# Chapter 3 Numerical method

To fully resolve the flow field around moving particles and to make a direct coupling between the particles and the fluid motion, an immersed boundary method computer code for axisymmetric flow is developed in cylindrical coordinates based on a fast immersed boundary projection method.

# 3.1 Immersed boundary method

The immersed boundary method, a popular numerical method for simulating incompressible flow with fluid-structure interaction, was introduced by Peskin in 1970's (see Peskin, 1972). In the present study, the immersed boundary method is used because it is good at handling a moving or deforming body with complex surface geometry. In this method, the flow field is solved on an Eulerian coordinate system that does not need to conform to the solid boundaries. Typically, a Cartesian mesh is used. The solid boundaries are represented by a set of Lagrangian points which can move freely through the fixed Cartesian mesh. Over those Lagrangian points, appropriate forces are applied to enforce the non-slip boundary condition. The forces exist in the continuous Navier-Stokes equations as singular functions, and convolutions with Dirac delta functions are used to exchange information between the Eulerian grid and the Lagrangian points. The computational domain with Eulerian grid discretization and immersed boundary formulation is shown in figure (3.1).

The immersed boundary method was first used to simulate blood flow patterns around heart valves by considering the solid surfaces as flexible elastic membranes with a constitutive rela-



Figure 3.1: Two-dimensional computational domain  $\mathscr{D}$  with Eulerian grid and immersed boundary formulation,  $\mathscr{B}$ , for a body. Lagrangian points,  $\boldsymbol{\xi}_k$ , are shown along  $\partial \mathscr{B}$  with filled dots where boundary forces,  $\mathbf{f}_k$ , are applied.

tion (see Peskin, 1972); Hooke's law relates the forces to the motion of the Lagrangian points. Beyer & LeVeque (1992) and Lai & Peskin (2000) extended it to rigid surfaces by taking the spring constant to be a large value. However, the above techniques need an appropriate choice of gain (stiffness), which has an obvious influence on the numerical results. A large value of gain in a constitutive relation limits the choice of the time step and a small gain leads to slip error at the solid surface. Higher order extensions are developed to get better results, see Mittal & Iaccarino (2005) for a general review.

# 3.2 Immersed boundary projection method

To circumvent the tuning parameters in immersed boundary method, an alternative is to consider the boundary forces as Lagrangian multipliers. Similar to the method in which the discretized pressure is treated as Lagrange multiplier to satisfy the continuity constraint (see Chang *et al.*, 2002), the values of the boundary forces are chosen to satisfy the non-slip constraint (see Glowinski *et al.*, 1998). Based on this idea, Taira & Colonius (2007) developed an immersed boundary projection method to solve the incompressible Navier-Stokes equations with boundary forces as given below:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \frac{1}{Re} \nabla^2 \mathbf{u} + \int_s \mathbf{f}(\boldsymbol{\xi}(s,t)) \delta(\boldsymbol{\xi} - \mathbf{x}) ds$$
(3.1)

$$\nabla \cdot \mathbf{u} = 0 \tag{3.2}$$

$$\mathbf{u}(\boldsymbol{\xi}(s,t)) = \int_{\mathbf{x}} \mathbf{u}(\mathbf{x}) \delta(\mathbf{x} - \boldsymbol{\xi}) d\mathbf{x} = \mathbf{u}_B(\boldsymbol{\xi}(s,t))$$
(3.3)

where  $\mathbf{u}$ , p and Re are the suitably non-dimensionalized velocity vector, pressure and the Reynolds number, respectively. Spatial variable  $\mathbf{x}$  represents position in the flow field,  $\mathscr{D}$ , and  $\boldsymbol{\xi}$  denotes coordinates along the immersed boundary,  $\partial \mathscr{B}$ , with a velocity of  $\mathbf{u}_B$ . The third term on the right hand side of equation (3.1) is the boundary force added at the Lagrangian points.

