,\theta,z) &= \nu_t(\Delta r/2,\theta+\pi,z) \quad \text{or} \quad \nu_{t\ 1,j,k} = \nu_{t\ 2,N_y/2+j,k}. \qquad (2.57)
\end{aligned}$$

where the boundary conditions for p are not directly utilized in the current formulation.

Inspection of Eqs. (2.53)∼(2.56) to locate the potential problematic terms due to the $1/r$ factors shows that many terms turn out to be free of the singularity problem because of the r factors in front of the fluxes or derivatives (with the assumption that those fluxes and derivatives are finite values, which are usually sound in physical sense). However, a centerline value for u_r is still required for those terms in red color. Notice that we write the diffusive terms in r direction in the forms appeared in Eq. (2.53) in order to be able to use Crank-Nicolson scheme in radial direction [66]. Here, the centerline bounary condition for u_r is obtained by averaging values of the opposing u_r across the centerline:

$$u_r(r=0,\theta,z) = \frac{1}{2}\left[u_r(\Delta r,\theta,z) - u_r(\Delta r,\theta+\pi,z)\right] \qquad (2.58)$$

or

$$u_{1,j,k} = \frac{1}{2}\left(u_{2,j,k} - u_{2,N_y/2+j,k}\right). \qquad (2.59)$$

The above approximation is a linear interpolation over a distance of $2\Delta r$. Therefore, the grid needs to maintain good quality near the centerline to avoid large discretization error. For all simulations in this work, this boundary condition is used and no obvious problem in the solution was found due to this centerline treatment.

2.2.3 Time Advancement Schemes

The fractional step method, which is discussed in the previous section, is used to integrate the governing equations in cylindrical coordinates in time. The extremely small grid size in azimuthal direction near the centerline imposes severe timestep constraints if all terms are treated explicitly. To remove this limitation, the diffusive terms in the azimuthal direction are treated implicitly. In particular, a second-order Adams-Bashforth (AB2) scheme or the low-storage third-order Runge-Kutta (RK3) scheme is used for terms treated explicitly, and the second-order Crank-Nicholson (CN2) scheme is used for terms treated implicitly:

$$\begin{aligned}
&\left[1-\frac{\alpha_k\Delta t}{2}\frac{1}{r^2}\frac{\partial}{\partial\theta}\left(\nu\frac{\partial}{\partial\theta}\right)\right]\hat{u}_r^k = \mathrm{RHS}_r^{k-1} \\
&= \quad u_r^{k-1}+\gamma_k\Delta t A_r^{k-1}+\rho_k\Delta t A_r^{k-2}+\frac{\alpha_k\Delta t}{2}B_r^{k-1}-\alpha_k\Delta t\frac{\partial p^{k-1}}{\partial r}, \qquad (2.60)
\end{aligned}$$

$$\begin{aligned}
&\left[1-\frac{\alpha_k\Delta t}{2}\frac{1}{r^2}\frac{\partial}{\partial\theta}\left((\nu+\nu_t)\frac{\partial}{\partial\theta}\right)\right]\hat{u}_\theta^k = \mathrm{RHS}_\theta^{k-1} \\
&= \quad u_\theta^{k-1}+\gamma_k\Delta t A_\theta^{k-1}+\rho_k\Delta t A_\theta^{k-2}+\frac{\alpha_k\Delta t}{2}B_\theta^{k-1}-\alpha_k\Delta t\frac{1}{r}\frac{\partial p^{k-1}}{\partial\theta}, \qquad (2.61)
\end{aligned}$$

$$\begin{aligned}
&\left[1-\frac{\alpha_k\Delta t}{2}\frac{1}{r^2}\frac{\partial}{\partial\theta}\left(\nu\frac{\partial}{\partial\theta}\right)\right]\hat{u}_z^k = \mathrm{RHS}_z^{k-1} \\
&= \quad u_z^{k-1}+\gamma_k\Delta t A_z^{k-1}+\rho_k\Delta t A_z^{k-2}+\frac{\alpha_k\Delta t}{2}B_z^{k-1}-\alpha_k\Delta t\frac{\partial p^{k-1}}{\partial z}. \qquad (2.62)
\end{aligned}$$

where the coefficients are the same as in the previous section, and the operators A and B are defined as:

$$
\begin{align}
A_r &= -\left[\frac{1}{r}\frac{\partial(ru_ru_r)}{\partial r}+\frac{1}{r}\frac{\partial(u_\theta u_r)}{\partial\theta}+\frac{\partial(u_zu_r)}{\partial z}\right] \nonumber\\
&+ \frac{\partial}{\partial r}\left[(\nu+\nu_t)\frac{1}{r}\frac{\partial(ru_r)}{\partial r}\right]+\frac{\partial}{\partial z}\left(\nu\frac{\partial u_r}{\partial z}\right)+\frac{1}{r}\frac{\partial}{\partial\theta}\left(\nu_t\frac{\partial u_\theta}{\partial r}\right)+\frac{\partial}{\partial z}\left(\nu_t\frac{\partial u_z}{\partial r}\right) \nonumber\\
&+ \frac{u_\theta^2}{r}-(2\nu+\nu_t)\frac{1}{r^2}\frac{\partial u_\theta}{\partial\theta}-\frac{u_\theta}{r^2}\frac{\partial\nu_t}{\partial\theta}-\frac{2u_r}{r}\frac{\partial\nu_t}{\partial r}, \tag{2.63}\\
A_\theta &= -\left[\frac{1}{r}\frac{\partial(ru_ru_\theta)}{\partial r}+\frac{1}{r}\frac{\partial(u_\theta u_\theta)}{\partial\theta}+\frac{\partial(u_zu_\theta)}{\partial z}\right] \nonumber\\
&+ \frac{1}{r}\frac{\partial}{\partial r}\left(\nu r\frac{\partial u_\theta}{\partial r}\right)+\frac{\partial}{\partial z}\left(\nu\frac{\partial u_\theta}{\partial z}\right)+\frac{1}{r}\frac{\partial}{\partial r}\left(\nu_t\frac{\partial u_r}{\partial\theta}\right)+\frac{1}{r}\frac{\partial}{\partial z}\left(\nu_t\frac{\partial u_z}{\partial\theta}\right) \nonumber\\
&- \frac{u_ru_\theta}{r}+(2\nu+\nu_t)\frac{1}{r^2}\frac{\partial u_r}{\partial\theta}-\nu\frac{u_\theta}{r^2}+\frac{2u_r}{r^2}\frac{\partial\nu_t}{\partial\theta}-\frac{u_\theta}{r}\frac{\partial\nu_t}{\partial r}, \tag{2.64}\\
A_z &= -\left[\frac{1}{r}\frac{\partial(ru_ru_z)}{\partial r}+\frac{1}{r}\frac{\partial(u_\theta u_z)}{\partial\theta}+\frac{\partial(u_zu_z)}{\partial z}\right] \nonumber\\
&+ \frac{1}{r}\frac{\partial}{\partial r}\left(\nu r\frac{\partial u_z}{\partial r}\right)+\frac{\partial}{\partial z}\left[(\nu+\nu_t)\frac{\partial u_z}{\partial z}\right] \nonumber\\
&+ \frac{1}{r}\frac{\partial}{\partial r}\left(\nu_t r\frac{\partial u_r}{\partial z}\right)+\frac{1}{r}\frac{\partial}{\partial\theta}\left(\nu_t\frac{\partial u_\theta}{\partial z}\right), \tag{2.65}
\end{align}
$$

$$
\begin{align}
B_r &= \frac{1}{r^2}\frac{\partial}{\partial\theta}\left(\nu\frac{\partial u_r}{\partial\theta}\right), \tag{2.66}\\
B_\theta &= \frac{1}{r^2}\frac{\partial}{\partial\theta}\left[(\nu+\nu_t)\frac{\partial u_\theta}{\partial\theta}\right], \tag{2.67}\\
B_z &= \frac{1}{r^2}\frac{\partial}{\partial\theta}\left(\nu\frac{\partial u_z}{\partial\theta}\right). \tag{2.68}
\end{align}
$$

The spatial discretization of above equations results in a series of cyclic tridiagonal equations, which are solved using the solver for cyclic tridiagonal systems from *Numerical Recipes* [51]. No boundary condition is required for the variables at the intermediate sub-steps due to the periodicity in azimuthal direction.

The following stability criterion from [1] is adopted here

$$CFL=\Delta t\left[\frac{|u_r|}{\Delta r}+\frac{|u_\theta|}{r\Delta\theta}+\frac{|u_z|}{\Delta z}+4\nu\left(\frac{1}{\Delta r^2}+\frac{1}{\Delta z^2}\right)\right]. \tag{2.69}$$

In the simulations of flow past a sphere, $CFL = 1.2$ is used.

2.3 Poisson Equation

The Poisson equation Eq. (2.31) can be rewritten here in vector form as

$$\nabla^2\phi = f = \frac{1}{\alpha_n \Delta t}\nabla\cdot\hat{u}_i^n, \quad \text{where} \quad \phi = p^n - p^{n-1}, \tag{2.70}$$

where n is sub step index for RK3 scheme or time step index for AB2 scheme. The discrete form of the above equation in Cartesian coordinates is

$$\left(\frac{\delta^2}{\delta^2 x} + \frac{\delta^2}{\delta^2 y} + \frac{\delta^2}{\delta^2 z}\right)\phi_{i,j,k} = f_{i,j,k}, \tag{2.71}$$

and in cylindrical coordinates is

$$\left[\frac{1}{r}\frac{\delta}{\delta r}\left(r\frac{\delta}{\delta r}\right) + \frac{1}{r^2}\frac{\delta^2}{\delta^2\theta} + \frac{\delta^2}{\delta^2 z}\right]\phi_{i,j,k} = f_{i,j,k}. \tag{2.72}$$

The discrete operators above have to be consistent with the ones used in the momentum equations. Otherwise, mass conservation cannot be guaranteed. Here only the solution procedure in cylindrical coordinates will be discussed because it is evident that the equation for Cartesian coordinates will be recovered when r in Eq. (2.72) is set to unity. The discretized form of Eq. (2.72) can be written as:

$$\begin{aligned}
& \frac{1}{r_p|_i}\xi_x|_i^p\frac{1}{\Delta\xi^2}\left\{\left[r_u|_i\xi_x|_i^u\left(\phi_{i+1,j,k}-\phi_{i,j,k}\right)\right] - \left[r_u|_{i-1}\xi_x|_{i-1}^u\left(\phi_{i,j,k}-\phi_{i-1,j,k}\right)\right]\right\} \\
+ \quad & \frac{1}{r_p^2|_i}\eta_y|_j^p\frac{1}{\Delta\eta^2}\left\{\left[\eta_y|_j^v\left(\phi_{i,j+1,k}-\phi_{i,j,k}\right)\right] - \left[\eta_y|_{j-1}^v\left(\phi_{i,j,k}-\phi_{i,j-1,k}\right)\right]\right\} \\
+ \quad & \zeta_z|_k^p\frac{1}{\Delta\zeta^2}\left\{\left[\zeta_z|_k^w\left(\phi_{i,j,k+1}-\phi_{i,j,k}\right)\right] - \left[\zeta_z|_{k-1}^w\left(\phi_{i,j,k}-\phi_{i,j,k-1}\right)\right]\right\} \\
= \quad & f_{i,j,k},
\end{aligned} \tag{2.73}$$

or

$$
\begin{aligned}
& am_i\, \phi_{i-1,j,k} + bm_i\, \phi_{i,j,k} + cm_i\, \phi_{i+1,j,k} \\
+ \quad & a\, l_j\, \phi_{i,j-1,k} + b\, l_j\, \phi_{i,j,k} + c\, l_j\, \phi_{i,j+1,k} \\
+ \quad & an_k\, \phi_{i,j,k-1} + bn_k\, \phi_{i,j,k} + cn_k\, \phi_{i,j,k+1} = f_{i,j,k}, \qquad (2.74)
\end{aligned}
$$

with the coefficients

$$
\begin{aligned}
am_i &= \frac{1}{\Delta\xi^2}\frac{1}{r_p|_i}\xi_x|_i^p r_u|_{i-1}\xi_x|_{i-1}^u, \\
cm_i &= \frac{1}{\Delta\xi^2}\frac{1}{r_p|_i}\xi_x|_i^p r_u|_i\xi_x|_i^u, \\
bm_i &= -am_i - cm_i, \\
a\, l_j &= \frac{1}{\Delta\eta^2}\frac{1}{r_p^2|_i}\eta_y|_j^p\eta_y|_{j-1}^v, \\
c\, l_j &= \frac{1}{\Delta\eta^2}\frac{1}{r_p^2|_i}\eta_y|_j^p\eta_y|_j^v, \\
b\, l_j &= -a\, l_j - c\, l_j, \\
an_k &= \frac{1}{\Delta\zeta^2}\zeta_z|_k^p\zeta_z|_{k-1}^w, \\
cn_k &= \frac{1}{\Delta\zeta^2}\zeta_z|_k^p\zeta_z|_k^w, \\
bn_k &= -an_k - cn_k. \qquad (2.75)
\end{aligned}
$$

The above equation is solved using a combination of a fast Fourier transform (FFT) method from FFTPACK [58] and a direct solution procedure from FISHPACK [57]. The major advantage of this direct solver is that it can converge the solution to machine accuracy with only one iteration, which is a great saving comparing with the iterative solvers. However, to be able to utilize Fourier transforms, the computational grid must be uniform in the direction in which the FFT is performed. In this work, the grid is uniform in the spanwise direction, i.e., y or θ

direction. Therefore, the fast Fourier transform in the spanwise direction can transform Eq. 2.70 into a set of two-dimensional Helmholtz equations in the uncoupled wave number space:

$$\left[\frac{1}{r}\frac{\delta}{\delta r}\left(r\frac{\delta}{\delta r}\right)+\frac{\delta^2}{\delta^2 z}+\frac{1}{r^2}k_l'\right]\hat{\phi}_{i,j,l}=\hat{f}_{i,l,k}, \tag{2.76}$$

and the modified wave number k_l' is defined as

$$k_l'=\frac{2}{\Delta\theta^2}\left[1-\cos\left(\frac{2\pi l}{N_\theta}\right)\right] \tag{2.77}$$

where l is the wave number, N_θ is the number of grid cells (not including ghost cells) and $\Delta\theta$ is the cell size in the spanwise direction. Eq. (2.74) can be rewritten as

$$\begin{aligned} & am_i\,\hat{\phi}_{i-1,l,k}+\left(bm_i-\frac{k_l'}{r_p^2|_i}\right)\hat{\phi}_{i,l,k}+cm_i\,\hat{\phi}_{i+1,l,k} \\ + \quad & an_k\,\hat{\phi}_{i,l,k-1}+bn_k\,\hat{\phi}_{i,l,k}+cn_k\,\hat{\phi}_{i,l,k+1}=\hat{f}_{i,l,k}. \end{aligned} \tag{2.78}$$

The above equations can be solved separately for each wave number. Both the real and imaginary part of each wave number are solved using the "BLKTRI" routine, in which a generalized cyclic reduction algorithm [57] is implemented, in the FISHPACK library.

In cases where periodic boundary conditions are used in both streamwise and spanwise directions, an FFT is also applied in the streamwise direction, and the resulting series of (cyclic) tridiagonal equations are solved using the (cyclic) tridiagonal system solver from *Numerical Recipes* [51].

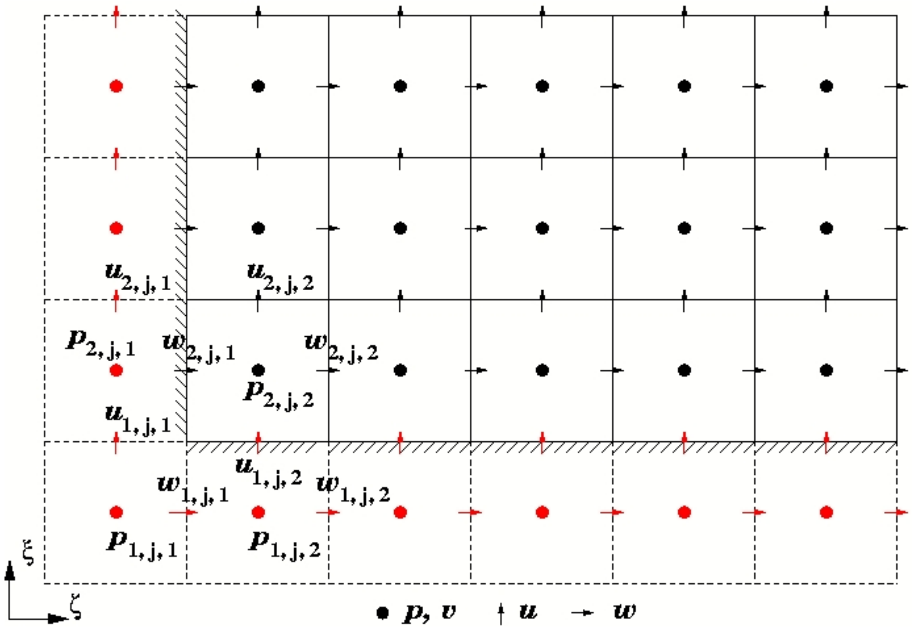


Figure 2.6: The implementation of Dirichlet and Neumann boundary condition.

2.4 Boundary Conditions

In this work, ghost cells are used to implement all boundary conditions. The advantages of the ghost cell approach are that the grid spacing near the boundary is continuous and the code can have a generalized form for all grid points at which the equations are solved. In addition, it provides a natural way to implement parallelization via domain decomposition technique.

2.4.1 Dirichlet and Neumann Boundary Conditions

The implementation of Dirichlet and Neumann boundary conditions is is shown in Fig. 2.6, in which the lower-left corner of a $\xi - \zeta$ plane of the computational domain and the collocation of the variables is given. For velocity component normal to the wall, e.g., u to the lower boundary and w to the left boundary in the figure, the Dirichlet condition can be directly enforced as

$$u_{1,j,k} = u_b, \quad \text{and} \quad w_{i,j,1} = w_b, \tag{2.79}$$

where u_b and w_b are the prescribed normal velocity components on the lower wall and left wall, respectively.

The non-slip conditions for the wall-tangential components are implemented through the use of ghost cells:

$$\begin{aligned} v_t &= \frac{1}{2}\left(v_{1,j,k} + v_{2,j,k}\right) \quad \text{or} \quad v_{1,j,k} = 2v_t - v_{2,j,k}, \\ w_t &= \frac{1}{2}\left(w_{1,j,k} + w_{2,j,k}\right) \quad \text{or} \quad w_{1,j,k} = 2w_t - w_{2,j,k}, \end{aligned} \tag{2.80}$$

where v_t and w_t are prescribed tangential components in y and z direction, respectively. For stationary wall without tangential movement, $v_t = 0$ and $w_t = 0$.

The Neumann boundary condition for an arbitrary variable φ can be written as

$$\frac{\partial \varphi}{\partial n} = f, \tag{2.81}$$

where n is the normal direction of the boundary, and f is a known function. Referring to the lower boundary in Fig. 2.6, this condition can be implemented with the help of ghost cells as follows:

$$\varphi_{1,j,k} = \varphi_{2,j,k} - f\Delta x, \tag{2.82}$$

or

$$\varphi_{1,j,k} = \varphi_{2,j,k} \tag{2.83}$$

for homogeneous boundary condition.

2.4.2 Convective Boundary Condition

The convective boundary condition proposed by Orlanski [41] is used for outflow boundaries in this study. This condition has been found very successful in convecting structures out of the domain without distorting the flow in the computational domain. In such a case the boundary velocity can be obtained from the following equation:

$$\frac{\partial u_i}{\partial t} + U_{conv}\frac{\partial u_i}{\partial z} = 0, \tag{2.84}$$

where u_i is any velocity component, and U_{conv} is the convective velocity, which is set to the mean streamwise velocity at the exit plane. Here, the outflow boundary is always normal to the streamwise direction, which is z direction for both Cartesian and cylindrical coordinates. However, the corresponding term from the continuity equation has to be used instead of $\partial u_i/\partial z$ when the exit plane is normal to the radial or azimuthal direction in cylindrical coordinates.

Eq. (2.84) is discretized using explicit Euler scheme in time, one-sided difference formulas for the streamwise velocity component and central difference for the other two velocity components in space. Note that the predicted streamwise velocity component is adjusted every timestep to globally conserve mass.

2.4.3 Periodic Boundary Conditions

The implementation of periodic boundary condition through ghost cells is also very simple. As shown in Fig. 2.7, the solution values on the left side are directly

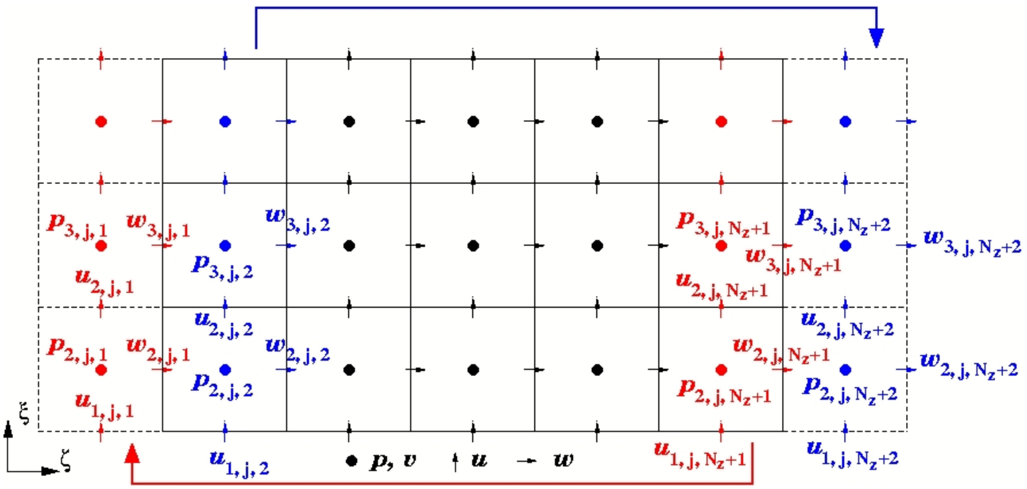


Figure 2.7: The implementation of periodic boundary condition. Arrows show the directions of data duplication.

copied to the ghost cells on the right side and vise versa:

$$u_{i,j,N_z+2} = u_{i,j,2},\; v_{i,j,N_z+2} = v_{i,j,2},\; w_{i,j,N_z+2} = w_{i,j,2},\; p_{i,j,N_z+2} = p_{i,j,2};$$

$$u_{i,j,1} = u_{i,j,N_z+1},\; v_{i,j,1} = v_{i,j,N_z+1},\; w_{i,j,1} = w_{i,j,N_z+1},\; p_{i,j,1} = p_{i,j,N_z+1}. \qquad (2.85)$$

This way periodicity of the function and its first derivative is imposed.

2.5 Validation of Basic Solver

The basic solver for the Navier-Stokes equations in Cartesian coordinates has been validated extensively [3, 4, 5]. Here the basic solver in cylindrical coordinates is validated for the case of fully developed turbulent pipe flow. This problem has not been studied numerically as extensively as its Cartesian equivalent, the turbulent channel flow. Although the geometry is equally simple, the presence of a singularity at the axis gives rise to numerical difficulties, making it a more challenging problem.

In the following we will present LES of this problem and will compare our results with the DNS data reported in [16].

As already discussed, a semi-implicit scheme (third-order Runge-Kutta scheme for explicit terms and second-order Crank-Nicolson scheme for implicit terms) is used for the time advancement. To maximize the timesteps the following procedure proposed in [1] is adapted in this study. Basically, the computational domain is divided into two separate regions: the core-region, which includes the axis and extends to a given radius (usually in the middle of the radius R), and the outer-region, which includes all the rest. Fig. 2.8 shows how this dividing is performed for a typical computational grid. Then all the convection and diffusion terms in the azimuthal direction are treated implicitly in the core-region, while all other terms are treated explicitly. The azimuthal momentum equation is linearized and solved first to provide a predicted velocity component for the other two equations. On the other hand, all the convection and diffusion terms in the radial direction are treated implicitly in the outer-region with all other terms treated explicitly. Again the radial momentum equation is linearized and solved before the other two equations as the predicted radial velocity is needed by those two equations. By doing this, the time step can be twenty times larger than that with only the diffusion terms in the azimuthal direction treated implicitly. Both the plane-averaged and the Lagrangian dynamic SGS model are implemented in cylindrical coordinates in this study.

The computational domain is the same as in [14] and [1]: $0 \leq z \leq L_z = 10R$, $0 \leq r \leq R$, and $0 \leq \theta \leq 2\pi$ with R the radius of the pipe. Periodic boundary conditions are applied in the streamwise and azimuthal directions. Computations

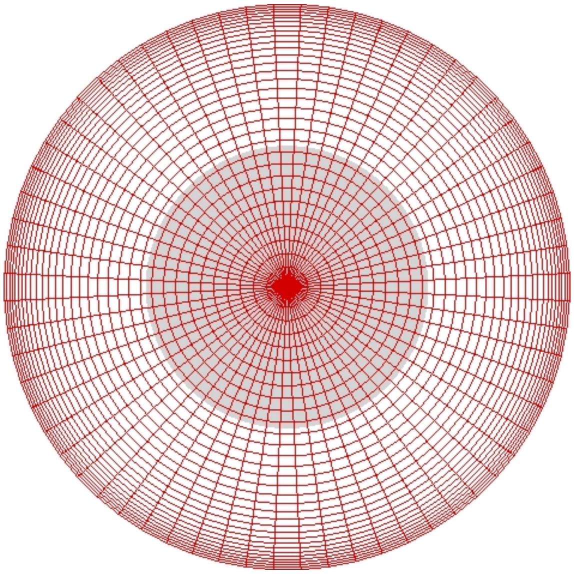


Figure 2.8: Typical computational grid. Shaded area is the core-region; un-shaded area is the outer-region.

on three grids, $32 \times 64 \times 48$, $48 \times 64 \times 64$, and $64 \times 96 \times 64$ ($N_r \times N_\theta \times N_z$), have been performed. Here only the results on grid 1 and 3 will be reported as the results on grid 2 are always in the middle of the other two grids. The grid spacing is uniform in the streamwise and azimuthal directions, and stretched in the radial direction to cluster points near the wall. The Reynolds number based on the friction velocity u_τ and the pipe radius R is 180.

Table 2.1 summarizes the grid resolution in wall units for the three different grids used in this works. The corresponding resolution for the reference DNS and LES has been added for comparison. Some mean flow properties together with the data from [1] are listed in Table 2.2. All data have been normalized using the friction velocity u_τ and the radius R. In the table U_{cl} is the center line velocity and U_{bulk} is the bulk velocity. C_f is the friction coefficient based on the bulk velocity and

Table 2.1: Grid resolution in wall units compared with data in [1]

.

$N_r \times N_\theta \times N_z$	Δz^+	$(R\Delta\theta)^+$	$(\Delta r\Delta\theta)^+$	Δr^+_{wall}	Δr^+_{cl}	Δr^+_{max}
Grid 1: $32 \times 64 \times 48$	37.5	17.7	0.85	1.54	8.62	8.62
Grid 3: $64 \times 96 \times 64$	28.13	11.8	0.29	0.39	4.39	4.39
$68 \times 128 \times 256$ DNS [1]	7.03	8.84	0.13	0.17	2.61	5.94
$38 \times 64 \times 32$ LES [1]	56.3	17.7	0.42	0.69	4.28	8.87

Table 2.2: Mean flow properties compared with data in [1]

.

$N_r \times N_\theta \times N_z$	U_{cl}	U_{bulk}	U_{cl}/U_{bulk}	$C_f \times 10^3$
$32 \times 64 \times 48$ LES	19.27	15.15	1.27	8.50
$64 \times 96 \times 64$ LES	19.00	14.76	1.29	9.05
$68 \times 128 \times 256$ DNS [1]	19.32	14.70	1.31	9.25
$38 \times 64 \times 32$ LES [1]	19.15	15.11	1.27	8.76

calculated according to $C_f = \tau_w / \left(\frac{1}{2}\rho U^2_{bulk}\right)$. From Table 2.2 it is evident that our flows match closely those in [1].

The mean streamwise velocity profile in wall units is shown in Fig. 2.9. Results from grid 3 agree well with the DNS data. As expected the LES on grid 1 results in a high intercept of the log-law.

Radial, azimuthal, and axial turbulence intensities are shown in Figs. 2.10, 2.11, and 2.12, respectively (only the resolved part of the intensities plotted here). The

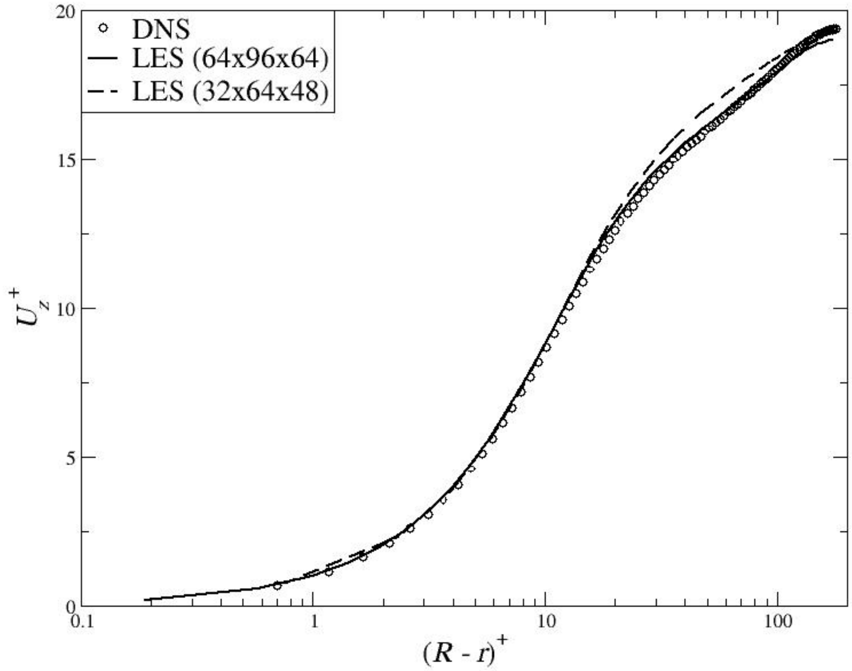


Figure 2.9: Mean streamwise velocity profile in wall unit. DNS data from [16]

overall agreement between our simulations the DNS data is very good. However, the intensities near the center line are under-predicted due to the coarse grid spacing there. Figure 2.13 shows the resolved turbulent shear stress and the total stress balance in the pipe. The viscous shear stress, resolved shear stress, and the SGS shear stress are summed up to obtained the total shear stress. The results from both grids shown here give total stress very close to the DNS data. The finer grid performs better.

Figure 2.14 shows the ratio of the eddy viscosity to molecular viscosity as a function of radius. As expected, fine grid gives lower eddy viscosity. Due to the coarse grid spacing near the center line, there is a bump of the eddy viscosity from the plane-averaged model near the center line. Also, it shows a sharp decrease to

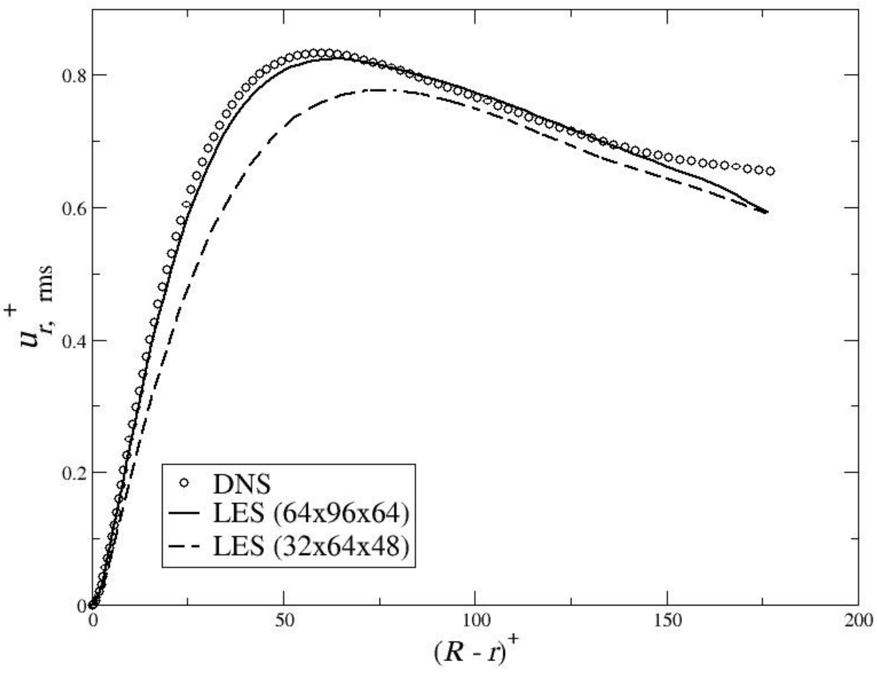


Figure 2.10: Radial turbulence intensity. DNS data from [16]

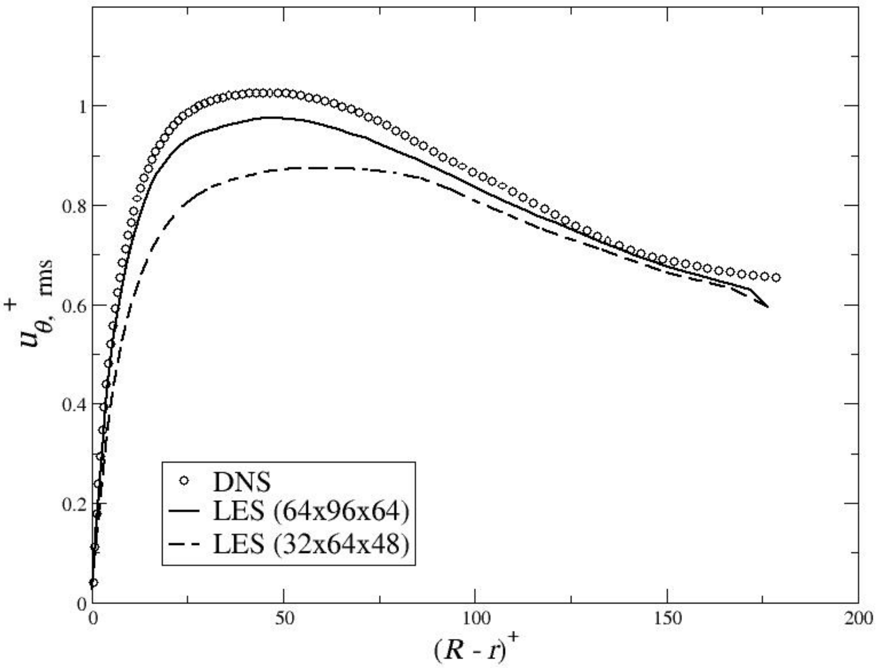


Figure 2.11: Azimuthal turbulence intensity. DNS data from [16].

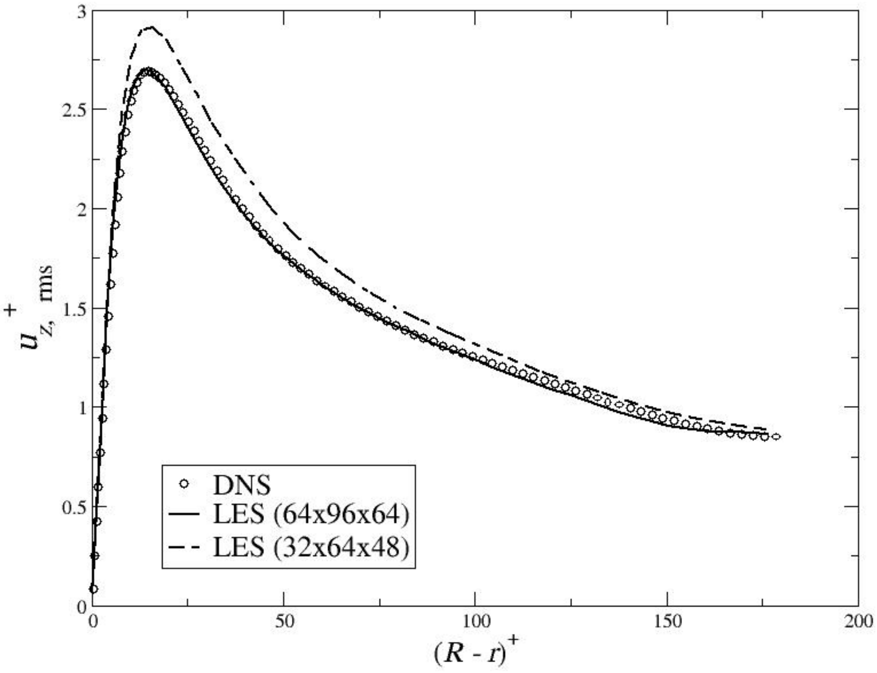


Figure 2.12: Axial turbulence intensity. DNS data from [16].

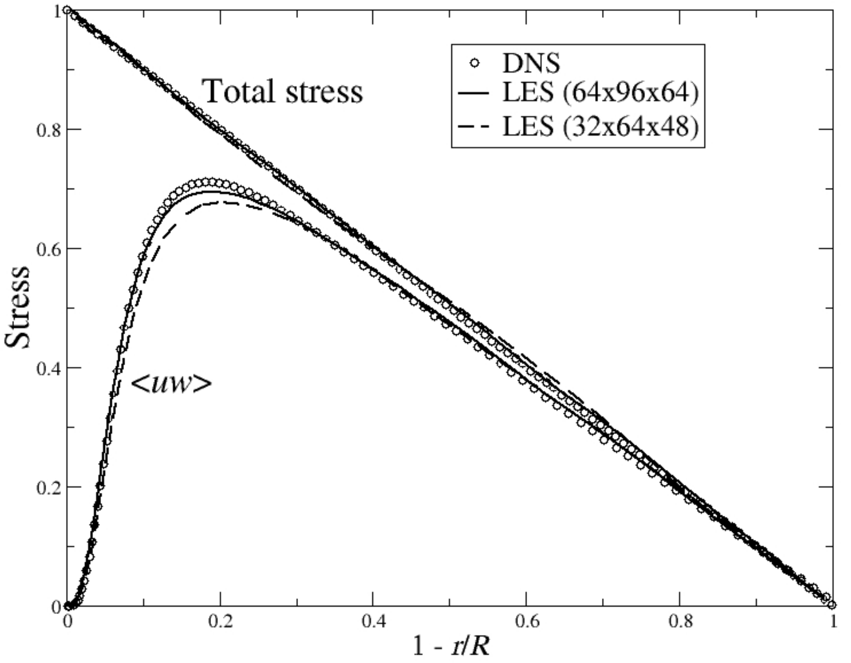


Figure 2.13: Total stress balance. DNS data from [16].

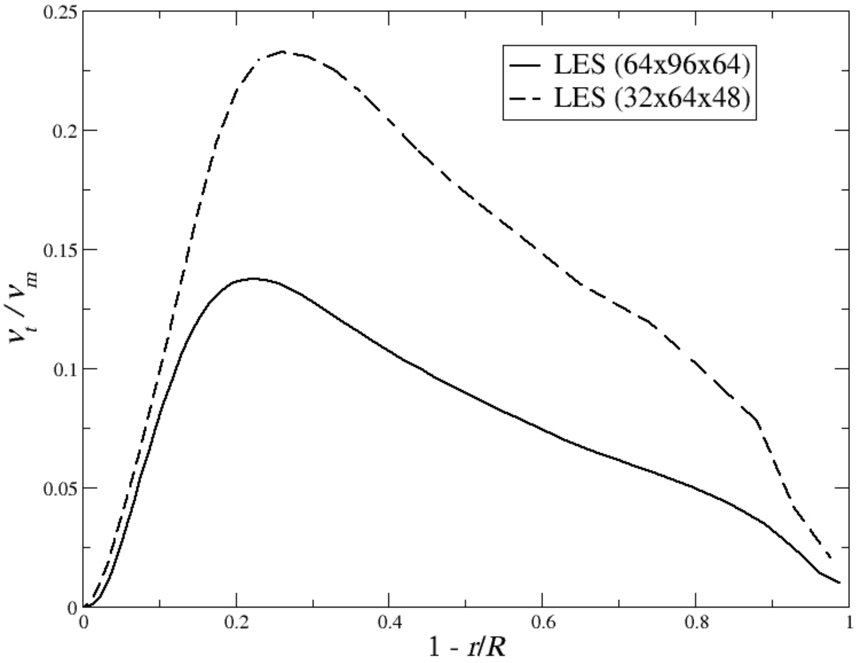


Figure 2.14: Ratio of eddy viscosity to molecular viscosity.

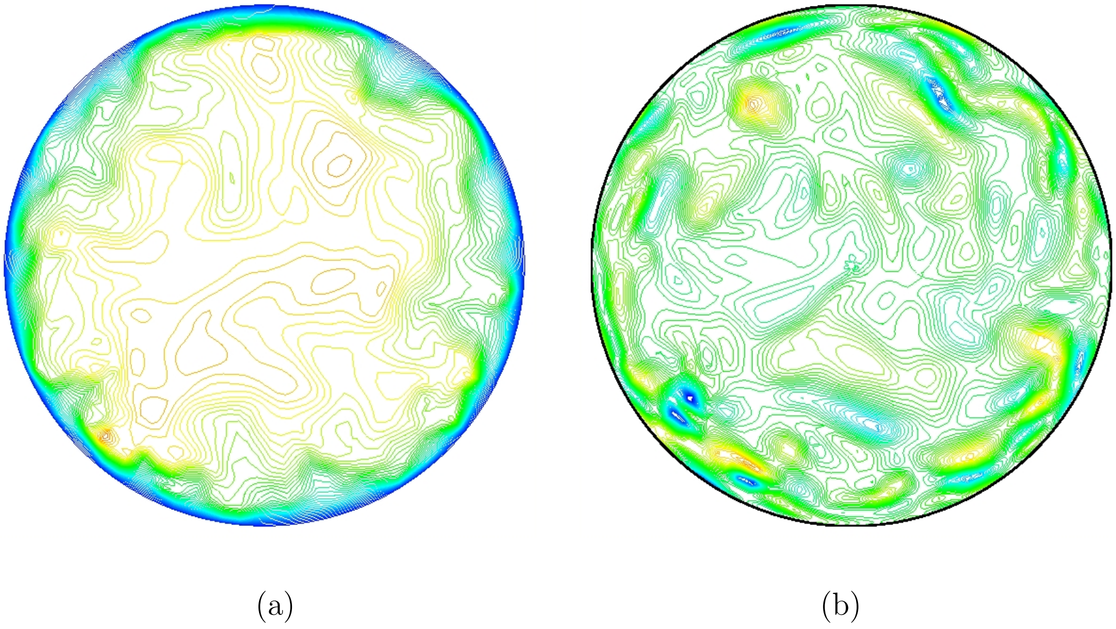


Figure 2.15: Instantaneous field in a polar plane. (a) Axial velocity; (b) Axial vorticity.

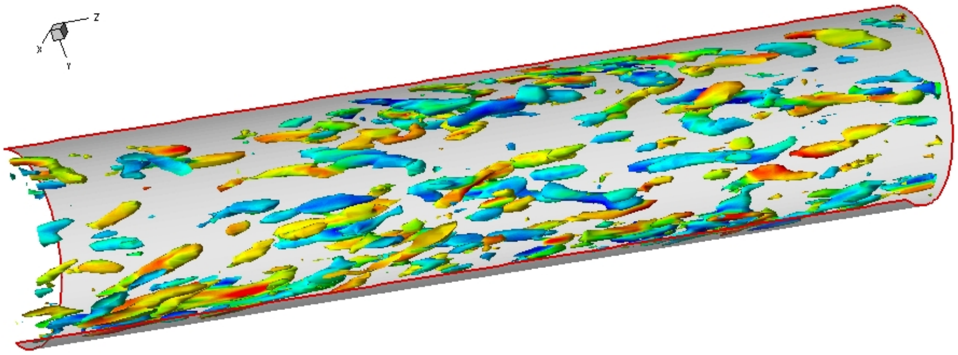


Figure 2.16: Instantaneous vortical structure by the Q-criterion (colored by axial vorticity).

the center line due to the decreased strain-rate as we average the axial velocity component over the center line.

Fig. 2.15a shows an instantaneous snapshot of the streamwise velocity in a polar plane and Fig. 2.15b shows an instantaneous snapshot of the streamwise vorticity in the same plane. The latter gives a very good illustration of the near-wall flow structure. A snapshot of the instantaneous vortical structure visualized by the Q-criterion (For details on the identification of coherent structures using the "*Q*-criterion" the reader is referred to [25]) is shown in Fig. 2.16. The quasi-streamwise structures that are responsible for most turbulent production can be observed.

Chapter 3

Embedded Boundary Method

In this chapter, the embedded-boundary formulation is introduced to extend the Cartesian/cylindrical grid solver to cases with complex moving boundaries. In Sec. 3.1, establishment of the interface-grid relation is discussed. Then the treatment of stationary immersed boundaries and moving immersed boundaries is given in Sections 3.2 and 3.3, respectively. The surface force calculation procedure, which computes the hydrodynamic loads to the structure, will be introduced in Sec. 3.4. Then the fluid-structure coupling scheme is discussed in Sec. 3.5. To utilize the current high-performance parallel computing platforms, the above algorithms have been parallelized. The details are given in Sec. 3.6.

3.1 Interface-grid Relation

3.1.1 Interface Description

There are two approaches for interface description: Eulerian and Lagrangian methods. In Eulerian methods, such as level set [42], the interface is implicitly given by a field function, namely, the signed shortest distance to the interface. In the Lagrangian methods on the other hand, such as front tracking and marker particles, the interface is explicitly described independent of the underlying grid. In the present study a front tracking scheme originally proposed for multi-phase flow problems is used for the description of the fluid/solid interface [62].

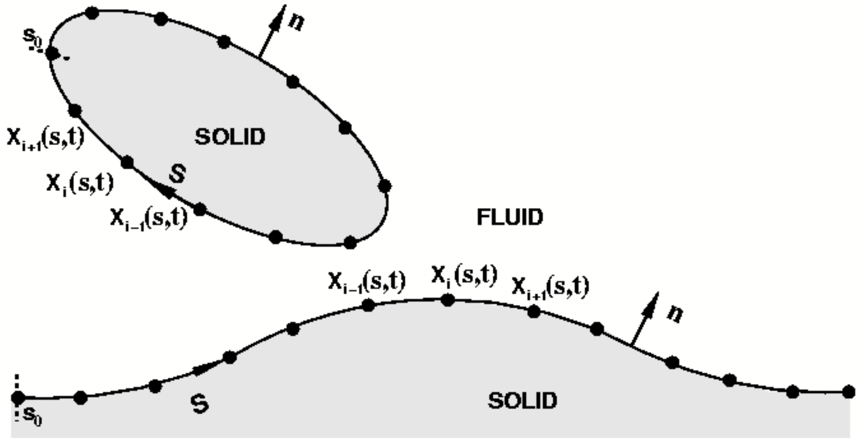


Figure 3.1: The parametrized description of interfaces of arbitrary shapes using marker particles.

As shown in Fig. 3.1, a two-dimensional immersed interface, Ψ, can be represented by a series of interfacial marker particles, which are defined by arc length coordinates $\vec{X}(s,t)$. The immersed interface can have arbitrary shapes and it can be open or closed. The marker particles are evenly attached to the interface with a spacing approximating the local grid size, and the beginning of the arclength coordinate is defined such that the fluid (or interested side of the interface) is always to the left of the observer as one moves along the interface toward increasing s. For each marker particle with arclength coordinates $\vec{X}_i$, the functions defining the coordinates can be written as

$$x(s,t) = a_x s^2 + b_x s + c_x \quad \text{and} \quad y(s,t) = a_y s^2 + b_y s + c_y. \tag{3.1}$$

These functions are generated at each sub-step of the splitting scheme for moving interfaces. The coefficients $a_{x,y}$, $b_{x,y}$, $c_{x,y}$ can be obtained by fitting quadratic polynomials to particle (i) and its two neighbors $(i-1)$ and $(i+1)$.

The normal from any location on the interface to the fluid can be calculated by the following equations:

$$n_x = \frac{-y_s}{\sqrt{(x_s^2 + y_s^2)}}, \quad n_y = \frac{x_s}{\sqrt{(x_s^2 + y_s^2)}}, \tag{3.2}$$

where the derivatives, x_s, y_s can be evaluated from the functions in Eq. (3.1) above as follows,

$$x_s(s,t) = 2a_x s + b_x, \quad y_s(s,t) = 2a_y s + b_y. \tag{3.3}$$

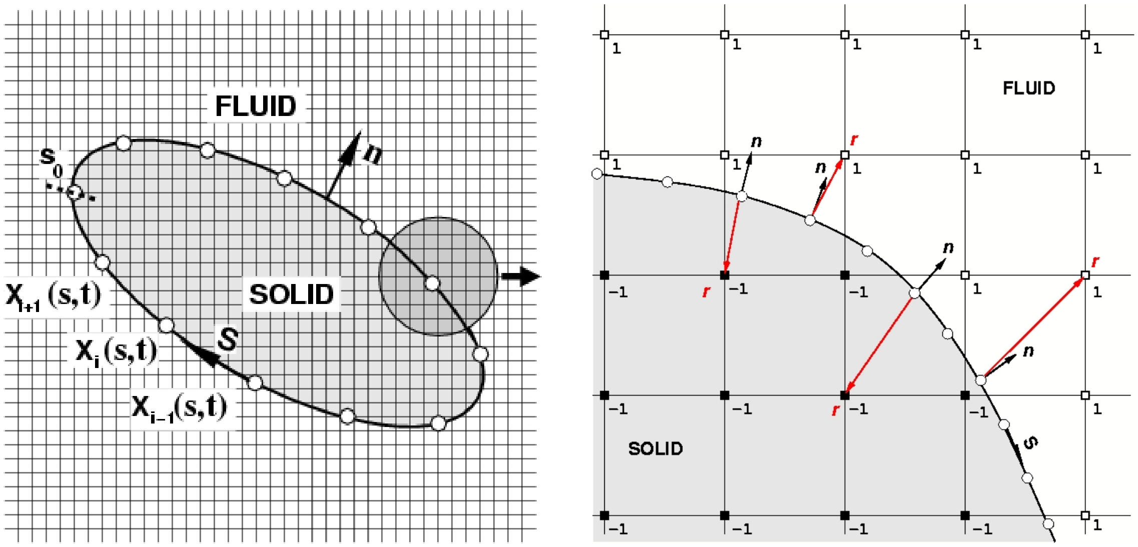
The coordinates $\vec{X}(s,t)$ and the coefficients are stored for each marker particle on the interface. For three-dimensional interfaces Bi-spline fitting can be used.

3.1.2 Tagging of Points on the Eulerian Grid

Having defined the immersed interface as a series of marker particles, one can now establish the relationship between these particles and the underlying Eulerian grid. Fig. 3.2a shows the parametrized interface immersed in a Cartesian grid. Here an approach similar to that presented in [4] is adopted for the tagging process and is summarized as follows:

1. Determine the part of the grid occupied by the interface as the coordinates of each marker particle are known from the previous Section.

2. For those grid points, a search for the closest marker particle, s_b, is performed. A ray, $\mathbf{r}$, is shot from this marker particle to the grid point, and the dot product of $\mathbf{r}$ and $\mathbf{n}_b$ is calculated.



(a) (b)

Figure 3.2: Grid–interface relation. (a) Parametrized interface immersed in the underlying Cartesian grid; (b) Zoom in the vicinity of interface where the inside/outside status of the Eulerian grid points are shown. □ Fluid points; ■ Solid points.

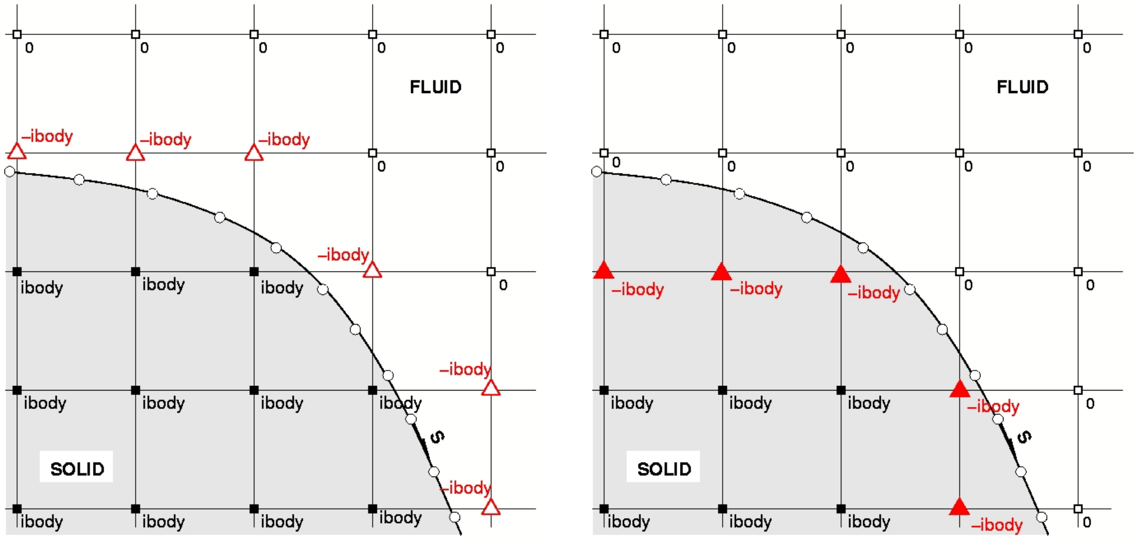
3. If $\mathbf{r} \cdot \mathbf{n}_b < 0$, then this grid point is inside the interface and assigned a tag of -1; otherwise, it is outside the interface and maintains its initial tag of 1.

Fig. 3.2b shows a zoom-in of the vicinity of the interface with the inside/outside status of the Eulerian grid points shown. In the next step the boundary points are identified, which are points in the fluid phase with at least one neighboring point in the solid. These points will be central to the reconstruction procedure described in the next section. An example of the result of the flagging process is shown in Fig. 3.3a where all Eulerian grid points are split into three different categories: (a) *forcing points*, which are grid points in the fluid phase that have one or more neighboring points in the solid phase; (b) *solid points*, which are all the points in the solid phase; (c) *fluid points*, which are all the remaining points in the fluid phase. In the solution procedure, the *fluid points* are the unknowns, the *forcing points* are boundary points, while the *solid points* do not influence the rest of the computation. For a stationary boundary the above tagging and flagging process is done only once at the beginning of the computation.

For a moving body the process is repeated at each timestep. In addition, another set of flags is used for the field extension treatment, which is shown in Fig. 3.3b. Again, all Eulerian grid points are split into three different categories: (a) *pseudo-fluid points*, which are grid points in the solid phase that have one or more neighboring points in the fluid phase; (b) *solid points*, which are all other remaining points in the solid phase; (c) *fluid points*, which are all the grid points in the fluid phase. In the field extension procedure, the solution at those *pseudo-fluid points* are extrapolated from the known solution of the *fluid points* and the interface. It is obvious that the forcing points and the pseudo-fluid points are the points closest to the interface inside the fluid and inside the solid, respectively. The proper manipulation of those boundary points is very important to the success of the embedded boundary formulation.

3.1.3 Establishment of Interface-Normal Intersections

With the boundary points identified, the next task is to establish the information required for the reconstruction procedure. Central to this algorithm is the normal from the boundary points to the interface. This normal passes through



(a) Momentum Forcing

(b) Field Extension

Figure 3.3: Identification of boundary points. (a) △ Forcing points, □ fluid points, and ■ solid points for momentum forcing procedure; (b) ▲ Pseudo-fluid points, □ fluid points, and ■ solid points for field extension procedure.

boundary point (x_i, y_j) and intersects the interface at s_n, or (x_n, y_n). Therefore, the unit normal vector of this line is identical to the unit normal vector of the interface on point (x_n, y_n) and can be written as,

$$\begin{aligned} \frac{x_i - x_n}{\sqrt{(x_i - x_n)^2 + (y_j - y_n)^2}} &= n_x = \frac{-y_s}{\sqrt{(x_s^2 + y_s^2)}}, \\ \frac{y_j - y_n}{\sqrt{(x_i - x_n)^2 + (y_j - y_n)^2}} &= n_y = \frac{x_s}{\sqrt{(x_s^2 + y_s^2)}}. \end{aligned} \tag{3.4}$$

Obviously, from the above equations it can be deduced that

$$(x_i - x_n)\, x_s + (y_j - y_n)\, y_s = 0, \tag{3.5}$$

and, substituting Eq. (3.3) into Eq. (3.5), it gives

$$(x_i - x_n)\,(2a_x s_n + b_x) + (y_j - y_n)\,(2a_y s_n + b_y) = 0, \tag{3.6}$$

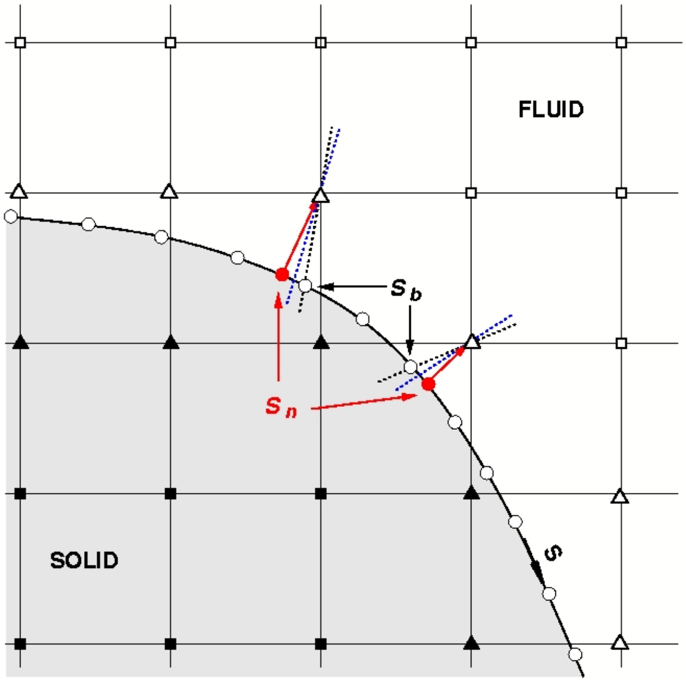


Figure 3.4: Schematic of the solution procedure for interface-normal intersections. △ Forcing points, ▲ Pseudo-fluid points, □ fluid points, and ■ solid points

or

$$s_n = \frac{-b_x(x_i - x_n) - b_y(y_j - y_n)}{2a_x(x_i - x_n) + 2a_y(y_j - y_n)}. \tag{3.7}$$

With the substitution of Eq. (3.1) into Eq. (3.7) above, an equation with only s_n unknown can be obtained:

$$\begin{aligned} (2a_x^2 + 2a_y^2) \quad & s_n^3 \\ +(3a_xb_x + 3a_yb_y) \quad & s_n^2 \\ +(2a_xc_x + 2a_yc_y + b_x^2 + b_y^2 - 2a_xx_i - 2a_yy_j) \quad & s_n \\ +(b_xc_x + b_yc_y - b_xx_i - b_yy_j) \quad & \quad = 0. \end{aligned} \tag{3.8}$$

The equation is solved iteratively using the Newton-Raphson method. The initial solution to this equation is the closest interfacial marker particle, s_b. Fig. 3.4 shows the schematic of the solution procedure from s_b to s_n. After obtaining s_n, the interface-normal intersection coordinates (x_n, y_n) and the unit normal vector $\mathbf{n}$ at the intersection can be calculated from Eq. (3.1) and Eq. (3.2).

3.2 Treatment of Stationary Immersed Boundaries

3.2.1 Momentum Forcing

To demonstrate the basic philosophy of the embedded-boundary formulation let us examine the special case where an Eulerian grid point coincides with the immersed interface, Ψ, on which a Dirichlet boundary condition, u_Ψ, has to be imposed (see Case 2 in Fig. 3.5). The magnitude of external forcing function that will enforce the above boundary condition can be obtained from Eq. (2.30) by simply setting, $\hat{u}_i^k = u_\Psi$, and solving for f_i^k:

$$f_i^k = \frac{u_\Psi - u_i^{k-1}}{\Delta t} - \mathrm{RHS}_i^k, \tag{3.9}$$

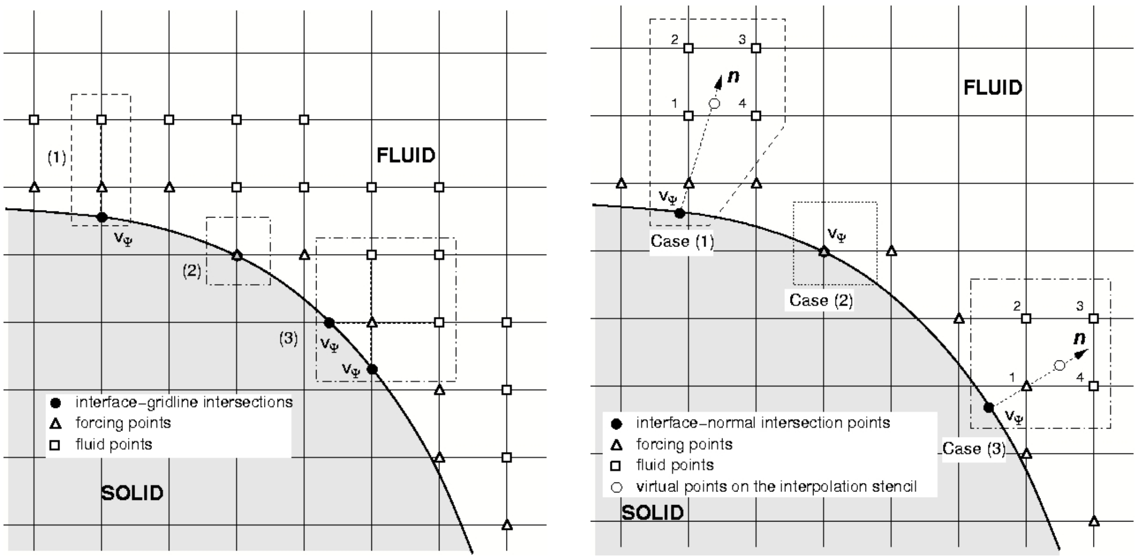
In the framework of the explicit fractional-step method discussed in the previous chapter, the use of f_i, given by Eq. (3.9) will enforce the proper boundary conditions on the predicted velocity, $\hat{u}_i^k$, (substitution of (3.9) into (2.30) gives $\hat{u}_i^k = u_\Psi$). This choice does not compromise the overall temporal accuracy of the splitting scheme, and u_i^k will also satisfy the same boundary condition to the order of Δt^2 [29].

A specific problem, which is related to the evaluation of $\mathbf{f}^k$ for the embedded boundary formulation, is raised when the implicit scheme is used, i.e., the momentum forcing $\mathbf{f}^k$ is no longer able to be obtained from the predicted field $\hat{\mathbf{u}}$, which is obtained using the explicit scheme in the previous section. That is why the momentum forcing terms are omitted in the above equations. Kim *et al.* [29] proposed a provisional explicit step for the evaluation of $\mathbf{f}^k$, which uses RK3 for A_i and the forward Euler scheme for B_i in the above equations:

$$\begin{aligned}\frac{\mathbf{u}^k - \mathbf{u}^{k-1}}{\Delta t} &= \gamma_k A\left(\mathbf{u}^{k-1}\right) + \rho_k A\left(\mathbf{u}^{k-2}\right) + \alpha_k B\left(\mathbf{u}^{k-1}\right) \\ &- \alpha_k \nabla p^k + \mathbf{f}^k. \qquad (3.10)\end{aligned}$$

This approach is adopted in the code and it works well for various cases. Notice that, for the flow past a sphere reported in Chapter 5, the geometry is two-dimensional in computational space without any variation in the azimuthal direction. Therefore, the implicit treatment in the azimuthal direction will not be affected by the requirement of fulfilling boundary conditions on embedded boundaries. This fact can be utilized to simplify the time advancement procedure and the above provisional explicit step can be omitted. On the other hand, the simplification is not applicable any more when the diffusive terms in the radial direction are also treated implicitly.

The detailed implementation has been discussed in Sec. 2.2.3. As mentioned before, for the flow past a sphere reported in this work, direct forcing of the predicted velocity field without the preliminary Euler step still can be applied due to the two-dimensionality of the geometry in cylindrical coordinates. Nevertheless, simulations of the flow past a sphere have also been conducted with the preliminary step for



(a)

(b)

Figure 3.5: Previous Interpolation schemes. (a) One-dimensional scheme in [15]; (b) Two-dimensional scheme in [4]. Cases (1) and (3) illustrate two possible interpolation stencils depending on the interface topology and local grid size.

the momentum forcing and the results have been found to be the same as the other approach used.

3.2.2 Generalized Interpolation Scheme

The above idealized case, although it demonstrates the basic approach, rarely appears in practical applications, where the Eulerian grid nodes almost never coincide with the immersed boundary. In such cases, f_i, has to be computed at grid points near — and not exactly on — the interface. For reasons that will became apparent in the next section, we will compute f_i at all fluid points that have at least one neighbor in the solid phase (these points were labeled forcing points in the previous section). In this case, however, u_Ψ is not known and has to be reconstructed using information from the interface and the surrounding velocity field.

In Fig. 3.5a, the interpolation stencil used by [15] is shown. They employed a second order linear interpolation stencil along the grid lines. But this method has ambiguity as shown in case (3) in the figure that there is no unique direction like case (1) over which the interpolation can be performed. This ambiguity might not decrease the accuracy of the method when the boundaries are stationary. However, it might be a problem, as the flow has a determined direction near the immersed interface when the interface is moving.

To eliminate this ambiguity, Balaras [4] proposed to perform the interpolation along the well defined line normal to the boundary as illustrated in Fig. 3.5b: initially a virtual point is located along the normal; then, the virtual point together with the point on the interface is used to perform the linear interpolation to find u_Ψ in Eq. (3.9) at the location of the forcing point. The velocity at the virtual point is computed from the surrounding fluid nodes using bi-linear interpolation. In this last step, one also has to impose the constraint that the stencil does not involve other forcing points, which can be easily achieved by gradually moving the virtual point further away from the boundary (see for example Case 1 in Fig. 3.5b). Implementation details and a series of applications in laminar and turbulent flows are given in [4].

The interpolation stencil in this method, however, involves a search procedure for the suitable fluid points and those points in the stencil may contain points that do not belong to the neighboring grid lines of the forcing point considered as shown

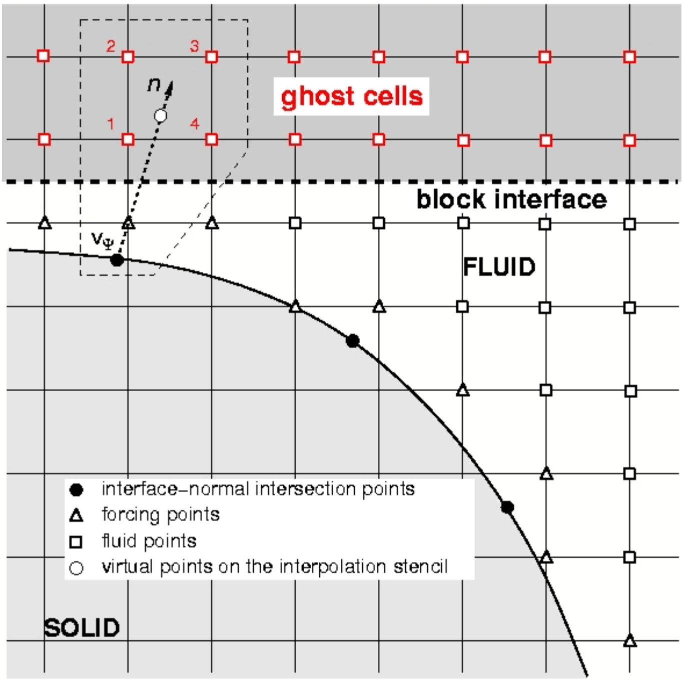

2
3
1
4
n
ghost cells
block interface
v_Ψ
FLUID
● interface–normal intersection points
Δ forcing points
□ fluid points
○ virtual points on the interpolation stencil
SOLID

Figure 3.6: Extra ghost cells required for the interpolation scheme in [4] when domain decomposition technique used for parallelization.

in Fig. 3.6. When domain decomposition is employed in parallel computing, the neighboring block should provide boundary conditions to the current block. In a second order algorithm on Cartesian grids, only one layer of ghost cells are required for the boundary conditions in the same way as a ”real” boundary treated in a finite difference method. But the example case requires extra information from the neighboring block. Of course, the information can be provided through extra communication, provided that the structure of the variable arrays will be modified.

The reconstruction scheme that is used in the present study is based on the work reported in [4], where u_Ψ at the forcing points is computed by means of linear interpolation along the well defined line normal to the boundary. In this study a

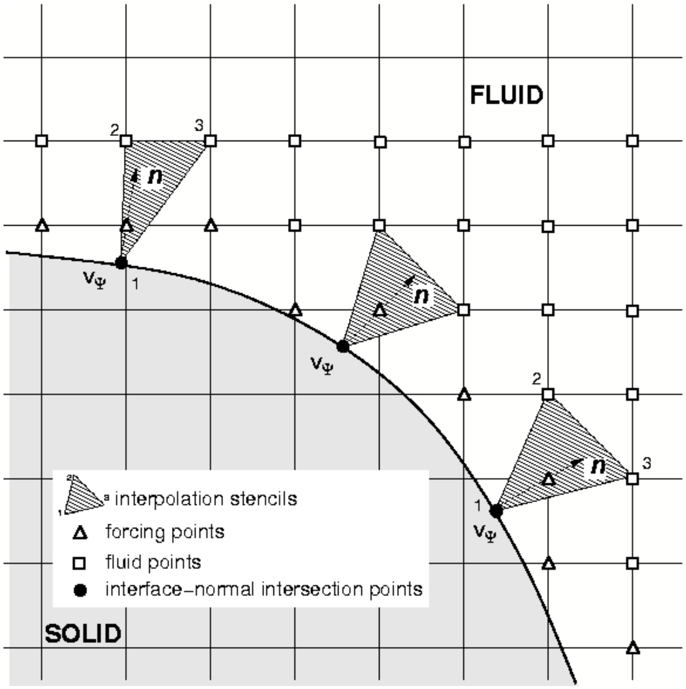


Figure 3.7: Generalized Interpolation stencil.

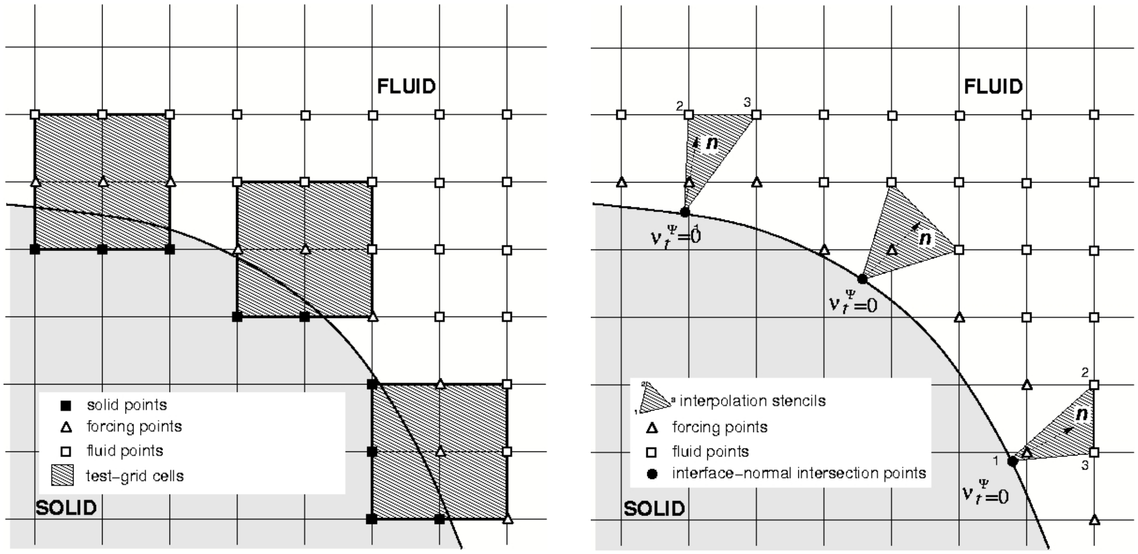
variation of the above method is adopted that is better suited to moving boundary problems. As will be demonstrated in the results section, the proposed method is as accurate and robust as the one in [4]. It utilizes, however, a more compact stencil and allows for the computation of all components of the strain rate tensor on the interface in a straightforward manner. The later feature is particularly attractive in fluid/structure interaction problems, while the former simplifies parallelization. The similarities between the two approaches became apparent when comparing Fig. 3.5b and Fig. 3.7. In the present approach we basically omit the virtual point and replace it with two points on an x-grid line, y-grid line, or along the diagonal. Consequently, the interpolation procedure is now a single-step process that involves two points from the fluid and one on the interface (shaded area in Fig. 3.7). The interpolation coefficients for the case of linear interpolation can be computed as follows: let us assume that any variable ϕ in the two-dimensional space can be written in the following form,

$$\phi = b_1 + b_2 x + b_3 y. \tag{3.11}$$

The coefficients b_1, b_2, and b_3 in Eq. (3.11) can be found by solving the following system:

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \mathbf{A}^{-1} \left| \begin{matrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{matrix} \right] = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}^{-1} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix}, \tag{3.12}$$

where (x_1, y_1), (x_2, y_2), and (x_3, y_3) in the 3×3 matrix $\mathbf{A}$ are the coordinates of the three points in the interpolation stencil shown in Fig. 3.7. For the case of a stationary body, the inversion of matrix, $\mathbf{A}$, at every forcing point is performed in the beginning of the simulation and then stored in memory. For moving boundaries, however, the inversion is performed at every time-step, since new forcing points emerge as a consequence of the boundary motion. The extension of the above interpolation procedure to three dimensions is straightforward (one needs to add a term of the form, $b_4 z$, to the polynomial in Eq. (3.11) to reflect the dependence of the solution near the interface on an additional spatial dimension. Higher-order reconstructions can also be readily achieved using the same overall procedure (see for example [35, 61]).



(a) (b)

Figure 3.8: Treatment of turbulent eddy viscosity for LES. (a) *Test-grid* cells for the boundary points involving solid points from the interior; (b) Interpolation stencils for ν_t at the boundary points based the well-defined values from the fluid points away from the immersed boundary and the boundary condition for ν_t at the interface.

3.2.3 Treatment of Eddy Viscosity

For the simulation of laminar flows or direct numerical simulation of turbulence, the embedded boundary formulation discussed above can satisfy the boundary conditions on the immersed interfaces “exactly” to the second-order accuracy of the spatial discretization scheme for the Navier-Stokes equations. Also, the pressure boundary conditions are implicitly satisfied as the results of linearization of the velocity near the interface. However, for the SGS models the treatment of the turbulent viscosity near the immersed boundary has to be considered.

The eddy viscosity ν_t at the boundary points is required in the computation of diffusion fluxes near the interface, and its calculation is no longer straightforward due to the presence of solid body on the fixed grid. In the case of a dynamic SGS model for example, the evaluation of all test-filtered quantities in the vicinity of the immersed boundary, involves points from the interior of the solid body. A schematic of the test-grid cells on a Cartesian grid with an immersed boundary is shown in Fig. 3.8a. A direct computation ignoring the interface can be problematic and the resulting eddy viscosity may be contaminated by the non-physical solution at the solid points. Explicit modification of the filtering operator or redefining the test-grid cells as in a cut-cell formulation can be considered, but a large number of cases will be introduced and the algorithm becomes complicated.

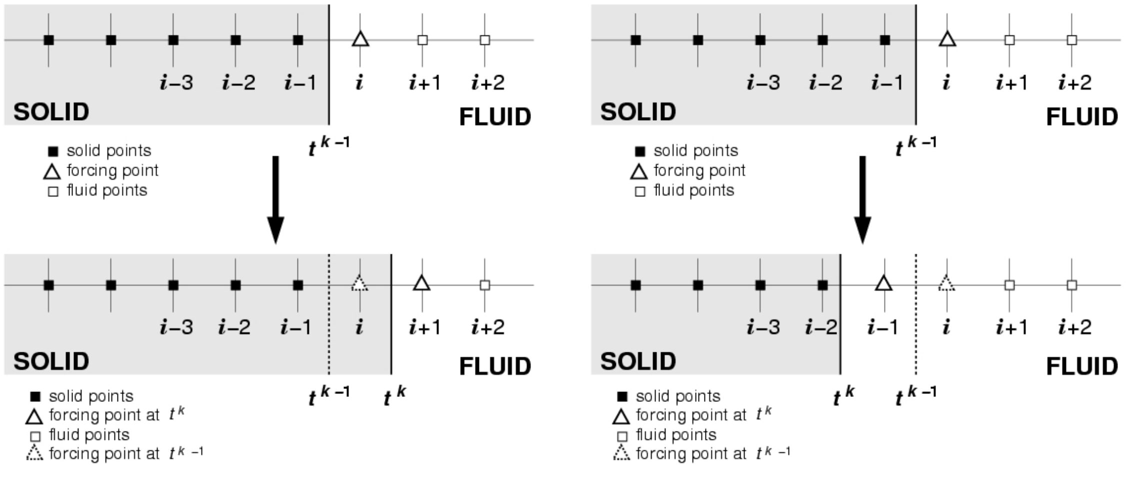
To avoid complex modifications of the filtering operator at these points, a linear reconstruction procedure similar to that used for the velocity field is also applied on ν_t, which follows the same method proposed in [4]. The interpolation stencil for ν_t is shown in Fig. 3.8. The linearized eddy viscosity is only an approximation of the realistic value, which usually does not follow a linear distribution near the solid boundary. In [4] extensive testing of this approach in LES over immersed boundaries has been performed and the error in the results due to the linearized eddy viscosity near the solid boundary has been found to be small.

3.3 Treatment of Moving Immersed Boundaries

3.3.1 Complications in Time Advancement

The general algorithm outlined in sections 3.1 and 3.2 above, is directly applicable to moving boundary problems provided that the grid-interface relation and the interpolation coefficients are re-evaluated every time the location of the interface is updated. In moving boundary problems, however, complications that are usually related to the time advancement scheme can also arise. This is due to the fact that the role of the Eulerian grid points near the interface changes from timestep to timestep (for example a forcing point can became a fluid point at the next timestep and vise-versa), as the solid body moves through the fixed grid. For the time advancement scheme used in the present study, the evaluation of the RHS of the momentum equation at substep k, requires physical values of the velocity vector and pressure, as well as their derivatives from substep $k-1$ at all fluid points (see Eq. 2.30). Due to the fact that the interface changes locations, it is possible that some of the required values from substep $k-1$ are not physical.

To identify the specific cases that are potential sources of error during the time advancement procedure, let us consider a simple, one-dimensional problem where the solid-phase: 1) encroaches upon the fluid phase, and 2) withdraws from the fluid phase as shown in Figs. 3.9a and 3.9b respectively. Due to the CFL restriction of the present scheme the interface cannot move by more than one gird cell in each substep, resulting to two possible changes in the flags of the points near the interface. For case 1: (a) forcing points $\Longrightarrow$ solid points, and (b) fluid points $\Longrightarrow$ forcing points.



(a) (b)

Figure 3.9: A simplified one-dimensional model problem that demonstrates the possible changes of flags of the Eulerian grid nodes near a moving interface. (a) Solid phase encroaches upon the fluid phase; (b) Solid phase withdraws from the fluid phase.

Both of the above do not cause problems to the time advancement scheme. For case 1a, the solution at t^k in all forcing points will be reconstructed anyway, and does not depend on the history of the velocity or pressure field. Case 1b, where Eulerian grid points move into the solid, also does not cause problems since the interior treatment of the body does not influence the solution.

In case 2, where the solid phase withdraws from the fluid (see Fig. 3.9b), the possible flag changes are the following: a) solid points $\Longrightarrow$ forcing points; (b) forcing points $\Longrightarrow$ fluid points. The former case does not generate any problem for the same reason that was discussed above (case 1a). In case 2b, however, it is apparent from Eq. (2.30) that for all the newly defined fluid points the RHS which involves

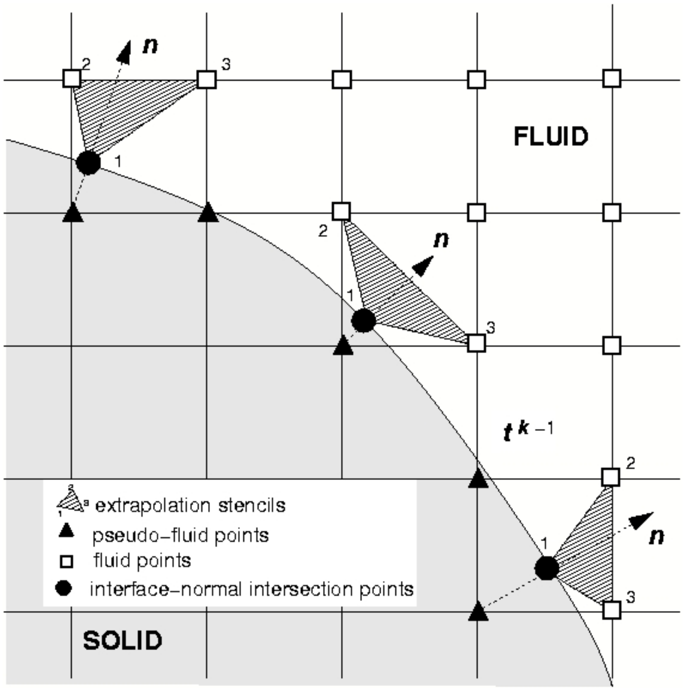


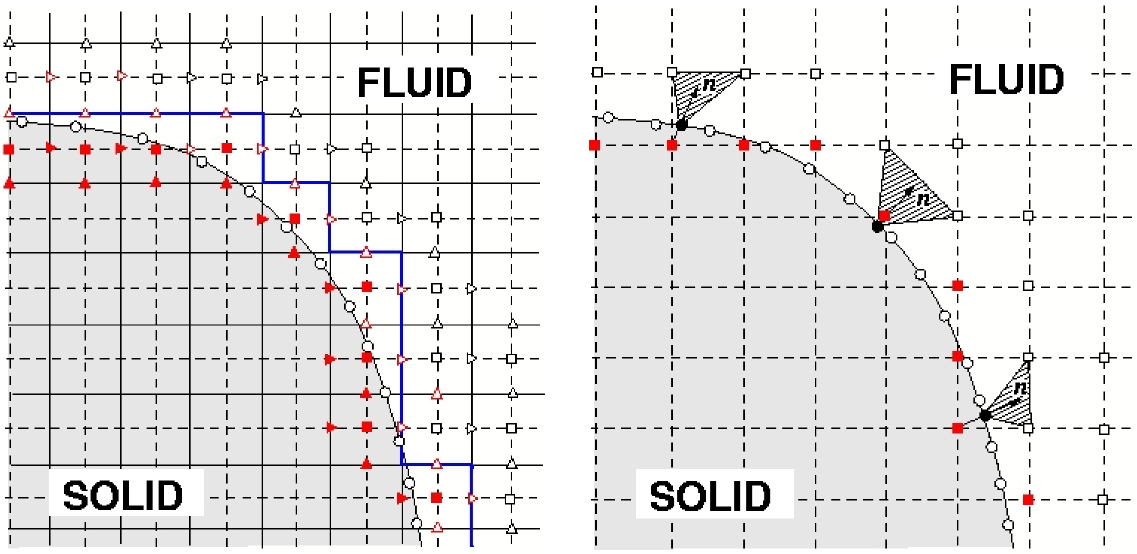
Figure 3.10: The field-extension procedure. Shaded triangles are possible extrapolation stencils for the pseudo-fluid points at substep t^{k-1} where the solution will be extended.

derivatives from previous sub-steps will not be correct. This is because these points at substep $k-1$ were forcing points and although they have the correct values for the velocity and pressure, their derivatives will not have physical values since they involve points from the solid phase. These unphysical values can introduce spurious vorticity near the boundary, leading to large errors.

3.3.2 Field Extension

To avoid forbiddingly complex special treatments of the computation of these derivatives every time such a change of flag is detected during the time advancement procedure, we propose a field-extension procedure, in which the velocity and pressure fields are "extended" in the solid phase at the end of each substep. Practically, with this procedure the velocity fields are extrapolated at the Eulerian nodes defined as pseudo-fluid points in Sec. 3.1 in the solid as illustrated in Fig. 3.10. The pressure fields are also extended into the domain where the pressure solution is non-physical. This way, not only the velocity and pressure at the forcing points at substep $k-1$, but also their derivatives will have physical values, eliminating problems in the computation of the RHS in the next substep for points in Case $2b$. The values of the velocity at the pseudo-fluid points is reconstructed using a procedure which is similar to the one used for the forcing points that is described in section 3.2. Central to this procedure is again the point on the interface where the normal from the corresponding pseudo-fluid point intersects it. In addition to this point the stencil involves two more points from the fluid phase as shown in Fig. 3.10. The interpolation coefficients are computed by solving a system of equations identical to that given by Eq. (3.12). The overall field-extension procedure is similar to the generalized ghost-cell approach discussed in [26, 61]. In both these cases, however, extrapolation is used to enforce boundary conditions on a stationary boundary.

We should also note that due to the staggering of the mesh the selection of points on the pressure grid where the "field-extension" is performed is not only based on their relative location with the interface but also on their association to velocity values that are non-physical. As a result the pressure is also "extended" to some nodes on the pressure grid that would have been classified as "boundary" nodes on the velocity grid as shown in Fig. 3.11



(a) (b)

Figure 3.11: Extension of pressure field. (a) Definition of pseudo-fluid points for pressure; (b) Extrapolation stencils for pressure pseudo-fluid points. ▷ Forcing points for u velocity component, △ Forcing points for v velocity component, ▶ Solid points for u velocity component, ▲ Solid points for v velocity component, □ fluid points for pressure, and ■ pseudo-fluid points for pressure.

3.3.3 Numerical Procedure

Having established the treatment of all Eulerian points in the vicinity of a stationary or moving interface, we can summarize the overall algorithm as follows:

- Given the location of the interface at step k, identify fluid, forcing, solid, and pseudo-fluid points on the Eulerian grid. This procedure needs to be performed only once in the beginning of the computation for problems with stationary boundaries.

- Calculate the predicted velocity field $\hat{u}_i^k$ from Eq. (2.30).
- Reconstruct the predicted velocity $\hat{u}_i^k$ at the forcing points as discussed in Section 3.2.
- Solve the pressure Poisson equation (Eq. 2.31).
- Update the velocity field to u_i^k (Eq. 2.32) and pressure field to p^k (Eq. 2.33).
- Perform the field-extension procedure described above for the pseudo-fluid points. This step is omitted for stationary boundary problems.

As it will be demonstrated in the results section the above algorithm is very robust and satisfies the boundary conditions on an immersed interface "exactly", within the overall accuracy of the discretization scheme. Also, contrary to alternative reconstruction procedures (i.e. generalized ghost cell approaches) it does not require special treatment of the grid cells emerging from the solid, which is usually *ad-hoc* (see [63] for a more detailed discussion on this issue). We should also note that in the above algorithm, boundary conditions for the pressure near the interface are not imposed directly, but they are essentially implicit into the RHS of the Poisson equation (2.31). To clarify this statement one can consider the projection of the momentum equation of a fluid particle on the solid boundary in the direction normal to the interface:

$$\frac{D\mathbf{u}}{Dt} \cdot \mathbf{n} = -\nabla p \cdot \mathbf{n} + \frac{1}{Re} \nabla^2 \mathbf{u} \cdot \mathbf{n}. \tag{3.13}$$

With the present reconstruction procedure the viscous term turns out to be zero as all velocity components are linearized in the vicinity of Ψ (see Sections 3.2 and 3.3), and the above reduces to:

$$\frac{\partial p}{\partial n} = -\frac{D\mathbf{u}}{Dt} \cdot \mathbf{n}, \tag{3.14}$$

Eq. (3.14) represents the boundary condition for the pressure which is usually enforced "directly" in boundary-conforming or cut-cell formulations involving moving boundaries (see for example [63]). In these cases, however, it is obtained from the projection of the inviscid momentum equation in the direction normal to the boundary. A discussion on the behavior of the pressure in embedded-boundary formulations is also given in [15]. In the results section, we will show that the above argument is sound and the pressure behaves correctly in the vicinity of the immersed boundaries.

3.4 Surface Force Calculation

In the case of non-boundary-conforming formulations the computation of the local forces on the surface of the body is a non-trivial problem. A good discussion of this issue can be found in the study by Lai and Peskin [30], where the accuracy of several different force calculation algorithms is discussed in the framework of their immersed-boundary formulation. In the present work we propose a method to compute the local force distribution on a complex body that is based on the same reconstruction stencil used for the velocity field. In general, the force **f** per unit area on a surface element with an outward normal **n** can be written as:

$$f_i = \tau_{ji} n_j = \left[-p\delta_{ij} + \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right] n_j, \tag{3.15}$$

where f_i is the surface force component in the x_i direction, τ_{ji} is stress tensor,

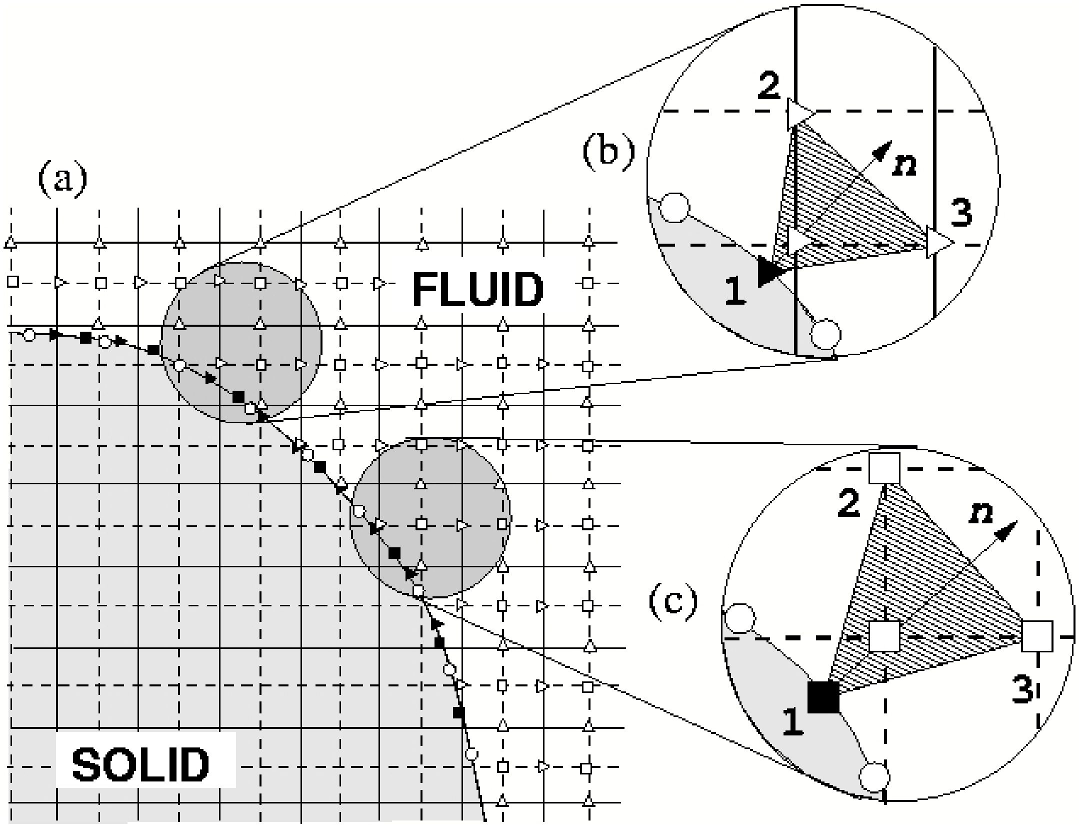


Figure 3.12: Surface force calculation. (a) Arrangement of flow variables on the staggered grid: ▷ u velocity, △ v velocity, □ pressure, ◯ marker particles. The corresponding filled symbols on the interface represent the location where normal intersects the boundary. (b) detail in the vicinity of the boundary for the grid of u velocity component. (c) detail in the vicinity of the boundary for the the grid of pressure. Interpolation stencil is the shaded triangle and line with arrow is the normal to the interface.

μ is the dynamic viscosity of the fluid, and n_j is the direction cosine of $\mathbf{n}$ in x_j direction. To compute $\mathbf{f}$ from Eq. (3.15) one would need to find p and $\partial u_i/\partial x_j$ on the body surface. In the framework of the present scheme, the later term can be computed in a straightforward manner using the stencil and interpolation coefficients that were utilized to reconstruct the velocity field near the interface (Eqs. 3.11 and 3.12). Due to grid staggering, however, the derivatives for each velocity component are computed on different points on the interface. Fig. 3.12a, for example, shows the staggered grid arrangement of all flow variables in the vicinity of a solid body and the corresponding points on the interface where $\partial u_i/\partial x_j$ is computed for each velocity component. In a detailed view around a u velocity point (see Fig. 3.12b), one can identify the interpolation stencil that was used in the reconstruction step. It involves points (2) and (3) from the flow, and point (1) on the interface. Considering Eq. (3.11) at point (1) and differentiating with respect to x_j, we can compute the desired velocity derivatives as follows:

$$\frac{\partial u}{\partial x} = b_2 \text{ and } \frac{\partial u}{\partial y} = b_3, \tag{3.16}$$

where the coefficients b_2 and b_3 have been already computed form Eq. (3.12). In a similar manner, the derivatives for v and w can be found at the interface-normal intersection points of each component using the corresponding interpolation stencils.

The computation of the surface pressure, p, is a little more complicated if one is to avoid simple extrapolation from the interior, which can result in large errors. In the present study we have tested several different strategies and found that an accurate and cost/efficient approach to compute the surface pressure is the one that utilizes the momentum equation normal to the boundary (Eq. 3.14), in addition to information from the interior of the flow. In particular, Eq. (3.14) can be used to obtain an estimate of $\partial p/\partial n$ on any point on the interface, since the material derivative in the RHS of can be computed directly from the known boundary velocities. The normal derivative of pressure can then be expressed as:

$$\frac{\partial p}{\partial n} = \frac{\partial p}{\partial x} n_x + \frac{\partial p}{\partial y} n_y, \tag{3.17}$$

where n_x and n_y are the components of normal unit vector, $\mathbf{n}$, in the x and y directions, respectively. Referring to the stencil shown in Fig. 3.12c, differentiation of Eq. (3.11) at point (1) yields:

$$\frac{\partial p}{\partial x} = b_2 \text{ and } \frac{\partial p}{\partial y} = b_3. \tag{3.18}$$

This stencil is built as if the pressure near the interface will be reconstructed in a manner similar to the velocity field, even though it is only used for the surface pressure computation.

Using equations (3.17) and (3.18) we can rewrite Eq. (3.12) as follows:

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \mathbf{A}^{-1} \begin{vmatrix} \frac{\partial p}{\partial n} \\ p_2 \\ p_3 \end{bmatrix} = \begin{bmatrix} 0 & n_x & n_y \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial p}{\partial n} \\ p_2 \\ p_3 \end{bmatrix}. \tag{3.19}$$

We can now use Eq. (3.19) to compute b_1, b_2 and b_3, and then compute the surface pressure in a straightforward manner from Eq. (3.11). As also mentioned above, due to the grid staggering the velocity derivatives and surface pressure have to be computed on different points on the interface. We then use cubic spline interpolation to compute the corresponding values at the locations of the marker particles. Finally the local forces can be computed from Eq. (3.15) at the marker particle location. Numerical integration is used to calculate integral quantities like drag and lift forces.

3.5 Fluid-Structure Interaction

Up to this point the imposition of boundary conditions on a body moving through a fixed orthogonal grid has been discussed in detail. The boundary conditions were assumed to be known functions of space and time. This information would be sufficient to solve problems with prescribed motion. However, in cases where the location of the body is determined by the fluid force, additional ODEs governing the motion of the body have to be considered. Also the coupling between the Navier-Stokes equations and ODEs becomes an important aspect.

In the present study the fluid and the structure will be treated as elements of a single dynamical system, and all governing equations will be integrated simultaneously, and iteratively in the time-domain. A fundamental complication with the application of a time-domain approach to fluid-structure interaction problems like the one described above is that the prediction of the hydrodynamic loads requires knowledge of the motion of the structure and vice-versa. To overcome this complication an iterative predictor-corrector scheme that accounts for the interaction between the hydrodynamic loads and the motion of the structure has been developed. The procedure is based on Hamming's 4^{th}-order, predictor-corrector method [9]. Hamming's scheme was adapted to the present problem to avoid evaluating the

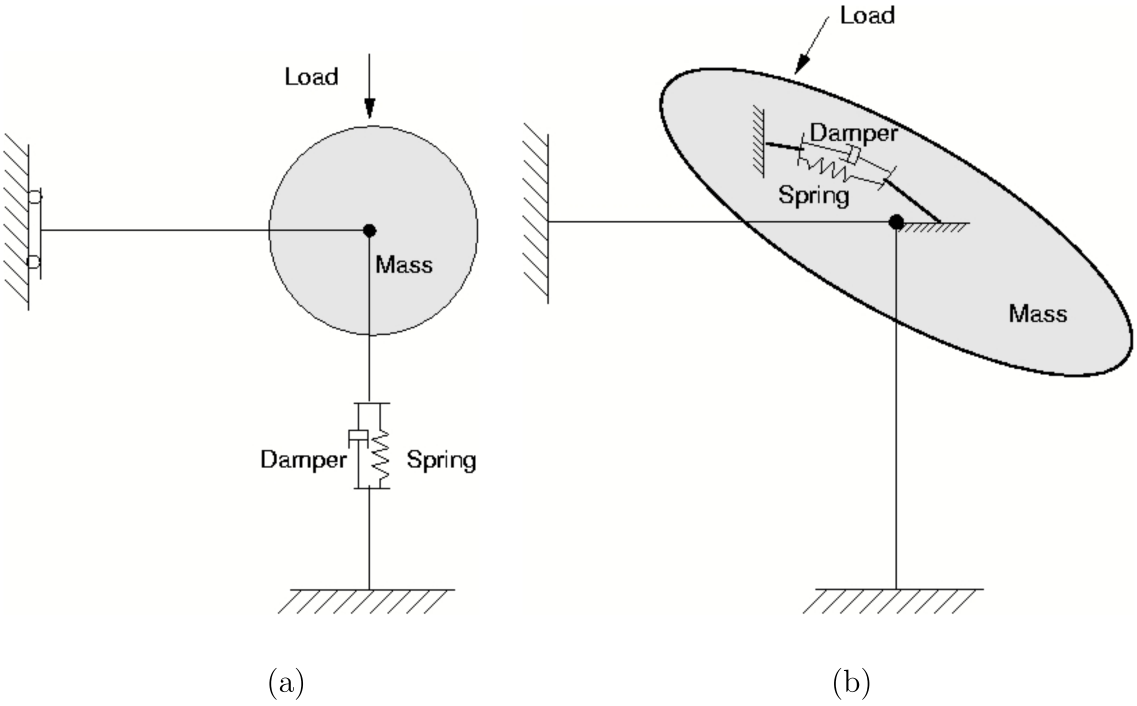


Figure 3.13: Schematic of rigid-body motion. (a) One degree of freedom of translation in vertical direction; (b) One degree of freedom of rotation about the origin.

hydrodynamic loads at fractional time steps and to accommodate hydrodynamic loads that are proportional to the acceleration (the so called added-mass effect). In the following paragraphs the structural solver and the coupling scheme will be discussed in detail. A schematic of rigid-body motion is shown in Fig. 3.13. The structure is modeled as a mass-spring-damper system. Fig. 3.13a shows one degree of freedom of translation in the vertical direction, and Fig. 3.13b shows the one degree of rotation about the origin. The motions are governed by the same ODE. Although in Chap. 5 the rigid-body rotation of the leaflets will be considered, the motion is prescribed instead of governed by a separated equation. Therefore, in this work only the problem of vortex-induced vibrations for a circular cylinder is considered. Here we introduce the equation of motion that generally governs the motion of the cylinder oscillating in the $(X - Y)$ plane:

$$\mathbf{M}\ddot{\mathbf{x}}\left(t\right) + \mathbf{C}\dot{\mathbf{x}}(t) + \mathbf{K}\mathbf{x}\left(t\right) = \mathbf{f}\left[\mathbf{x}\left(\tau\right), \dot{\mathbf{x}}\left(\tau\right), \ddot{\mathbf{x}}\left(t\right)\right], \qquad \forall\tau \in 0 \leq \tau \leq t \tag{3.20}$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{C}$ is the damping matrix, $\mathbf{K}$ is the stiffness matrix and $\mathbf{f}$ is the vector of generalized hydrodynamic forces per unit span, and $\mathbf{x}\left(t\right) = X_0\left(t\right)\mathbf{i} + Y_0\left(t\right)\mathbf{j} + \Theta(t)\mathbf{k}$ is the vector of generalized structural displacements. Here, $X_0\left(t\right)$ and $Y_0\left(t\right)$ are the displacements of the center of mass of the cylinder in the x and y directions, respectively, and $\Theta(t)$ is the rotation of the cylinder around an axis perpendicular to the $(X - Y)$ plane. Equations (2.1) and (2.2) governing the dynamics of the fluid and equation (3.20) governing the dynamics of the body need to be solved in a coupled manner. The overall algorithm will be given later.