The equations are discretized with a staggered-mesh finite volume formulation. By using appropriate interpolation and regularization operators and certain time marching schemes for the viscous term and the convective terms, the equations can be written as a system of algebraic equations:

$$\begin{bmatrix} A & G & -H \\ D & 0 & 0 \\ E & 0 & 0 \end{bmatrix} \begin{pmatrix} q^{n+1} \\ \phi \\ f \end{pmatrix} = \begin{pmatrix} r^n \\ 0 \\ u_B^{n+1} \end{pmatrix} + \begin{pmatrix} bc_1 \\ bc_2 \\ 0 \end{pmatrix}$$
(3.4)

where  $q^{n+1}$  and  $\phi$  are the discretized velocity flux and pressure vectors;  $A = \frac{1}{\Delta t}M - \alpha_L L$  results from the implicit treatment of the vector term,  $\Delta t$  is the discrete time step, and M and L are the mass matrix and discrete Laplacian operator; G and D are the discrete gradient and divergence operators and are related as  $G = -D^T$ . The operators H and E are the regularization and interpolation operators resulting from the regularized Dirac delta function in equations (3.1) and (3.3). They are introduced to smear the singular boundary force over a few cells and interpolate velocity values defined on the staggered grid onto the Lagrangian points, respectively. Hf corresponds to the last term in equation (3.1) with  $f = (f_x, f_y)^T$ .  $Eq^{n+1} = u_B^{n+1}$  enforces the non-slip constraint. A superscript for time level is not placed on  $\phi$  and f because they are treated as Lagrange multipliers that do not play a direct role in time advancement but minimize the system energy and satisfy the kinematic constraints. After introducing a transformed force,  $\tilde{f}$ , the operators are related as  $Hf = -E^T \tilde{f}$ . The choice of E and H will be discussed in detail later in section 3.4. Here  $r^n$ ,  $bc_1$  and  $bc_2$  are the explicit terms in the momentum equation, the boundary condition vector resulting from the Laplacian operator, and the boundary condition vector generated from the divergence operator, respectively. Thus, the linear system (3.4) can be written as:

$$\begin{bmatrix} A & G & E^T \\ G^T & 0 & 0 \\ E & 0 & 0 \end{bmatrix} \begin{pmatrix} q^{n+1} \\ \phi \\ \tilde{f} \end{pmatrix} = \begin{pmatrix} r^n \\ 0 \\ u_B^{n+1} \end{pmatrix} + \begin{pmatrix} bc_1 \\ -bc_2 \\ 0 \end{pmatrix}$$
(3.5)

By grouping the pressure and the force variables as  $\lambda$  and reorganizing the sub-matrices in the following form:

$$Q \equiv [G, E^T], \quad \lambda \equiv \begin{pmatrix} \phi \\ \tilde{f} \end{pmatrix}, r_1 \equiv r^n + bc_1, \quad r_2 \equiv \begin{pmatrix} -bc_2 \\ u_B^{n+1} \end{pmatrix},$$

it becomes an algebraic system of equations that is recognized as a Karush-Kuhn-Tucker system:

$$\begin{bmatrix} A & Q \\ Q^T & 0 \end{bmatrix} \begin{pmatrix} q^{n+1} \\ \lambda \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}.$$
 (3.6)

A fractional step method can be used to solve the overall system. After performing an LU decomposition, equation (3.6) becomes:

$$\begin{bmatrix} A & 0 \\ Q^T & Q^T B^N Q \end{bmatrix} \begin{bmatrix} I & B^N Q \\ 0 & I \end{bmatrix} \begin{pmatrix} q^{n+1} \\ \lambda \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix} + \begin{pmatrix} -\frac{\Delta t^N}{2^N} (LM^{-1})^N Q \lambda^{n+1} \\ 0 \end{pmatrix}$$

where,  $B^N$  is the Nth order Taylor series expansion of  $A^{-1}$  and can be made symmetric and positive definite with appropriate choices of  $\Delta t$  and N. There is an Nth order splitting error in the second term of the right hand side which is the leading order error resulting from the truncation in  $B^N$  so a third-order expansion for  $B^N$  is required, see Taira & Colonius (2007).