Equation (3.20) can be rewritten in non-dimensional form, as a system of $2n$ first-order, non-linear, ordinary-differential equations (n is the number of degrees-of-freedom of the structure) as follows:

$$\dot{\mathbf{y}}\left(t\right) = \mathbf{F}\left[\mathbf{y}\left(\tau\right), \dot{\mathbf{y}}\left(\tau\right)\right], \qquad \forall\tau \in 0 \leq \tau \leq t, \tag{3.21}$$

where half of the vector $\mathbf{F}$ represents generalized velocities and the other half represents generalized forces divided by the corresponding inertias. In general, the loads depend explicitly on $\mathbf{y}$ and implicitly on the history of the motion and the acceleration of the structure. For this reason Hamming's 4^{th}-order, predictor-corrector method [9] is used to integrate equation (3.21) in the time domain. The details of the basic numerical procedure used to determine the current value of the vector $\mathbf{y}$ are given next:

1. Let $t_j = j\Delta t$ denote the time at the j-th time step, where Δt is the time-step size used to obtain the numerical solution, and

$$\mathbf{y}^{j} = \mathbf{y}\left(t_j\right), \quad \dot{\mathbf{y}}^{j} = \dot{\mathbf{y}}\left(t_j\right), \quad \mathbf{F}^{j} = \mathbf{F}\left[\mathbf{y}\left(t_j\right)\right] \tag{3.22}$$

2. Compute the predicted solution, ${}^{p}\mathbf{y}^{j}$, and modify it, ${}^{1}\mathbf{y}^{j}$, using the local truncation error, $\mathbf{e}^{j-1}$, from the previous time step:

$${}^{p}\mathbf{y}^{j} = \mathbf{y}^{j-4} + \frac{4}{3}\Delta t\left(2\mathbf{F}^{j-1} - \mathbf{F}^{j-2} + 2\mathbf{F}^{j-3}\right), \quad {}^{1}\mathbf{y}^{j} =^{p} \mathbf{y}^{j} + \frac{112}{9}\mathbf{e}^{j-1} \tag{3.23}$$

3. Correct the modified, predicted solution by:

$${}^{k+1}\mathbf{y}^{j} = \frac{1}{8}\left[9\ \mathbf{y}^{j-1} - \mathbf{y}^{j-3} + 3\Delta t\ \left(\ {}^{k}\mathbf{F}^{j} + 2\ \mathbf{F}^{j-1} - \mathbf{F}^{j-2}\right)\right] \tag{3.24}$$

where ${}^{k}\mathbf{F}^{j} = \mathbf{F}\left({}^{k}\mathbf{y}^{j}\right)$ and ${}^{1}\mathbf{y}^{j} = {}^{p}\mathbf{y}^{j}$. k is the iteration index. Convergence is achieved when the iteration error, $e^{j} = \left\|\ {}^{k+1}\mathbf{y}^{j} - {}^{k}\mathbf{y}^{j}\right\|_{\infty}$ is less than a prescribed tolerance ϵ.

4. Compute the local truncation error, $\mathbf{e}^{j}$, and the final solution, $\mathbf{y}^{j}$, at step j and advance to the next timestep:

$$\mathbf{e}^{j} = \frac{9}{121}\left(\ {}^{k+1}\mathbf{y}^{j} - {}^{p}\mathbf{y}^{j}\right), \quad \mathbf{y}^{j} = {}^{k+1}\mathbf{y}^{j} - \mathbf{e}^{j} \tag{3.25}$$

Note that in the above procedure information from four previous timesteps is required. Therefore, starting from the initial conditions, Euler, Adams-Moulton, and Adams-Bashforth methods were used.

The fluid solver described in previous chapter is integrated into the above procedure as follows:

1. Find the predicted location and velocity of the body using equation (3.23).

2. Find the predicted fluid velocity and pressure fields using equations (2.30)–(2.33) with the boundary conditions provided by step 1. Then, compute the resulting loads on the structure.

3. Compute the new location and velocity of the body using equation (3.24).

4. Check for convergence. If e^j is greater than the prescribed tolerance, ϵ, repeat steps 2 to 4. If convergence is achieved go to step 5.

5. Find final position and velocity of the body using equation (3.25), and the final fluid pressure and velocity fields from equations (2.30)–(2.33).

In all computations reported in this study a tolerance of $\epsilon = 10^{-8}$ was used. The number of iterations required for convergence at each time step varied from 2 to 6 depending from the stiffness of the problem.

3.6 Parallelization

The parallelization is implemented via slab decomposition and the parallel communication between among blocks is done using the MPI library. To minimize the communications at the block (slab) interface, the decomposition is carried out in the direction with the largest number of points (usually the streamwise direction). Fig. 3.14a shows the slab decomposition in the streamwise direction for the momentum equations. Here, each block is equally sized, which greatly simplifies the mesh partition and provides optimal load balance. Minor modifications are

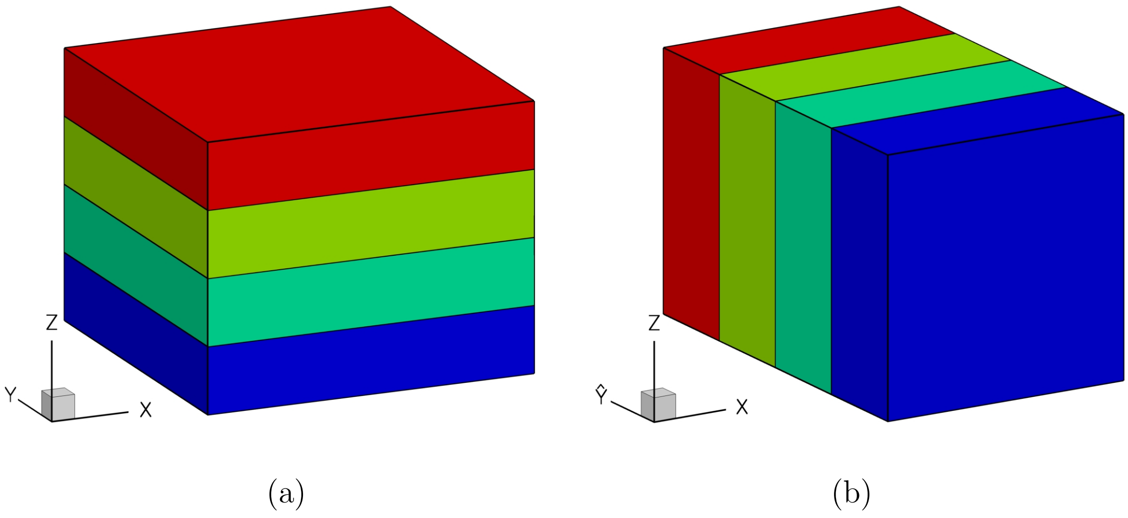


Figure 3.14: Domain decomposition parallelization. (a) Slab decomposition in streamwise direction for momentum equations; (b) Slab decomposition in wavenumber space for the Poisson equation.

introduced to the original serial solver, as ghost cells at the block interfaces are utilized to provide information from neighboring blocks. A schematic of message communication among slabs is shown in Fig. 3.15. In the current implementation, only one layer of ghost cells from the neighboring block are required as the standard second-order central difference scheme is used for the spatial discretization. Usually the solution of the pressure Poisson equation is the most expensive part of a Navier-Stokes solver, and the parallelization via domain decomposition adds further complications. In the present study, the computational grid is always uniform in the spanwise direction; thus the fast Fourier transformation (FFT) can be applied to the Poisson equation in the spanwise direction and a series of decoupled two-dimensional problems are obtained and solved using the direct solver from

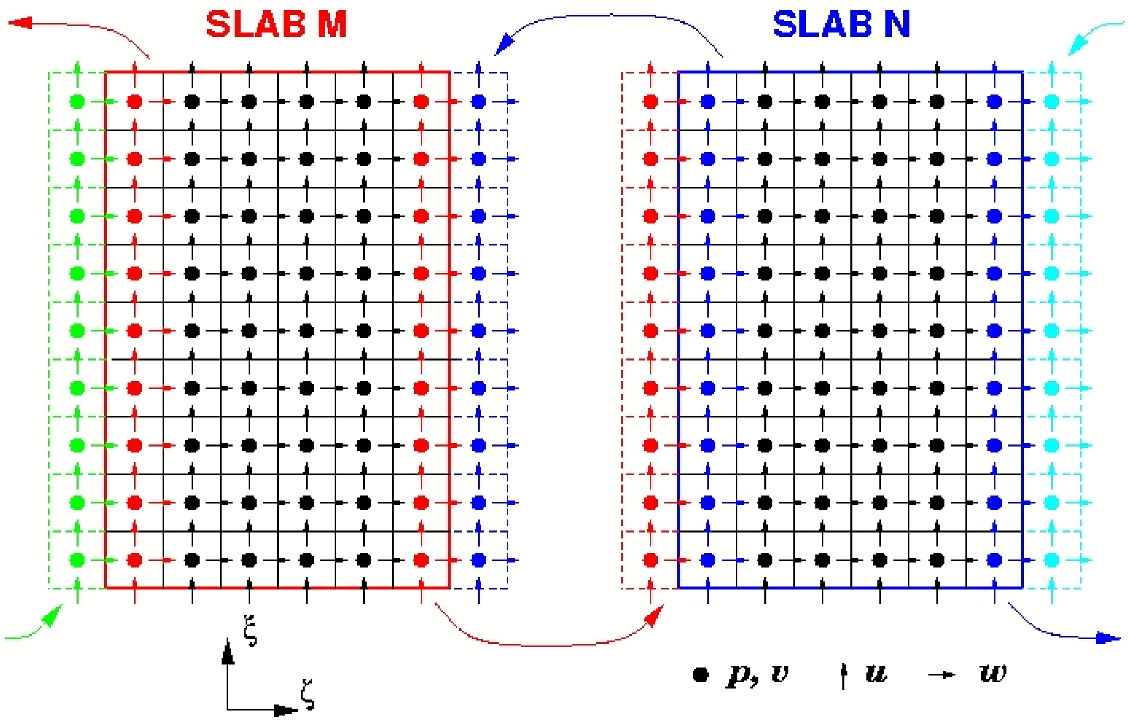


Figure 3.15: Inter-slab communication via layers of ghost cells.

FISHPACK [57]. The FFT is performed in each block and no information from neighboring blocks is required. However, global communication is required to swap the slabs in th estreamwise direction into slabs in wavenumber space, and swap back after the direct solution of the two-dimensional problem for each wavenumber (Fig. 3.14b shows a slab decomposition in wave number space). Nevertheless, the overhead for the global communication is small and, as shown in the results which follow, the speedup for the problem considered is linear.

The issue of the parallelization of the embedded boundary formulation has been addressed in Sec. 3.2.2 and will be discussed here.

The parallel speedup for a simulation including the embedded boundary treatment is shown in Fig. 3.16. Linear speedup is obtained up to 16 processors, which

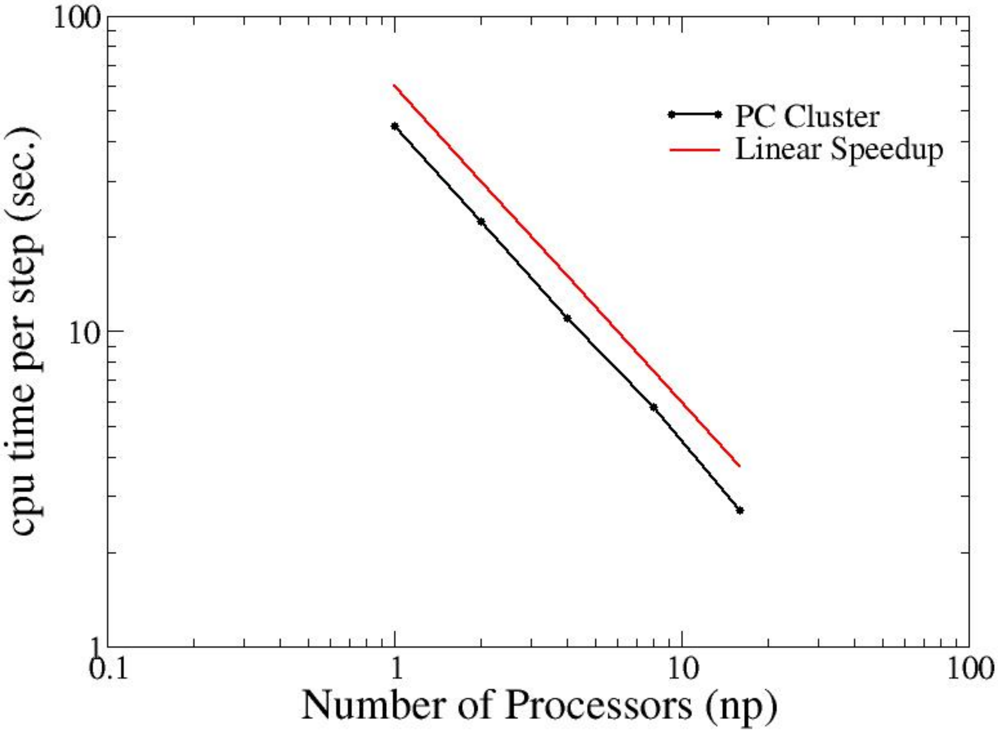


Figure 3.16: Parallel Speedup for a 96 × 64 × 384 Grid.

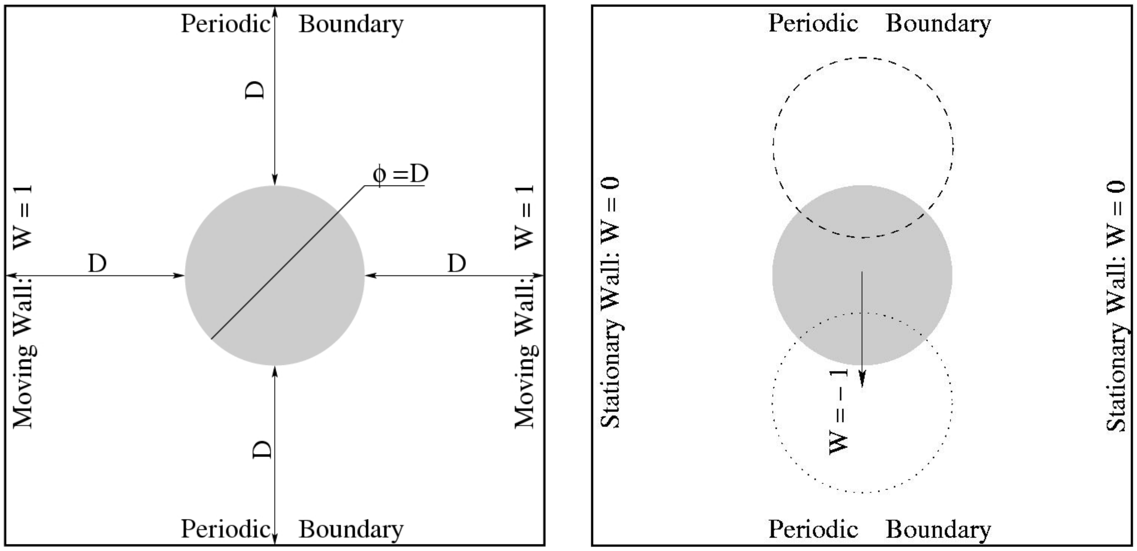
was the limit of the cluster where the preliminary testes were carried out.

Chapter 4

Two-Dimensional Validation Cases

To evaluate the accuracy of the proposed methodology several cases with increasing complexity have been considered. The first case in Sec. 4.1, which simulates the two dimensional flow around one element of a moving cylinder-array in a planar channel, is a demonstration of the formal accuracy of the method. Then, laminar flow problems involving prescribed motions of two-dimensional bodies are computed in Sec. 4.2: the flow induced by the harmonic in-line oscillation of a circular cylinder in a quiescent fluid (Sec. 4.2.1), and the flow from a transversely oscillating cylinder in a free-stream (Sec. 4.2.2). Both cases are characterized by complex vortex-vortex and vortex-structure interactions and present a challenge to non-boundary conforming formulations. In addition, detailed experimental data and numerical results from boundary-fitted methods are available in the literature, allowing for a detailed validation. Then, in Sec. 4.3 vortex-induced vibrations of circular cylinders will be used in the present study to evaluate the accuracy and efficiency of the proposed fluid-structure interaction scheme. Two different configurations are considered: one degree-of-freedom oscillations in the cross-stream direction (Sec. 4.3.1); and two degree-of-freedom oscillations in both the streamwise and cross-stream directions (Sec. 4.3.2). In both cases the selected parametric space was as close as possible to well documented laminar two-dimensional flow problems in the literature.

4.1 Formal Accuracy

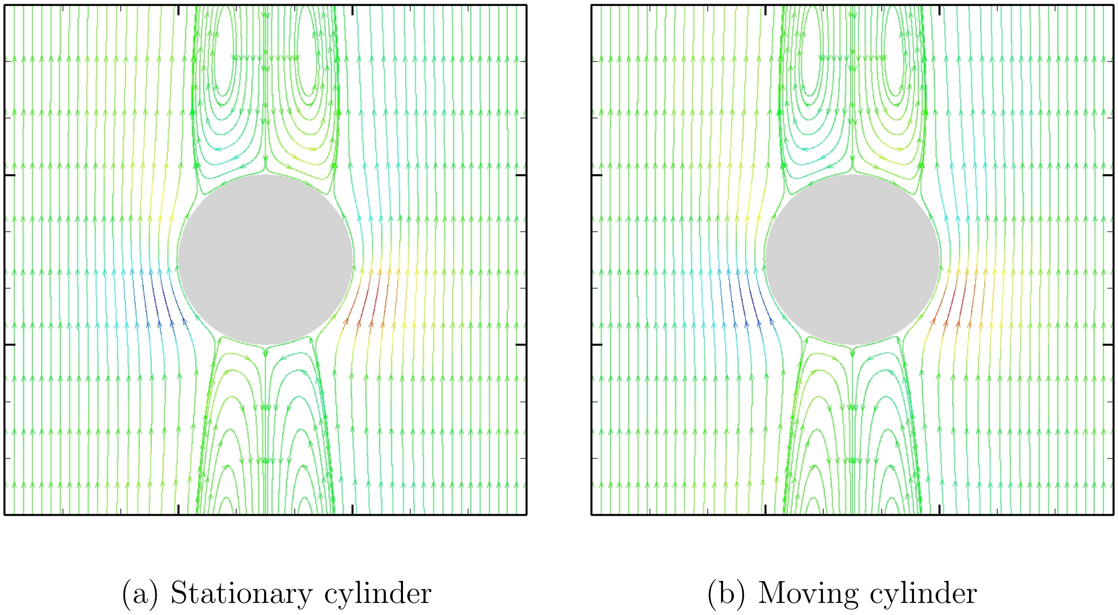


(a) Stationary cylinder

(b) Moving cylinder

Figure 4.1: Flow around a periodic array of cylinders in a plane channel: Sketch of the computational domain.

There are no cases in fluid dynamics that involve complex moving boundaries and have analytical solutions, to serve as reference for the formal accuracy study of the proposed method. For this reason we have designed a test problem which involves a body moving through a fixed grid, and performed a detailed numerical accuracy study. In particular, we have considered the flow around a periodic array of cylinders moving with constant velocity $W = -1$ in a planar channel. A sketch of the computational domain and boundary condition setup is shown Fig. 4.1a. Non-slip conditions are used on the walls and periodic boundary conditions in the direction of motion. When the reference frame moves with the cylinder, the case



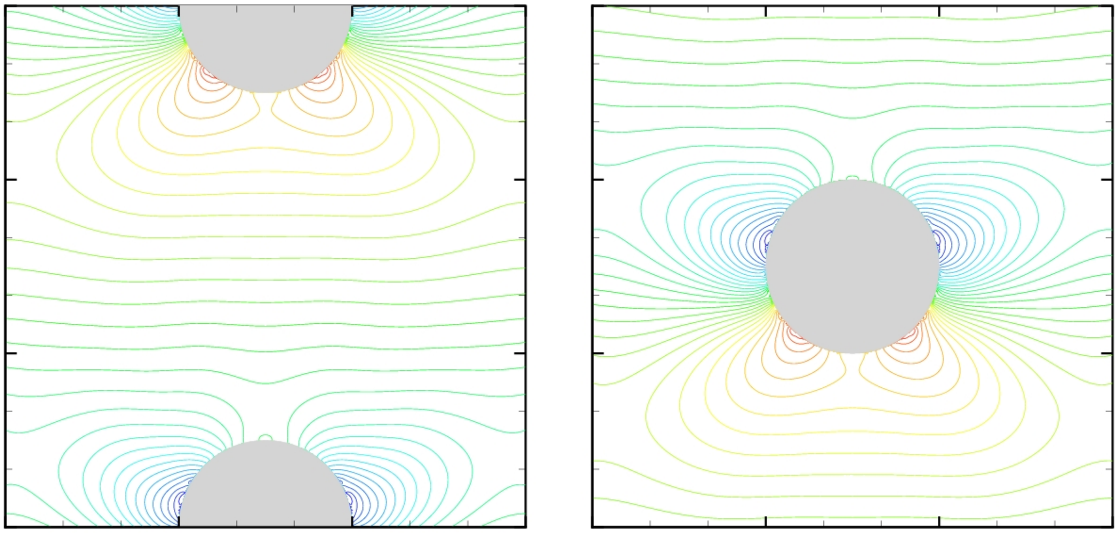
(a) Stationary cylinder

(b) Moving cylinder

Figure 4.2: Streamlines on a 150×150 uniform grid colored by u velocity component (a reference frame moving with the body is used for the moving cylinder case).

shown in Fig. 4.1b is recovered, where the cylinder is stationary and the walls have a velocity of $W = 1$. The Reynolds number based on the diameter and velocity of the cylinder (or the velocity of the moving wall) is 100. The corresponding flow fields are visualized in Fig. 4.2 where streamlines are shown. A recirculation bubble is formed and restricted between two cylinders, which keeps the flow two-dimensional and steady. Note that for the moving cylinder a reference frame following the cylinder is used. In both cases, the flow is identical even though we have two distinct computational configurations.

Fig. 4.3 shows the instantaneous pressure contours when the cylinder is at two different locations: the center and the boundary of the computational domain. The almost identical pressure distributions in Fig. 4.3a and Fig. 4.3b also verify



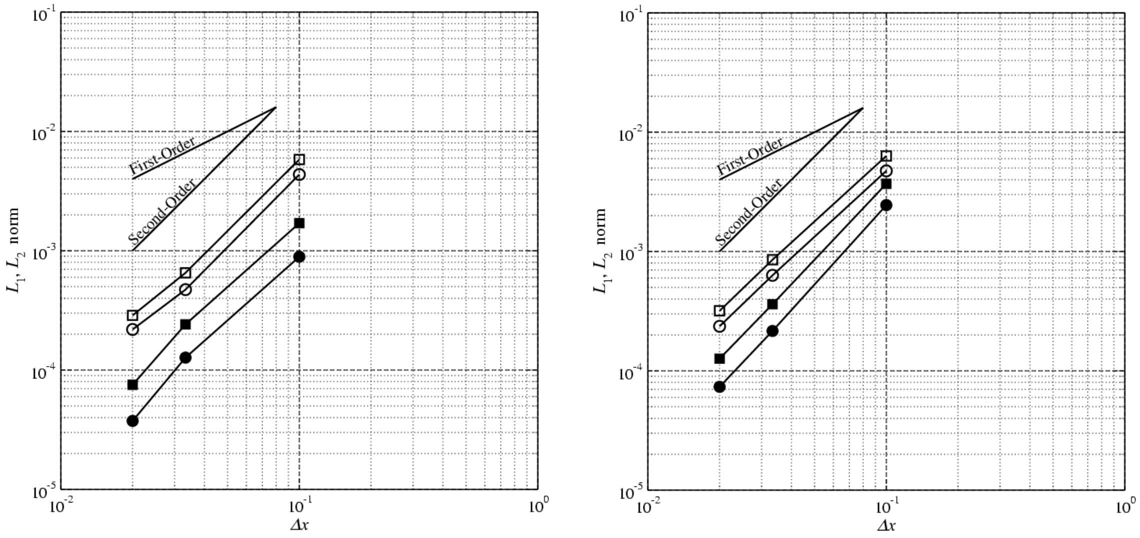
(a) $t/T = 0.5$

(b) $t/T = 1.0$

Figure 4.3: Instantaneous pressure contours at two different locations.

the stableness and accuracy of the current method.

Calculations on four levels of uniform grids were carried out: 30×30, 90×90, 150×150, and 450×450. The above choice -there is not the usual doubling of grid points in the next finer grid level- is due to the staggered variable arrangement. We have selected the specific grids in a way that every node on coarser grids can be directly located on the reference, finest, grid to avoid interpolations when comparing the two solutions. The L_1 and L_2 norms of the error, which measure the difference between the solutions from coarser grids and the reference grid, versus the corresponding grid resolution are shown in Fig. 4.4. The results from the stationary cylinder case and the moving cylinder case are given in Fig. 4.4a and Fig. 4.4b, respectively. Note that we use the solution on the finest grid from the stationary cylinder case for both plots presented. For the moving cylinder case, the L_1 and L_2



(a) Stationary cylinder

(b) Moving cylinder

Figure 4.4: □ L_1 norm of the error; ○ L_2 norm of the error. Open symbols are for w and filled for u.

norms of the error computed using the solution on the 450×450 grid are almost the same as those in Fig. 4.4b. In the figure, lines representing the first- and second-order accuracy slopes are included. It is clearly shown that the error decreases with Δx^2, verifying the second-order accuracy of the present method. This means that, within the order of the accuracy of the discretization scheme, the moving boundaries are represented "exactly". By comparing Fig. 4.4b with Fig. 4.4a, we can see that the error of the moving cylinder case is of the same order as the stationary cylinder case.

4.2 Forced Vibrations of a Cylinder

4.2.1 In-Line Oscillating Cylinder in a Fluid at Rest

The interaction of an oscillating circular cylinder with a fluid at rest is a well documented model-problem in the literature because of its relevance to a variety of applications. The two key parameters in this flow are the Reynolds number, $Re = \rho U_{\max} D/\mu$, and the Keulegan-Carpenter number, $KC = U_{\max}/fD$ ($U_{\max}$ is the maximum velocity of the cylinder, D is the diameter of the cylinder, ρ is the density of the fluid, μ is the dynamic viscosity of the fluid, and f is the characteristic frequency of the oscillation). The parametric space we selected in the present work corresponds to the LDA experiments and numerical simulations reported by Dütsch *et al.* [13]. In particular, the cylinder's translational motion is described by a simple harmonic oscillation, $x(t) = -A\sin(2\pi ft)$, where A is the amplitude of the oscillation. The Reynolds and Keulegan-Carpenter numbers were set to $Re = 100$ and $KC = 5$ respectively. For this setup the flow remains two-dimensional with periodic vortex shedding. The size of the computational domain is $50D \times 30D$ in x and z, respectively, with the cylinder located at the center. Radiative boundary conditions are applied to all the far-field boundaries. Three different grid levels were used (240×120, 480×240, and 960×480) to investigate the sensitivity of the results to numerical resolution. Near the cylinder the grid is approximately uniform in both directions with local spacing of $0.05D$, $0.025D$, and $0.0125D$ respectively.

All computations were started from a quiescent field and the integration in time was performed until periodic vortex shedding was established. In Fig. 4.5

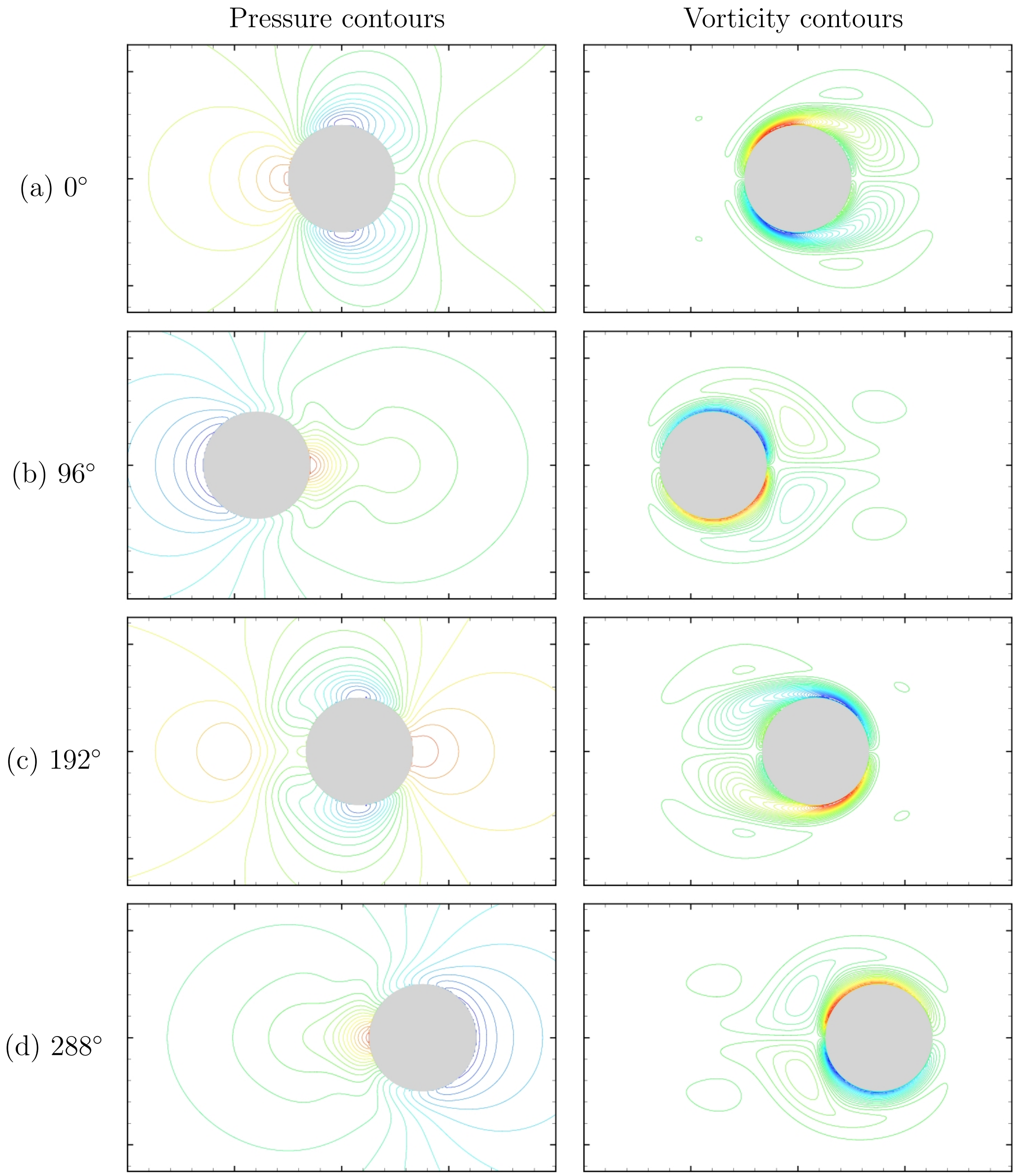
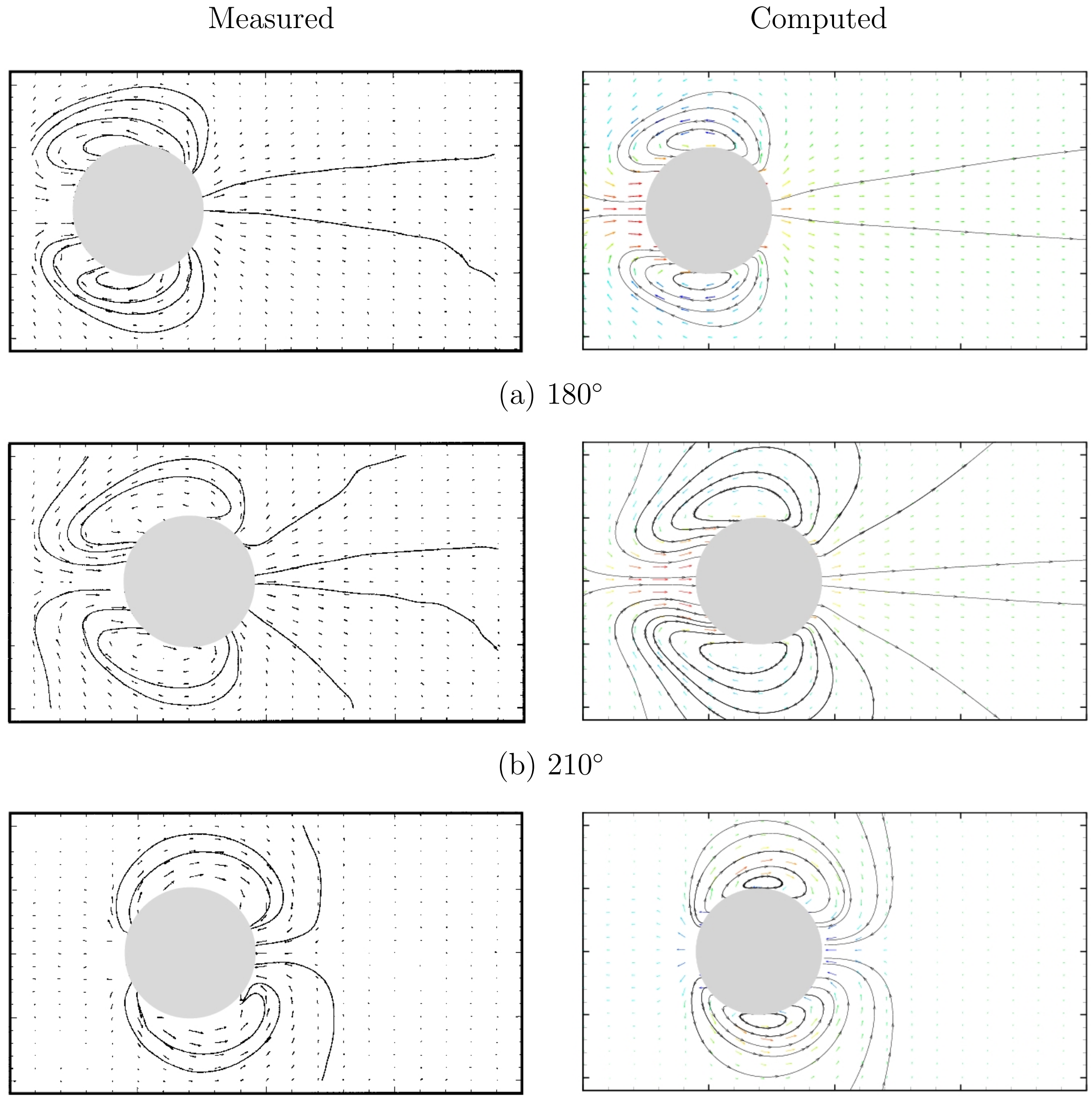


Figure 4.5: In-line oscillating cylinder in a fluid at rest ($Re = 100$ and $KC = 5$). Pressure and vorticity contours at four different phase-angles. $-1.1 < P < 0.6$ with intervals of 0.09 and $-26 < \omega_y < 26$ with intervals of 0.95.



(a) 180°

(b) 210°

(c) 330°

Figure 4.6: Measured (left) and computed (right) velocity vectors and streamlines in the vicinity of the cylinder at $Re = 100$ and $KC = 5$.

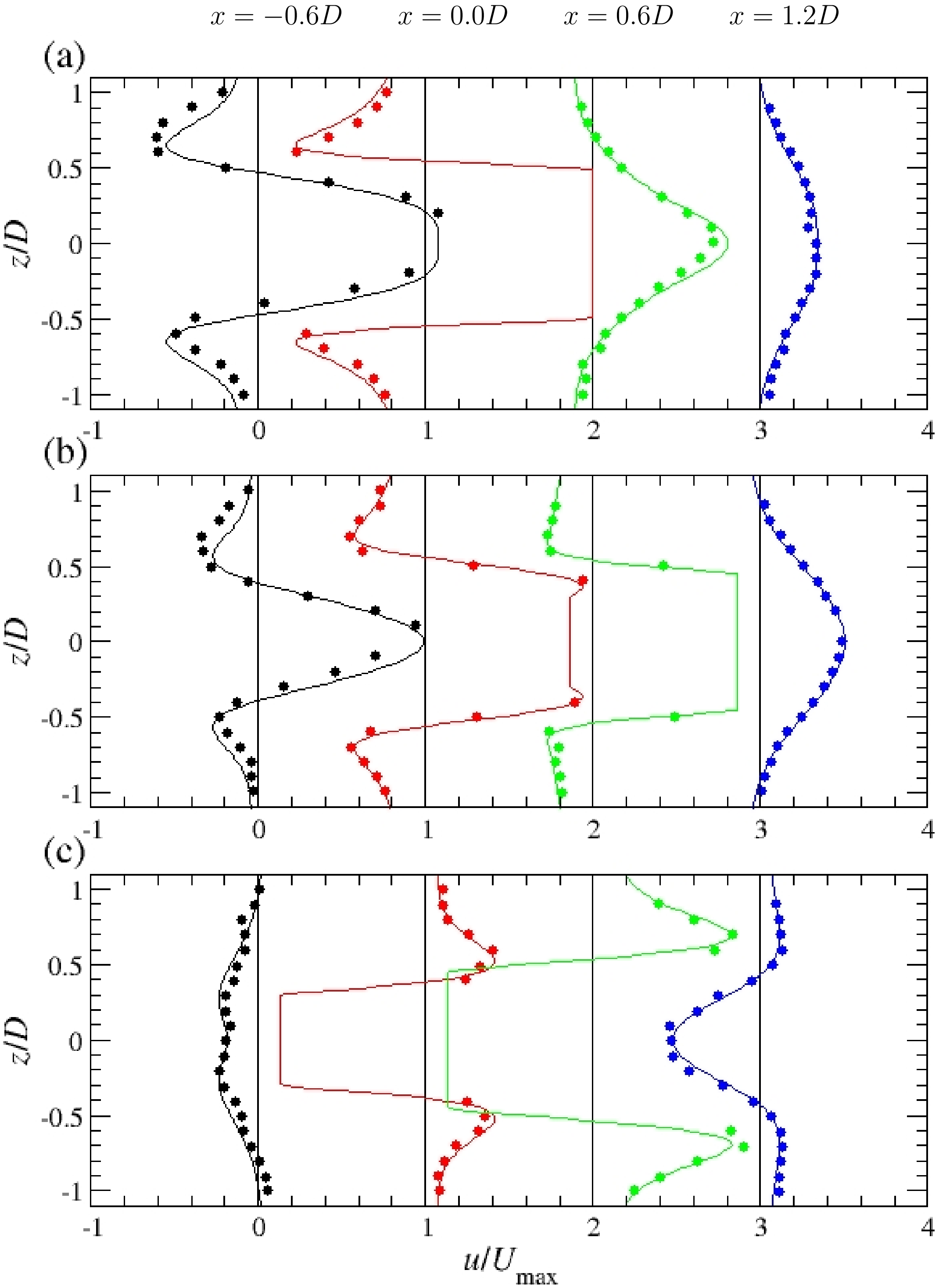


Figure 4.7: In-line oscillating cylinder in a fluid at rest (Re = 100 and KC = 5). Profiles of u velocity component at: (a) 180°; (b) 210°; (c) 330°. Symbols: experiment; Lines: Computation.

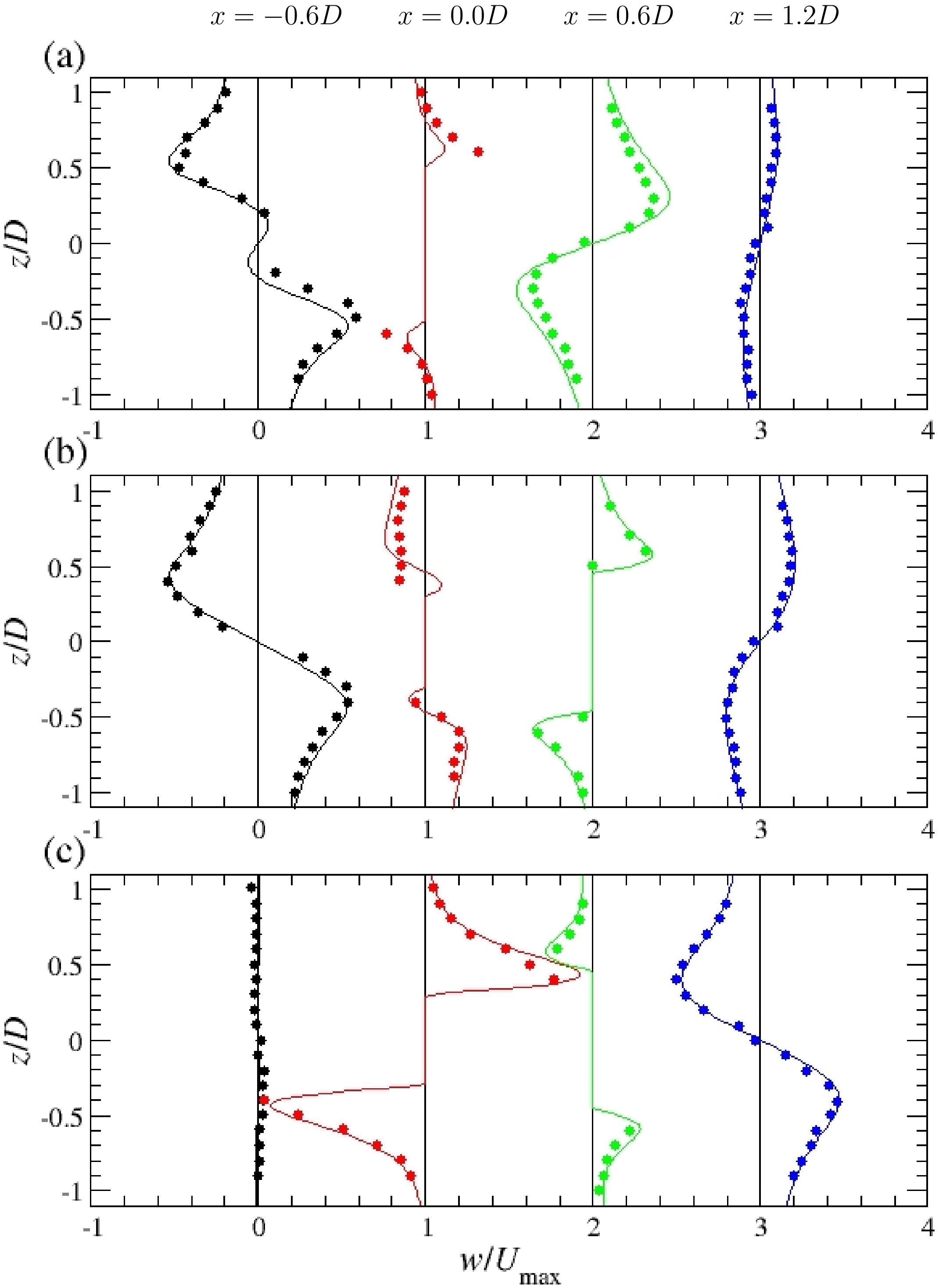


Figure 4.8: In-line oscillating cylinder in a fluid at rest (Re = 100 and KC = 5). Profiles of w velocity component at: (a) 180°; (b) 210°; (c) 330°. Symbols: experiment; Lines: Computation.

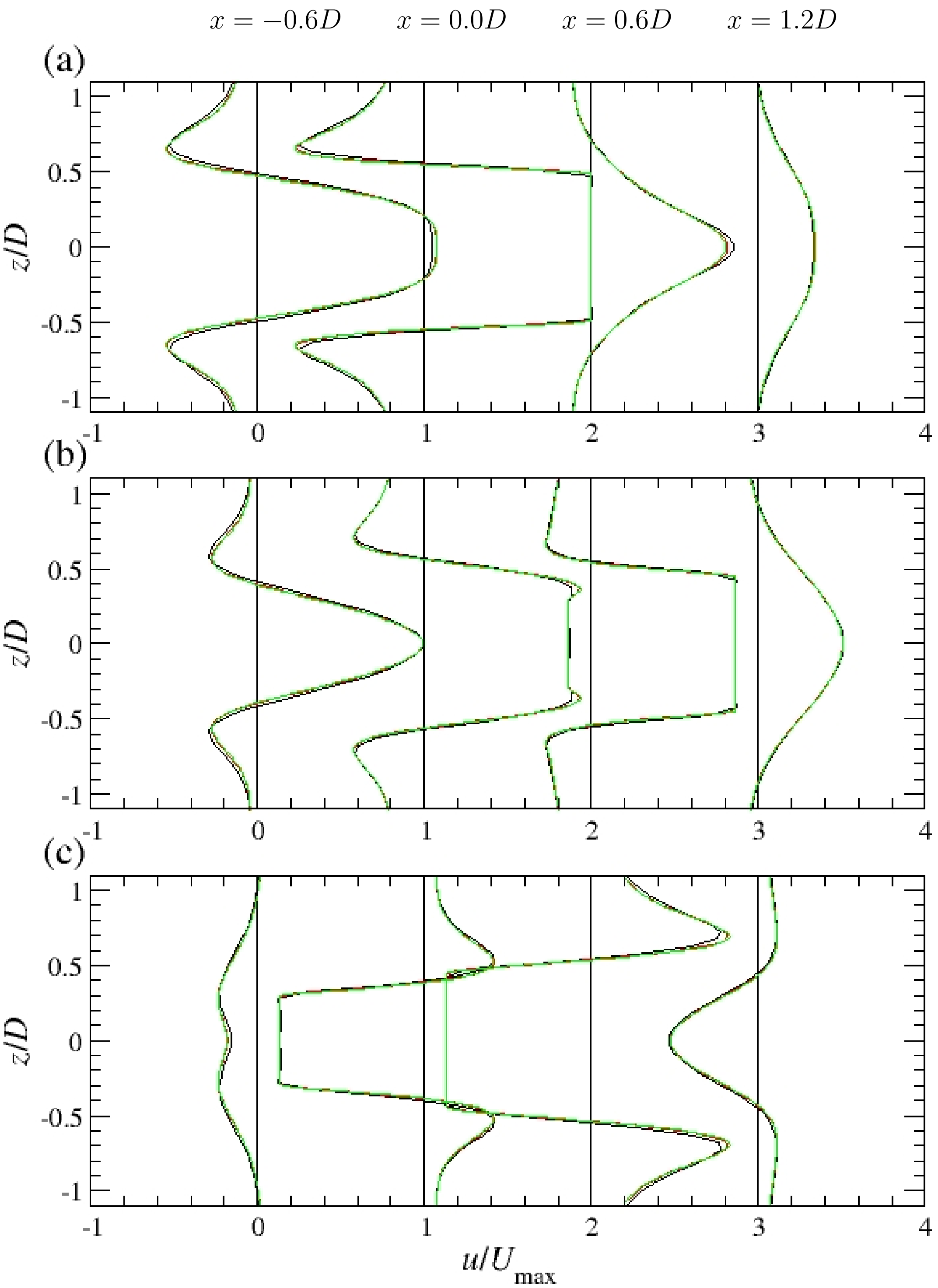


Figure 4.9: Convergence of the u velocity component for different grid levels at $Re = 100$ and $KC = 5$. (a) 180°; (b) 210°; (c) 330°. —— : grid 240 × 120; —— : grid 480 × 240 ; —— : grid 960 × 480.

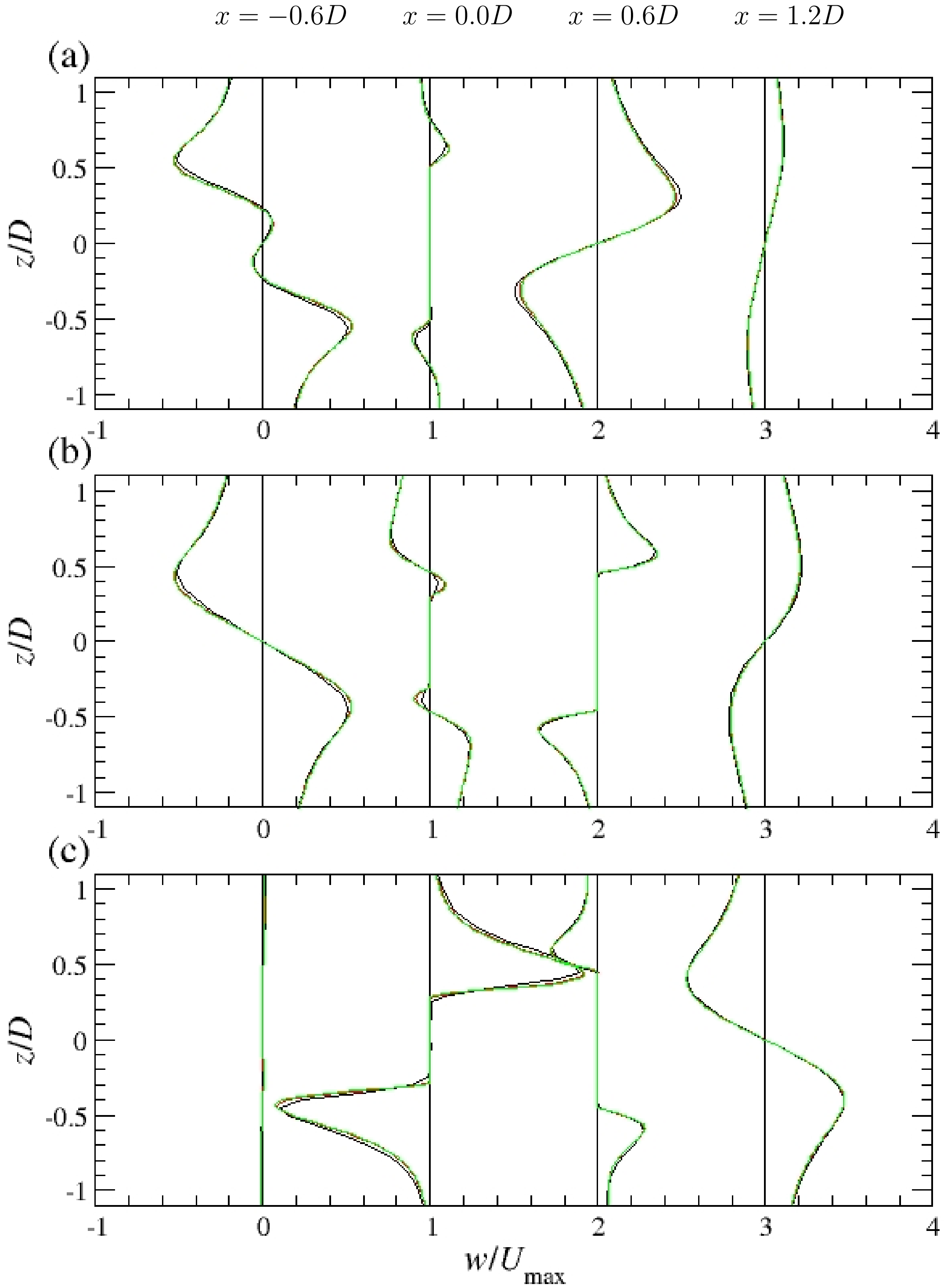


Figure 4.10: Convergence of the w velocity component for different grid levels at $Re = 100$ and $KC = 5$. (a) 180°; (b) 210°; (c) 330°. —— : grid 240 × 120; —— : grid 480 × 240 ; —— : grid 960 × 480.

the pressure and vorticity contours at four different phase-angles are shown. One can observe the two fixed stagnation points at both ends of the cylinder that also define the axis of symmetry of the flow. As the cylinder moves to the left, two thin boundary layers develop on the upper and lower sides (see Fig. 4.5a), that eventually separate giving rise to two counter-rotating vortexes of exactly the same strength. The vortex generation procedure stops as the body reaches its extreme left location, as shown in Fig. 4.5b. Then the cylinder moves backwards, and the same process takes place on its right side. During this part of the cycle, the vortex pair generated earlier is split and pushed away the cylinder (see Fig. 4.5c). These results are in very good agreement with the corresponding results reported in [13], demonstrating that the present method can properly capture the dynamics of the vorticity field. Fig. 4.6 shows the comparisons of measured and computed velocity vector fields at three different phase angles of the cylinder motion. Very good agreement between the experimental and computed results is emphasized by the coincidence of the measured and computed streamlines in the vicinity of the cylinder.

To further examine the accuracy of the proposed method, quantitative comparisons with the experimental results are considered. Fig. 4.7 and Fig. 4.8 show the computed velocity profiles on the intermediate grid, at four x locations and three different phase-angles, in comparison with the experiments. The agreement is very good. We also investigate the grid independence of our solution by comparing the velocity profiles obtained from the three grids in Fig. 4.9 and Fig. 4.10. It is very clear that even the coarse grid 120×240 gives very good results. As the grid is refined, there is a obvious convergence of the solution. In Fig. 4.11 the evolution in

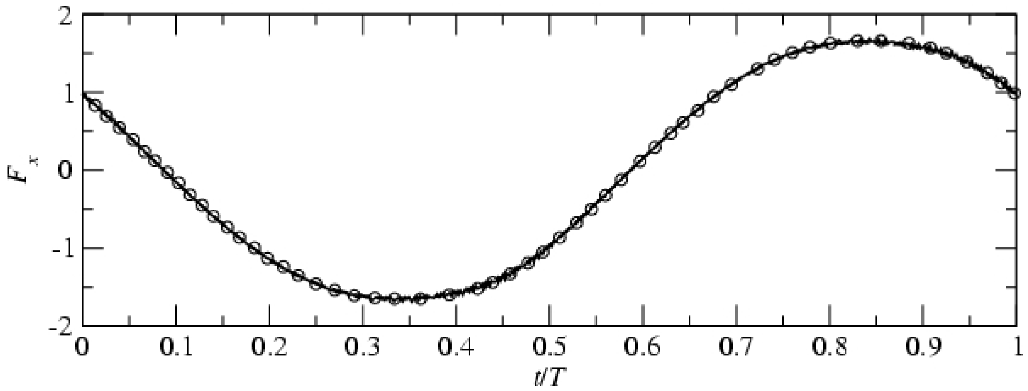
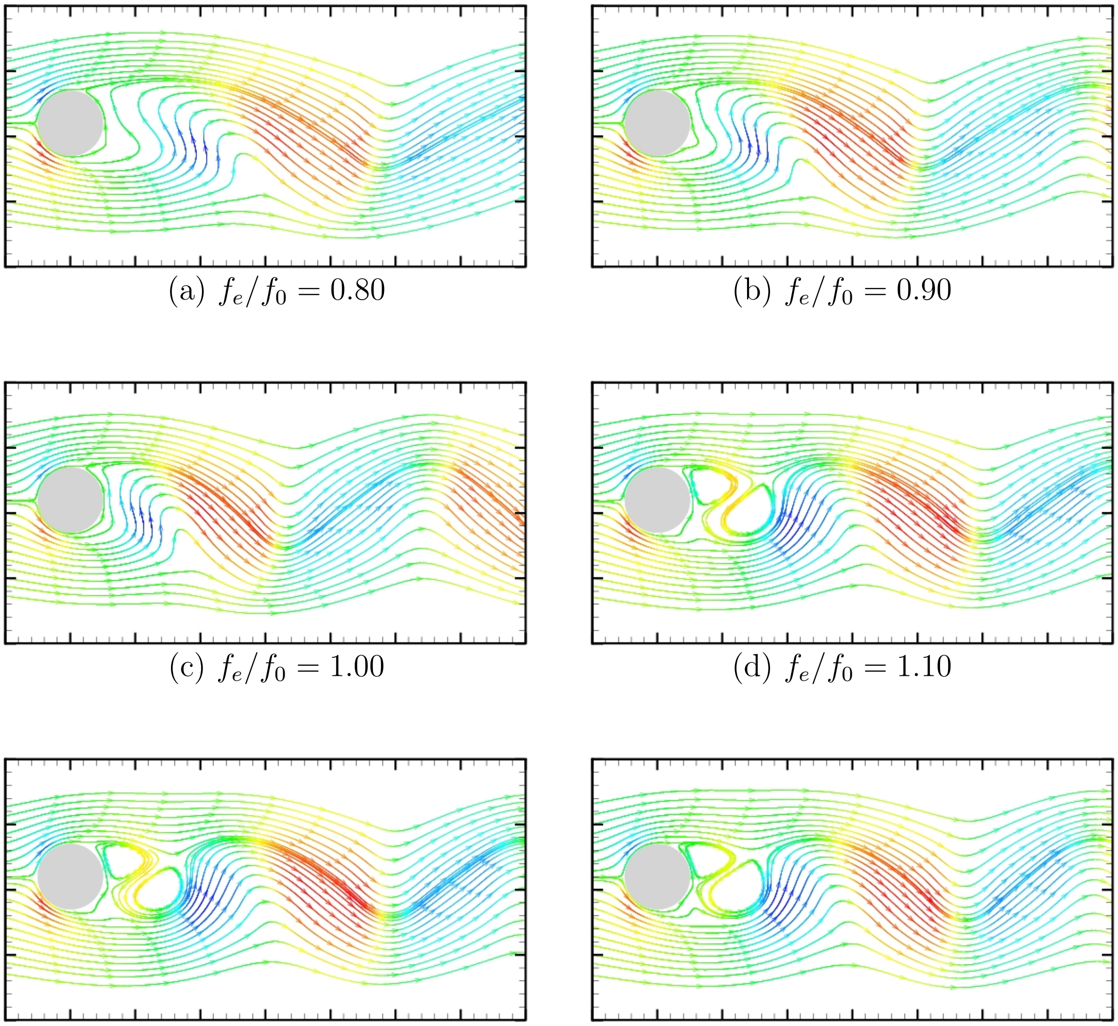


Figure 4.11: Evolution in time of the in-line force acting on a cylinder oscillating in a fluid at rest at $Re = 100$ and $KC = 5$. ∘ Boundary-conforming simulation [13]; ········ present computation 240 × 120; - - - present computation 480 × 240; —— present computation 960 × 480.

time of the in-line force, $F_x(t)$, acting on the cylinder is shown for the two finest grids in comparison with the reference simulation results. Details on the actual definition and normalization of $F_x(t)$ can be found in [13]. Here it is important to point out that $F_x(t)$ is practically calculated by integrating the surface shear stress and pressure using the algorithm described in Section 3.4. On both fine grids the agreement with the reference computations is very good (results are within 2%) demonstrating that the forces on the cylinder's surface (or at least their integral at an instant in time) can also be predicted very accurately with the proposed approach.

4.2.2 Transversely Oscillating Cylinder in a Free-Stream

The fluid-structure interaction problem becomes more complicated when the oscillating cylinder is in a free-stream. In the present study we only consider cases of



(a) $f_e/f_0 = 0.80$

(b) $f_e/f_0 = 0.90$

(c) $f_e/f_0 = 1.00$

(d) $f_e/f_0 = 1.10$

(e) $f_e/f_0 = 1.12$

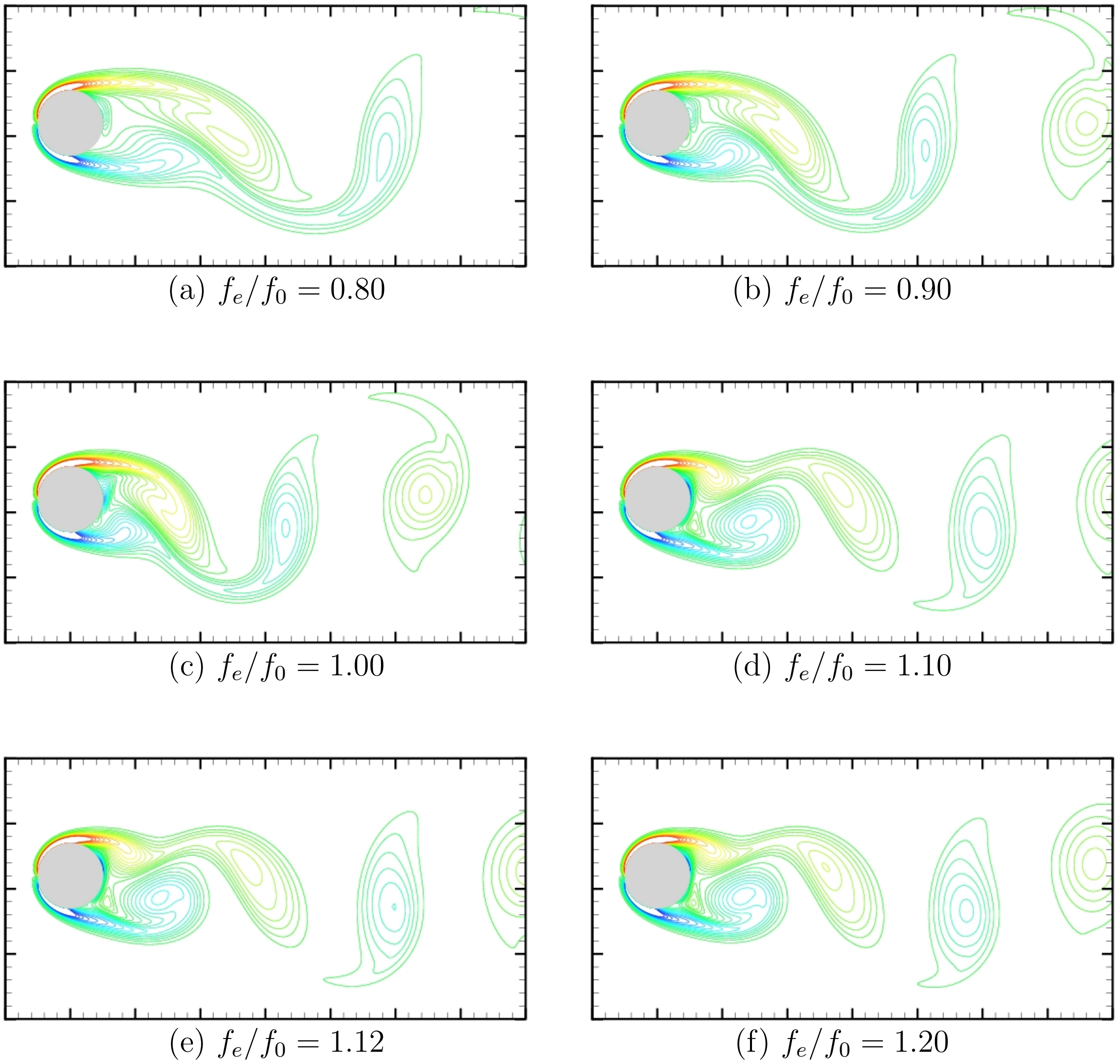
(f) $f_e/f_0 = 1.20$

Figure 4.12: Cylinder oscillating in a free-stream: instantaneous streamline patterns when cylinder is located at its extreme upper position. Streamlines are colored with transverse velocity component values.

transverse oscillation. The relevant parameters for such a problem are the Reynolds number $Re = \rho U_\infty D/\mu$ (U_∞ is the free-stream velocity), the excitation frequency, f_e, and the oscillation amplitude, A. When A is larger than a threshold value the synchronization condition can appear -manifested by a jump in the phase-angle, ϕ, between the lift force and the body motion- when the frequency of the oscillation is varied around the natural shedding frequency, f_0, of the stationary cylinder. The variations in this phase angle are associated with the change in sign of the energy transfer between the fluid and the cylinder. Capturing the physics of these flows, even at low Reynolds numbers, requires a very accurate prediction of the vorticity production and dynamics on the surface of the body, which makes their simulation a very challenging task, especially using non-boundary-conforming formulations.

The choice of parameters in this flow reproduces the conditions in the experiments by Gu *et al.* [22] and the subsequent simulations by Guilmineau & Queutey [23], where an accurate boundary-conforming numerical method is used. To conduct quantitative comparisons, however, we mainly use the numerical results since the amount of quantitative information reported in the experiments is limited. To match the above conditions the prescribed cylinder motion is chosen to be $z(t) = A\cos(2\pi f_e t)$, where $z(t)$ is the cross-stream location of the cylinder as a function of time. Six different cases were considered at $Re = 185$, $A = 0.2D$ and $0.8 \leq f_e/f_0 \leq 1.2$. This way we cover a wide spectrum of the conditions in [22, 23].

The computational domain is $50D \times 30D$, in the streamwise and cross-stream directions respectively. The cylinder is located at a distance of $10D$ from the the inlet boundary, and $15D$ from the upper and lower boundaries. At the outflow



(a) $f_e/f_0 = 0.80$

(b) $f_e/f_0 = 0.90$

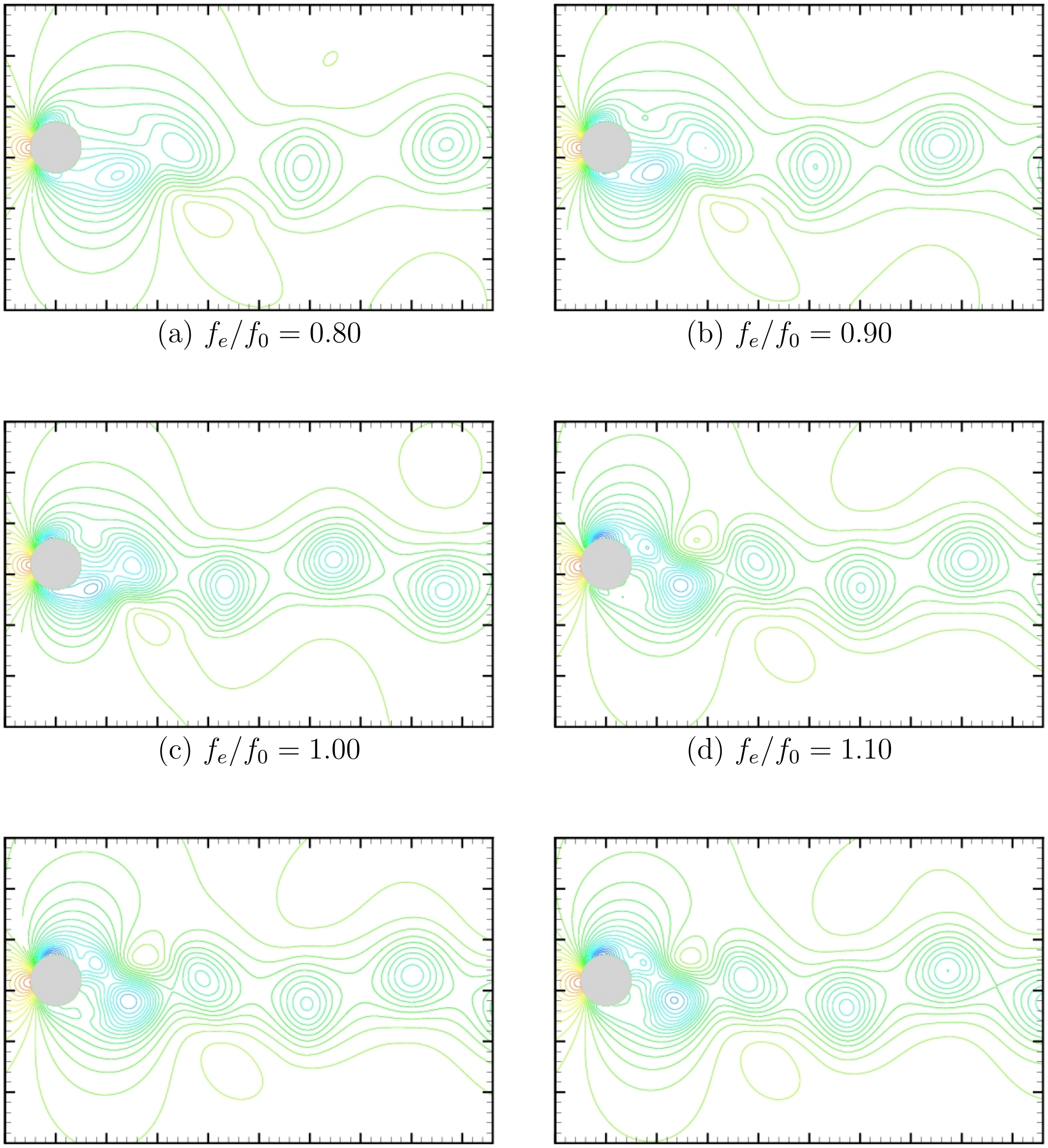
(c) $f_e/f_0 = 1.00$

(d) $f_e/f_0 = 1.10$

(e) $f_e/f_0 = 1.12$

(f) $f_e/f_0 = 1.20$

Figure 4.13: Cylinder oscillating in a free-stream: instantaneous vorticity contours when cylinder is located at its extreme upper position. $-8 < \omega_y < 8$ with intervals of 0.5.



(a) $f_e/f_0 = 0.80$

(b) $f_e/f_0 = 0.90$

(c) $f_e/f_0 = 1.00$

(d) $f_e/f_0 = 1.10$

(e) $f_e/f_0 = 1.12$

(f) $f_e/f_0 = 1.20$

Figure 4.14: Cylinder oscillating in a free-stream: instantaneous pressure contours when cylinder is located at its extreme upper position.

boundary a convective boundary condition is used, and radiative conditions are imposed at the freestream boundaries. The grid is stretched in both directions to cluster points in the vicinity of the body. To investigate the dependency of our results on grid resolution three different meshes with an increasing number of nodes are considered: 200×160, 400×320, and 800×640 points in the streamwise and cross-stream directions respectively. For all three grids the flow past a fixed cylinder at $Re = 185$ is initially computed, to obtain the natural shedding frequency, f_0. As for the previous test problem the results converge with grid refinement. For example, $\overline{C}_D = 1.366$, ${C_D}^{\mathrm{rms}} = 0.029$, and ${C_L}^{\mathrm{rms}} = 0.461$ on the finest grid, which are within 1% of the values on the previous grid level. The Strouhal number is 0.197 for all the three grids.

Fig. 4.12 shows the instantaneous streamline patterns, Fig. 4.13 shows the vorticity contours on the finest grid, and Fig. 4.14 shown the instantaneous pressure contours when the cylinder is at the extreme upper position. Six cases are shown, where f_e/f_0 is gradually increased from 0.8 to 1.2. A sudden change in the topology of the near wake can be observed at $f_e/f_0 = 1.10$, (see Fig. 4.12d, 4.13d, and 4.14d), as manifested by the appearance of concentric, closed streamlines, and remains the same up to the highest value of excitation frequency. This topology suggests the existence of concentrated vorticity in the near wake, which can be verified from the vorticity contours shown in the same figure. The different wake structure is also reflected in the lift and drag coefficients shown in Fig. 4.15. For values of f_e/f_0 greater than 1.0, both the drag and lift exhibit signs of the influence of a higher harmonic. This behavior can be attributed to the strengthening of the opposite-

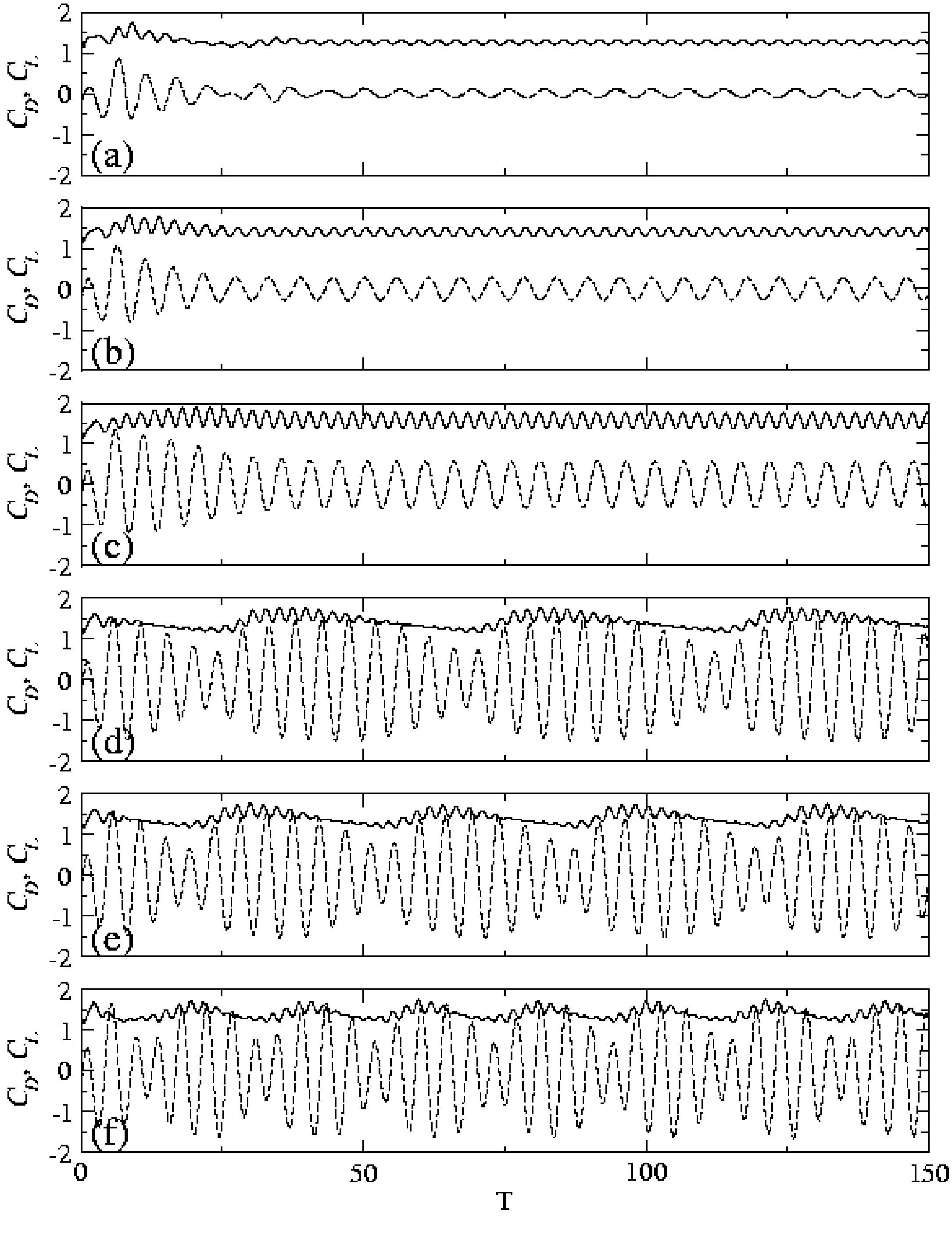


Figure 4.15: Evolution of the drag and lift coefficients in time for the case of a cylinder oscillating in a free-stream: —— C_D; --- C_L. (a) $f_e/f_0 = 0.8$; (b) $f_e/f_0 = 0.9$; (c) $f_e/f_0 = 1.0$; (d) $f_e/f_0 = 1.1$; (e) $f_e/f_0 = 1.12$; (f) $f_e/f_0 = 1.2$.

sign vorticity formed at the base of the cylinder as f_e/f_0 increases, which alters the vorticity in the top and bottom shear layers. The above results are in excellent agreement with reference data in the literature [22, 23].

Quantitative comparisons of the predicted drag and lift coefficients for all frequencies with the corresponding values from the simulations by Guilmineau & Queutey [23] are shown in Fig. 4.16a. The variation of the average drag coefficient, $\overline{C}_D$, with f_e/f_0 is in agreement with the reference data, although it is a little higher in our simulations (approximately 5%) for all frequency ratios. In the reference simulations, however, $\overline{C}_D$ was found to be very sensitive to grid resolution and they reported higher values when the grid was refined for the case of $f_e/f_0 = 1.10$. The $r.m.s$ values of the drag and lift coefficients are less sensitive to grid resolution and are in excellent agreement with the reference data. The same applies to the phase angle, ϕ, between the lift coefficient and the vertical displacement of the cylinder that is shown in Fig. 4.16b. Both ϕ and ${C_L}^{rms}$ exhibit a sudden change at $f_e/f_0 = 1.10$ when the vortex switching occurs as shown in [22, 23].

Next, we will examine how accurately the local forces on the cylinder's surface are predicted. The fact that $\overline{C}_D$, ${C_D}^{rms}$, and ${C_L}^{rms}$ above are in good agreement with the reference simulations, does not guaranty the same agreement on the local skin friction distribution, for example, which contributes a small percentage in the above quantities. Guilmineau & Queutey [23] reported the local distribution of surface pressure and vorticity, when the cylinder is located at the extreme upper position. A comparison with our results on all grids for a range of frequencies is shown in Fig. 4.17 and 4.18. The angle θ on the x axis, is measured clockwise from

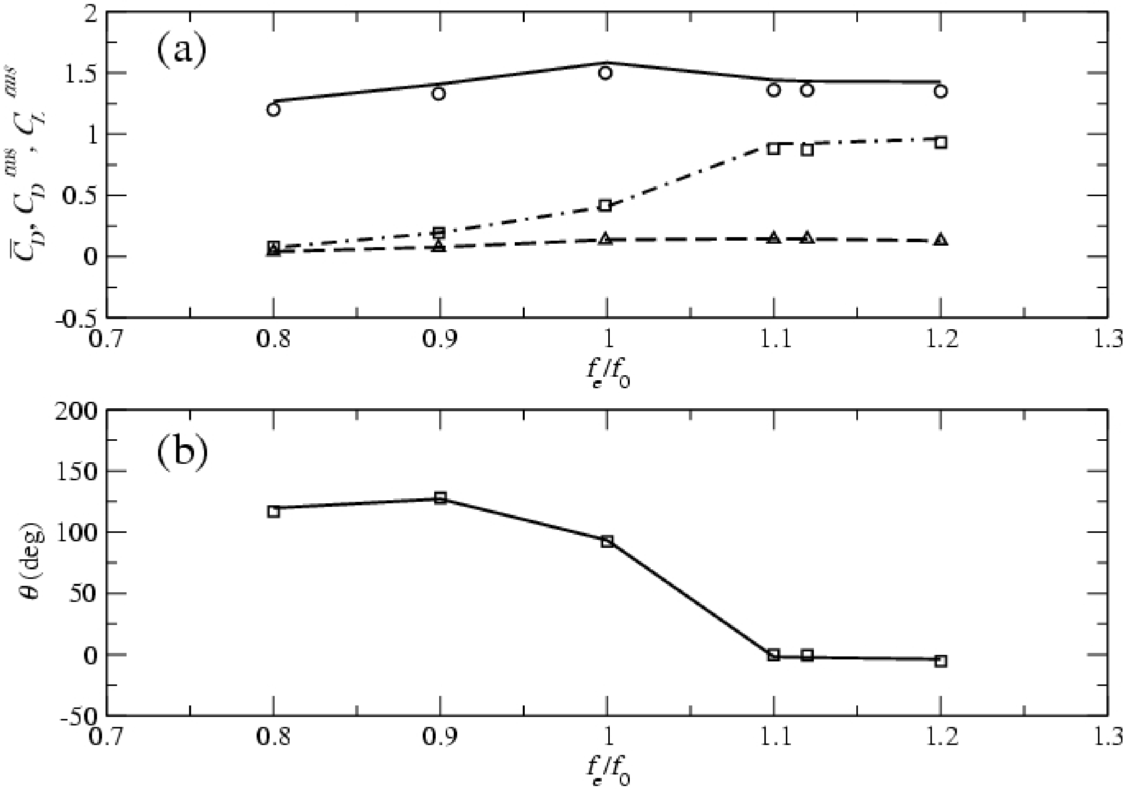


Figure 4.16: (a) Comparison of force coefficients for the case of the cylinder oscillating in a free-stream: —— & ○ $\overline{C}_D$; - - - & △ ${C_D}^{\text{rms}}$; —·— & □ ${C_L}^{\text{rms}}$; lines are present results on the 800 × 640 grid and symbols are the corresponding data from [23]. (b) Comparison of the phase angle between the lift coefficient and the vertical displacement: —— present results; □ [23]

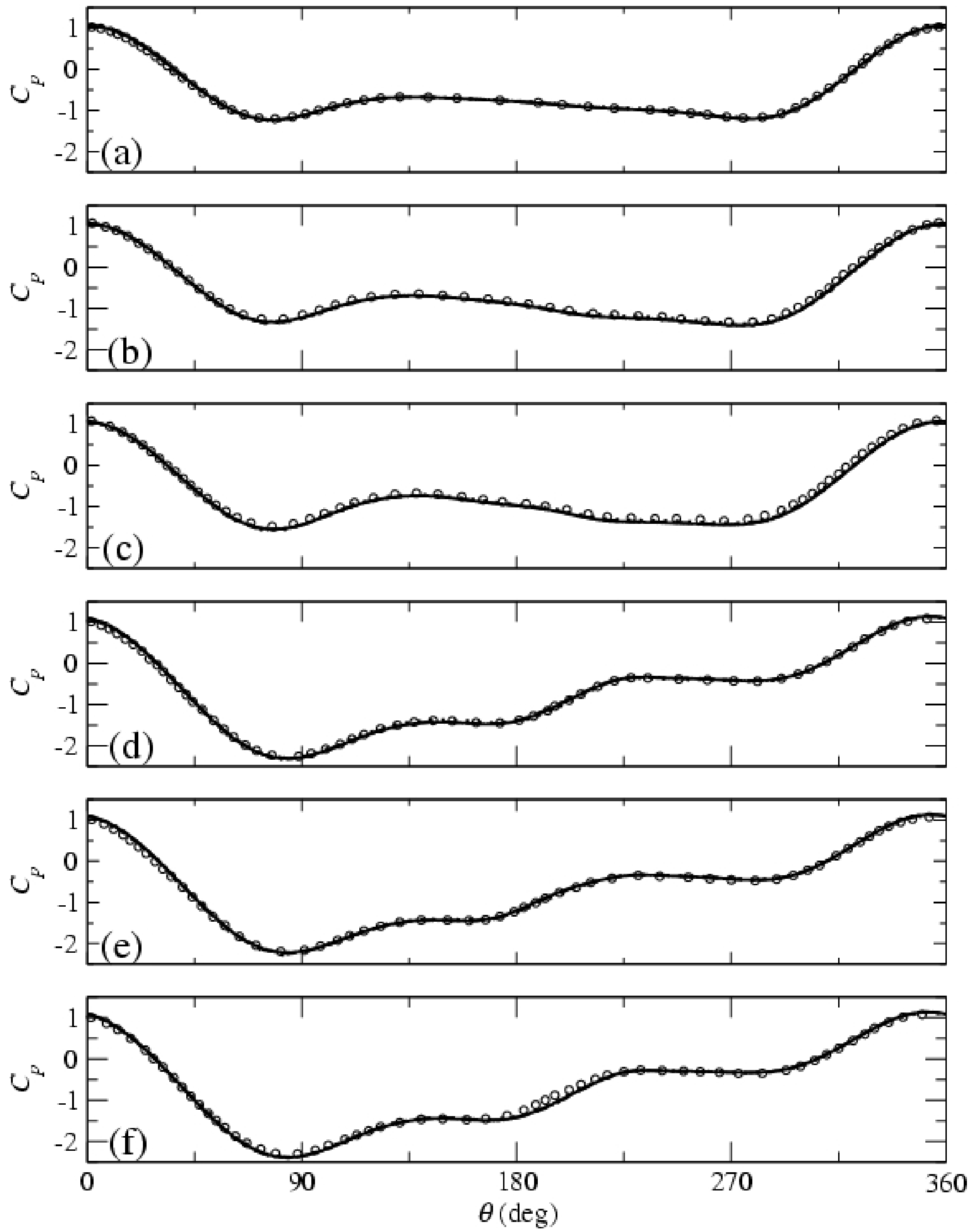


Figure 4.17: Distribution of the pressure coefficients on the cylinder's surface, when located at the extreme upper position. —— grid 800 × 640; - - - grid 400 × 320; ········ grid 200×160; ∘ body-fitted reference computation in [23]. (a) $f_e/f_0 = 0.80$; (b) $f_e/f_0 = 0.90$; (c) $f_e/f_0 = 1.00$; (d) $f_e/f_0 = 1.10$; (e) $f_e/f_0 = 1.12$; (f) $f_e/f_0 = 1.20$.

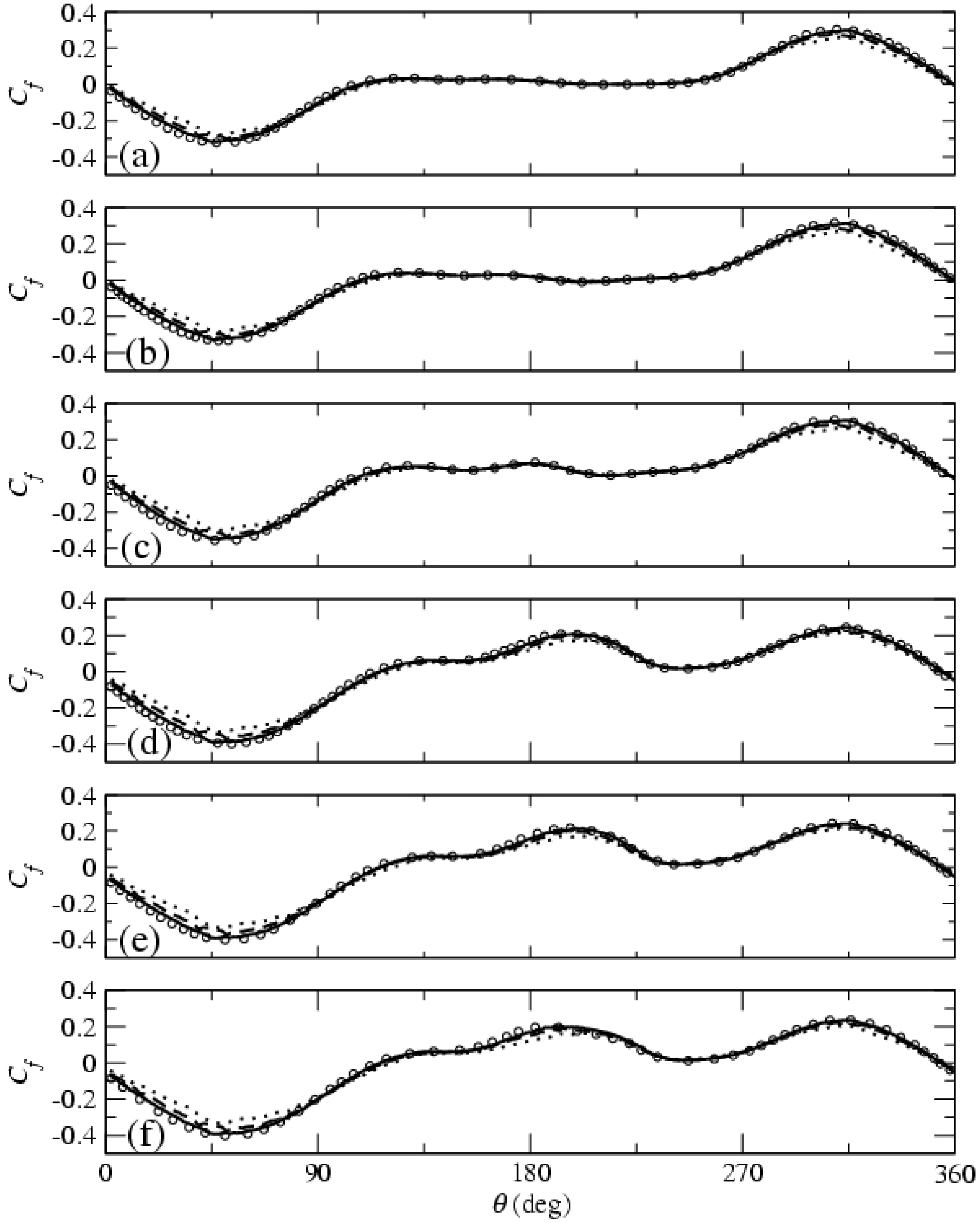


Figure 4.18: Distribution of the skin friction coefficients on the cylinder's surface, when located at the extreme upper position. —— grid 800×640; - - - grid 400×320; ········ grid 200×160; ○ body-fitted reference computation in [23]. (a) $f_e/f_0 = 0.80$; (b) $f_e/f_0 = 0.90$; (c) $f_e/f_0 = 1.00$; (d) $f_e/f_0 = 1.10$; (e) $f_e/f_0 = 1.12$; (f) $f_e/f_0 = 1.20$.

the stagnation point. It can be seen that our results follow the reference data very well especially on the finest grid. As expected the skin friction is more sensitive to grid resolution and is under-predicted near the peaks on the coarser grids. The pressure coefficient on the other hand is fairly insensitive and almost identical on all grids. We should also note that the body-fitted grid that is used in [23] is much finer near the cylinder's surface compared to our finest grid. The local grid spacing normal to the boundary is $0.001D$ in [23] and $0.005D$ for our 800×640 grid, which further demonstrates the accuracy of the overall approach.

4.3 Vortex-Induced Vibrations of an Elastic Cylinder

Vortex-induced vibration of a circular cylinder is a problem that has been extensively studied both experimentally and numerically (see [67] for a recent review), and will be used in the present study to evaluate the accuracy and efficiency of the proposed methodology. Two different configurations are considered: 1. Free oscillations in the cross-stream direction (one degree-of-freedom); 2. Free oscillations in both the streamwise and cross-stream directions (two degree-of-freedom). In both cases the selected parametric space was as close as possible to well documented laminar two-dimensional flow problems in the literature.

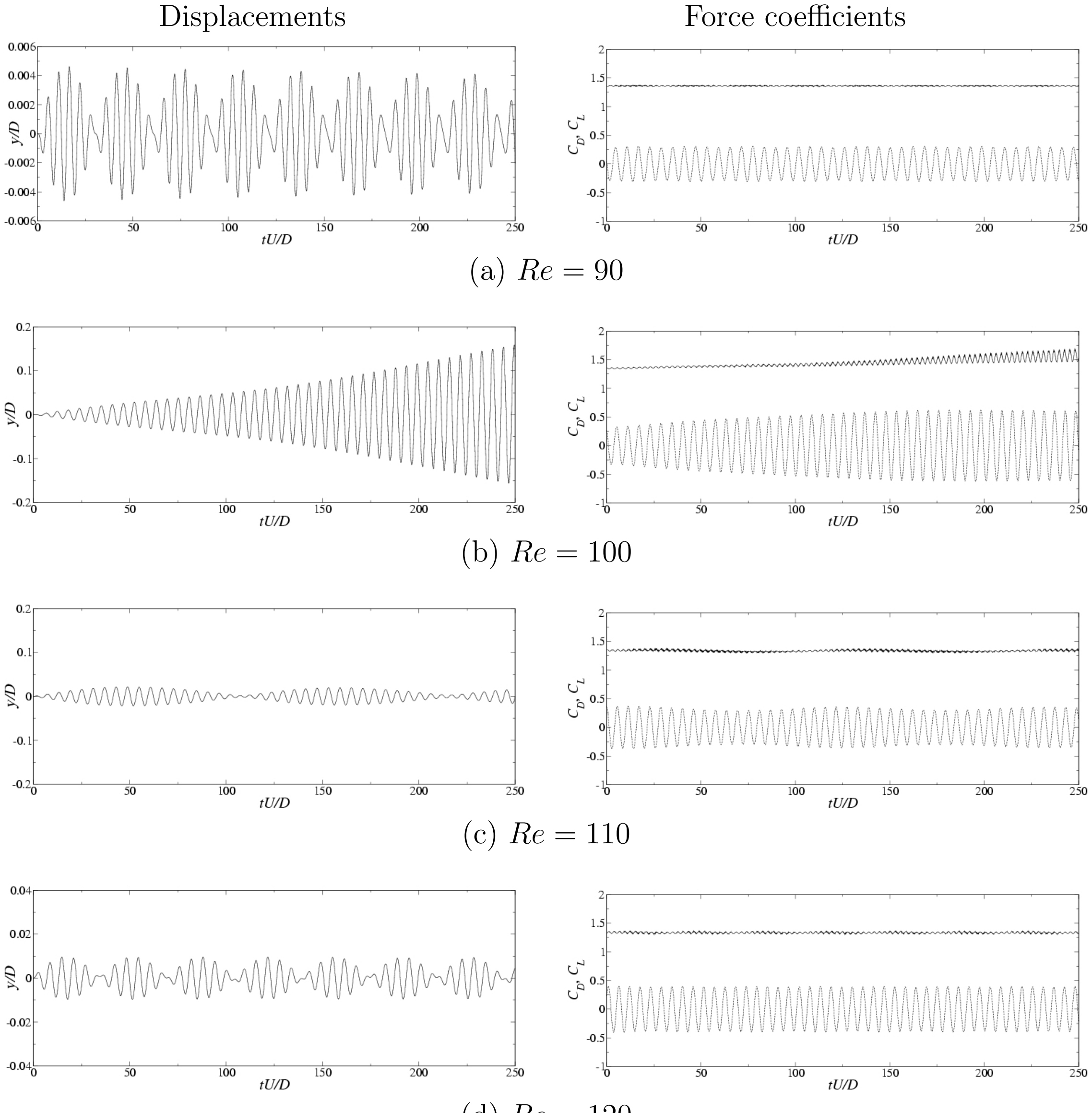
4.3.1 One Degree of Freedom

For a cylinder freely oscillating in the cross-stream direction, which is modeled as mass-damper-spring system, the non-dimensional form of equation (3.20) becomes:

$$\ddot{y} + 2\zeta\left(\frac{2\pi}{U_{red}}\right)\dot{y} + \left(\frac{2\pi}{U_{red}}\right)^2 y = \frac{1}{2n}C_L, \tag{4.1}$$

where $y = Y_o/D$ is the dimensionless vertical displacement, $\zeta = \frac{c}{2\sqrt{km}}$ is the damping ratio, $C_L = f_y/(\frac{1}{2}\rho DLU^2)$ is the lift force coefficient, $n = m/\rho D^2$ is the mass ratio with ρ the fluid density, and $U_{red} = U/f_nD$ is the reduced velocity with $f_n = \frac{1}{2\pi}\sqrt{k/m}$ the natural vibrating frequency of the structure. In the above m is system mass, c the damping coefficient, k the spring constant, Y_o the transverse displacement of the system centroid, and f_y the instantaneous lift force on cylinder. Note that the same reference scales as in the Navier-Stokes equations (the cylinder diameter, D, and the freesream velocity, U), were used to obtain the dimensionless form of the equations.

The parametric space was selected to match an experiment by Anagnostopoulos and Bearman [2], which has been used for validation in several other numerical simulations (see for example [53, 31]). The mass ratio was set to $n = 117.10$ and the damping ratio to $\zeta = 0.0012$. The computational domain is $40D \times 20D$, in the streamwise and cross-stream directions, respectively. The cylinder is located at a distance of $10D$ from the the inlet boundary, where a uniform velocity is specified, and $10D$ from the upper and lower boundaries, where radiative conditions are imposed. At the outflow boundary a convective boundary condition is used [41]. The number of grid points in all computations was 320×240 in the streamwise and transverse directions, respectively. The grid was stretched in both directions and the resulting resolution near the cylinder was approximately $0.02D \times 0.02D$. A



(a) $Re = 90$

(b) $Re = 100$

(c) $Re = 110$

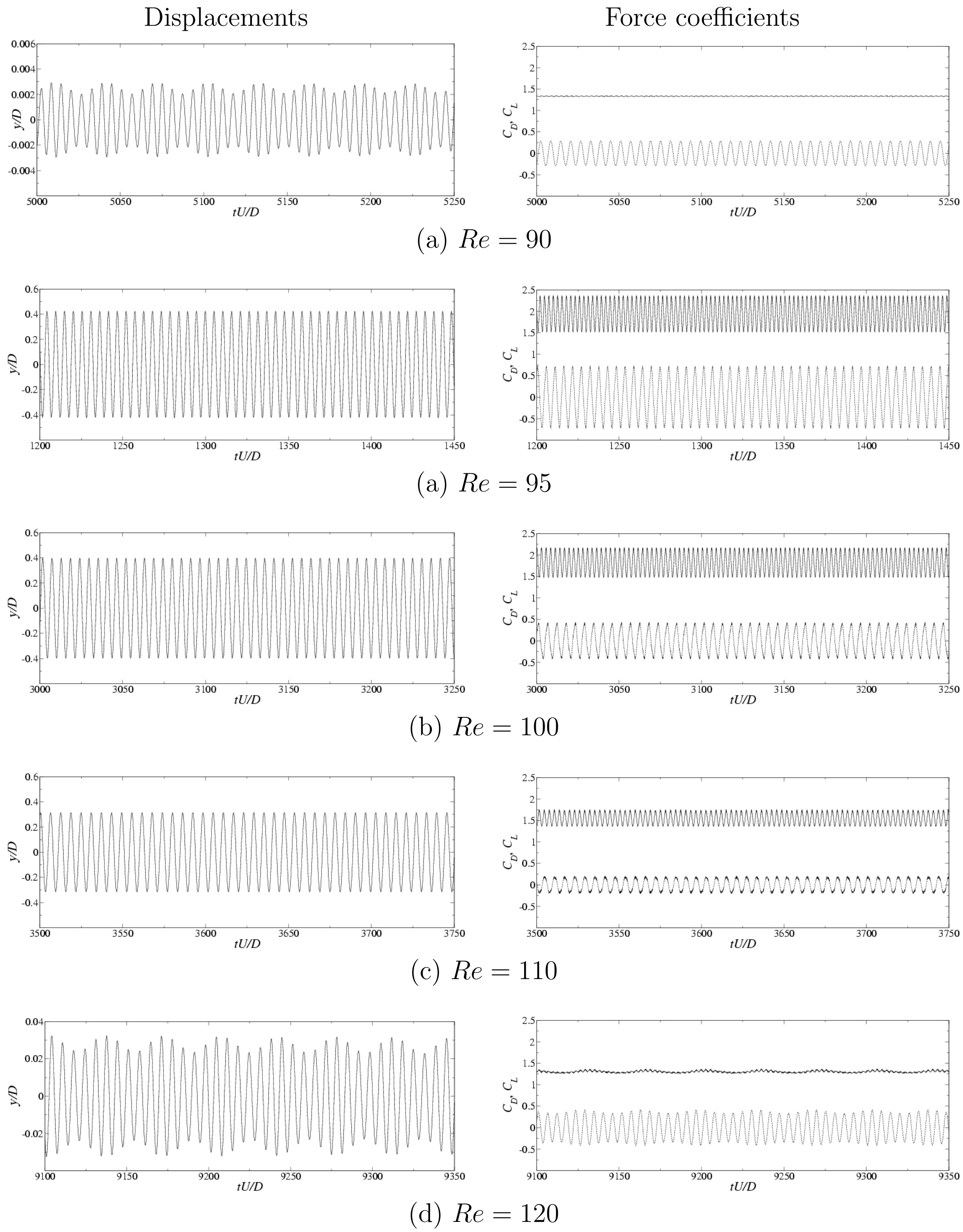
(d) $Re = 120$

Figure 4.19: Time history of the displacements for the freely vibrating cylinder in the cross-stream direction: start-up phase.

constant time step of $0.005D/U$ was used for all cases.

For all cases the flow over a stationary cylinder for the same Reynolds number was initially computed. Then, the cylinder was allowed to move freely in the transverse direction. Integration in time was performed until a periodic state of constant maximum amplitude was reached. The large mass ratio of this system makes the problem fairly stiff and very long integration times were necessary for most cases. It is interesting to investigate how the cylinder behaves after the release. Fig. 4.19 shows the start-up phase and Fig. 4.20 shows the stable phase of the cylinder displacement and corresponding drag and lift force coefficients. Outside the lock-in regime the vibration amplitude is modulated and has a very small magnitude, e.g., as shown in Fig. 4.19a for $Re = 90$. Correspondingly, the drag coefficient is oscillating in a very narrow range. Also, the time histories of cylinder displacement and the force coefficients do not change much even after reaching the stable phase as shown in Fig. 4.20a. In the lock-in regime, for most cases the amplitudes are also modulated (see Fig. 4.19c,d). However, for $Re = 100$, one can observe a monotonically growing oscillation amplitude, as shown in Fig. 4.19b. Nevertheless, all cases in the lock-in regime approach the stable periodic states. Similar phenomena had been observed by [31], but with a small shift of the range of Reynolds numbers.