The matrices can be recombined such that an intermediate velocity  $q^*$  is formed as

$$\left(\begin{array}{c} q^* \\ \lambda \end{array}\right) = \left[\begin{array}{c} I & B^N Q \\ 0 & I \end{array}\right] \left(\begin{array}{c} q^{n+1} \\ \lambda \end{array}\right)$$

Finally, three steps are performed to solve the whole system:

$$Aq^* = r_1 \tag{3.7}$$

$$Q^T B^N Q \lambda = Q^T q^* - r_2 \tag{3.8}$$

$$q^{n+1} = q^* - B^N Q\lambda \tag{3.9}$$

Equations (3.7) and (3.8) can be solved by using conjugate gradient method since the matrices in these equations are symmetric and positive definite. The third equation is the projection step resulting in the solution of the whole system. In this method, the divergence-free constraint and the non-slip boundary condition can be satisfied to an arbitrary accuracy implicitly at the next time level, and the CFL number is only limited by the choice of the time marching schemes of the viscous and convective terms. The method is second order accurate in time and better than first order accurate in space in  $L_2$  norm.

To apply the above method and the relative code (courtesy of Kunihiko Tiara) in simulating the evolution of the flow field during a wet collision process, the motion of the particle needs to be known prior to the simulation. As a first step, the motion of the sphere was described with experimental data. In the pendulum-wall collision experiments performed by Joseph *et al.* (2001), the trajectory of a sphere before and after colliding with a stationary wall in a glycerol-water mixture was recorded by a CCD camera. As an example, during a collision process at St = 186, the position of the sphere center can be plotted as functions of time, as shown in figure (3.2).

In the experiment record, the time interval between two neighbor points was dt = 0.001 s because the frame speed of the camera was 1000 fps. Based on the points near the wall, the impact and



Figure 3.2: Trajectory of a steel particle with diameter 7.39mm during the collision process in a mixture of glycerol and water with specific density 1.143, viscosity 7.305cp. The points are the positions of the center of the sphere at different time.

rebound velocities were obtained as shown in figure (3.3).

The collision position of the sphere center  $x_c$  is taken to be the intersection of the two lines fitting the trajectory before and after the collision. Because of the uncertainty in the location of the wall, the position of the wall was determined artificially. Leweke, Thompson & Hourigan (2004) took the minimum distance between the sphere and wall as 0.005D when they calculated the collision with Re  $\geq 1000$ . Both experimental and computational tests were conducted in Leweke *et al.* (2004) to verify that this had negligible influence on the predictions. For our cases with Re < 250, as a first try, the minimum distance between the sphere and the wall was taken as 0.02D. So, the wall position was taken as  $x_c + 0.5D + 0.02D$  in our cases. With linear interpolation, more points were inserted into the trajectory so that the time interval was decreased to dt = 0.25 ms. The simulation was performed for different cases with different Stokes numbers. For example, in one case with St = 186, the initial grid geometry is sketched in figure (3.4). Then, for every time step, the positions of the Lagrangian points on the particle surface are updated according to the experiment record. The collision happened at the 279th time step which corresponds to the maximum value of the position  $x_c$  in figure (3.3).

An accurate description of the evolvation of the flow field was obtained. Figure (3.5) presents the streamline configuration before and after the collision. The direction of the streamline around



Figure 3.3: Details of the trajectory close to the wall. "+" represents the recorded position. "." represents the interpolated point. The arrows give the impact and the rebound velocities before and after the collision. The intersection of the arrows is considered as the final collision point,  $x_c$ .



Figure 3.4: Initial grid geometry of the case with St = 186.



Figure 3.5: (a) Stream line distribution 0.25 ms before the collision. (b) Stream line distribution 0.25 ms after the collision.

the particle and close to the wall changes after the collision, resulting in a high shear stress in those regions. The distribution of the vorticity field before and after the collision is shown in figure (3.6).

The shear stress generated along the wall changes with time, as shown in figure (3.7). The variables are non-dimensionalized with the diameter of the sphere. As the solid body approaches the wall the amplitude of the shear stress along the wall increases and the position corresponding to the maximum shear stress moves toward the final contact point. After the collision, as the solid body moves away from the wall, the surrounding liquid reenters the growing gaps between the solid body and the wall. Thus, the shear stress changes to opposite direction and its magnitude decreases as the gap increases.

Because the simulation is two dimensional, it is impossible to quantitatively verify whether the simulated results are right. Experimentally, it is difficult to perform an experiment in which a two dimensional cylinder collides with a planar wall exactly parallel as it impacts the wall and rebounds. When trying to apply the numerical method to a three dimensional case, the limitation of this method makes it unfeasible. To solve equation (3.8), the conjugate gradient method requires a large number of iterations to make the convergence error small to satisfy the divergence-free and no-slip constraints. Thus, in a three dimensional case, a significantly longer time is required to yield a



Figure 3.6: Vorticity distribution of the flow field before and after the collision.



Figure 3.7: Shear stress distributions along the wall. The dash-dot and solid lines represent the shear stress distributions before and after the collision, respectively.

suitable solution.

# 3.3 Fast immersed boundary projection method

To improve the above numerical method, Colonius & Taira (2008) used a null space approach and a multi-domain technique to develop a fast immersed boundary projection method.

To eliminate the most computationally expensive component of the projection method, the pressure Poisson solver for equation (3.8), the nullspace approach is used by applying discrete curl operator, C, to the Navier-Stokes equations, such that

$$q = Cs, \quad \gamma = C^T q.$$

where q, s and  $\gamma$  are the discrete velocity flux vector, streamfunction and circulation. Here, C is a constructed null space for the discrete divergence operator. Thus:

$$DC \equiv 0$$

which corresponds to the relation  $\nabla \cdot \nabla \times \equiv 0$ . Hence,

$$Dq^{n+1} = DCs^{n+1} \equiv 0$$

which means the divergence-free constraint can be automatically satisfied.

When using a uniform Cartesian mesh and simply boundary conditions, the discrete Laplacian can be diagonalized by a sine transform. Thus, after applying the implicit trapezoidal rule on the viscous term with  $\alpha_L = \frac{1}{2}$  and second-order Adam-Bashforth method on the convective term, the transformed system of equations has only one linear system that need to be solved. This system with a positive definite and symmetric left-hand side operator can be solved by conjugate gradient method. Moreover, by applying this curl operator to the momentum equation, the pressure variable is eliminated so that the dimension of equation (3.8) is dramatically decreased from  $N \times N$  to



Figure 3.8: Schematic of a 3-level multi-domain solution of the Poisson equation. The figure is taken from figure 5 in Colonius & Taira (2008).

 $N_f \times N_f$ , where N is the Eulerian grid size and  $N_f$  is the number of boundary forces which is two times the number of Lagrangian points in a two dimensional case.  $N_f \ll N$ . Many fewer iterations are required than the original modified Poisson equation. Thus, this approach saves significant computation time.

However, applying uniform grid over the whole computational domain with simple boundary conditions results in low computational efficiency if the grid size is fine everywhere as that for the region of interest. Thus, to achieve high efficiency with uniform grid and simultaneously get sufficient resolution at the region of interest, a multi-domain technique is employed, as shown in figure (3.8).

33

A series of computation domains are used. The solution in a larger domain with simplified far-field boundary conditions and coarser grid is interpolated along the inner boundary of a smaller domain and provides the boundary condition for the smaller domain with finer grid. Thus, the domain of interest is solved with high efficiency but still enough resolution and with boundary values known in a simple form. Colonius & Taira (2008) should be consulted for further details.

# **3.4** Modification for axisymmetric system

The purpose of this study is to understand the contact mechanism of an immersed collision process. A normal collision is good enough to capture the essence of the relevant physics. An axisymmetric flow field surrounding the sphere particle and the horizontal wall resulted from a normal collision is simpler to solve than the three-dimensional problem. To apply the fast immerse boundary projection method to an axisymmetric system, modifications are made to the Navier-Stokes equations and the algorithms to solve the discrete linear systems.

#### 3.4.1 Discrete Navier-Stokes equations for axisymmetric system

For an axisymmetric incompressible flow in cylindrical coordinates, the velocity  $\mathbf{u}(u_z, u_r, 0)$ , the vector potential  $\mathbf{B}(0, 0, \psi/r)$  and the vorticity  $\mathbf{w}(0, 0, \omega)$  are related as  $\mathbf{u} = \nabla \times \mathbf{B}$ ,  $\mathbf{w} = \nabla \times \mathbf{u}$ , and  $\mathbf{w} = -\nabla^2 \mathbf{B}$ . The relation between the scalar variables  $(u_z, u_r)$  and streamfunction  $\psi$  are:

$$u_z = \frac{1}{r} \frac{\partial \psi}{\partial r}, \quad u_r = -\frac{1}{r} \frac{\partial \psi}{\partial z}.$$