The vorticity and pressure contours for one of the cases in the lock-in regime is shown in Fig. 4.21. The expected $2S$ mode pattern in Fig. 4.21b, c(two single vortexes per cycle of motion) can be seen in the wake [68]. Also, the smoothness of the vorticity contours and pressure countours near the cylinder is a farther indication that the method properly captures the complex dynamics of the flow near near the



(a) $Re = 90$

(a) $Re = 95$

(b) $Re = 100$

(c) $Re = 110$

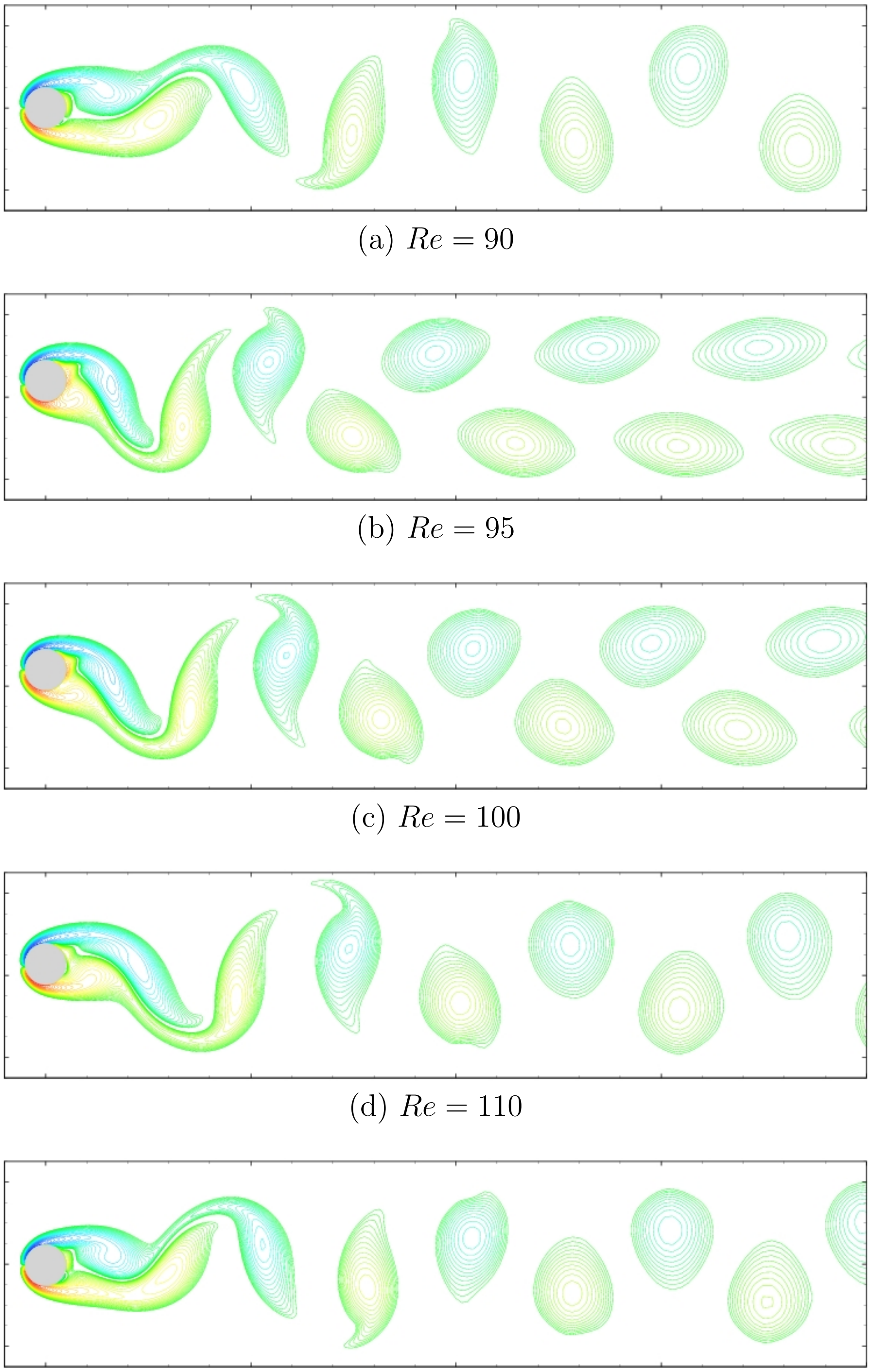
(d) $Re = 120$

Figure 4.20: Time history of the displacements for the freely vibrating cylinder in the cross-stream direction: stable periodic phase.

immersed interface.

The comparison of $\overline{C}_D$, C_D^{rms}, and C_L^{rms} between the fixed cylinder cases and the freely oscillating cylinder cases is shown in Fig. 4.23. As the Reynolds number increases, the rms values of the drag and lift coefficients increase slowly, and the mean values of the drag coefficient decrease slowly. However, for the freely oscillating cylinder cases, in the lock-in regime, a significant different plot is presented, while outside the lock-in regime, there is little difference between the fixed and the freely oscillating cases.

The maximum oscillation amplitudes and frequencies are shown in Fig. 4.24. The corresponding experimental results [2] and previous computations [53, 31] have been added for comparison. For all simulations the maximum vibration amplitude in the lock-in regime is lower that one in the experiment, where also the critical Reynolds number for lock-in is higher (approximately $Re = 103$, while in most computations it is below 95). Schulz & Kallinderis [53] speculated that this difference could be due to three-dimensional effects induced by the absence of end-plates in the experimental apparatus. Discrepancies can also be observed between the simulations. Our results are in good agreement with the computations reported in [31], where a boundary conforming methodology utilizing a moving frame of reference was used. This indicates the accuracy of the proposed methodology. An extensive grid refinement study, however, remains to be performed to clarify this issue.



(a) $Re = 90$

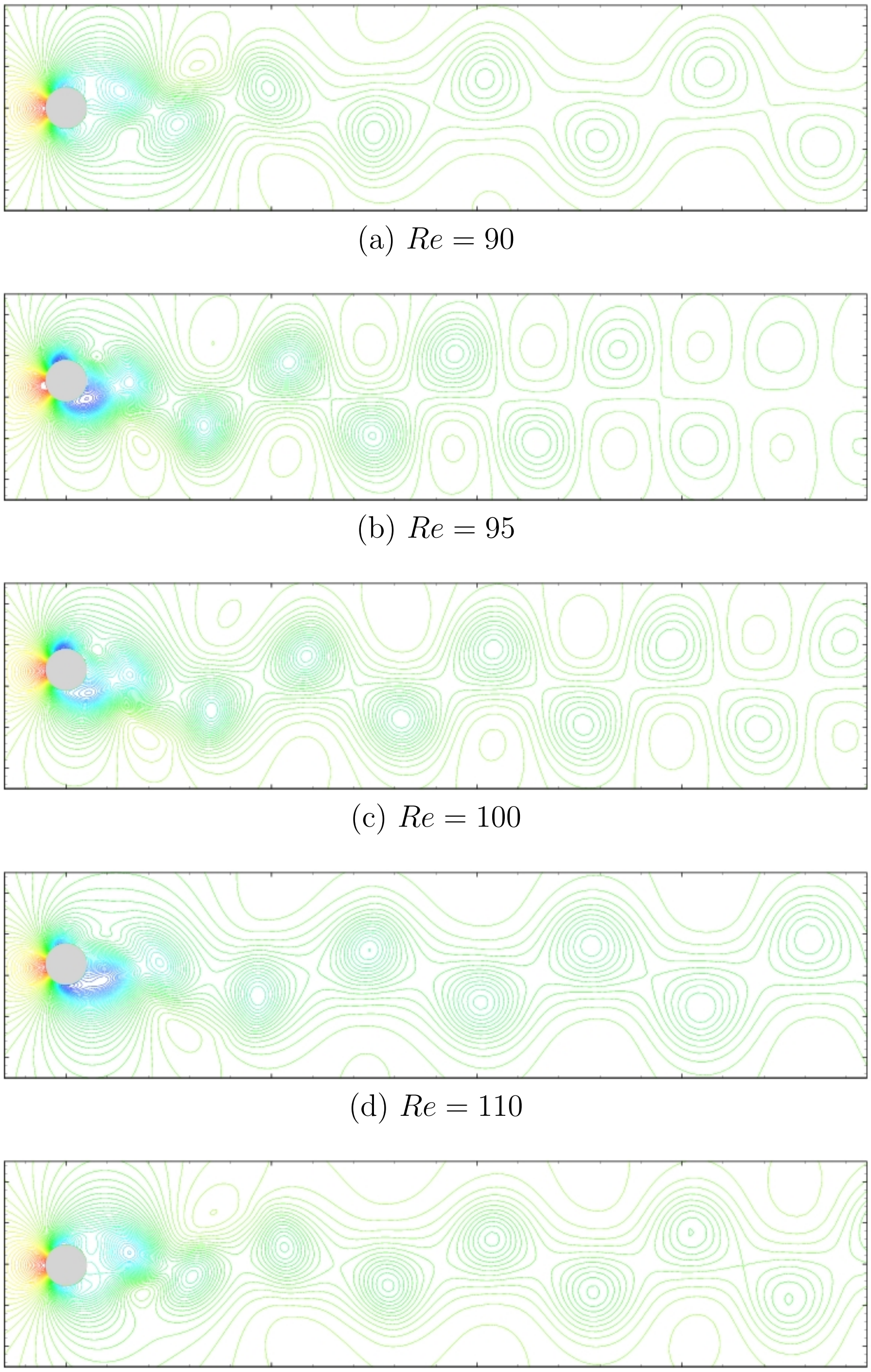
(b) $Re = 95$

(c) $Re = 100$

(d) $Re = 110$

(e) $Re = 120$

Figure 4.21: Freely vibrating cylinder in the cross-stream direction: instantaneous vorticity contours.



(a) $Re = 90$

(b) $Re = 95$

(c) $Re = 100$

(d) $Re = 110$

(e) $Re = 120$

Figure 4.22: Freely vibrating cylinder in the cross-stream direction: instantaneous pressure contours.

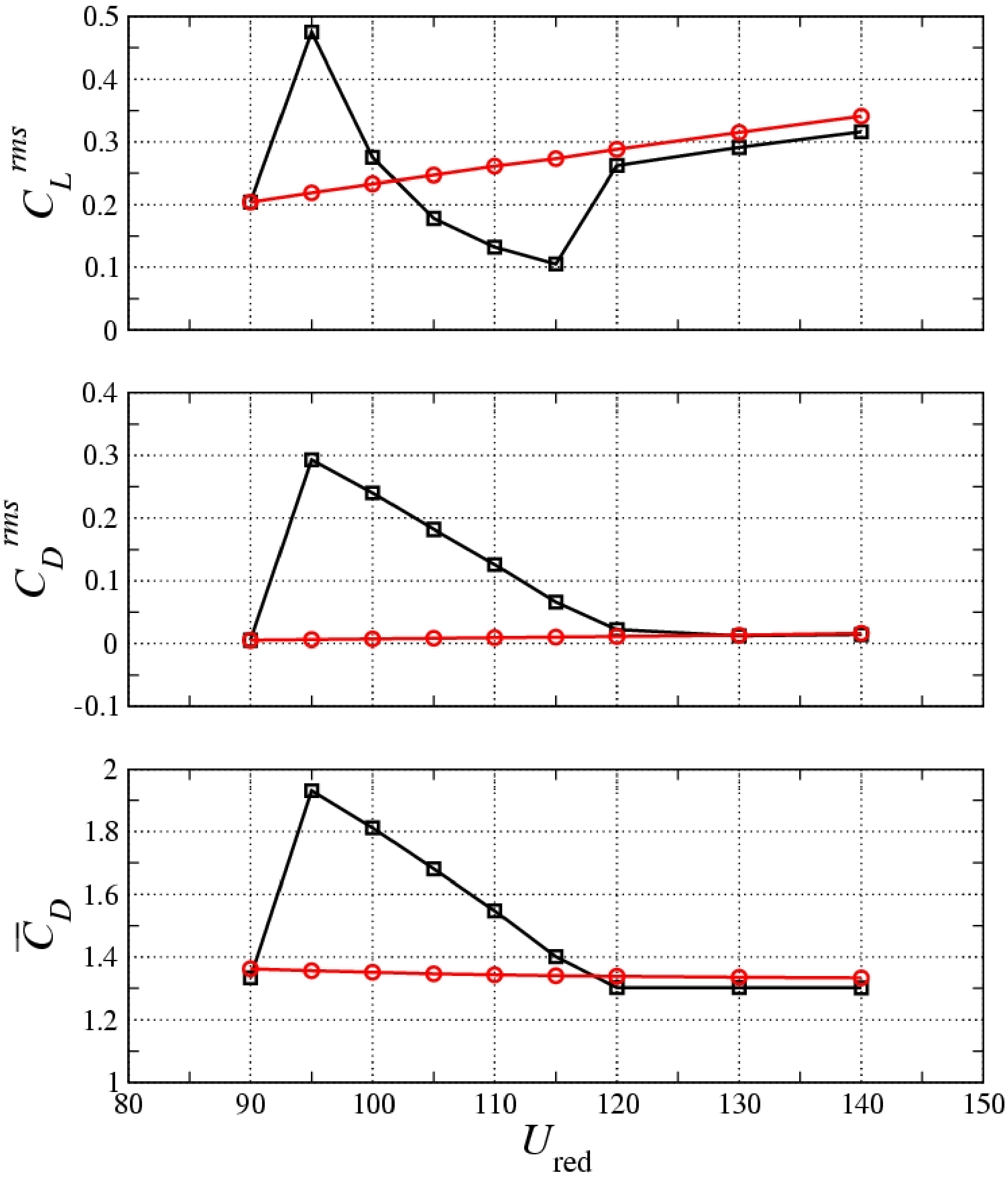


Figure 4.23: Comparison of the force coefficients for the cases of fixed cylinder and the freely oscillating cylinder. ○ fixed cylinder; □ freely oscillating cylinder

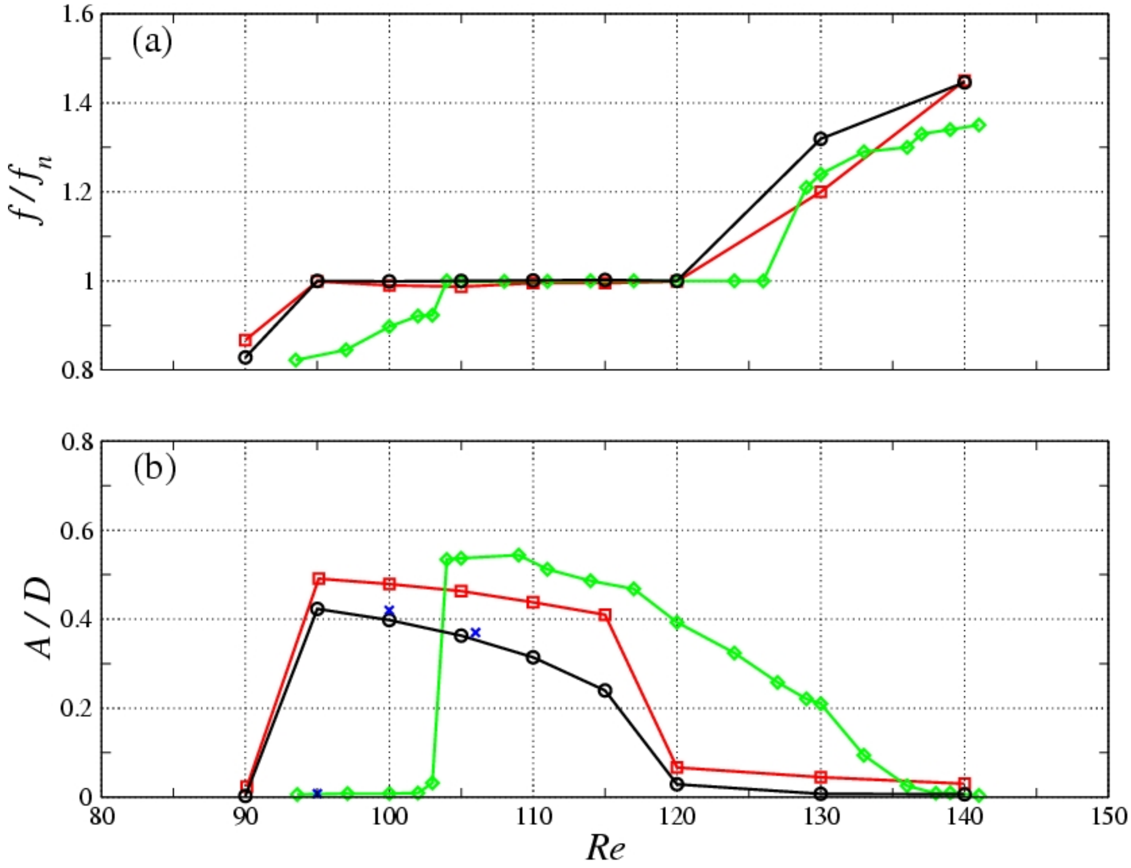


Figure 4.24: Comparison of the oscillation frequency and the maximum oscillation amplitude. ○ present computations; ◇ experiment in [2]; □ computation in [53]; × computation in [31]

4.3.2 Two Degrees of Freedom

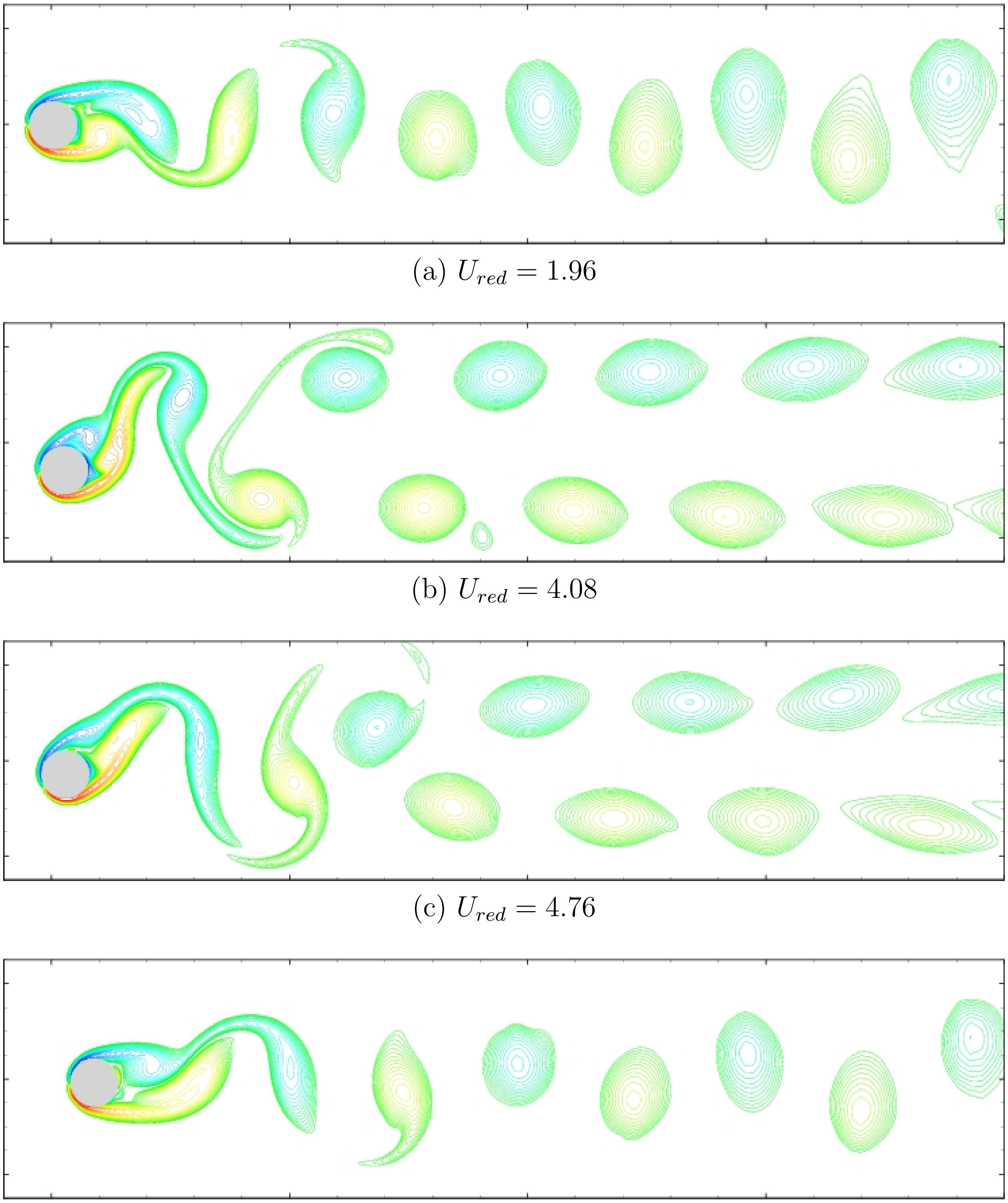
Although the one degree of freedom case has shed light on the physics of the vortex-induced vibration problems, there are questions to be answered regarding to what extent the freedom of oscillating in streamwise direction can change the response of the system. Recently, a paper by Jauvtis & Williamson [27] reported a substantial increase of the oscillation amplitude for systems with two degrees of freedom when the mass ratio is smaller than a critical value in their water channel experiments with Reynolds numbers ranged from 1,000 to 15,000. In the laminar flow regime, whether such a dramatical change exists or not is still in question since there are no experiment results on systems with two degrees of freedom reported in the literature for low Reynolds numbers. Zhou *et al.* [74] carried out numerical simulations using the discrete vortex method on vortex-induced vibrations of a circular cylinder with two degrees of freedom at Reynolds number 200. They compared these results to those from one degree of freedom cases, and found no substantial changes.

Here we shall not aim at reproducing the results in [27] as the moderate Reynolds number flows in their experiments require fully three dimensional numerical simulations. By contrast, we shall stay in the low Reynolds number, laminar flow regime, and select a low mass ratio $n = 2.04$ and a low damping ratio $\zeta = 0.00425$ for the system, which are in the range used in [27]. The Reynolds number is set to $Re = 200$. The reduced velocity U_{red}, however, was varied from 1 to 11. The computational domain is $36D \times 20D$, in the streamwise and cross-stream directions, respectively. The cylinder is located at a distance of $6D$ from the the inlet boundary, where a uniform velocity is specified, and $10D$ from the upper and lower boundaries, where radiative conditions are imposed. At the outflow boundary a convective boundary condition is used [41]. The number of grid points in all computations was 400×360 in the streamwise and transverse directions, respectively. The grid was stretched in both directions and the resulting resolution near the cylinder was approximately $0.02D \times 0.02D$, which is the same as the one degree of freedom case above. The time step was set to $\Delta t = 0.0025D/U$ (for $U_{red} = 4.08$, $\Delta t = 0.002D/U$).

As for the previous case, in all computations the corresponding stationary cylinder problem was first solved and then free vibrations in both directions were allowed. Due to the low mass ratio in this case, a stable periodic state was achieved much faster compared to the previous problem.

Fig. 4.25 and Fig. 4.26 show the instantaneous vorticity and pressure contours for several cases with increasing U_{red}. Outside the lock-in regime, e.g., $U_{red} = 1.96$ and $U_{red} = 10.71$, the contour patterns are very similar to those of the fixed cylinder cases. Whithin the lock-in regime, the vibrations of the cylinder modify the vortex shedding patterns substantially. However, for the current low Reynolds number cases, only the "2S" mode is captured as in the one degree of freedom cases.

The X, Y trajectories of several cases are shown in Fig. 4.27. All these figures illustrate highly periodical behaviors and symmetries in the transverse direction. They all exhibit the well-known "figure eight" type of motion, while the different shapes reflect the differences of phase angles between the X and Y displacements. In the lock-in regime, very high amplitudes are observed with a maximum value of 0.597



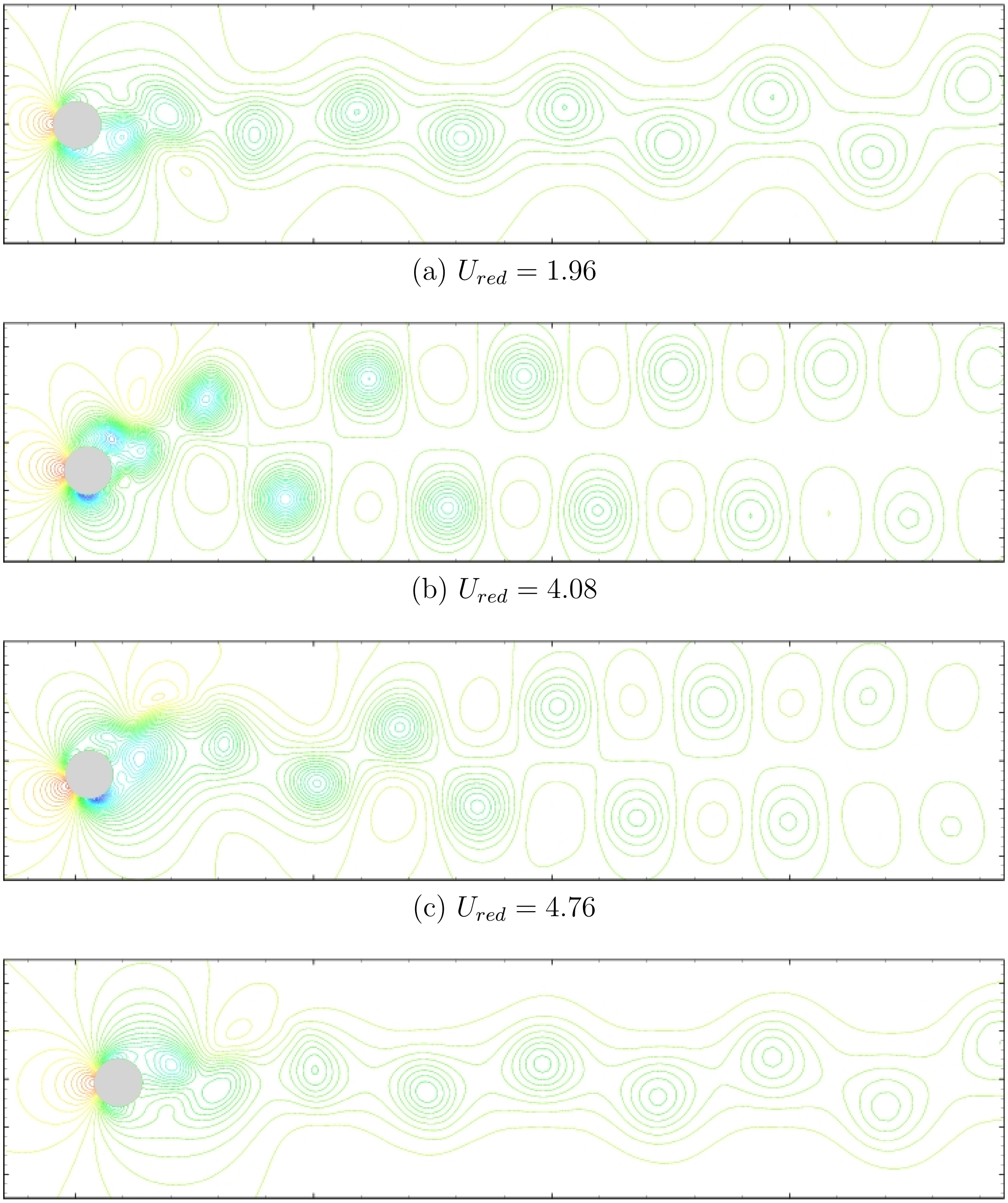
(a) $U_{red} = 1.96$

(b) $U_{red} = 4.08$

(c) $U_{red} = 4.76$

(d) $U_{red} = 10.71$

Figure 4.25: Vortex-induced vibration of an elastic cylinder with two degrees of freedom: instantaneous vorticity contours.



(a) $U_{red} = 1.96$

(b) $U_{red} = 4.08$

(c) $U_{red} = 4.76$

(d) $U_{red} = 10.71$

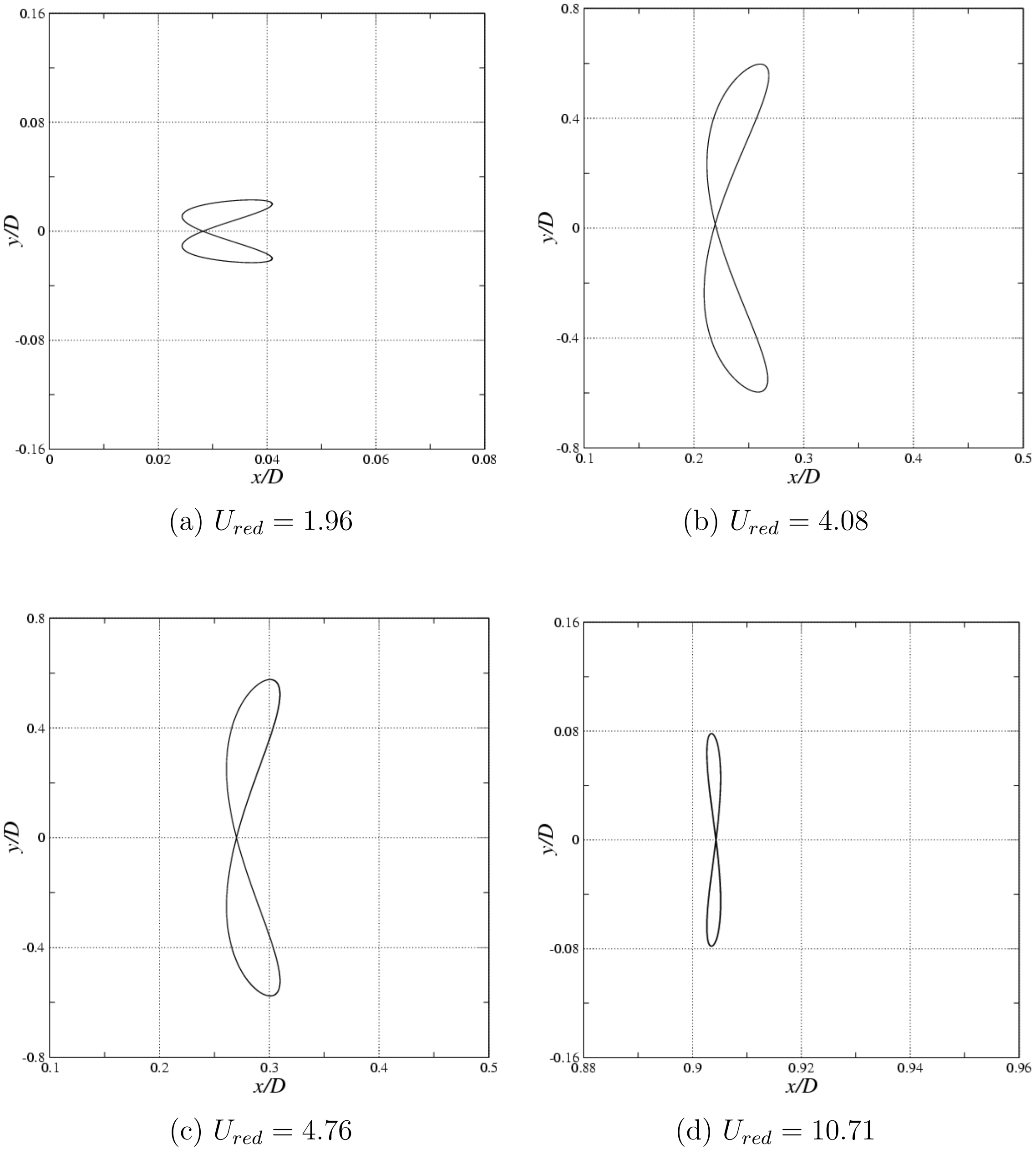
Figure 4.26: Vortex-induced vibration of an elastic cylinder with two degrees of freedom: instantaneous pressure contours.

at $U_{red} = 4.08$, which agrees with the saturation maximum amplitude for laminar flows ($A \sim 0.6$) reported in [27] very well. They reported a crescent trajectory for the "super-upper" branch for their moderate Reynolds number experiments, which is not presented in our low Reynolds number simulations.

Fig. 4.28 shows the time history of the displacements and lift coefficients for several cases with increasing U_{red}, and Fig. 4.29 shows the corresponding phase plots of displacement and lift coefficient. Outside the lock-in regime, both the force and the displacement are very close to sinusoidal; while in the lock-in regime, the shape of displacement doesn't vary much from sinusoidal, but the force is quite different from it, which illustrates a superposed oscillating frequency along with the fundamental oscillating frequency.

The phase plots of drag and lift coefficients are presented in Fig. 4.30. Outside the lock-in regime, the phase plots give the similar "figure eight" as phase plots of the streamwise and transverse displacement. However, the phase plots for the cases in the lock-in regime are much more complex and resemble the corresponding plots of the "super-upper" branch discussed in [27] for much higher Reynolds numbers. The intricate shapes indicate the development of oscillating frequencies different from the fundamental oscillating frequencies in both the streamwise and transverse directions.

Fig. 4.31 shows the oscillation frequency, transverse oscillation amplitude, and the streamwise oscillation amplitude as functions of the reduced velocity. A lock-in regime ranging from $U_{red} \approx 4 \sim 6$ is evident as the transverse and streamwise oscillation amplitudes are prominent in this range. Also, the oscillation frequencies



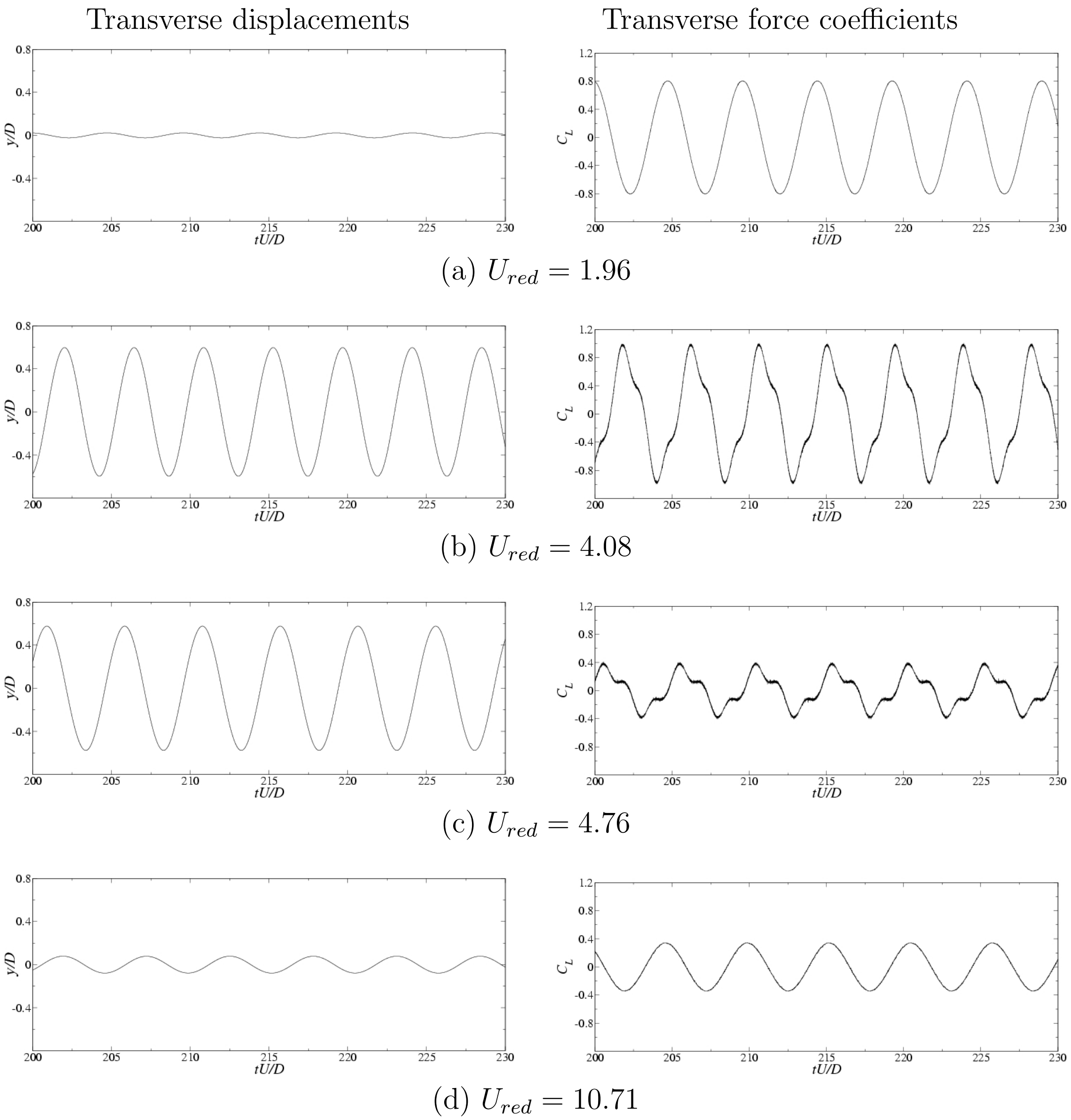
(a) $U_{red} = 1.96$

(b) $U_{red} = 4.08$

(c) $U_{red} = 4.76$

(d) $U_{red} = 10.71$

Figure 4.27: X-Y phase plots. X and Y are in different scales.



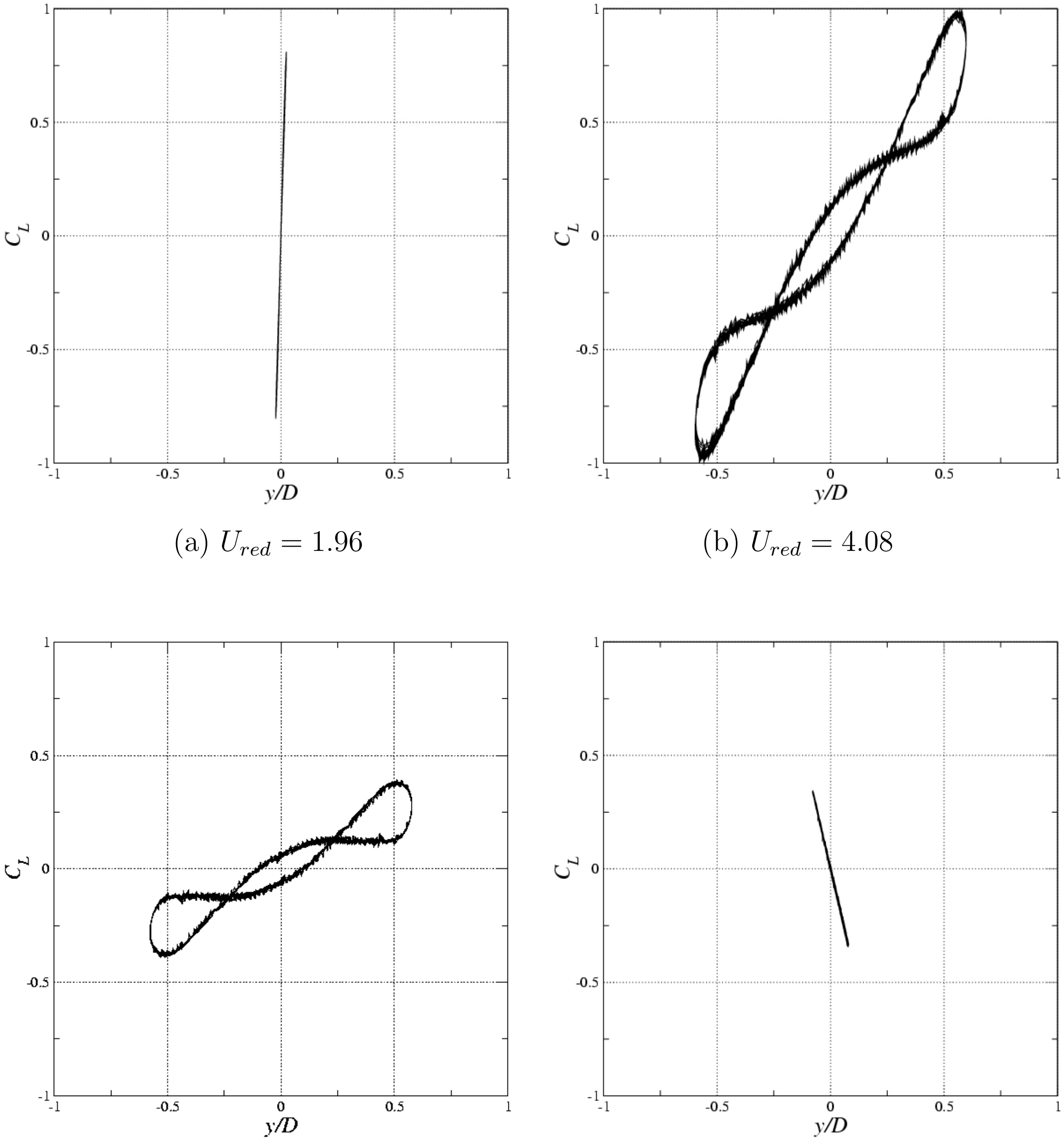
(a) $U_{red} = 1.96$

(b) $U_{red} = 4.08$

(c) $U_{red} = 4.76$

(d) $U_{red} = 10.71$

Figure 4.28: Time history of the displacements and lift coefficients.



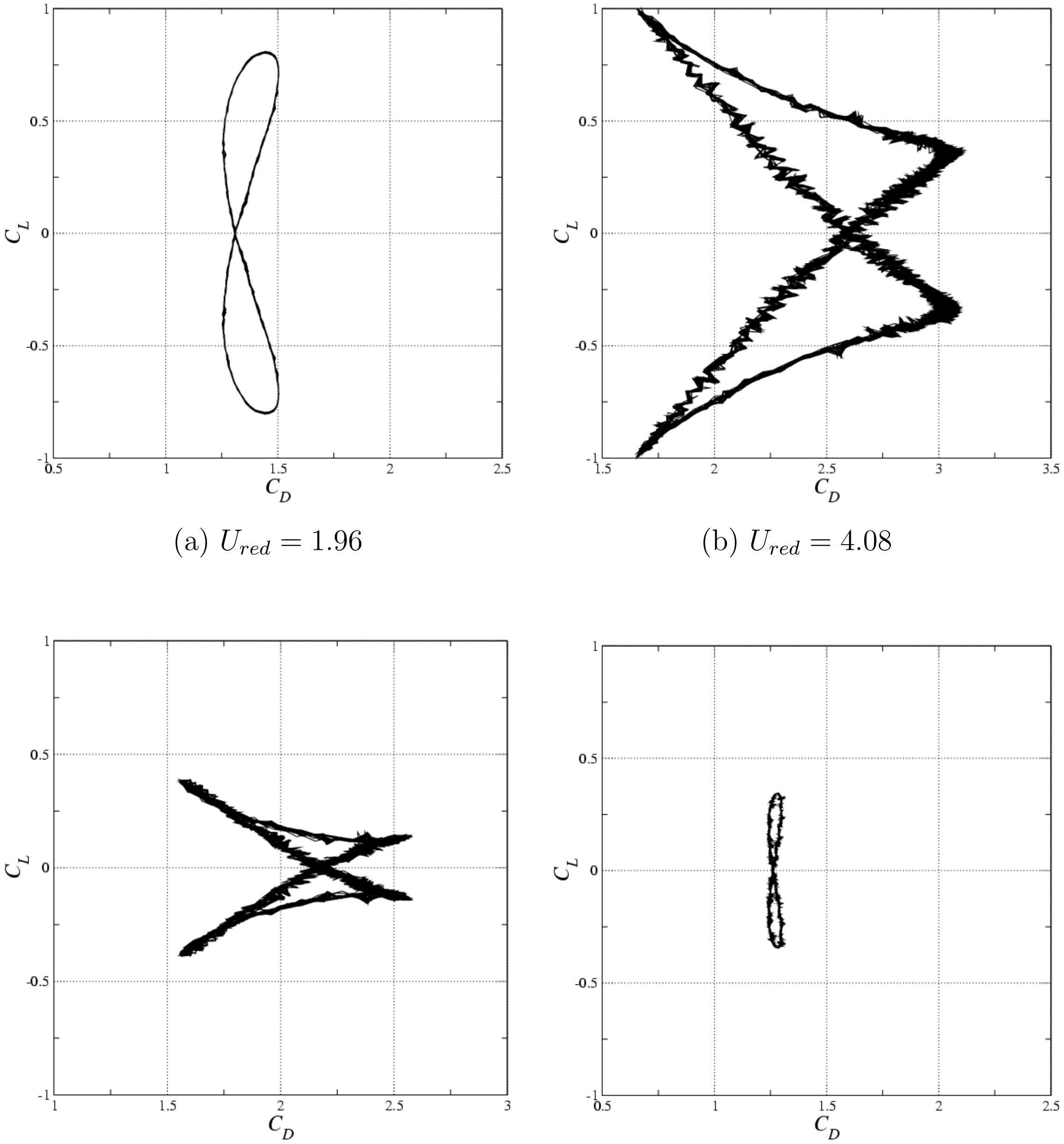
(a) $U_{red} = 1.96$

(b) $U_{red} = 4.08$

(c) $U_{red} = 4.76$

(d) $U_{red} = 10.71$

Figure 4.29: Phase plots of transverse displacement, lift coefficient.



(a) $U_{red} = 1.96$

(b) $U_{red} = 4.08$

(c) $U_{red} = 4.76$

(d) $U_{red} = 10.71$

Figure 4.30: Phase plots of drag, lift coefficients.

are about the same values as the natural oscillation frequencies of the structure. We also put the results in terms of reduced velocity from the one degree of freedom case in the previous section. There is an apparent overlapping lock-in regime. However, the two degrees of freedom case has a wider lock-in range, which is probably a result of the much lower mass ratio.

It is also interesting to compare $\overline{C}_D$, C_D^{rms}, and C_L^{rms} between the one degree-of-freedom and the two degree-of-freedom cases, as shown in Fig. 4.32. Again, there is significant similarity between these two cases. Just like the oscillation frequency and amplitude data, the two degree-of-freedom case has wider lock-in regime than that of the one degree-of-freedom case.

4.4 Summary

In this chapter, the formal second-order accuracy of the embedded boundary method presented in the previous chapter has been demonstrated via a specially-designed case: a periodic array of moving cylinders in a two-dimensional planar channel. This case is characterized by rigid bodies moving through the computational domain, which presents an extra difficulty to body-fitted methods.

After the establishment of the formal accuracy, the study has been concentrated on the vortex-induced vibrations of a circular cylinder. First, the forced vibrations are studied, i.e., the flow induced by the harmonic in-line oscillation of a circular cylinder in a quiescent fluid, and the flow from a transversely oscillating cylinder in a free-stream. Both cases are well documented in the literature,

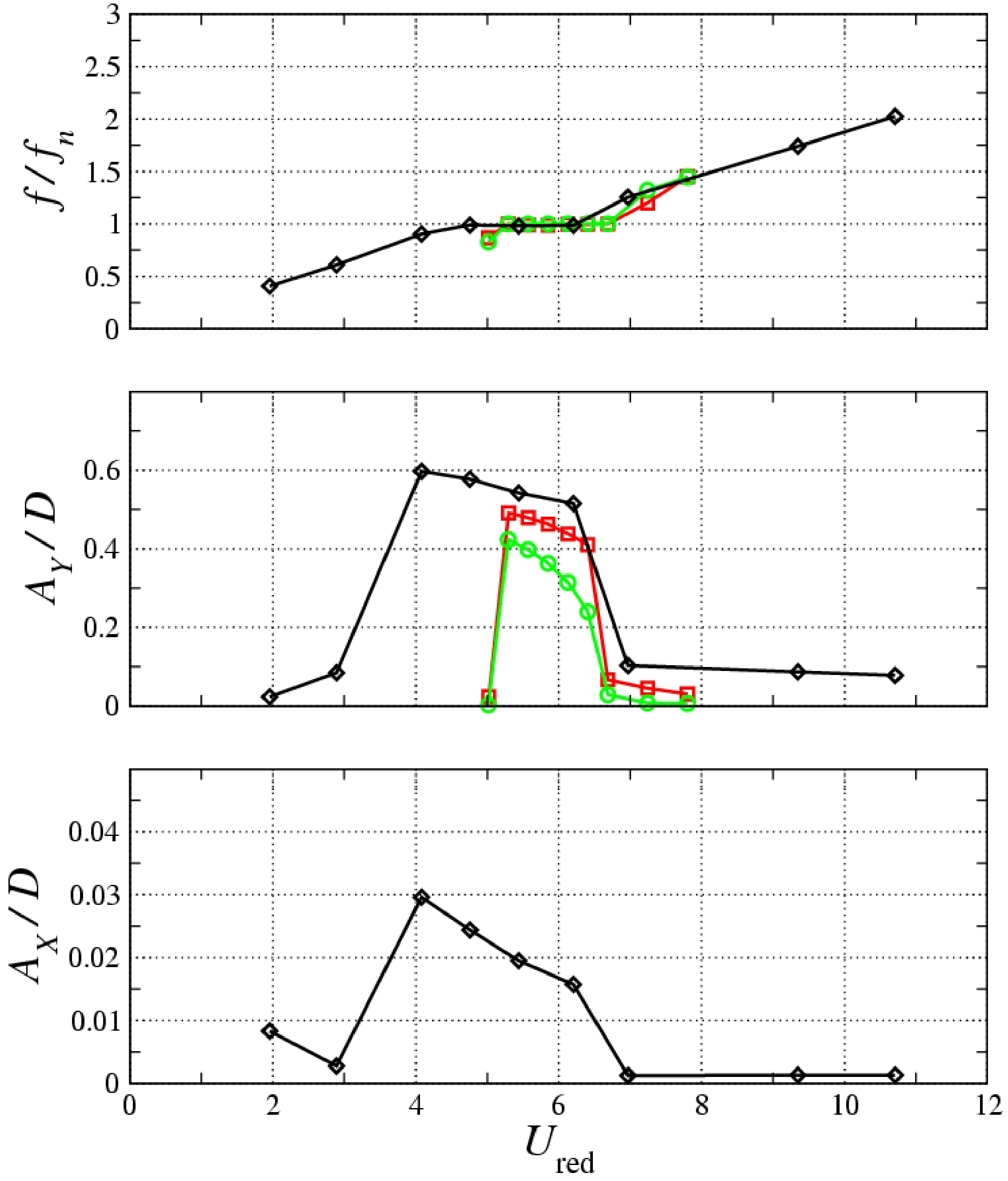


Figure 4.31: The oscillation frequency,transverse oscillation amplitude and streamwise oscillation amplitude as functions of reduced velocity. ◇ two degrees of freedom cases; ○ one degree of freedom cases from Sec. 4.3.1; □ one degree of freedom cases from [53]

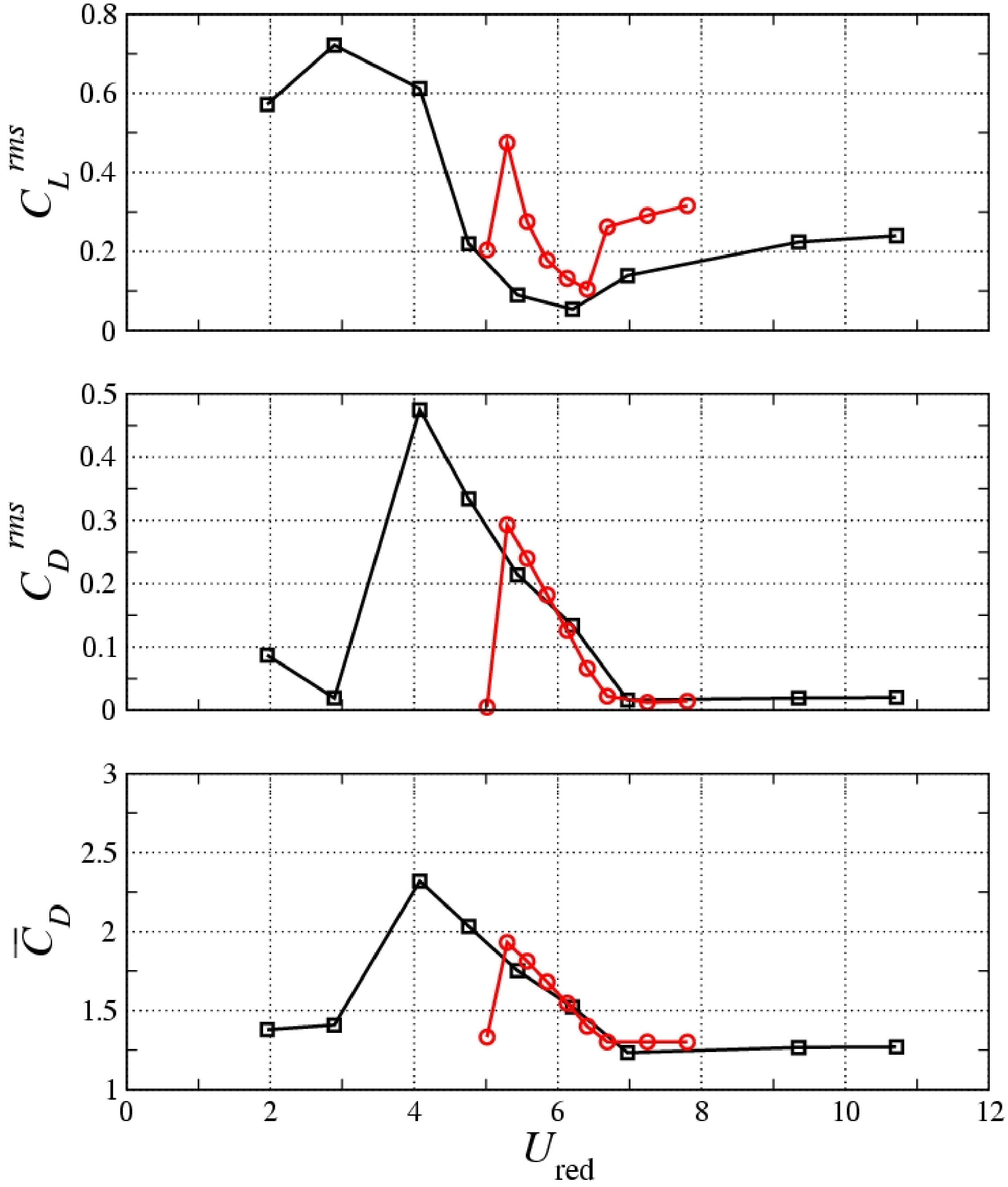


Figure 4.32: Comparison of force coefficients of the freely oscillating cylinders with one degree of freedom and two degrees of freedom. □ two degrees of freedom cases; ○ one degree of freedom cases from Sec. 4.3.1

and experimental and numerical results are available for comparison. The proposed method was found to reproduce all features of the flow, including the detailed force distribution on the body with the same degree of accuracy as the reference boundary-conforming computations.

Then the proposed strong coupling scheme for fluid-structure interaction problems are verified through the studies of vortex-induced vibrations of an elastic circular cylinder with one degree of freedom and two degrees of freedom. For the one degree-of-freedom case, the lock-in regime is accurately captured. The computational results have been compared with experimental data and other simulations using body-fitted grids. The current results are in good agreement with other simulations and reasonable agreement with the experiment. For the two degree-of-freedom case at Reynolds number 200, the results are similar to the one degree-of-freedom case but with a wider lock-in regime in reduced velocity space and higher oscillation amplitude in the lock-in regime.

Chapter 5

Three-Dimensional Applications

5.1 Flow Past a Sphere

The flow past the immersed bluff-body is of interest in many engineering applications. Here, the three-dimensional case of the flow past a sphere for a Reynolds numbers ranging from 50 to 1000 will be studied using the method developed in this work. The flow is steady and axisymmetric for Reynolds numbers up to 200. While at $Re = 250$ the flow is still steady, the axisymmetry breaks down. At $Re = 300$ the flow is no longer steady and is dominated by periodic vortex shedding. For $Re = 1000$, the flow becomes chaotic and transitions to turbulence.

5.1.1 Axisymmetric Flow

Cylindrical coordinates are used for a more efficient distribution of the grid points for the simulations of the flow past a sphere. For the cases of $Re = 50, 100, 150,$ and 200, ($Re = \rho U D/\mu$, where U is the freestream velocity, D the diameter of the sphere), the size of the computational box in the streamwise direction is $30D$ with the sphere located in the middle, and $15D$ in the radial direction. To investigate the influence of grid resolution on the results three different grids have been considered for these low Reynolds number cases. Grid 1 involves $40 \times 40 \times 100$ computational points in the radial, azimuthal, and streamwise directions, respectively. In

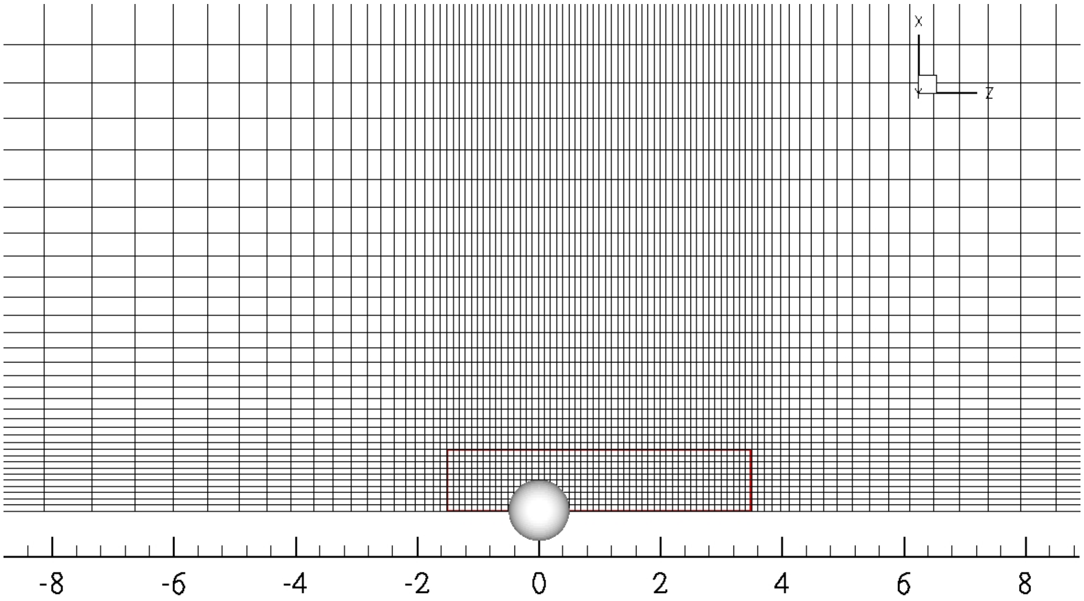


Figure 5.1: Part of grid $40 \times 40 \times 100$ for axisymmetric flow past the sphere. Uniform grid in the red box.

the other two cases the grid is refined in the streamwise and radial directions (Grid 2: $80 \times 40 \times 200$ and Grid 3: $160 \times 40 \times 400$, respectively). A part of the computational grid 1 for the flow around the sphere is shown in Fig. 5.1, in which a uniform grid distribution is adopted around the sphere. All three grids are stretched in the streamwise and radial directions to cluster points near the surface of the sphere. The resulting average grid spacing near the sphere for the three different resolutions is approximately $0.1D$, $0.05D$ and $0.025D$, respectively. In all cases a uniform velocity field is specified at the inflow plane. A convective boundary condition is used at the outflow boundary [41], and radiative boundary conditions are applied at the freestream boundary.

Table 5.1 shows the present results in comparison with the experimental results

Table 5.1: Prediction of C_D for axisymmetric flow past the sphere

Re	50	100	150	200
Grid 1: 40 × 40 × 100	1.708	1.230	1.045	0.944
Grid 2: 80 × 40 × 200	1.610	1.118	0.920	0.807
Grid 3: 160 × 40 × 400	1.586	1.095	0.894	0.776
Experiment [10]	1.574	1.087	0.889	0.776
Reference Simulations [28]	1.575	1.100	0.900	0.775

in [10], and the well-resolved simulations by Johnson and Patel [28] where body-fitted grids are used. For all Reynolds numbers and grid resolutions the main features of the flow are properly captured. The drag coefficient on the coarsest grid, is a little higher (approximately 8%) in comparison with the reference experimental and numerical data. As the grid is refined, however, the agreement is very good. The error in drag coefficient, which is computed using the experimental results in [10] as the reference values, as a function of grid spacing near the sphere is shown in Fig. 5.2. It can be seen that the error reduces with a second order slope, which is consistent with the order of accuracy of the method. Streamlines for the these Reynolds numbers are shown in Fig. 5.3. The flow direction is from left to right. A steady separation bubble is formed behind the sphere for all four Reynolds numbers. The flow is axisymmetric and has the same topological structure for all four cases. The only differences are the separation angle, vortex center, and the length of the recirculation bubble.

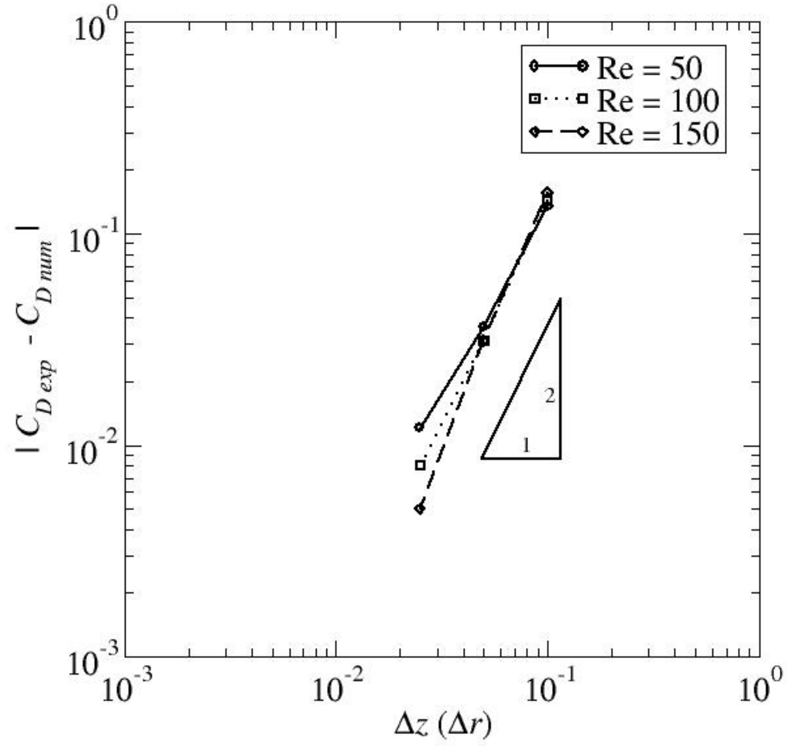


Figure 5.2: Error in the drag coefficient for the axisymmetric flow past the sphere as a function of grid resolution.

The only non-zero vorticity component is in the azimuthal direction and its contours are shown in Fig. 5.4. As the Reynolds number increases, the boundary layer becomes thinner and the vorticity extends to further downstream. Also, it is evident that the vorticity changes the sign across the separation point at the sphere surface.

5.1.2 Planar-Symmetric Flow

For $Re = 250$ and 300 cases the computational domain is expanded in streamwise direction, becoming $-10D \sim 40D$ with the sphere located at the origin, to capture the vortical structure in the wake. And the grid is refined to contain $112 \times 64 \times 420$ nodes in radial, azimuthal, and streamwise directions, respectively. Compared to the grid used for lower Reynolds number calculations, this grid is stretched more to cluster enough points near the sphere surface. With this resolution approximately 10 grid points are located in the boundary layers near the stagnation point.

At $Re = 250$ and 300, the flow no longer retains its axial symmetry. The flow past a sphere transits from aixsymmetric flow to three dimensional flow at about $Re = 211$ through a linear instability [60]. However, a plane of symmetry does still exist. In the current study, this plane of symmetry is almost (one grid cell difference) perpendicular to the boundary plane of the azimuthal direction, which is the $y-z$ plane in this work. Due to the use of ghost cells in the simulations, the plane of symmetry doesn't coincide with the $y-z$ plane exactly. Note that the results in

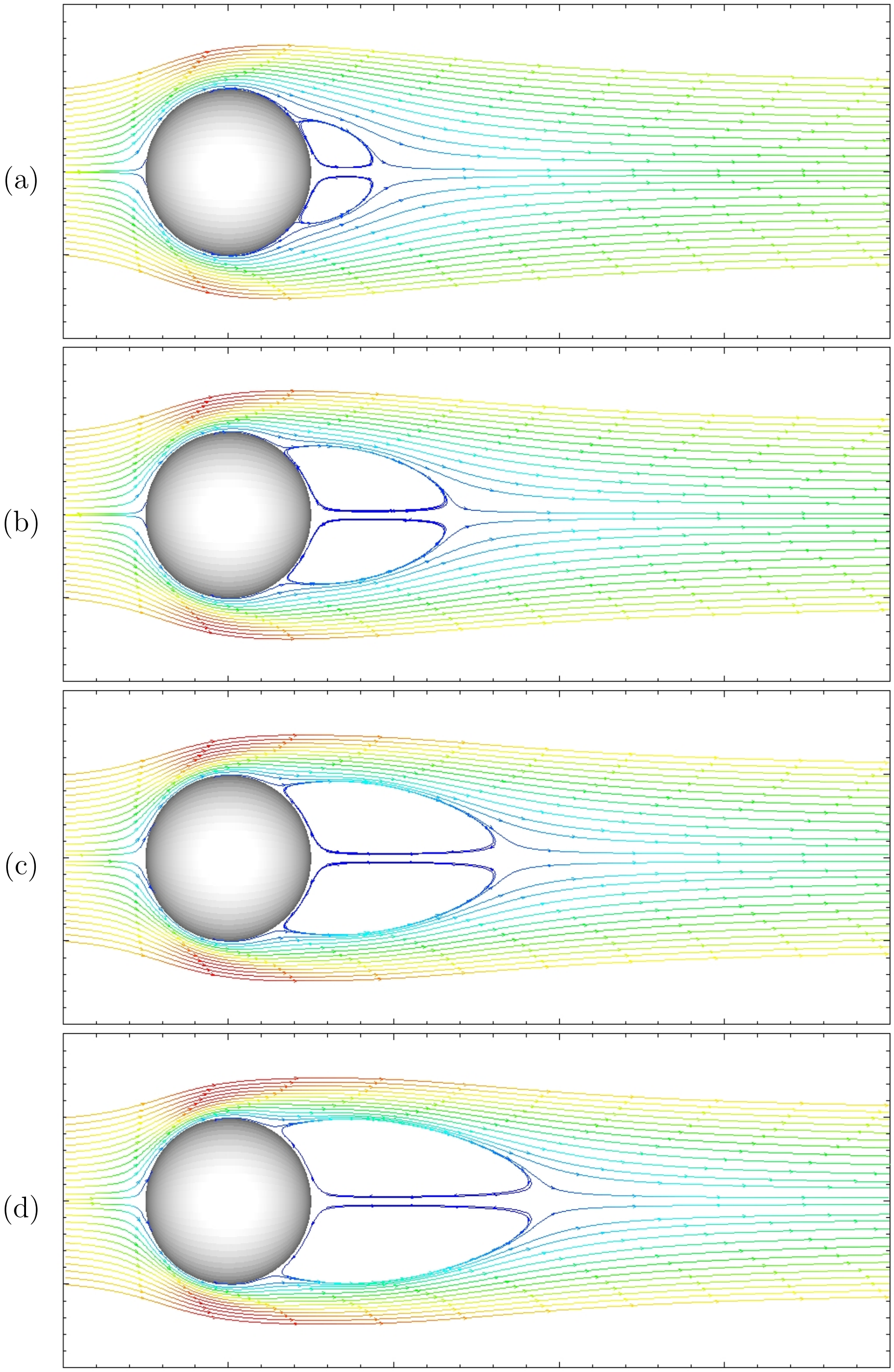


Figure 5.3: Axisymmetric streamlines past the sphere (colored by the streamwise velocity): (a) $Re = 50$; (b) $Re = 100$; (c) $Re = 150$; (d) $Re = 200$.

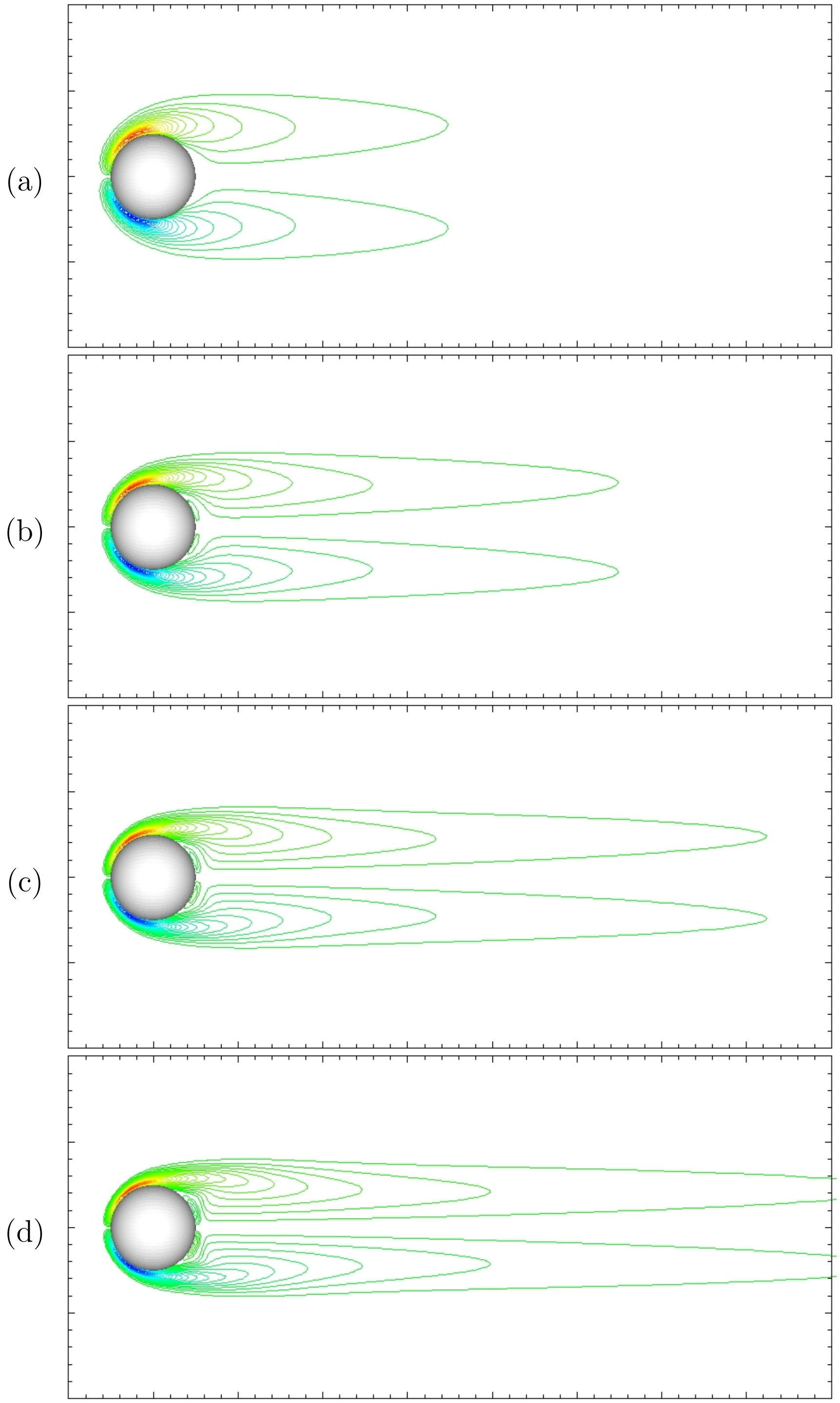


Figure 5.4: Vorticity contours for axisymmetric flow past the sphere: (a) $Re = 50$; (b) $Re = 100$; (c) $Re = 150$; (d) $Re = 200$.

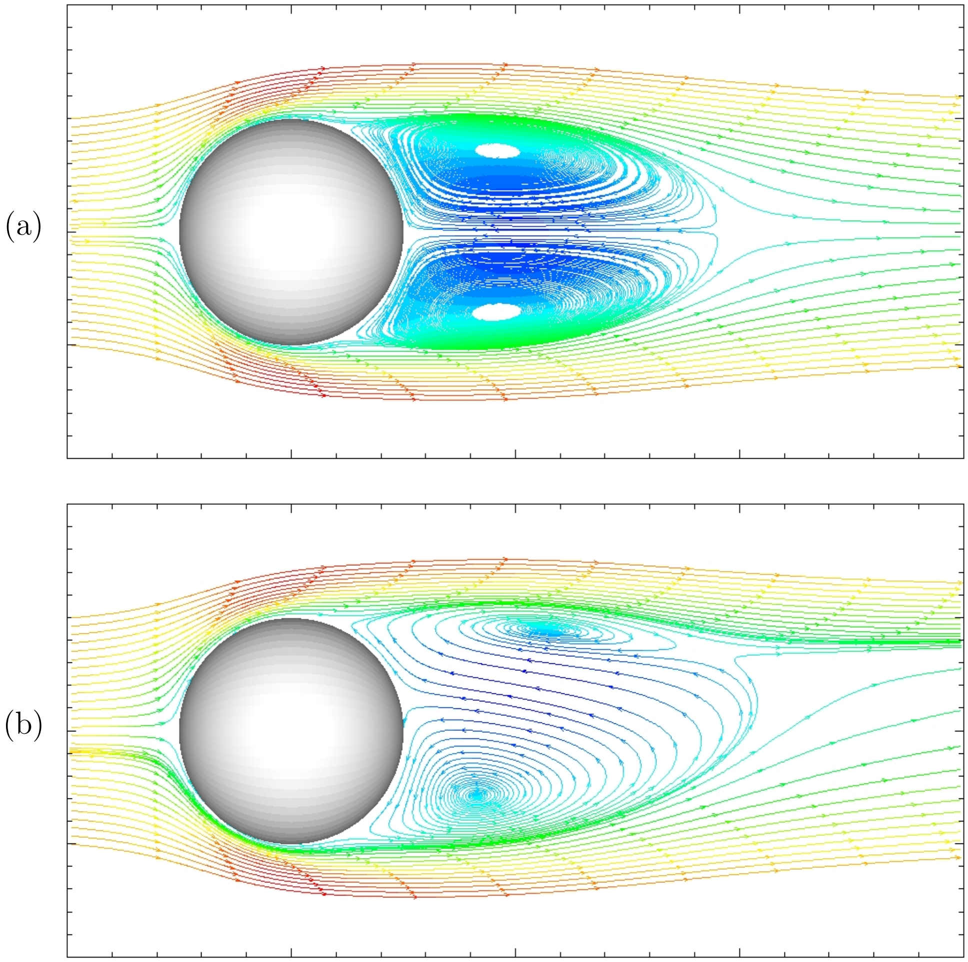
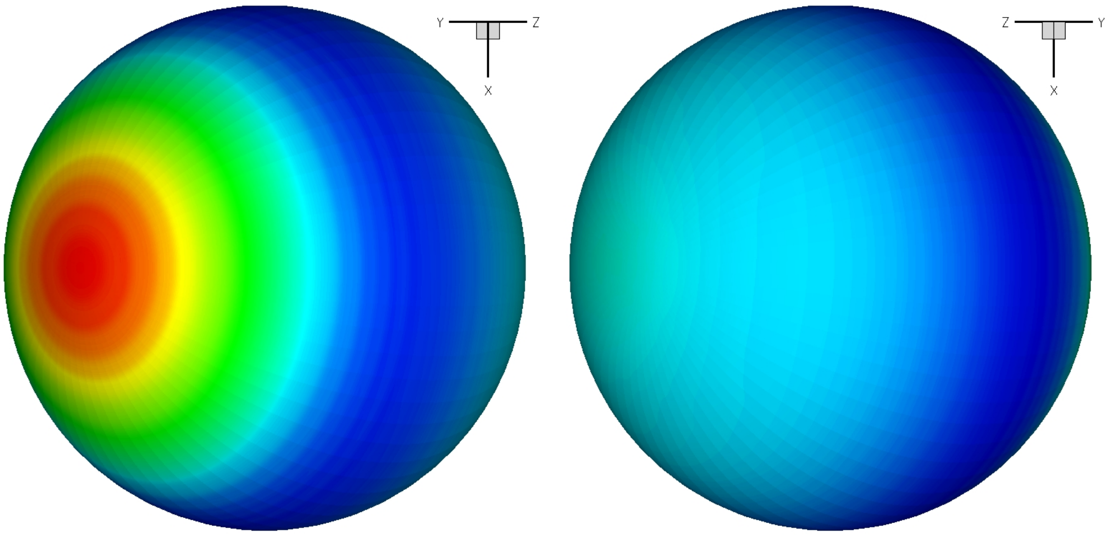


Figure 5.5: Streamlines of projected velocity field for flow past the sphere at $Re = 250$ (colored by the streamwise velocity): (a) $x - z$ plane; (b) $y - z$ plane.

this part have been rotated to have the symmetry plane coincide with the $y - z$ plane. In [28], the plane of symmetry is near the azimuthal boundary plane. This difference may be due to the different initial disturbances in two simulations.

For $Re = 250$, the streamlines obtained from the projected velocity field in both $x - z$ and $y - z$ planes are shown in Fig. 5.5. The symmetry of the streamlines in the $x - z$ plane is obvious with the upper part mirroring the lower part of the figure. Note that in the $y - z$ plane the velocity component perpendicular to this



(a)

(b)

Figure 5.6: Surface pressure distribution on the sphere at $Re = 250$: (a) front view; (b) rear view.

plane is non-zero and the streamlines are different from the real three-dimensional case due to the projection operation.

The contour plot of the surface pressure is shown in Fig. 5.6. The pressure distribution near the stagnation points is still axisymmetric. However, on the rear part of the sphere the pressure losses its axial symmetry. Note that the surface pressure is obtained using the surface force calculation procedure in Chap. 3.

The contours of the azimuthal vorticity in the $x-z$ and $y-z$ planes are shown in Fig. 5.7. It is evident that the vorticity in the $x-z$ plane are symmetric and very similar to those of the lower Reynolds number cases. However, this symmetry breaks down in the $y-z$ plane and one side of the vorticity contours is extended farther downstream than the other side.

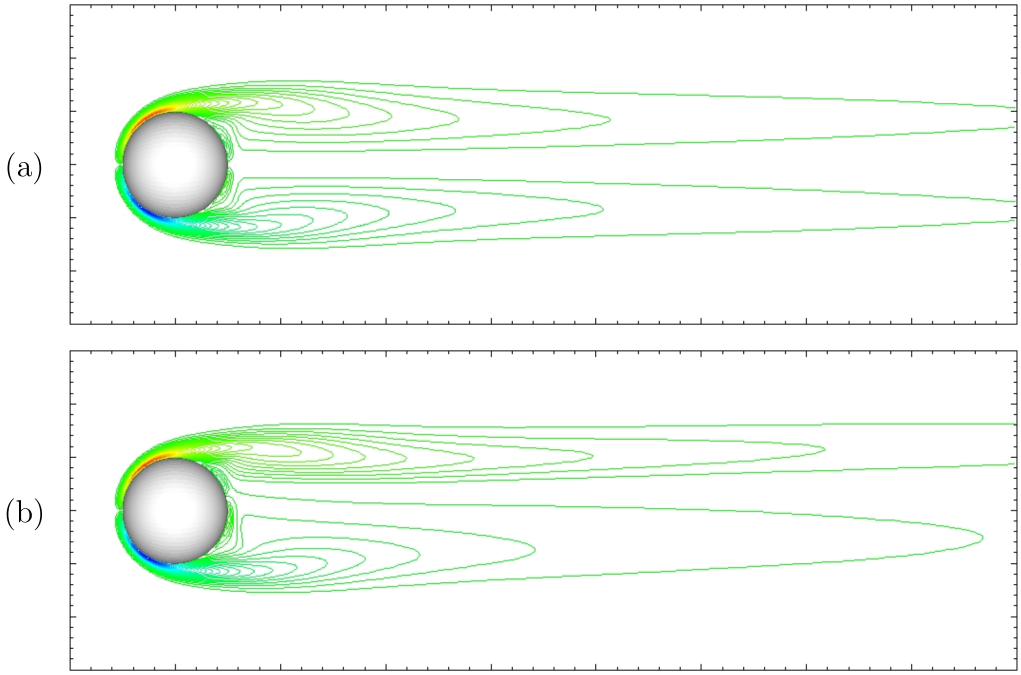


Figure 5.7: Vorticity contours for flow past the sphere at $Re = 250$: (a) Spanwise vorticity in $x - z$ plane; (b) Spanwise vorticity in $y - z$ plane.

The drag coefficient for $Re = 250$ is $C_D = 0.70$ from the present simulation, which is the same as in [28]. Due to the loss of axisymmetry, a non-zero lift force is generated. And the lift coefficient is $C_L = 0.062$, which is also the same as in [28].

For $Re = 300$ the flow past the sphere becomes unsteady and the vortices are shed, although the planar-symmetry is retained. The Strouhal number in the current study is $St = 0.133$, which is in good agreement with the Strouhal number of 0.137 in [28] (within 3%). Fig. 5.8 shows the force coefficients as a function of time. The mean drag coefficient is $C_D = 0.655$, which is the same as that in [28]. Note that in the figure the side force coefficients C_x and C_y are given directly without a rotation to have $y - z$ plane as the plane of symmetry. The combination of these two components gives a lift coefficient $C_L = 0.064$, which is again in very good agreement with other reference data ($C_L = 0.065$ in [28]).

The contours of the azimuthal vorticity component in the $x-z$ plane is shown in Fig. 5.9 for the four quarter periods. The definition of the period of shedding is chosen to be the same as in [28]. From the first to the fourth panel, the pinch-off and the convection downstream of a segment of vorticity is evident. A further check of Fig. 5.10, which shows the azimuthal vorticity component in the $y-z$ plane, reveals that the above phenomenon corresponds to a small vortex shed from the vortex attached to the lower part of the sphere. Also, the vortex shed from the upper part of the sphere has an interesting finger shape.

The vortical structures in the wake are shown in Fig. 5.11 and 5.12, where iso-surfaces of the second invariant of the velocity gradient tensor (or the Q-criterion) are used for their visualization. Hairpin like vortices originating from the surface of the sphere can be observed, with new vortical structures developing around their legs as they are convected downstream. This behavior is nearly the same as the one shown in [28].

5.1.3 Transitional Flow

The flow past the sphere loses its planar-symmetry as the Reynolds number further increases. A DNS of the flow at $Re = 1000$ has been conducted to establish the applicability of the current method in transitional and turbulent flows in cylindrical coordinates. The computational domain is chosen to the same as in [60]:

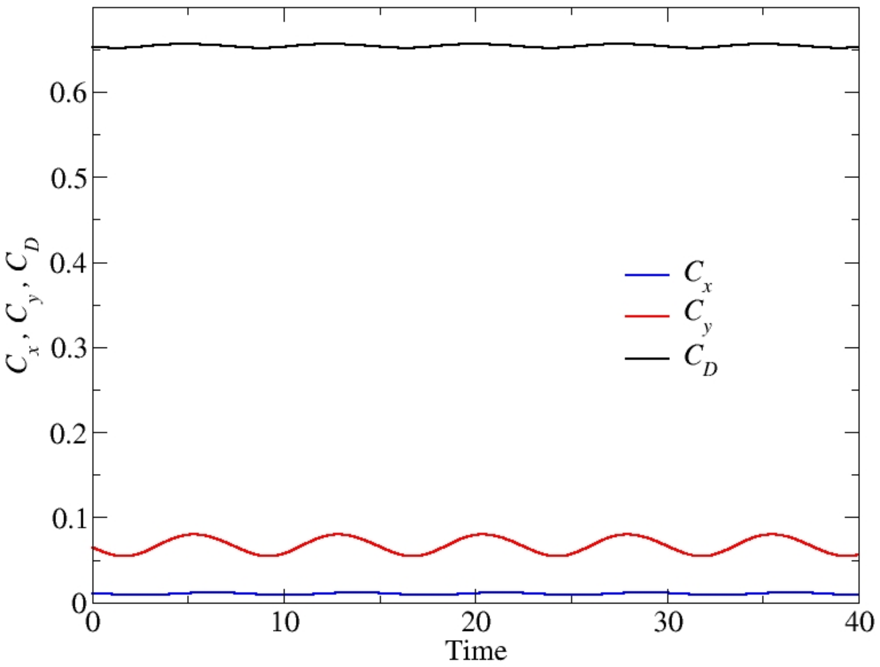
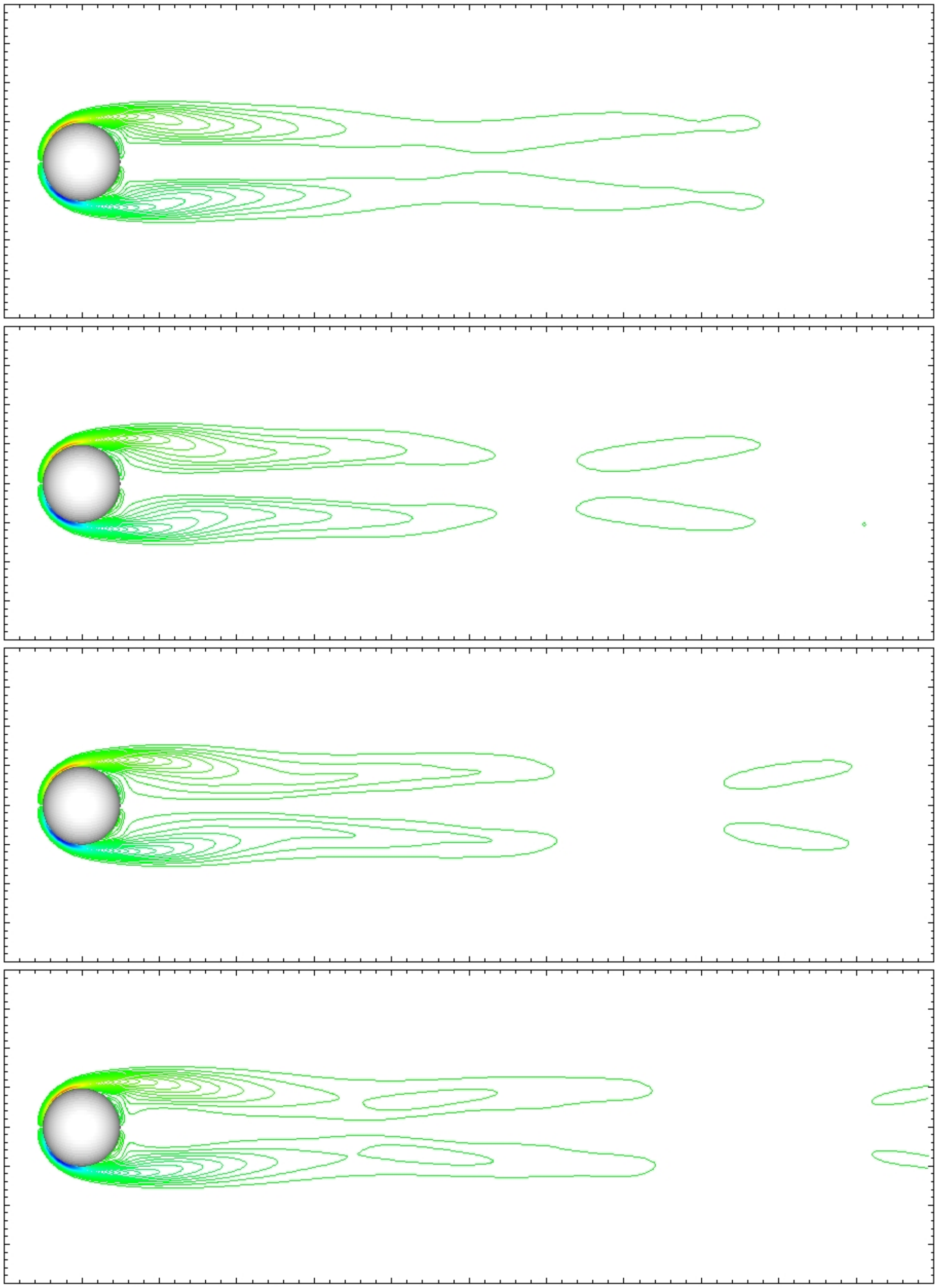


Figure 5.8: Time history of drag, x, and y side force coefficients at $Re = 300$.

$-4.5D \sim 25D$ in the streamwise direction with the sphere located at the origin, and $0 \sim 4.5D$ in radial directions. The computational grid is $300 \times 64 \times 768$ in the radial, azimuthal, and streamwise directions, respectively. The total number of points is nearly 15 million. The simulation is started from the interpolated flow field of $Re = 300$. Due to the limited computing resources, this simulation has not been run for enough time to obtain statistically stationary results: the simulation has been carried out for a total of about 100 time units with the first 60 time units discarded. For the current grid, a mean separation angle 100° is obtained, while in [60] the averaged separation angle was 102°. Although our result is close to that in [60], further refinement near the sphere surface may be necessary to capture the thin boundary layers.



(a) $t = 0T$

(b) $t = \frac{1}{4}T$

(c) $t = \frac{2}{4}T$

(d) $t = \frac{3}{4}T$

Figure 5.9: Spanwise vorticity contours for flow past the sphere at $Re = 300$ at the four quarter periods in the $x - z$ plane.

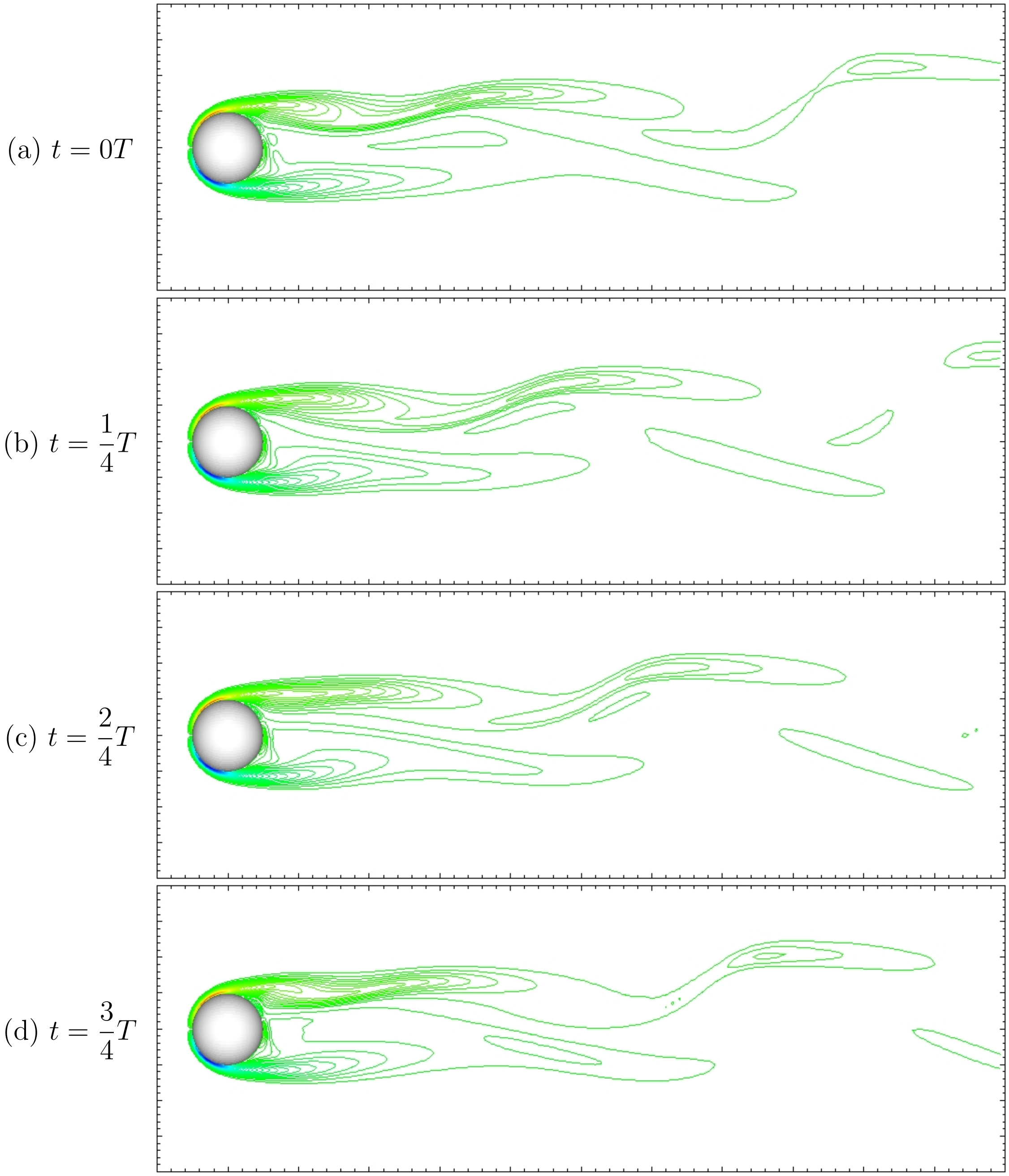
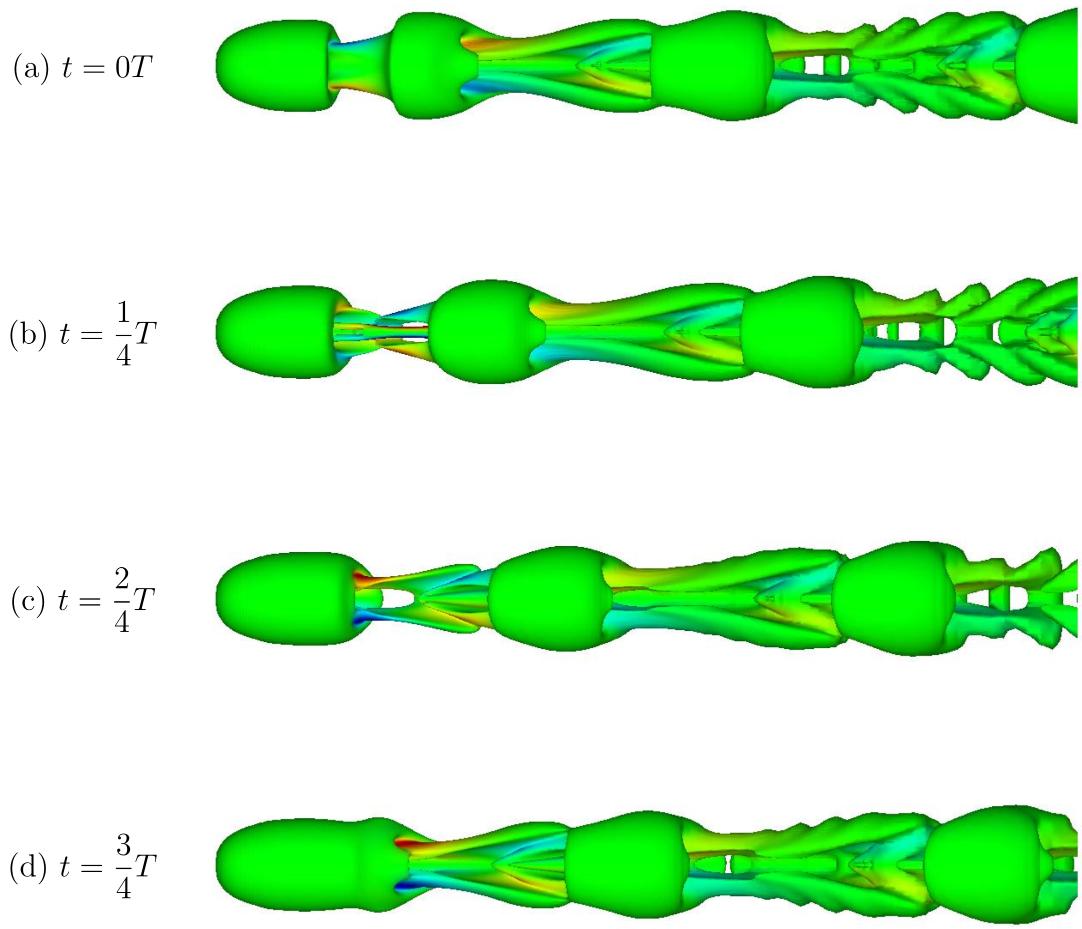


Figure 5.10: Spanwise vorticity contours for flow past the sphere at $Re = 300$ at the four quarter periods in the $y - z$ plane.



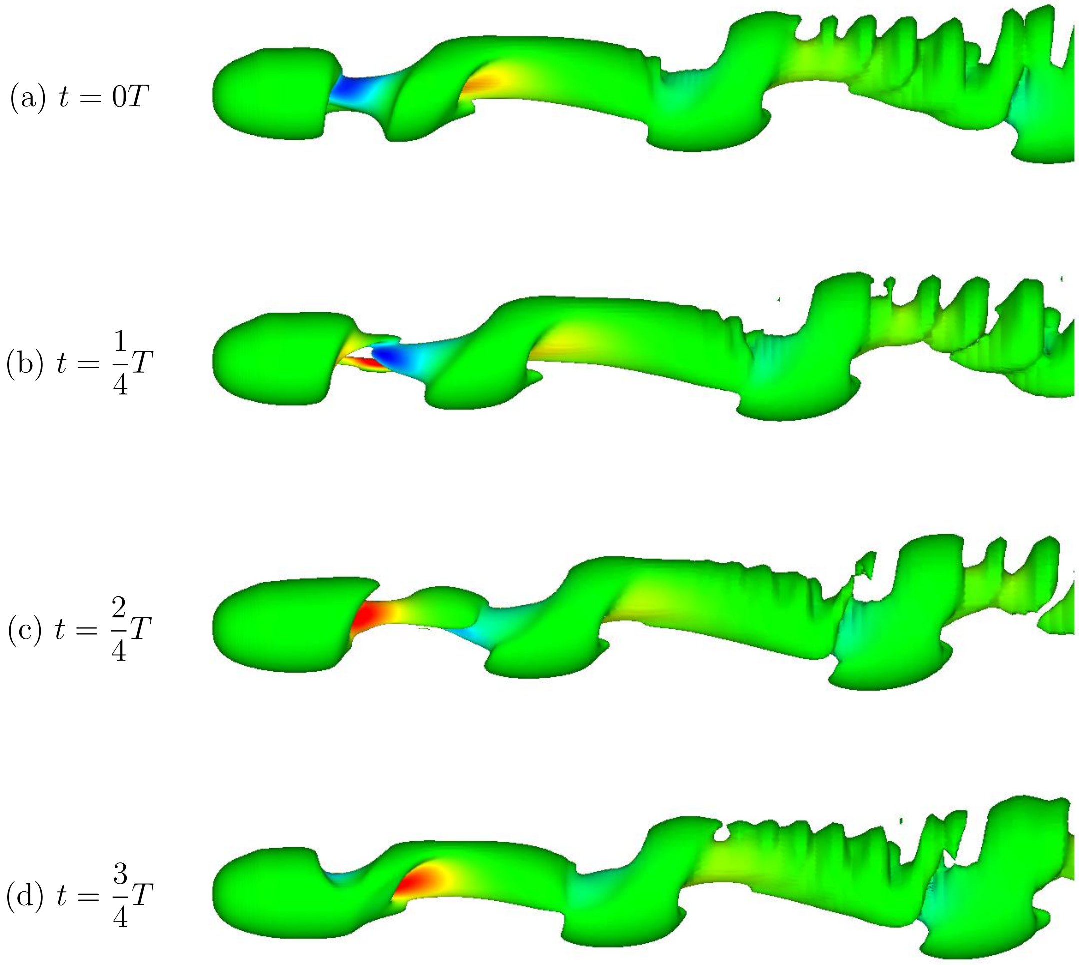
(a) $t = 0T$

(b) $t = \frac{1}{4}T$

(c) $t = \frac{2}{4}T$

(d) $t = \frac{3}{4}T$

Figure 5.11: Instantaneous vortical structures for flow past the sphere at $Re = 300$ (colored with the streamwise vorticity) at the four quarter periods: $x - z$ view.



(a) $t = 0T$

(b) $t = \frac{1}{4}T$

(c) $t = \frac{2}{4}T$

(d) $t = \frac{3}{4}T$

Figure 5.12: Instantaneous vortical structures for flow past the sphere at $Re = 300$ (colored with the streamwise vorticity) at the four quarter periods: $y - z$ view.

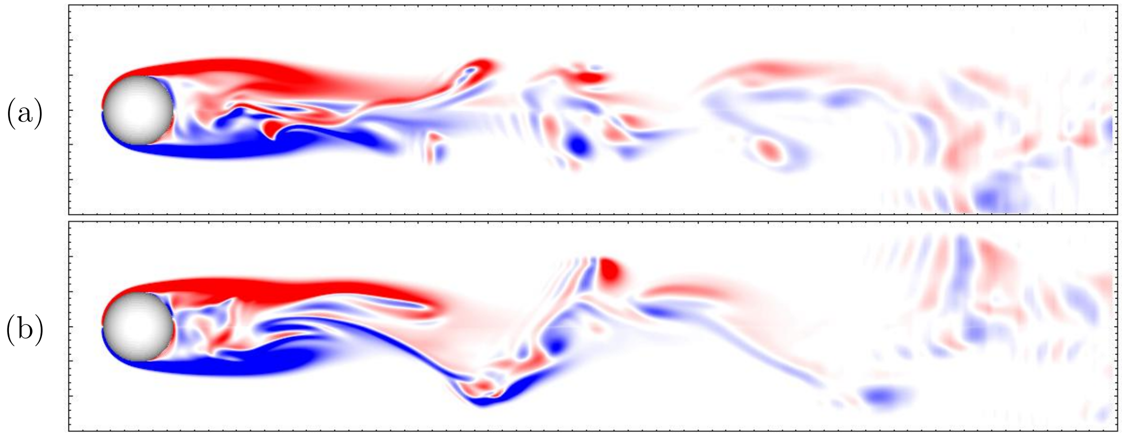


Figure 5.13: Instantaneous azimuthal vorticity contours for flow past the sphere at $Re = 1000$: (a) $x - z$ plane; (b) $y - z$ plane.