With those additional radial factors, the corresponding discrete variables have new relations:

$$q = \hat{R}^{-1} C R s \tag{3.10}$$

where,  $C = \begin{bmatrix} \frac{\partial}{\partial r} \\ -\frac{\partial}{\partial z} \end{bmatrix}$  is the discrete curl operator,  $R = \text{diag}(R_{i,j})$  is a diagonal matrix of the radial coordinates of the vertices of the cells where the discrete streamfunction s is defined and



Figure 3.9: Location of Eulerian variables on cylindrical coordinates and the definition of matrices  $\hat{R}$  and R.

$$\hat{R} = \begin{bmatrix} \operatorname{diag}(\hat{R}_{1i,j}) & 0\\ 0 & \operatorname{diag}(\hat{R}_{2i,j}) \end{bmatrix}$$
 is a diagonal matrix of the radial coordinates of the center points on the cell edges where the discrete velocity flux  $q$  is defined. Figure (3.9) shows the configuration of the Eulerian variables defined on cylindrical grids for an axisymmetric system  $(z,r)$  and the definition of matrices  $R$  and  $\hat{R}$ .

The relation between streamfunction  $\psi$  and vorticity  $\omega$  gives:

$$\omega = -\frac{\partial}{\partial r} \left( \frac{1}{r} \frac{\partial \psi}{\partial r} \right) - \frac{\partial^2}{\partial z^2} \left( \frac{\psi}{r} \right).$$

Then, discretely, the circulation  $\gamma$  corresponding to  $\omega$  is related to s as:

$$\gamma = -C^T \hat{R}^{-1} CRs. \tag{3.11}$$

So that

$$s = -(C^T \hat{R}^{-1} C R)^{-1} \gamma.$$
(3.12)

since  $C^T \hat{R}^{-1} CR$  is a square matrix and can be inverted. Thus, combining equation (3.10) and (3.12) yields:

$$q = \hat{R}^{-1} C R (C^T \hat{R}^{-1} C R)^{-1} \gamma, \qquad (3.13)$$

36

Now, considering the Navier-Stokes equation of momentum with the immersed boundary forces:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \frac{1}{Re} \nabla^2 \mathbf{u} + \int_s \mathbf{f}(\boldsymbol{\xi}(s,t)) \delta(\boldsymbol{\xi} - \mathbf{x}) ds$$

with the relation

$$\mathbf{u} \cdot \nabla \mathbf{u} = \nabla (\frac{1}{2}\mathbf{u}^2) - \mathbf{u} \times (\nabla \times \mathbf{u})$$

and

$$\nabla^2 \mathbf{u} = \nabla (\nabla \cdot \mathbf{u}) - \nabla \times (\nabla \times \mathbf{u}) = -\nabla \times (\nabla \times \mathbf{u})$$

which incorporates incompressible condition  $\nabla \cdot \mathbf{u} = 0$ , the momentum equation can be rewritten as:

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla (p + \frac{1}{2}\mathbf{u}^2) = \mathbf{u} \times \mathbf{w} - \frac{1}{Re} \nabla \times \mathbf{w} + \int_s \mathbf{f}(\boldsymbol{\xi}(s, t)) \delta(\boldsymbol{\xi} - \mathbf{x}) ds$$

After applying a curl operator, it becomes:

$$\frac{\partial \mathbf{w}}{\partial t} = \nabla \times (\mathbf{u} \times \mathbf{w}) - \frac{1}{Re} \nabla \times (\nabla \times \mathbf{w}) + \nabla \times \int_{s} \mathbf{f}(\boldsymbol{\xi}(s,t)) \delta(\boldsymbol{\xi} - \mathbf{x}) ds$$

Then, the semi-discrete circulation form of the Navier-Stokes equations is:

$$\frac{d\gamma}{dt} = C^T N(q) - \beta C^T \hat{R}^{-1} C R \gamma + C^T H f + b c_{\gamma}.$$
(3.14)

where  $C^T$  is the discrete curl operating on the variables defined at the centers of cell edges, C is the discrete curl operating on the variables defined at the vertices of cells,  $\beta$  is a constant equal to  $1/(Re\Delta z^2)$ , where  $\Delta z$  is the uniform grid spacing and  $\Delta r = \Delta z$ , and  $bc_{\gamma}$  is the inhomogeneous term depending on the particular boundary conditions. N(q) denotes discretized non-linear term ( $\mathbf{u} \times \mathbf{w}$ ), which is calculated as:

$$\begin{vmatrix} \hat{e_z} & \hat{e_r} & \hat{e_\theta} \\ u_z & u_r & u_\theta \\ 0 & 0 & \omega \end{vmatrix} = u_r \omega \hat{e_z} - u_z \omega \hat{e_r}$$