At this Reynolds number, small scales are present in the wake and the flow becomes chaotic further downstream. Those small scale structures are generated by the breakdown of the large cylindrical vortical structures that are formed just downstream of the sphere. The mechanism of this breakdown is due to a Kelvin-Helmholtz-like instability of the shear layer developed from the boundary layer separation of the sphere. The instantaneous azimuthal vorticity contours in the $x - z$ and $y - z$ plane and the instantaneous vortical structures are shown in Fig. 5.13 and 5.14, respectively. The roll-up of the shear layer, the development of large hairpin structures, and the breakdown of larger structures into smaller scales can be clearly observed. The flow becomes turbulent in the far wake. Those results are in very good agreement with the DNS using a high-order spectral method in [60].

The time history of the drag, and side force coefficients in x and y directions are shown in Fig. 5.15a. Unlike the $Re = 300$ case, which has a predominant side

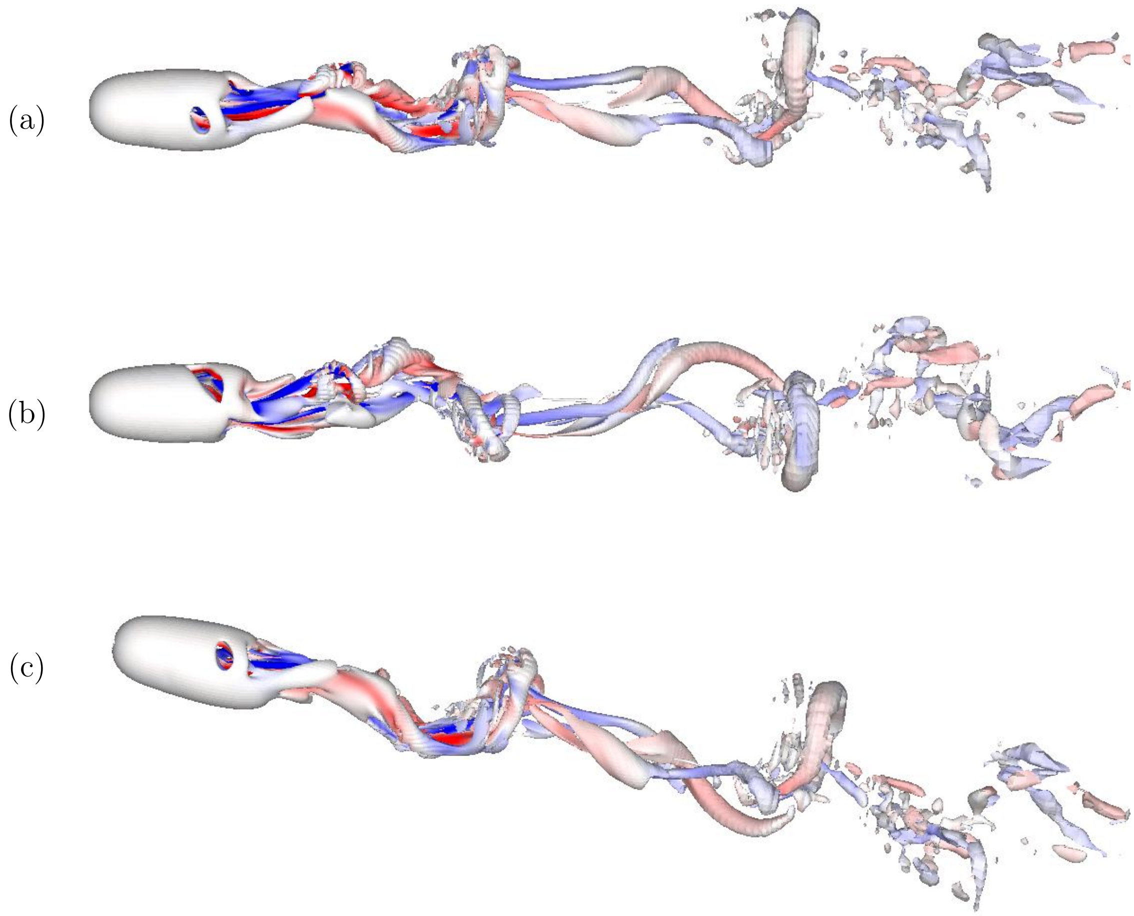
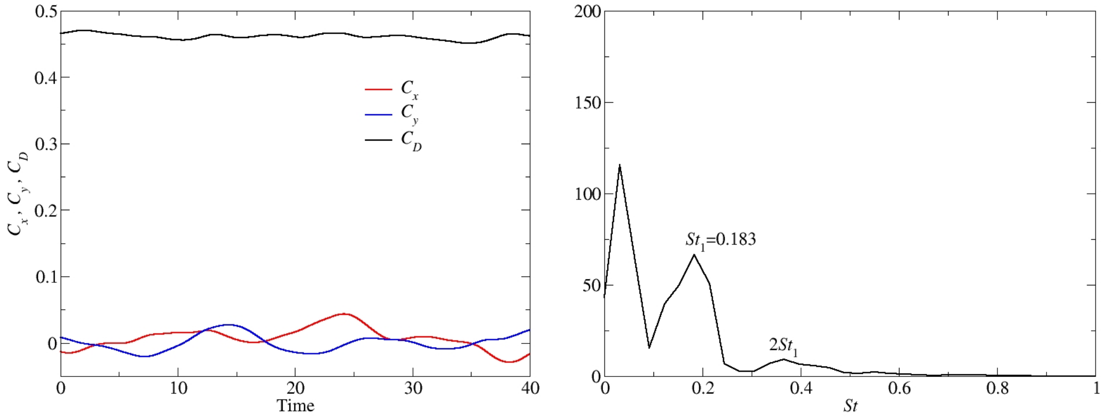


Figure 5.14: Instantaneous vortical structures for flow past the sphere at $Re = 1000$ (colored with the streamwise vorticity): (a) $x - z$ view; (b) $y - z$ view; (c) oblique view.

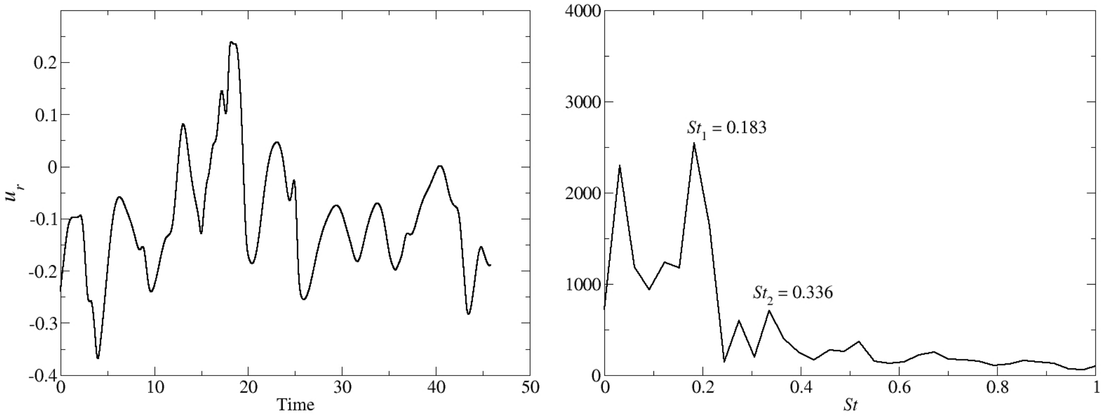


(a) Force coefficients (b) Frequency spectrum of C_D

Figure 5.15: Time history of drag, x, and y side force coefficients and frequency spectrum of C_D at $Re = 1000$.

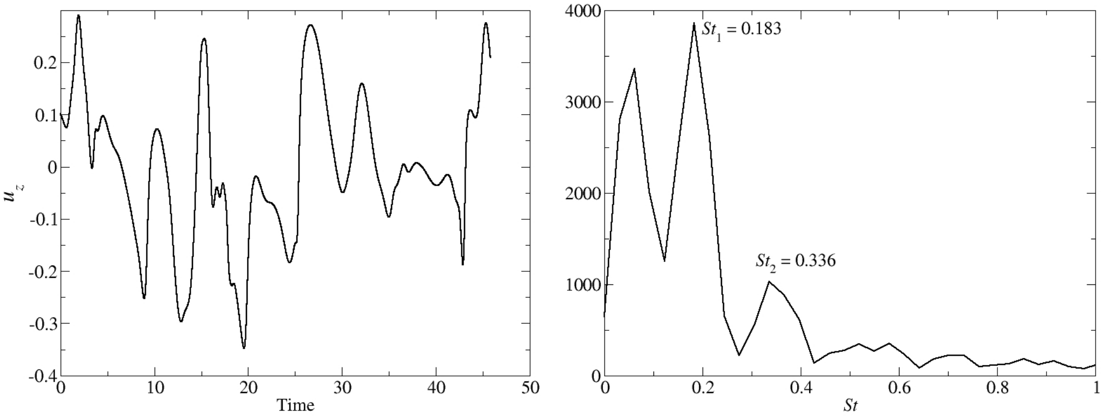
force which gives rise to a net lift force, for $Re = 1000$ both side forces in the x and y directions are oscillating around zero. A mean drag coefficient $C_D = 0.46$ is obtained for the limited sample size. The frequency spectrum of C_D is given in 5.15b. The Strouhal number is $St_1 = 0.183$, which is smaller than that in [60]. The reason for this difference may be the earlier separation in our simulation and the limited sample.

Fig. 5.16 and 5.17 show the time series of the axial and streamwise velocity components at $r = 0.3D$ and $z = 2.0D$ and their frequency spectra, respectively. Both components show a major Strouhal number of $St_1 = 0.183$; they also present a second higher frequency at $St_2 = 0.336$, which is again smaller than the corresponding value of $St_2 = 0.35$ in [60].



(a) (b)

Figure 5.16: Time history and frequency spectrum of u_r at $r = 0.3D$, $z = 2.0D$.



(a) (b)

Figure 5.17: Time history and frequency spectrum of u_z at $r = 0.3D$, $z = 2.0D$.

5.2 Transitional Flow Past an Airfoil

A more challenging test-case is the computation of transitional flow around an Eppler E387 airfoil. The angle of attack is $10°$ and the chord Reynolds number is $Re = \rho U c/\mu = 10,000$ with U the free-stream velocity and c the chord length of the airfoil. The Reynolds number regime is characteristic of bird flight.

The computational domain used is $-10D \sim 21D$ and $-10D \sim 10D$ in the streamwise and transverse direction, respectively. The airfoil is arranged such that there is a $10D$ distance to the inflow, the upper and the lower boundaries, and a $20D$ distance to the outflow boundary. This simulation is very demanding in terms of computing resources. A LES using the Lagrangian dynamic model has been performed on a grid with approximately 18×10^6 nodes ($560 \times 48 \times 672$ in the transverse, spanwise, and streamwise directions, respectively). The length scale in the spanwise direction is chosen to be $4c$. A part of the computational grid is shown in Fig. 5.18. The computation has been run at a constant $CFL = 1.2$, which gives a timestep varying in $4 \sim 8 \times 10^{-4} c/U$.

The flow field is characterized by the massive separation as shown in Fig. 5.19 where the mean streamlines are shown. A primary recirculation bubble, which starts from the laminar separation point and extends until the trailing edge can be observed. A second recirculation zone induced by the flow below the airfoil develops at the trailing edge. Interestingly, a third separation bubble, which is attached to the airfoil, beneath the large one can be observed. A careful examination of Fig. 5.20, which gives the pressure coefficient C_p on the upper and lower surface of the

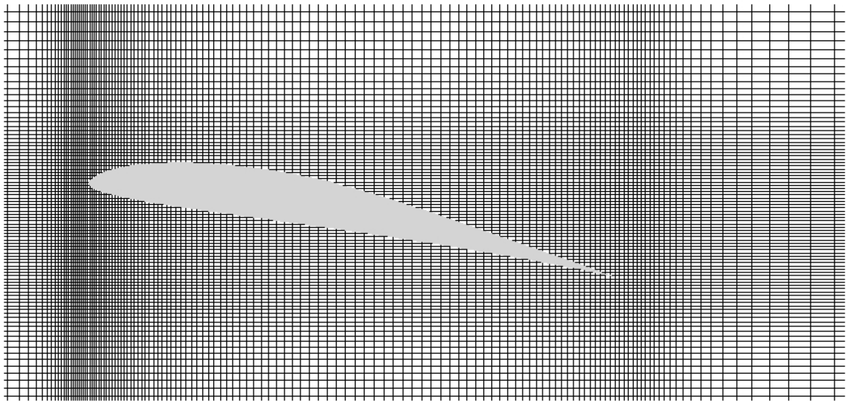


Figure 5.18: Part of computational grid in $x - z$ plane around the airfoil. Every three grid lines are shown.

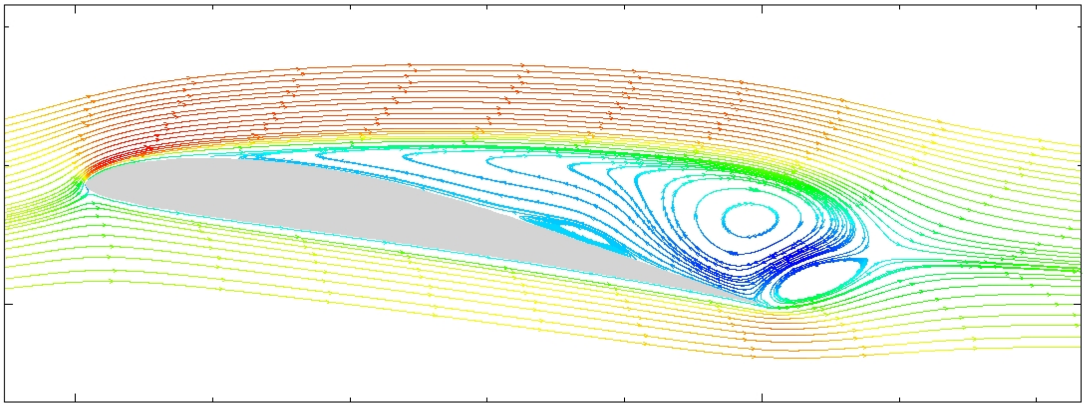


Figure 5.19: Mean streamlines of the flow past an E387 airfoil. Streamlines are colored with streamwise velocity component.

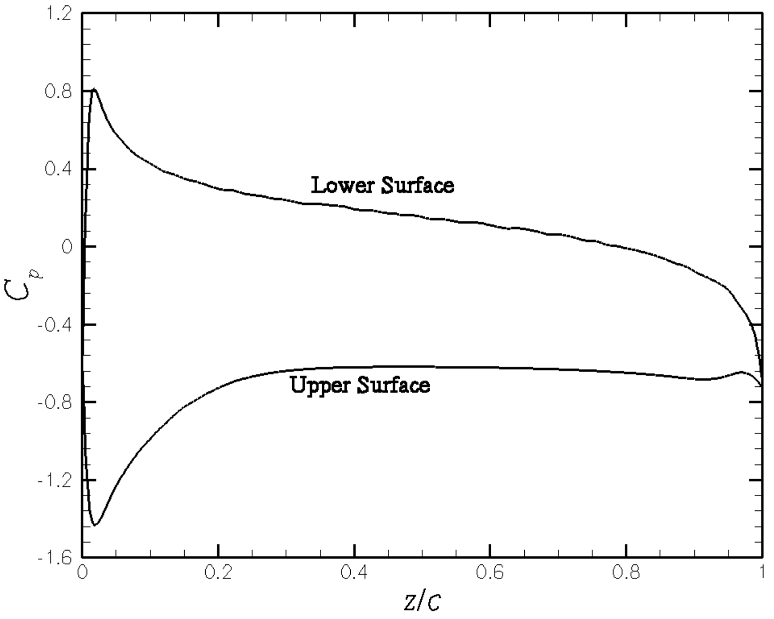


Figure 5.20: Mean pressure coefficient on the airfoil surface.

airfoil, shows that an adverse pressure gradient developes on the upper surface as the flow approaching this second separation point. In Fig. 5.20, a small bump of C_p is evident as a consequence of reattachment. The time history of the drag and lift force coefficients is given in Fig. 5.21. A mean drag coefficient $C_D = 0.13$ and a mean lift coefficient $C_L = 0.86$ can be derived from the limited time sample. The frequency sepctra of the drag and lift coefficients are shown in Fig. 5.22. The lower Strouhal number is $St_1 = 0.05$ and the averaged higher Strouhal number is $St_2 = 1.3$. These two different Strouhal numbers may be associated with the two different vortex sheddings: one resulting from the separation on top of the airfoil and the other from the trailing edge separation. However, a detailed analysis of the time series of the variables at different locations is necessary to obtain the timing of

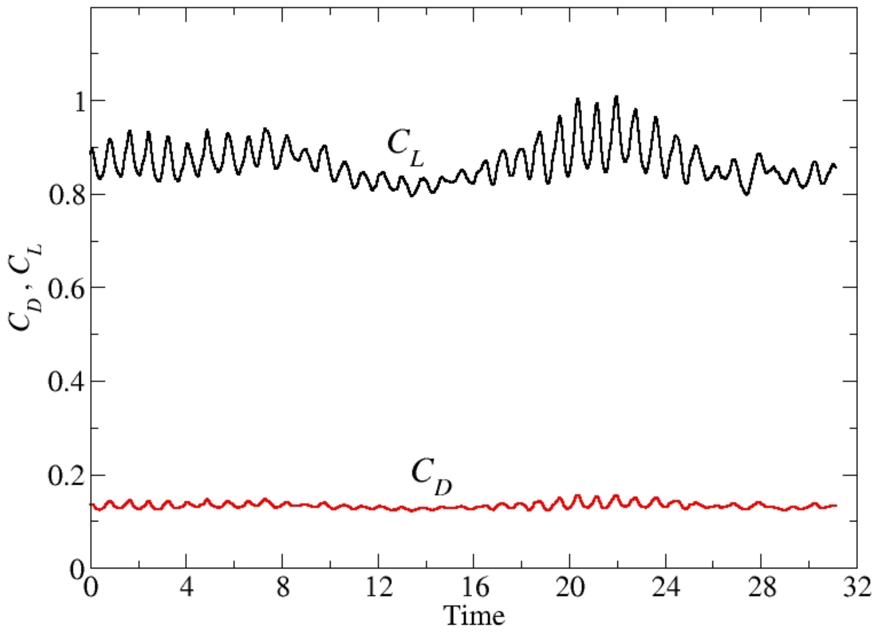
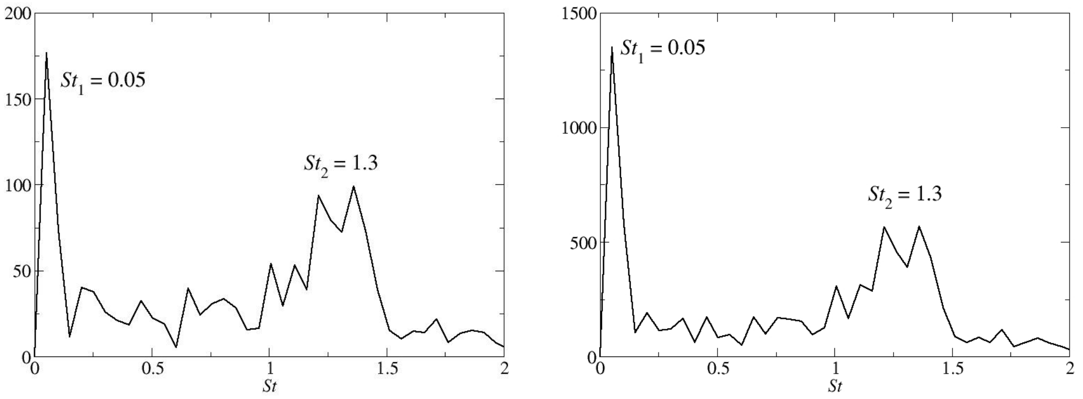


Figure 5.21: Time history of drag and lift coefficients for the flow past an airfoil.

the vortex shedding.

The turbulent intensities, turbulent shear stress, and the eddy viscosity are shown in Fig. 5.23 (Note that the turbulent intensities and shear stress only contain the resolved parts). The turbulence activities of the transverse and spanwise components exist mainly within the shear layer formed at the trailing edge, where the laminar flow beneath the airfoil interacts with the transitional flow resulting from the massive separation. The close-up views of streamwise turbulent intensity, the turbulent shear stress, and the mean eddy viscosity normalized using the molecular viscosity ν_m are given in Fig. 5.24. It is apparent that a weak shear layer is generated as the flow separates from the upper surface of the airfoil.

Fig. 5.25 shows the instantaneous spanwise and vorticity components, and Fig.



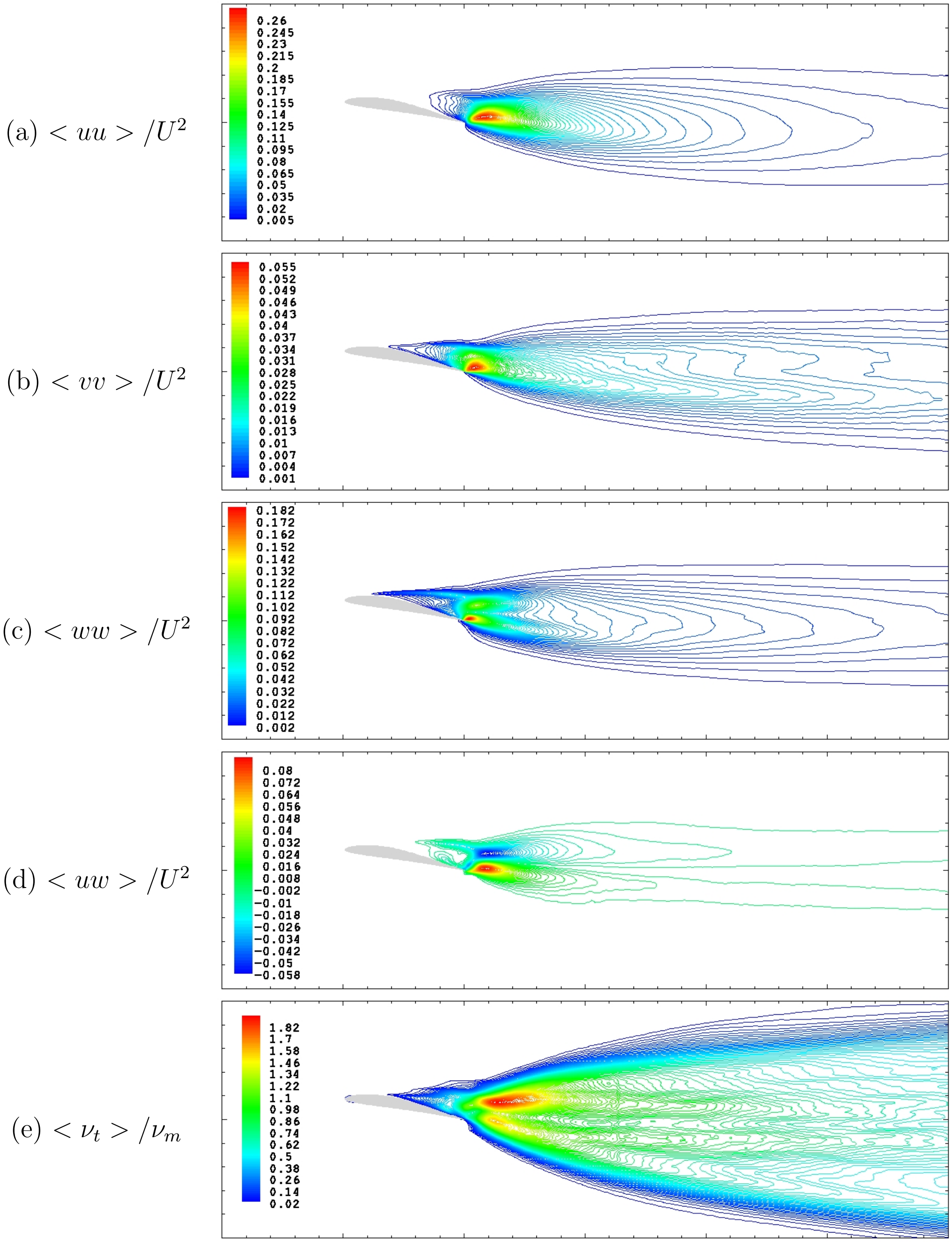
(a) (b)

Figure 5.22: Frequency spectra of the drag and lift coefficients for the flow past an airfoil.

5.26 shows the instantaneous vortical structures. At the early stage of separation, the vortices are well-organized and large rollers are generated. The flow becomes turbulent very soon after it separates from the trailing edge.

5.3 Turbulent Flow Over a Traveling Wavy wall

In this section we will investigate the accuracy and efficiency of the proposed approach in the framework of LES of turbulent flows. In particular, we will perform LES for the case of turbulent flow over a flexible wall undergoing streamwise traveling wave motion. A key parameter in this flow is the ratio of the wave speed, c, to the external mean flow, U, which has a substantial effect on turbulence production near the surface. Depending the value of c/U turbulence can be enhanced or fully suppressed leading to relaminarization. The parametric space in our compu-


(a) $< uu > /U^2$

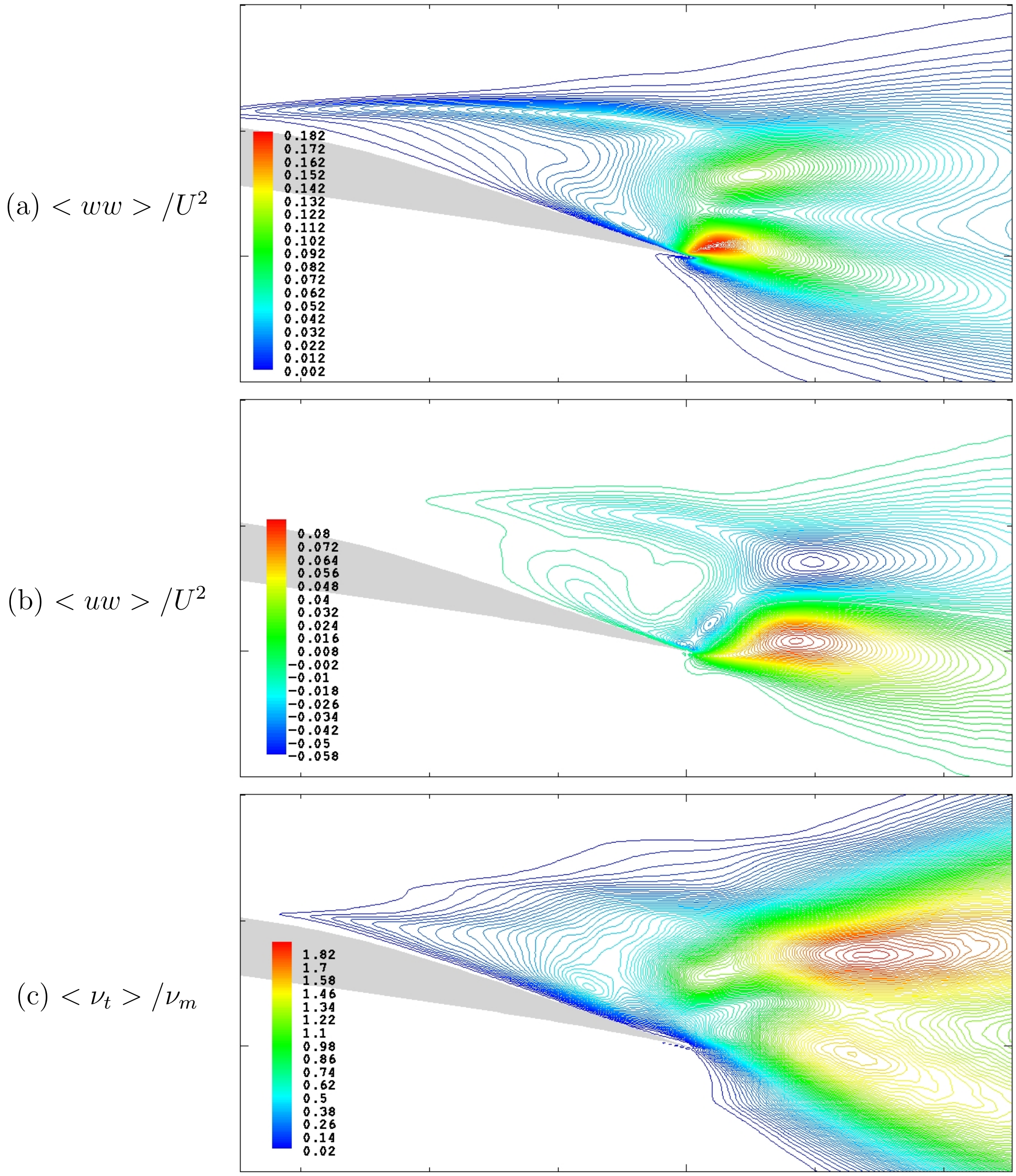
(b) $< vv > /U^2$

(c) $< ww > /U^2$

(d) $< uw > /U^2$

(e) $< \nu_t > /\nu_m$

Figure 5.23: Turbulent intensities, turbulent shear stress, and mean eddy viscosity for the flow past an airfoil.

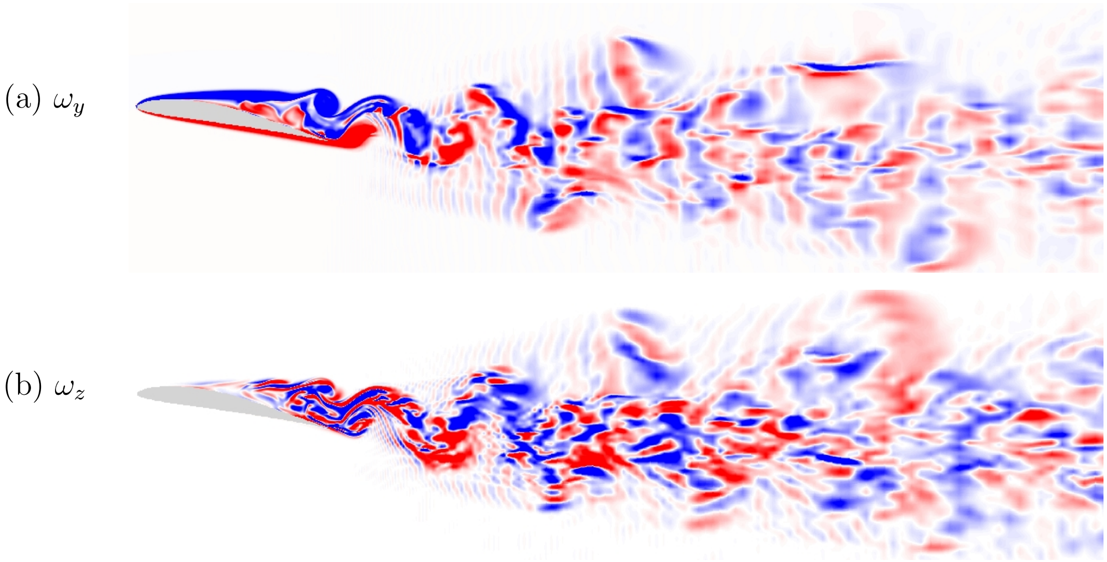


(a) $< ww > /U^2$

(b) $< uw > /U^2$

(c) $< \nu_t > /\nu_m$

Figure 5.24: Close-up views of the streamwise turbulent intensity, turbulent shear stress $< uw >$, and mean eddy viscosity $< \nu_t >$ for the flow past an airfoil.



(a) ω_y

(b) ω_z

Figure 5.25: Instantaneous contours of the spanwise and streamwise vorticity components.

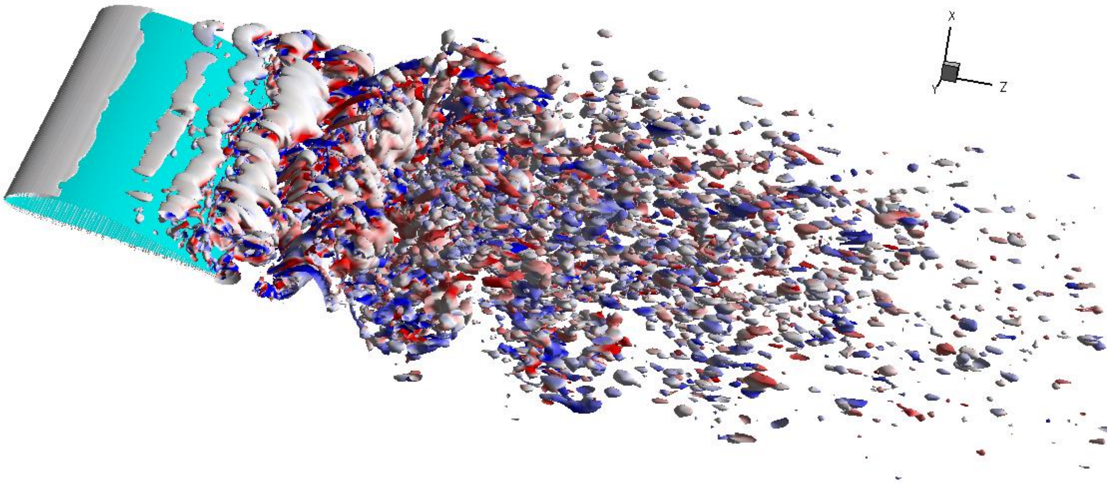


Figure 5.26: Instantaneous vortical structures using isosurfaces of $Q = 10$ colored by the streamwise vorticity for the flow past an airfoil.

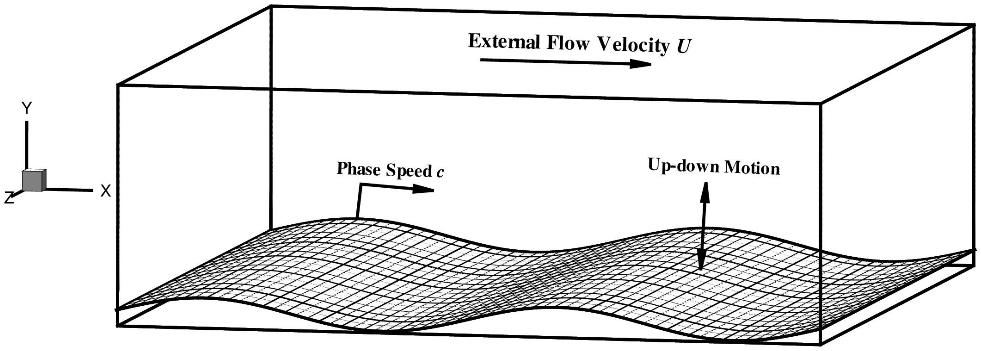
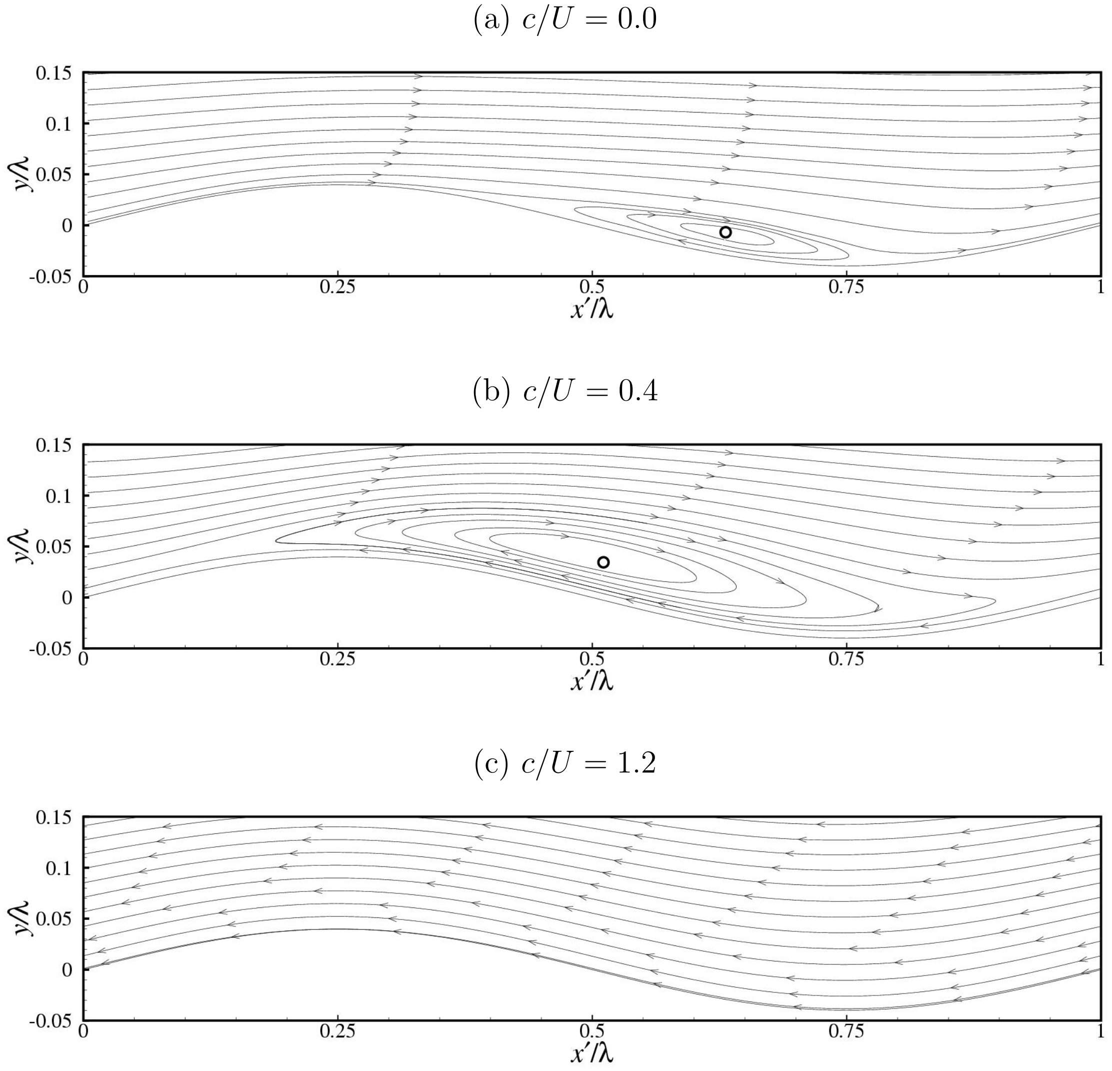


Figure 5.27: A sketch of the geometry for the case of turbulent flow over a traveling wavy wall.

tations is selected to match the conditions in the DNS by Shen *et al.* [54]. These simulations were performed using a pseudospectral method in the horizontal directions, and 2$^{\text{nd}}$ order finite-differences in the wall-normal direction. They also used a coordinate transformation to map the physical space into a computational space that eliminates the need to deform the grid with the wave motion. A sketch of the computational domain is shown in Fig. 5.27. The wavy wall is undergoing a vertical oscillation in the form of a streamwise-traveling wave. The location of the wall boundary as a function of time is: $y_w(t) = a \sin k(x - ct)$, where a is the magnitude of the oscillation, $k = 2\pi/\lambda$ is the wavenumber with λ the wavelength, and c is the phase-speed of the traveling wave. For such a configuration the only non-zero component of the velocity vector on the boundary is the vertical one given by: $v_w(t) = -\omega a \cos k(x - ct)$, where $\omega = kc$ is the frequency of oscillation. The components of the velocity vector are denoted by u, v, and w in the x (streamwise), y (wall-normal), and z (spanwise) directions respectively.


(a) $c/U = 0.0$

(b) $c/U = 0.4$

(c) $c/U = 1.2$

Figure 5.28: Mean streamline pattern in a frame of reference following the traveling wave. (a) $c/U = 0.0$; (b) $c/U = 0.4$; (c) $c/U = 1.2$. ○ indicates the center of the corresponding recirculation region in the reference DNS [54].

We have considered the same wave steepness, $ka = 0.25$, and Reynolds number, $Re = \rho U\lambda/\mu = 10,170$ (U is the mean freestream velocity), as in the DNS by Shen *et al.* [54]. Computations at three different c/U ratios are presented: $c/U = 0.0$, $c/U = 0.4$ and $c/U = 1.2$. The first case, where $c/U = 0.0$, corresponds to a stationary wavy wall, and it is included to highlight the effect of the wave motion on the flow physics. The size of the computational domain is $2\lambda \times 2/\pi\lambda \times 2\lambda$, in the streamwise, wall-normal and spanwise directions respectively. The corresponding number of points in each direction is $288 \times 88 \times 64$ for all simulations. The grid is uniform in the streamwise and spanwise directions, and is stretched in the wall-normal direction. Grid refinement studies for the case of a stationary wavy boundary under similar flow conditions reported in [4] guided the generation of the above grid. Following Shen *et al.* [54], periodic boundary conditions are used in the streamwise and spanwise directions and a slip-wall at the top boundary. Phase-averaged statistics with respect to the motion of the wavy boundary were accumulated for approximately $40\lambda/U$, according to the relation [54]: $x - ct = x' + n\lambda$, where n is an integer and $0 \leq x' < \lambda$. In Fig. 5.28 the mean streamline pattern in a frame of reference moving with the phase speed c is shown for all c/U ratios. For the stationary wall case (see Fig. 5.28a) the flow separates just after the crest and forms a large recirculating bubble on the downslope side of the wavy boundary. As the c/U increases, the recirculating region becomes more elongated and extends over the crest (see Fig. 5.28b). For larger values flow separation is altogether suppressed and the streamlines follow the wavy terrain as shown in Fig. 5.28c for $c/U = 1.2$. On the same figure the centers of the corresponding recirculation regions from the

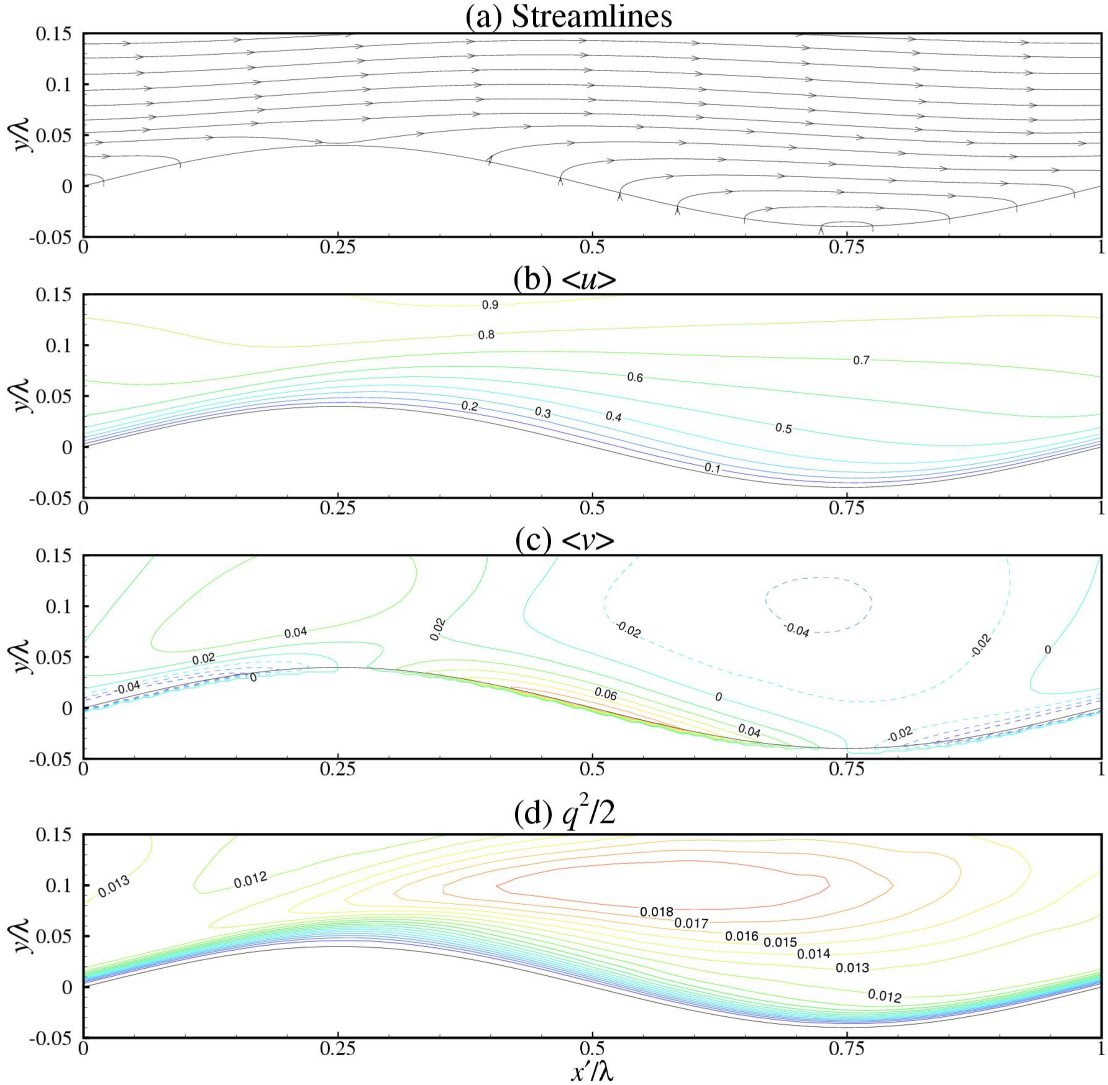


Figure 5.29: Phase-averaged statistics for $c/U = 0.4$.

reference DNS [54] are shown for comparison. The agreement with our results is very good indicating that the mean flow dynamics are properly captured with the present method.

In Figs. 5.29 and 5.30, detailed phase-averaged statistics in a fixed frame of reference for $c/U = 0.4$ and $c/U = 1.2$ are shown respectively. The differences compared to the stationary wavy wall case are more evident in this reference frame, with streamlines that begin from, and end on, the boundary as shown in part (a) of both figures. From the isolines of the wall-normal velocity component, $< v >$, it is also evident that as c/U increases from 0.4 to 1.2, the vertical flow induced by the moving wall increases substantially (see Figs. 5.29c and 5.30c). This has an effect on the shape of the streamlines which is concave for $c/U = 0$ (see Fig. 5.28a), and becomes flat for $c/U = 0.4$ and convex for $c/U = 1.2$. As expected, c/U, also affects the turbulent fluctuations. The turbulent kinetic energy is dramatically decreased and the maximum is relocated further downstream when c/U increases from 0.4 to 1.2, as shown in part (d) of both figures. A comparison of Figs. 5.29 and 5.30 to the corresponding ones in [54] -the scale and contour intervals are selected appropriately to facilitate direct comparisons- shows very good agreement with the reference results.

To further investigate the accuracy of the proposed approach the forces on the wavy boundary are computed, and then compared to the corresponding reference data. These quantities are very sensitive to the adopted numerical methodology and are a stringent test for the accuracy of the proposed formulation. Following Shen *et al.* [54], the local friction force, f_x^f, and local pressure force, f_x^p, on an element, ds,

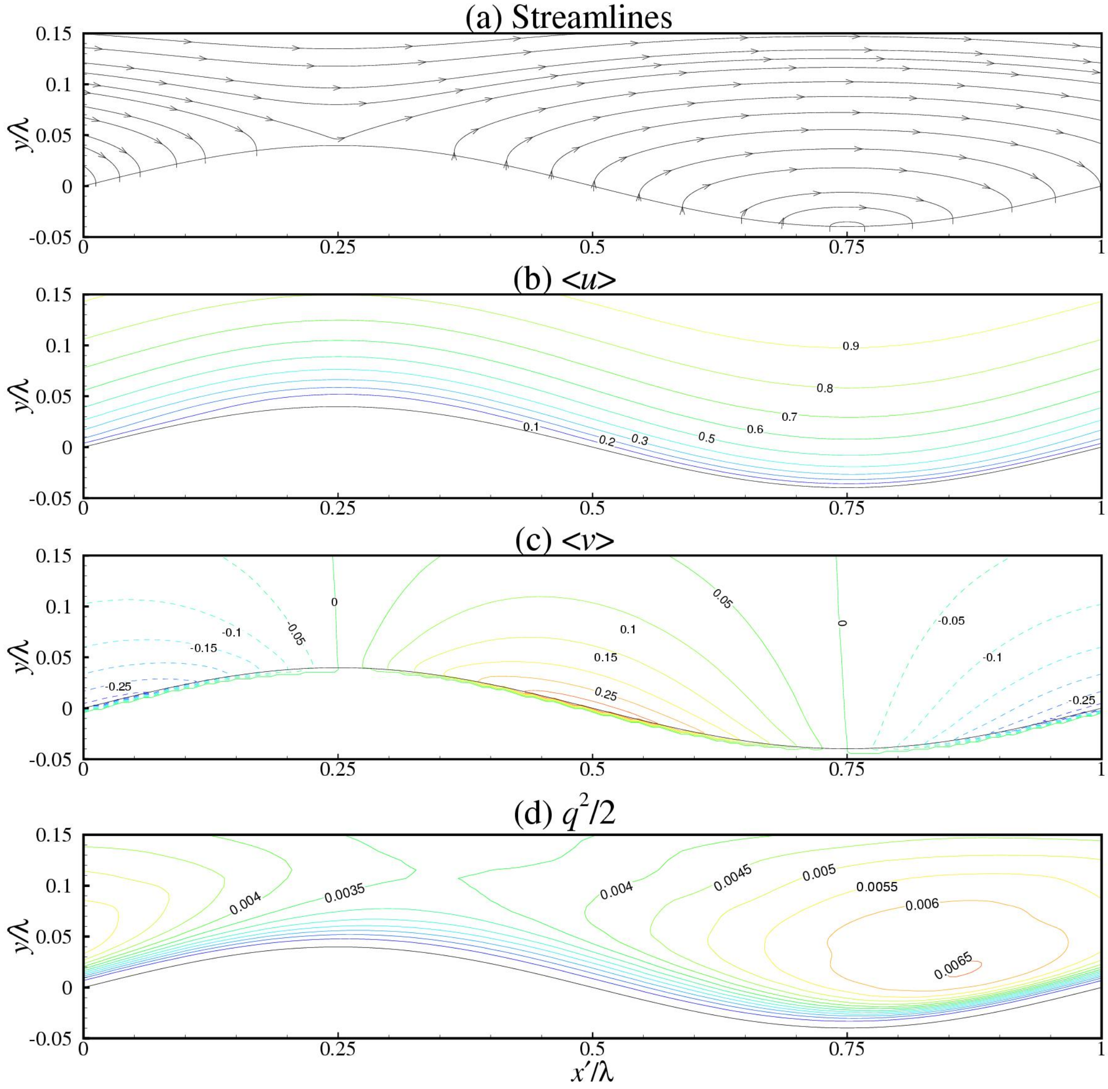


Figure 5.30: Phase-averaged statistics for $c/U = 1.2$.

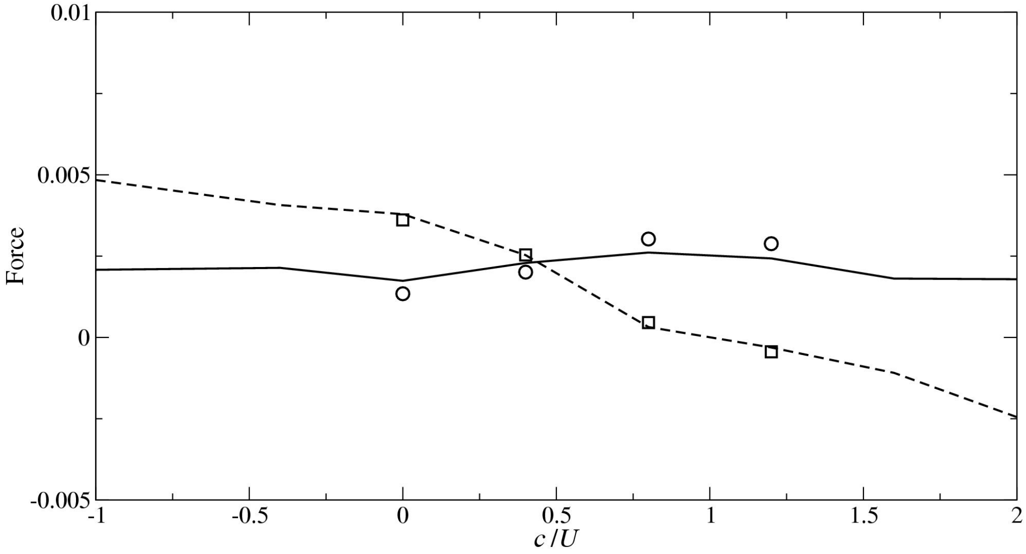


Figure 5.31: Mean force acting on the traveling wavy boundary as a function of c/U. —— , ○ is total friction force, F_f; - - - , □ is the total pressure force, F_p. Lines are the DNS results in [54], and symbols are the present results.

on the solid boundary can be written as:

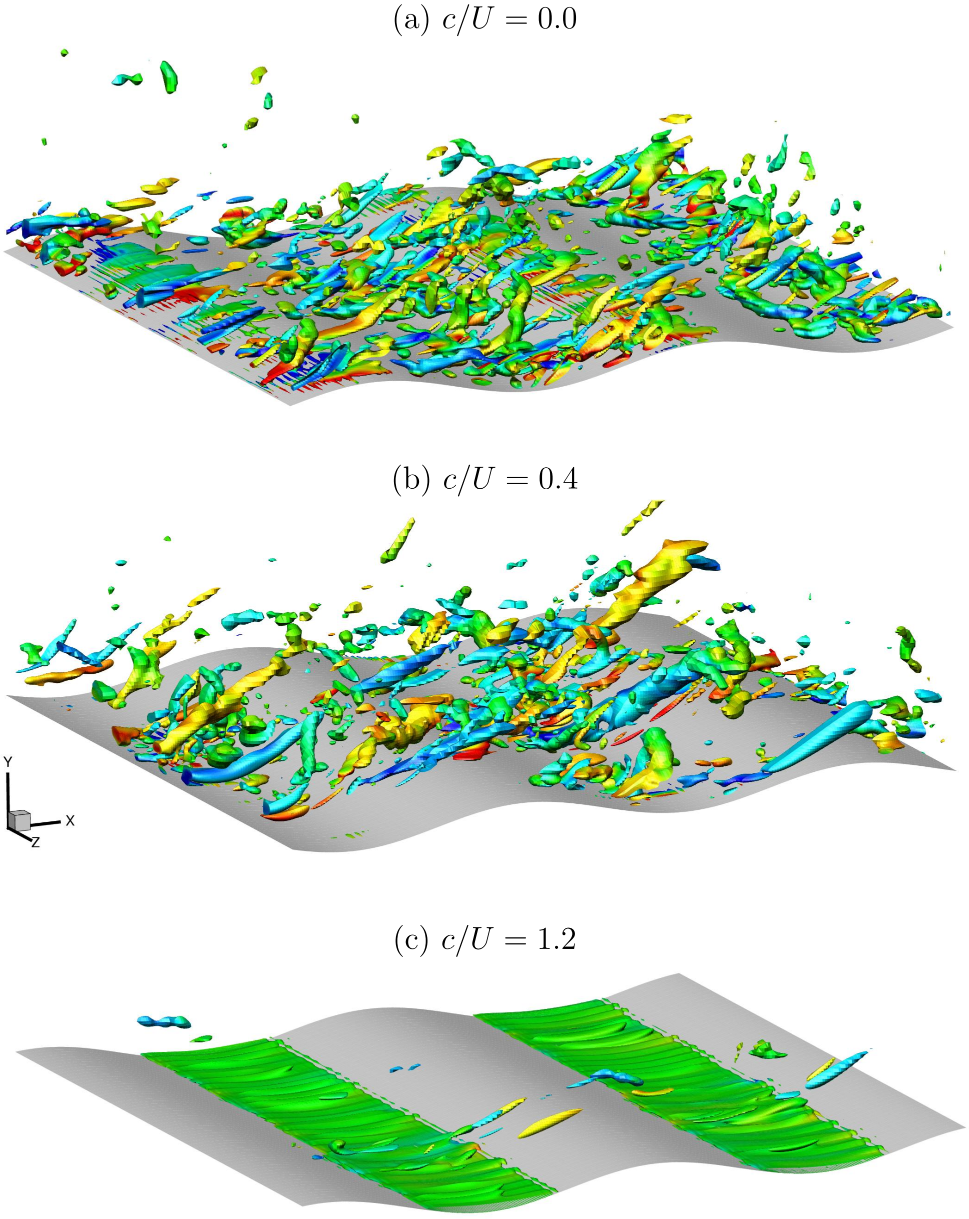
$$\begin{cases} f_x^f = \mu \left[-2\dfrac{\partial u}{\partial x}\dfrac{\mathrm{d}y_w}{\mathrm{d}x} + \left(\dfrac{\partial u}{\partial y} + \dfrac{\partial v}{\partial x} \right) \right] \dfrac{1}{\mathrm{d}s}, \\ f_x^p = p\dfrac{\mathrm{d}y_w}{\mathrm{d}x}\dfrac{1}{\mathrm{d}s}. \end{cases} \tag{5.1}$$

The velocity derivatives and surface pressure in Eq. (5.1) are computed using the method discussed in section 3.4. Then, one can integrate f_x^f and f_x^p over the wavy surface to obtain the total friction force, F_f, and total pressure force, F_p, respectively. A comparison of these quantities with the reference data is shown in Fig. 5.31. Note that we also include a simulation at $c/U = 0.8$, where all other parameters are the same as in the cases discussed above. Our results are in excellent agreement with DNS results. As in the reference simulations the variation of the friction force is relatively small and the value is always above zero, while the pressure force decreases as c/U increases and becomes negative around $c/U = 1.0$.

Finally, a series of instantaneous realizations from all cases were carefully examined to verify the smoothness of the velocity, vorticity and pressure fields near the moving boundary, and visualize the characteristic coherent vortical structures present in this flow. An example is shown in Fig. 5.32 where isosurfaces of the second invariant of the velocity gradient tensor, Q, are used to identify the near wall vortices. Three cases are included: $c/U = 0.0$, $c/U = 0.4$, and $c/U = 1.2$. For all cases the isosurfaces are colored with the streamwise vorticity values. For the stationary wavy boundary ($c/U = 0.0$) the presence of strong quasi-streamwise vortices that begin in the upslope area of the wave and extend over the trough is evident. As c/U increases, there are fewer and more elongated vortices. For the maximum ratio considered in the present study ($c/U = 1.2$) these structures are fully suppressed. This trend is in agreement with the phase-averaged turbulent statistics shown in Figs. 5.29 and 5.30, and the results reported in [54].

5.4 Pulsatile Flow Past a Prosthetic Bileaflet Heart Valve

To further demonstrate the capabilities of the present method in handling realistic turbulent and transitional flow problems that involve complex three-dimensional boundaries consisting of multiple moving parts, we have computed the flow past a bileaflet, mechanical heart valve in the aortic position. The geometry of the valve is shown in Fig. 5.33. The shape and size of the leaflets roughly mimics the St. Jude Medical (SJM) Standard bi-leaflet mechanical heart valve, which is commonly


(a) $c/U = 0.0$

(b) $c/U = 0.4$

(c) $c/U = 1.2$

Figure 5.32: Isosurfaces of $Q = 8.0$ colored with the streamwise vorticity at an instant in time.

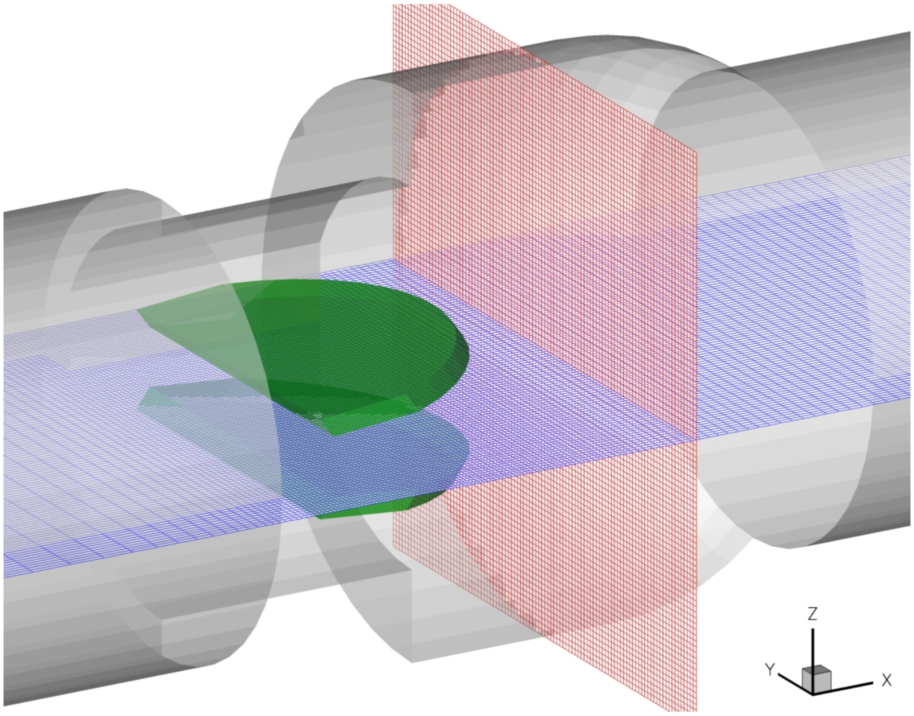


Figure 5.33: Geometry of the simplified bileaflet prosthetic heart valve placed in a rigid wall model of the aortic root. The leaflets are shown in their fully open position, and part of the housing wall has been removed for clarity. Two planes of the Cartesian grid are also shown.

used in clinical practice. In our case, however, the hinge region has been simplified to only allow one degree of freedom for each leaflet (rotation around a fixed axis in the y direction). Also, the leaflets contact the housing wall tightly and there is no gap between the hinge and the housing wall. The housing is also shown in Fig. 5.33, where a straight pipe with rigid walls expands and then contracts to mimic the geometry of the aortic root. The overall set-up resembles the ones commonly used in in-vitro experiments to test the hemodynamic performance of prosthetic valves in the aortic position (see for example [72] for a recent review).

In the actual case the motion of the leaflets is a product of their interaction with the pulsatile blood flow: the valve opens at the beginning of the systole and closes before the start of the diastole. When the leaflets are fully open the blood flow rapidly accelerates through the valve and reaches its peak velocity. Since the objective of the present simulations, however, is to demonstrate the applicability of the method to complex moving boundaries, the movement of the leaflets is prescribed according to a simplified law that resembles the real movement of the leaflets as determined by their interaction with the blood flow. The fully open position of the leaflet forms a 85° angle with the $y-z$ plane, while a rotation of 53° towards the housing wall closes the valves completely. In Fig. 5.34 the variation of the bulk velocity and the corresponding angle of the leaflets are shown for a full pulsatile cycle. The Reynolds number based on the pipe diameter and the maximum bulk velocity was set to $4,000$, which is a realistic value for this application.

The above configuration which involves the interaction of the flow with a stationary (the aortic chamber) and two moving (the leaflets) boundaries is a great challenge for any structured or unstructured boundary-conforming method. In the present computation the overall geometry is immersed into a Cartesian grid and the boundary conditions on the stationary and moving boundaries are imposed using the method described in the previous sections. The computational grid involves $420 \times 200 \times 200$ nodes in streamwise, spanwise (direction parallel to the rotation axis of the leaflets), and transverse directions, respectively. The mesh is stretched in the $x-y$ plane and is kept uniform in the z direction (see Fig. 5.33), in a way that the leaflets are moving in an approximately isotropic grid with average dimension of

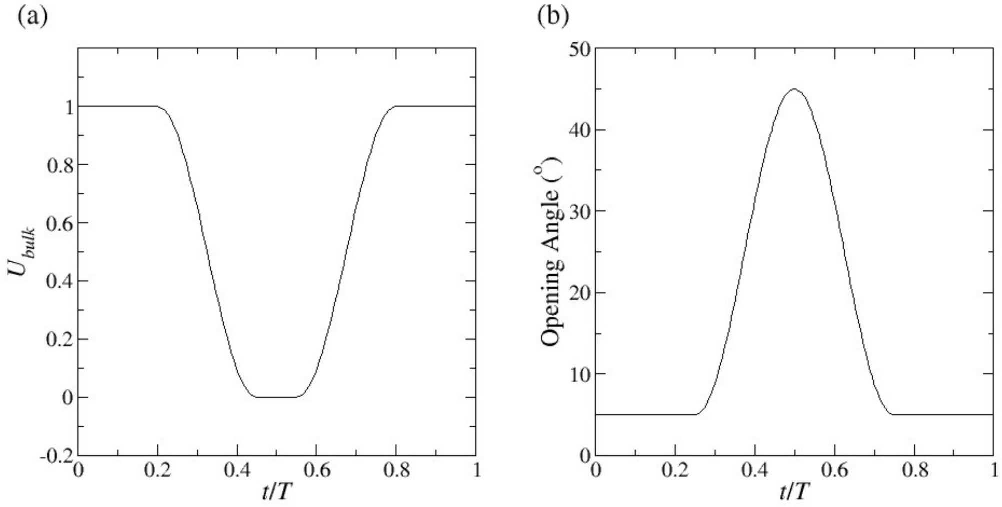
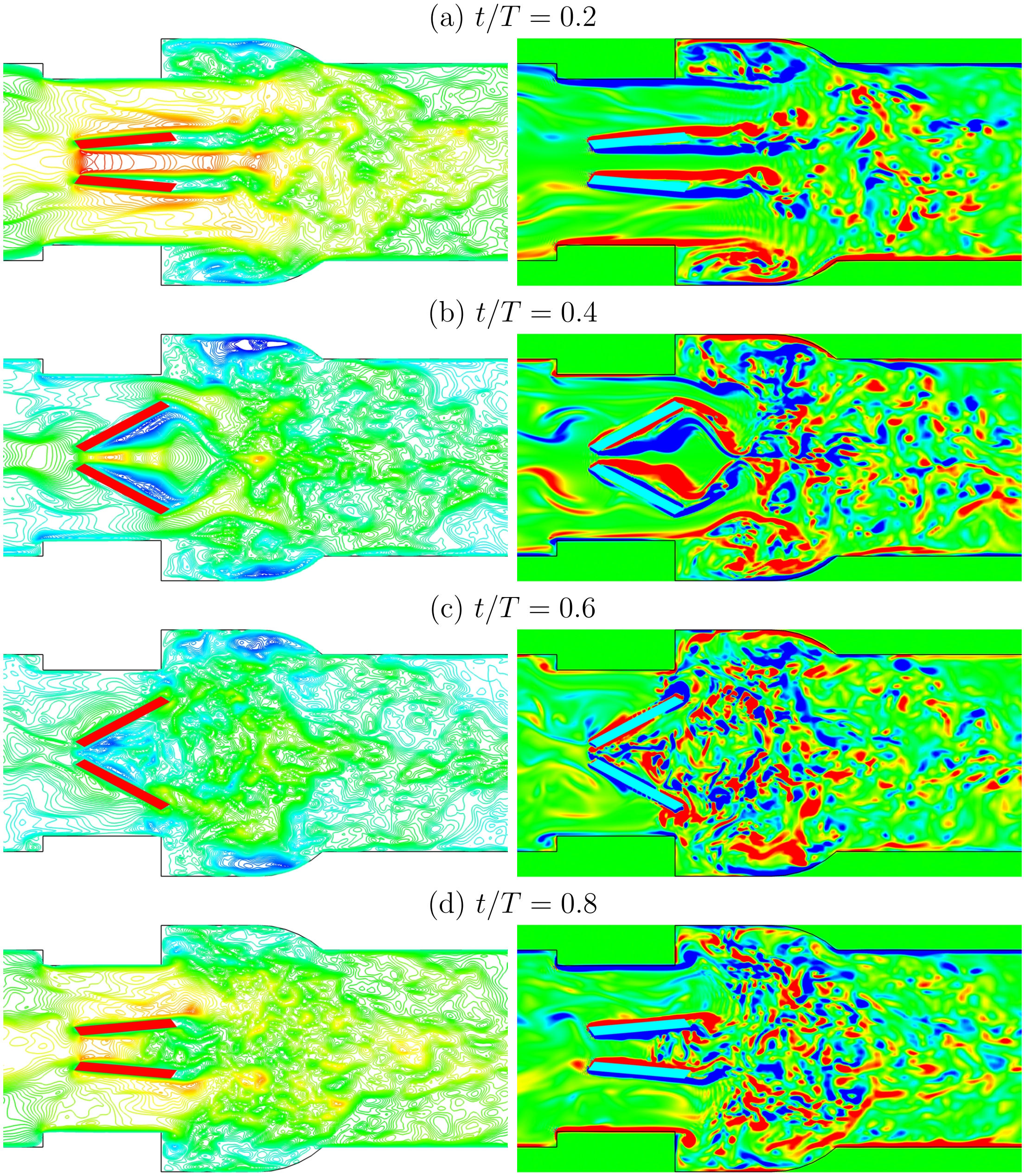


Figure 5.34: Imposed flow rate and leaflet kinematics for the heart valve problem: (a) Bulk velocity at the inlet; (b) opening angle of the leaflets, as functions of time.

$0.005D$. At the inflow boundary, located $4D$ upstream of the valve, fully developed flow corresponding to the flowrate variation shown in Fig. 5.34 is imposed. At the outflow boundary ($8D$ downstream of the valve) a convective boundary condition is used. The total number of nodes in this computation is 16.8 million.

The flow in the proximal area of the leaflets is very complicated, and it is dominated by intricate vortex-leaflet and vortex-vortex interactions. To illuminate basic flow patterns and demonstrate that the present method can accurately capture the thin shear layers that form on both moving and stationary immersed boundaries, instantaneous contours of the streamwise velocity and spanwise vorticity are shown in Fig. 5.35 for a few characteristic phase angles during the pulsatile cycle. When the leaflets are fully open, a strong jet is formed from the central orifice which interacts with the wake of the leaflets as can be seen from the velocity and vorticity contours



(a) $t/T = 0.2$

(b) $t/T = 0.4$

(c) $t/T = 0.6$

(d) $t/T = 0.8$

Figure 5.35: Instantaneous flow fields at an $x - z$ plane cutting through the middle of the leaflets. Left: streamwise velocity ($-1.5 < u/U_b^{max} < 3$, intervals are 0.05); Right: spanwise vorticity ($-20 < \omega_y D/U_b^{max} < 20$, intervals are 0.2). The phase reference, t/T, is from Fig. 5.34.

in Fig. 5.35a. Two shear layers are generated and roll-up into large vortices which interact with other vortices inside the chamber. A similar flow pattern has been observed downstream of a bi-leaflet valve in a recent highly resolved PIV experiment where a much more detailed model of the left ventricle was used [47]. There are several other locales where weaker shear layers are present: at the edge of the leaflet housing and outer surfaces of the leaflets, for example. In general, at this stage in the pulsatile cycle the flow has similar characteristics with what has been observed in steady experiments with the leaflets at the fully open position [8].

As the flow rate is decreasing and the leaflets begin to move towards the fully closed position (see Fig. 5.35b), the central orifice shrinks and the jet becomes thinner. The interaction of this jet and the ambient decelerating flow generates stronger shear layers than before. The flow rate keeps decreasing until it reaches zero, and the leaflets reach their maximum closing angle. The central jet during this stage becomes weaker and is finally dissipated as there is no flow through the central orifice. When the leaflets begin to open again and the flow rate starts to increase, the ring vortex at the edge of leaflet housing begins its development (see Fig. 5.35c). The flow accelerates until it reaches the maximum flow rate, and the leaflets reach their fully open position again as shown in Fig. 5.35d.

To better illuminated the highly three-dimensional nature of the complex vortex interactions just downstream of the leaflets, isosurfaces of Q are shown in Fig 5.36. In (a) and (b) of Fig. 5.36, two consecutive instantaneous realizations ($t/T = 0.3$ and $t/T = 0.4$) correspond to instances in the process of the closing of the leaflets. the motion of the leaflets generate strong vortices (structures B) at

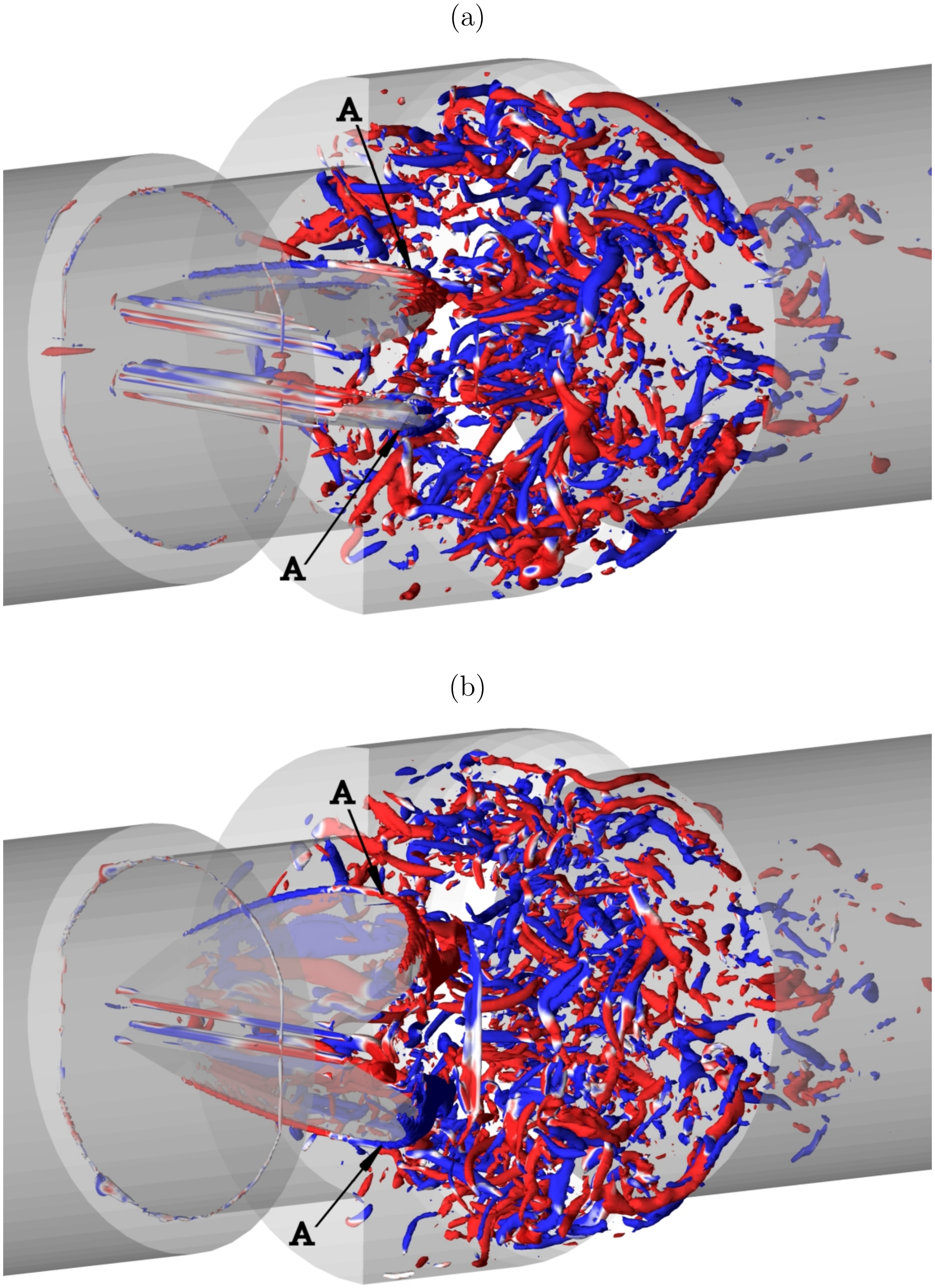


Figure 5.36: (a) and (b). For caption see next page.

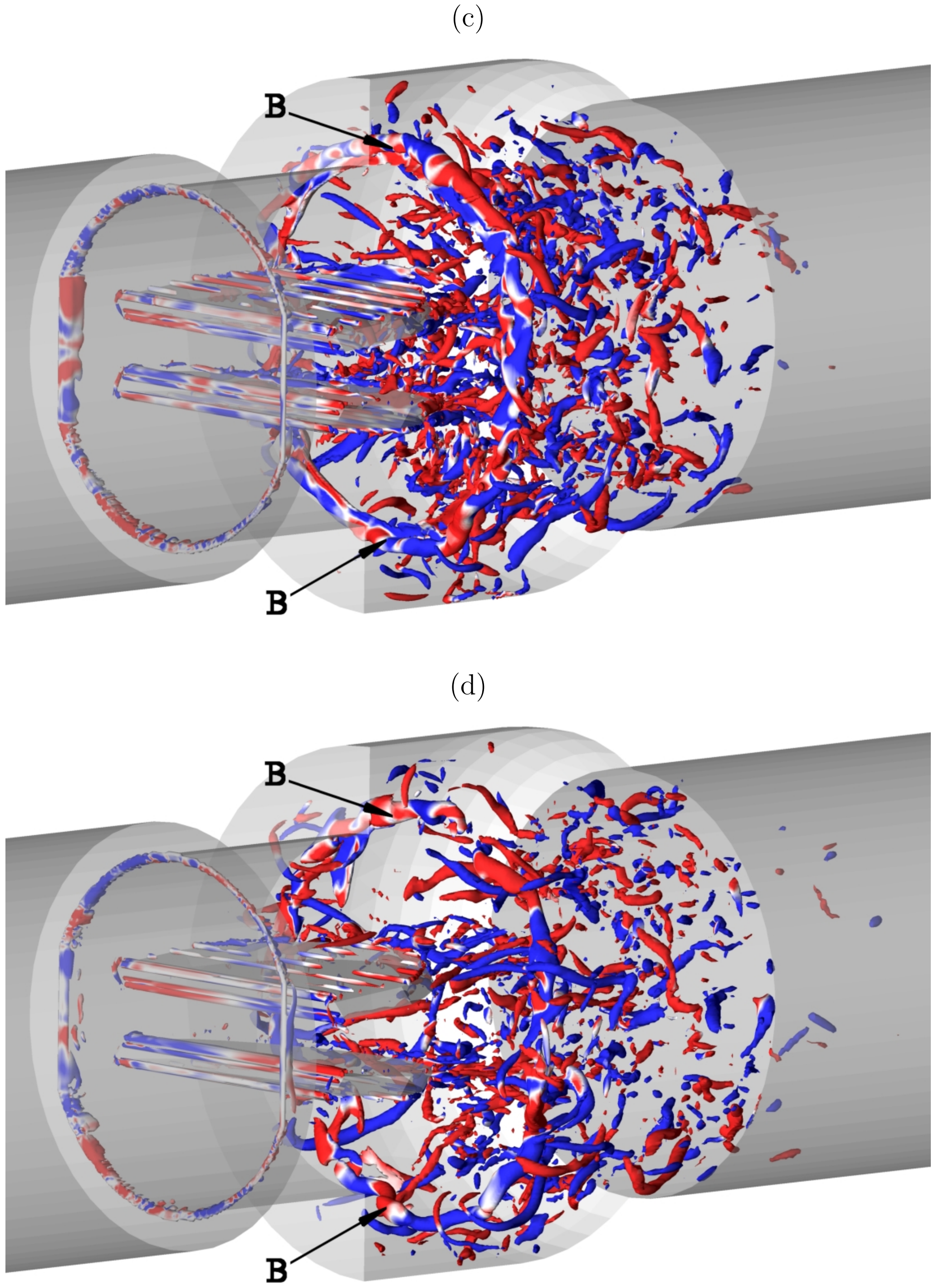


Figure 5.36: Instantaneous vortical structures visualized using isosurfaces of $Q = 600$ colored by the streamwise vorticity at: (a) $t/T = 0.3$; (b) $t/T = 0.4$; (c) $t/T = 0.8$; (d) $t/T = 0.9$. The phase reference, t/T, is from Fig. 5.34.

their tips. On the other hand, after the leaflets return to their fully open position, as the flow rate increases a ring vortex is generated at the edge of the leaflet housing Fig 5.36c ($t/T = 0.8$). At a later time (Fig. 5.36d, $t/T = 0.9$) this ring vortex is shed from the edge and interact with the surrounding flow, which again gives rise to complex vortex-vortex interactions.

Chapter 6

Conclusions and Future Directions

The objective of the present study is the development of a highly accurate, cost-effective strategy for the numerical simulations of a wide range of challenging problems in biological flows and other fields. Brief conclusions and further possible developement of current work are given in the chapter.

6.1 Conclusions

First, the original Cartesian solver [3] is extended to a generalized basic solver for the Navier-Stokes equations in both Cartesian and cylindrical coordinates. The singularity at the centerline is addressed and a simple approach is adopted. To remove the severe timestep constraint in the azimuthal direction near the centerline, the diffusive terms in the azimuthal direction in the momentum equations are treated implicitly. The solver is validated through a LES of fully developed turbulent pipe flow. Good agreement with the reference DNS data has been obtained.

Despite the increased attention that a variety of immersed boundary methods have been receiving in recent years, systematic studies examining their accuracy and applicability in cases of moving boundaries have not been reported. In the present study we initially identified problems that are unique to these formulations, and are a consequence of the boundary motion on a fixed grid. In particular, we have shown that in a typical fractional-step scheme , when a point near the interface (forcing point) moves into the flow (fluid point), carries with it unphysical information related to the pressure and velocity derivatives. As a result, the evaluation of the RHS of the momentum equations at this splitting cycle is problematic. To address this issue we proposed a robust field-extension procedure, where the velocity and pressure fields are "extended" in the solid phase at the end of each substep. This way the pressure and velocity derivatives at the problematic points have values that reflect the proper boundary conditions. The overall extension procedure, although it is tailored to our reconstruction scheme that utilizes the well defined normal to the interface [4], is applicable in a straightforward manner to any embedded-boundary formulation that reconstructs the solution around points in the fluid phase [15]. In cases where some type of a generalized ghost-cell approach is used [29, 61], then the grid points emerging from the solid need to be treated in a manner similar to that suggested, for example, in [63].

There are a variety of fluid-structure interaction problems that arise in engineering, biology and medicine. The numerical simulations of these problems require the solution of the Navier-Stokes equations for the fluid flows in a changing domain and the coupled dynamical equations for the structures that may be undergoing large deformation and/or displacement. The coupling between the flow and the structure leads to a highly nonlinear system, which presents a great challenge to numerical algorithms on accuracy, robustness, and efficiency. We have proposed a strong coupling scheme, where the fluid and the structure are treated as elements of a single dynamical system, and all of the governing equations are integrated simultaneously and interactively in the time domain. This algorithm utilizes an iterative predictor-corrector scheme that accounts for the interaction between the hydrodynamic loads and the motion of the structure.

In terms of validating the proposed methodologies, a variety of problems with increasing complexity were considered. First, the formal accuracy of the method (2^{nd} order) was established for the problem of an array of cylinders moving in a plane channel. Then, computations for two cases of laminar flow interacting with moving boundaries were conducted. The flow induced by the harmonic in-line oscillation of a circular cylinder in a quiescent fluid, and the flow from a transversely oscillating cylinder in a free-stream. Both cases are well documented in the literature, and experimental and numerical results are available for comparison. The proposed method was found to reproduce all features of the flow, including the detailed force distribution on the body with the same degree of accuracy as the reference boundary-conforming computations. In addition, vortex-induced vibrations of an elastic cylinder with one and two degrees of freedom in a free-stream are simulated. The results are compared with reference experiments and simulations. Good agreement was observed.

Several three-dimensional cases have been considered. Flows involving stationary bodies were considered. The laminar flow past a sphere was examined carefully and the results are in excellent agreement with the well-resolved DNS using a body-fitted grid. Transitional flows past a stationary sphere and an airfoil were simulated. Then, computations of the flow over a traveling wavy wall provided a validation of the method in LES of turbulent flows with moving boundaries. Finally LES of transitional flow around a prosthetic heart valve with moving leaflets demonstrated the robustness and applicability of the method in turbulent/transitional flows with multiple moving boundaries. Despite the simplification on the geometry the obtained results revealed flow patterns that have been reported in the literature.

6.2 Future Directions

In general the results in this study show that embedded-boundary formulations can be cost-efficient strategies especially for moving boundary problems without sacrificing accuracy. A limiting factor in all these methods, however, is the inflexibility in clustering grid points near a complex body. As the Reynolds number increases and more points need to be clustered near the body to resolve the thinner boundary layers, this can result in very large grids. To investigate if they are still cost efficient compared to boundary conforming formulations, one needs to look at the rate of increase of the total number of grid points for both categories of methods and the cost per node for each method as a function of the Reynolds number. This is a fairly difficult task since it depends on a variety of problem specific parameters. In general for the case of a single body, when using boundary-fitted grids, an increase in the Reynolds number requires refinement of the mesh in the wall-normal direction only; in non-boundary-conforming formulations like the present one, however, to achieve refinement normal to the body the grid usually needs to be refined in two or three directions. As a result the required number of grid points as a function of the Reynolds number increases faster for non-boundary-conforming formulations compared to boundary conforming ones. Actually, for laminar boundary layers it can be shown that the ratio of the boundary-conforming to the non-boundary-conforming grid size increases approximately as $Re^{1.5}$ [37]. However, given that methods like the present one are substantially less expensive per node than boundary-conforming ones (especially in case of boundaries undergoing large motions and/or deformations where grid re-generation is a major obstacle that compromises the accuracy and efficiency of boundary fitted approaches) they can be cost efficient for a fairly wide range of Reynolds numbers. The computation of the flow around the heart valve, for example, has been performed on a Linux desktop workstation equipped with four AMD Opteron 846 2GHz processors (each with 1M cache, and 8GB memory in total) and takes about 6 CPU hours for one pulsatile cycle. A 10 cycle simulation with 17 million points can be completed in less than 3 days. For a variety of problems, especially from biology where only low/moderate Reynolds numbers are encountered, methods such the present one have a lot of advantages.

In the near future, there are three major directions in the further extension of the current work: a) description of the geometries/evolution of immersed interfaces; b) dynamical grid adaptation; and c) introduction of new physics phenomena and models.

a) The immersed interfaces can be two and/or three dimensional; their motion can be prescribed or governed by the interaction between different phases separated by the interfaces; the interfaces can collapse or break up. For the description of evolving interfaces, both explicit and implicit methods can be applied, while their appropriateness in the framework of our method and in the context of the physical problems studied needs further investigation.

b) As the immersed body/interface interacting with the flow field, the location of interface becomes unpredictable and the dynamical grid adaptation will be imperative to a successful simulation. One key issue here is to extend the applicability of our embedded boundary method and at the same time to maintain the efficiency of Cartesian solvers.

c) Complex geometries and moving interfaces are encountered in many fields where fluid flow plays a major role. Some can be solved readily in the current framework with the addition of new boundary conditions at the interface, such as free surface and interfacial flows, etc. However, new physical models are required for most situations, e.g. fluid-structure interaction problems in biomimetic and physiological flows, reacting flows, and aero-acoustics among others.

BIBLIOGRAPHY

[1] K. Akselvoll and P. Moin, “Large-eddy simulation of turbulent confined coannular jets and turbulent flow over a backword-facing step.” *Report TF-63, Thermosciences Div.*, Dept. Mech. Eng., Stanford University, 1995.

[2] P. Anagnostopoulos and P.W. Bearman, “Response characteristics of a vortex-excited cylinder at low Reynolds numbers.” *J. Fluids Struct.*, 6:39–50, 1992.

[3] E. Balaras, “Large eddy simulation of high Reynolds number wall bounded shear layers.” *Ph.D. Dissertation*, Swiss Federal Institute of Technology, 1995.

[4] E. Balaras, “Modeling complex boundaries using an external force field on fixed Cartesian grids in large-eddy simulations.” *Comput. Fluids*, 33:375–404, 2004.

[5] E. Balaras, U. Piomelli, and J.M. Wallace, “Self-similar states in turbulent mixing layers.” *J. Fluid Mech.*, 446:1–24, 2001.

[6] E. Balaras and J. Yang, “Nonboundary conforming methods for large-eddy simulations of biological flows.” *ASME J. Fluids Eng.*, 127:851-857, 2005.

[7] S.A. Bayyuk, K.G. Powell, and B. van Leer, “A simulation technique for 2-D unsteady inviscid flows around arbitrary moving and deforming bodies of arbitrary geometry.” *AIAA Paper* 93-3391-CP, 1993.

[8] P. Browne, A. Ramuzat, R. Saxena, and A.P. Yganathan, “Experimental investigation of the steady flow downstream of the St. Jude bileaflet heart valve: a

comparison between LDV and PIV techniques." *Annals Biomed. Eng.*, 28:39–47, 2000.

[9] B. Carnahan, H.A. Luther, and J.O. Wilkes, *Applied Numerical Methods*, John Wiley & Sons, Inc., New York, 1969.

[10] R. Clift, J.R. Grace, and M.E. Weber, *Bubbles, Drops and Particles*, Academic Press, San Diego, CA, 1978.

[11] G.S. Constantinescu and S.K. Lele, "A highly Accurate Technique for the Treatment of Flow Equations at the Polar Axis in Cylindrical Coordinates Using Series Expansions." *J. Comput. Phys.*, 183:165–186, 2002.

[12] J.W. Deardorff, "A numerical study of three-dimensional turbulent channel flow at alrge Reynolds numbers." *J. Fluid Mech.*. 41:453–480, 1970.

[13] H. Dütsch, F. Durst, S. Becker, and H. Lienhart, "Low-Reynolds-number flow around an oscillating circular cylinder at low Keulegan-Carpenter numbers." *J. Fluid Mech.*. 360:249–271, 1998.

[14] J.G. Eggels, F. Unger, M.H. Weiss, J. Westerweel, R.J. Adrian, R. Friedrich, and F.T.M. Nieuwstadt, "Fully developed turbulent pipe flow: a comparison between direct numerical simulation and experiment." *J. Fluid Mech.*. 268:175–209, 1994

[15] E.A. Fadlun, R. Verzicco, P. Orlandi, and J. Mohd-Yusof, "Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations." *J. Comput. Phys.*, 161:35–60, 2000.

[16] K. Fukagata, and N. Kasagi, “Highly Energy-Conservative Finite Difference Method for the Cylindrical Coordinate System.” *J. Comput. Phys.*, 181:478–498, 2002.

[17] M. Germano, U. Piomelli, P. Moin, and W.H. Cabot, “A dynamic subgrid-scale eddy viscosity model.” *Phys. Fluids A*, 3:1760–1765, 1991.

[18] M. Germano, “Turbulence – The filtering approach.” *J. Fluid Mech.*, 238: 325–336, 1992.

[19] D. Goldstein, R. Handler, and L. Sirovich, “Modeling a no-slip flow boundary with an external force field.” *J. Comput. Phys.*, 105:354–366, 1993.

[20] D. Goldstein, R. Handler, and L. Sirovich, “Direct numerical simulation of turbulent flow over a modeled riblet covered surface.” *J. Fluid Mech.*, 302:333–, 1995.

[21] D. Goldstein, and T.C. Tuan, “Secondary flow induced by riblets.” *J. Fluid Mech.*, 363:115–, 1998.

[22] W. Gu, C. Chyu, and D. Rockwell, “Timing of vortex formation from an oscillating cylinder.” *Phys. Fluids*, 6:3677–3682, 1994.

[23] E. Guilmineau, and P. Queutey, “A Numerical simulation of vortex shedding from an oscillating circular cylinder.” *J. Fluids Struct.*, 16:773–794, 2002.

[24] H. H. Hu, N. A. Patankar, and M. Y. Zhu. "Direct numerical simulations of fluid-solid systems using the arbitrary Largrangian-Eulerian Technique." *J. Comput. Phys.*, 169:427–462, 2001.

[25] J.C.R. Hunt, A.A. Wray, and P. Moin, "Eddies, Streams and Convergence Zones in Turbulent Flows." , *Proceedings of the Summer Program 1988*, Center for Turbulence Research, 193, 1988.

[26] G. Iaccarino, and R. Verzicco, "Immersed boundary technique for turbulent flow simulations." *Applied Mechanics Reviews*, 56:331–347, 2003.

[27] N. Jauvtis, and C.H.K. Williamson, "The effect of two degrees of freedom on vortex-induced vibration at low mass and amping." *J. Fluid Mech.*, 509:23–62, 2004.

[28] T. A. Johnson and V. C. Patel, "Flow past a sphere up to a Reynolds number of 300." *J. Fluid Mech.*, 378:353–385, 1999.

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

[30] M.C. Lai and C.S. Peskin, "An immersed boundary method with formal second-order accuracy and reduced numerical viscosity." *J. Comput. Phys.*, 160:705–719, 2000.

[31] L. Li, S.J. Sherwin, and P.W. Bearman. "A moving frame of reference algorithm for fluid/structure interaction of rotating and translating bodies." *Int. J. Numer. Meth. Fluids*, 38:187–206, 2002.

[32] D.K. Lilly. "The representation of small-scale turbulence in numerical simulation experiments." *Proceddings of the IBM Scientific Computing Symposium on Environmental Sciences*. Yorktown Heights, N.Y., 1967.

[33] D.K. Lilly, "A proposed modification of the Germano subgrid-scale closure method." *Phys. Fluids A*, 4:633– , 1992

[34] T.S. Lund, "On the use of discrete filters for large eddy simulation." *Annual Research Briefs, 1997*, Center for Turbulence Research, 83–95, 1997.

[35] S. Majumdar, G. Iaccarino, and P. Durbin, "RANS solvers with adaptive structured boundary non-conforming grids." *Annual Research Briefs, 2001*, Center for Turbulence Research, 353–366, 2001.

[36] C. Meneveau, C.S. Lund, and W.H. Cabot, "A Lagrangian dynamic subgrid-scale model of turbulence." *J. Fluid Mech.*, 319:353–385, 1996.

[37] R. Mittal and J. Iaccarino, "Immersed boundary methods." *Annu. Rev. Fluid Mech.*, 37:239–261, 2005.

[38] Y. Morinishi, O.V. Vasilyev, and T. Ogi, "Fully conservative finite difference scheme in cylindrical coordinates for incompressible flow simulations." *J. Comput. Phys.*, 197:686–710, 2004.

[39] A.H. Nayfeh, S. Emam, S. Preidikman, and D.T. Mook. "An Exact Solution for the Natural Frequencies of Flexible Beams Undergoing Overall Motion." *J. Vibration and Control*, 9:1221–1229, 2003.

[40] D.J. Newman and G.E. Karniadakis. "A direct numerical simulation study of flow past a freely vibrating cable." *J. Fluid Mech.*, 344:95–136, 1997.

[41] I. Orlanski. "Simple boundary-condition for unbounded hyperbolic flows." *J. Comput. Phys.*, 21:251–269, 1976.

[42] S. Osher and R. Fedkiw. *Level set methods and dynamic implicit surfaces.* Springer, New York, 2003.

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

[44] C.S. Peskin and D.M. McQueen, "Numerical analysis of blood flow in the heart." *J. Comput. Phys.*, 25:220–252, 1980.

[45] C.S. Peskin and D.M. McQueen, "Cardiac fluid dynamics." *Critical Reviews in Biomedical Engineering*, 20:451–459, 1992.

[46] C.D. Pierce, "Progress-variable approach for large eddy simulation of turbulent combustion." *Ph.D. Dissertation*, Stanford University, 2001.

[47] O. Pierrakos, P.P. Vlachos, and D.P. Telionis, "Time-resolved DPIV analysis of vortex dynamics in a left ventricular model through bileaflet mechanical and porcine heart valve prostheses." *J. Biomech. Eng.*, 126:714–726, 2004.

[48] U. Piomelli, E. Balaras, and A. Pascarelli, “Turbulent structures in accelerating boundary layers.” *J. Turbulence*, 1:1–19, 2000.

[49] P. Ploumhans, G.S. Winckelmans, J.K. Salmon, A. Leonard, and M.S. Warren, “Vortex methods for direct numerical simulation of three-dimensional bluff body flows: Application to the sphere at Re=300, 500, and 1000.” *J. Comput. Phys.*, 178:427–463, 2002.

[50] S. Preidikman and D.T. Mook. “Time-Domain Simulations of Linear and Non-Linear Aeroelastic Behavior.” *J. Vibration and Control*, 6:1135–1176, 2000.

[51] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in Fortran 77: The Art of Scientific Computing.* Cambridge University Press, N.Y., 1992.

[52] J.J. Quirk, “An alternative to unstructured grids for computing gas dynamic flows around arbitrary two-dimensional bodies.” *Comput. Fluids* 23:125–, 1994.

[53] K.W. Schulz and Y. Kallinderis. “Unsteady flow structure interaction for incompressible flows using deformable hybrid grids.” *J. Comput. Phys.*, 143:569–597, 1998.

[54] L. Shen, X. Zhang, D.K.P. Yue, and M.S. Triantafyllou. “Turbulent flow over a flexible wall undergoing a streamwise travelling wave motion.” *J. Fluid Mech.*, 484:197–221, 2003.

[55] D. Shiels, A. Leonard, and A. Roshko. “Flow-induced vibration of a circular cylinder at limiting structural parameters.” *J. Fluids Struct.*, 15:3–21, 2001.

[56] J. Smagorinsky. “General circulation experiments with the primitive equations. I. The basic experiment.” *Monthly Weather Review*, 91:99–164, 1963.

[57] P.N. Swarztrauber, “Direct method for the discrete solution of separable elliptic equations.” *SIAM J. Num. Anal.*, 11:1136–, 1974.

[58] P.N. Swarztrauber, “Fast Fourier Transforms Algorithms for Vector Computers.” *Parallel Computing.* 45–63, 1984.

[59] F. Tessicini, G. Iaccarino, M. Fatica, M. Wang, and R. Verzicco, “Wall modeling for large-eddy simulation using an immersed boundary method.” *Annual Research Briefs 2002*, Center for Turbulence Research, 181–378, 2002.

[60] A.G. Tomboulides and S.A. Orszag, “Numerical investigation of transitional and weak turbulent flow past a sphere.” *J. Fluid Mech.*, 416:45–73, 2000.

[61] Y.-H. Tseng and J.H. Ferziger, “A ghost-cell immersed boundary method for flow in complex geometry.” *J. Comput. Phys.*, 192:593–623, 2003.

[62] H.S. Udaykumar, R. Mittal, and W. Shyy, “Computation of solid-liquid phase fronts in the sharp interface limit on fixed grids.” *J. Comput. Phys.*, 153:535–574, 1999.

[63] H.S. Udaykumar, R. Mittal, P. Rampunggoon, and A. Khanna, “A sharp interface Cartesian grid method for simulating flows with complex moving boundaries.” *J. Comput. Phys.*, 174:345–380, 2001.

[64] H.S. Udaykumar, W. Shyy, and M.M. Rao, “Elafint: A mixed Eulerian-Lagrangian method for fluid flows with complex and moving boundaries.” *Int. J. Numer. Methods Fluids* 22:691–, 1996.

[65] R. Verzicco, J. Mohd-Yusof, P. Orlandi, and D. Haworth, “Large-eddy simulation in complex geometric configurations using body forces.” *AIAA J.*, 38:427–, 2000.

[66] R. Verzicco and P. Orlandi, “A finite-difference scheme for three-dimensional incompressible flows in cylindrical coordinates.” *J. Comput. Phys.*, 123:402–414. 1996.

[67] C.H.K. Williamson and R. Govardhan. “Vortex-induced vibrations.” *Annu. Rev. Fluid Mech.*, **36**, 413–455, 2004.

[68] C.H.K. Williamson and A. Roshko. “Vortex formation in the wake of an oscillating cylinder.” *J. Fluids Struct.*, 2:355–381, 1988.

[69] C.T. Yamamoto, J.R. Meneghini, F. Saltara, R.A. Fregonesi, and J.A. Ferrari Jr., “Numerical simulations of vortex-induced vibration on flexible cylinders.” *J. Fluids Struct.*, 19:467–489, 2004.

[70] J. Yang and E. Balaras. “An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries.” *J. Comput. Phys.*, 2006.

[71] T. Ye, R. Mittal, H.S. Udaykumar, and W. Shyy, "An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries." *J. Comput. Phys.*, 156:209–240, 1999.

[72] A.P. Yoganathan, Z. He, and C.S. Jones, "Fluid mechanics of heart valves." *Annu. Rev. Biomed. Eng.*, 6:331–362, 2004.

[73] D. De Zeeuw, and K.G. Powell, "An adaptively refined Cartesian mesh solver for the Euler equations." *AIAA Paper* 91-1542-CP, 1991.

[74] C.Y. Zhou, R.M.C. So, and K. Lam. "Vortex-induced vibrations of an elastic circular cylinder." *J. Fluids Struct.*, 13:165–189, 1999.