Because the velocity variables are defined on the center of a cell edge and the vorticity is defined at the vertices of a cell, as shown in figure (3.9), then, in z direction:

$$(u_r\omega)_{i,j} = \frac{1}{2}(\bar{u}_{r(i,j+1)}\omega_{i,j+1} + \bar{u}_{r(i,j)}\omega_{i,j})$$

in r direction:

$$(u_z\omega)_{i,j} = \frac{1}{2}(\bar{u}_{z(i+1,j)}\omega_{i+1,j} + \bar{u}_{z(i,j)}\omega_{i,j})$$

where i and j are the indices for grids in z and r direction, respectively. The bar values are the averages at the center of an edge calculated as:

$$\bar{u}_{z(i,j)} = \frac{1}{2}(u_{z(i,j)} + u_{z(i,j-1)}),$$

and

$$\bar{u}_{r(i,j)} = \frac{1}{2}(u_{r(i,j)} + u_{r(i-1,j)})$$

from the values at the vertices. Thus, the nonlinear term finally is still defined at the vertices of cells.

The regularization operator, H, is related to the delta function and its form will be given as following together with the interpolation operator E. For the delta function, in two dimensional problems, a discrete form that is proposed by Roma *et al.* (1999) is used in Taira & Colonius (2007). A similar form is used here for an axisymmetric system in cylindrical coordinates:

$$d(r) = \begin{cases} \frac{1}{6\Delta r} \left[ 5 - 3\frac{|r|}{\Delta r} - \sqrt{-3(1 - \frac{|r|}{\Delta r})^2 + 1} \right] & \text{for } 0.5\Delta r \le |r| \le 1.5\Delta r; \\ \frac{1}{3\Delta r} \left[ 1 + \sqrt{-3\left(\frac{r}{\Delta r}\right)^2 + 1} \right] & \text{for } |r| \le 0.5\Delta r; \\ 0 & \text{otherwise.} \end{cases}$$
(3.15)

where  $\Delta r$  is the cell width of the staggered grid in the r-direction. This discrete delta function is specifically designed for use on staggered grids. In this form, the delta function is supported over only three cells which is an advantage for computational efficiency. Taira & Colonius (2007) compared the results for the current formulation with other discrete functions and did not find significant differences.

Then, for a two-dimensional axisymmetric case, discretizing the convolution of u and  $\delta$  in equation (3.3) yields

$$u_k = \Delta z \Delta r \sum_i u_i d(z_i - \xi_k) d(r_i - \eta_k)$$
(3.16)

where  $u_i$  is the discrete velocity vector defined on the staggered grid  $(z_i, r_i)$  and  $u_k$  is the discrete boundary velocity at the kth Lagrangian point $(\xi_k, \eta_k)$ . Using  $\alpha$  to denote the factor preceding the summation, the interpolation operator in equation (3.16) is defined as

$$\hat{E}_{k,i} = \alpha d(z_i - \xi_k) d(r_i - \eta_k).$$
(3.17)

Letting  $E \equiv \hat{E}R^{-1}$  for use of the velocity flux,  $q^{n+1} = Ru^{n+1}$ , the non-slip boundary condition can be represented as

$$\hat{E}_{k,i}u_i^{n+1} = E_{k,i}q_i^{n+1} = u_{B_k}^{n+1} \tag{3.18}$$

The regularization operator, H, is a discrete version of the convolution operator in equation (3.1) that passes information from the Lagrangian points,  $\boldsymbol{\xi}_k$ , to the neighboring Eulerian grid points,  $\boldsymbol{x}_i$ . In a similar manner as E, H is defined as

$$H_{i,k} = \beta \hat{M}_i d(z_i - \xi_k) d(r_i - \eta_k) = \frac{\beta}{\alpha} \hat{M}_i \hat{E}_{k,i}^T, \qquad (3.19)$$

where  $\beta$  is the numerical integration factor proportional to the interval between two neighboring Lagrangian points. The diagonal matrix  $\hat{M}$  is used for consistency with the fractional step formulation.

To relate the regularization operator with the interpolation operator in a similar manner as

mentioned in section 3.2, a transformed forcing function  $\tilde{f}$  is introduced so that

$$Hf = -E^T \tilde{f}.$$

Thus, the original boundary force can be retrieved by  $f = -(EH)^{-1}EE^T \tilde{f}$ . Because of the uniform grid with  $\Delta z = \Delta r$ , the relation simplifies to  $f = -\frac{1}{\Delta z^2} \frac{\alpha}{\beta} \tilde{f}$ .

So, the final semi-discrete momentum equation in circulation form, equation (3.14) can be written as:

$$\frac{d\gamma}{dt} + C^T E^T \tilde{f} = C^T N(q) - \beta C^T \hat{R}^{-1} C R \gamma + b c_{\gamma}.$$
(3.20)

After applying the second order Adam-Bashforth method to the nonlinear convective term:

$$C^T N(q) \rightarrow C^T \left[ \frac{3}{2} N(q)^n - \frac{1}{2} N(q)^{n-1} \right]$$

and implicit trapezoidal rule to the viscous term:

$$\beta C^T \hat{R}^{-1} C R \gamma \to \beta C^T \hat{R}^{-1} C R \frac{1}{2} (\gamma^{n+1} + \gamma^n),$$

the fully discrete system is

$$(I + \frac{\Delta t}{2}L)\gamma^{n+1} + \Delta t C^T E^T \tilde{f} = (I - \frac{\Delta t}{2}L)\gamma^n - \Delta t C^T \left[\frac{3}{2}N(q)^n - \frac{1}{2}N(q)^{n-1}\right] + \Delta t b c_\gamma \qquad (3.21)$$

where  $L = \beta C^T \hat{R}^{-1} C R$  is the discrete Laplacian operator for an axisymmetric system in cylindrical coordinates.

Now consider the boundary condition for the Navier-Stokes equations:

$$\int_{\mathbf{x}} \mathbf{u}(\mathbf{x}) \delta(\mathbf{x} - \boldsymbol{\xi}) d\mathbf{x} = \mathbf{u}_B(\boldsymbol{\xi}(s, t))$$

Using the relation (3.13), the discrete form is written as:

$$Eq^{n+1} = E\hat{R}^{-1}CR(C^T\hat{R}^{-1}CR)^{-1}\gamma^{n+1} = u_B^{n+1}$$
(3.22)

After combining the equation (3.21) and (3.22), the linear system for axisymmetric case is:

$$\begin{pmatrix} I + \frac{\Delta t}{2}L & \Delta t C^T E^T \\ E\hat{R}^{-1}CR(C^T\hat{R}^{-1}CR)^{-1} & 0 \end{pmatrix} \begin{pmatrix} \gamma^{n+1} \\ \tilde{f} \end{pmatrix} = \begin{pmatrix} rhs \\ u_B^{n+1} \end{pmatrix}$$
(3.23)

where rhs is a simplified denotation for the right hand side of equation (3.21).

Following the idea of the fast immersed boundary projection method and applying LU decomposition to the coefficient matrix yields:

$$\begin{pmatrix} I + \frac{\Delta t}{2}L & \Delta tC^T E^T \\ EQ & 0 \end{pmatrix} = \begin{pmatrix} I + \frac{\Delta t}{2}L & 0 \\ EQ & -EQ(I + \frac{\Delta t}{2}L)^{-1}\Delta tC^T E^T \\ 0 & I \end{pmatrix} \begin{pmatrix} I & (I + \frac{\Delta t}{2}L)^{-1}\Delta tC^T E^T \\ 0 & I \end{pmatrix}$$

where  $Q = \hat{R}^{-1} C R (C^T \hat{R}^{-1} C R)^{-1}$  for simplification. After defining the intermediate variable:

$$\begin{pmatrix} \gamma^* \\ \tilde{f} \end{pmatrix} = \begin{pmatrix} I & (I + \frac{\Delta t}{2}L)^{-1} \Delta t C^T E^T \\ 0 & I \end{pmatrix} \begin{pmatrix} \gamma^{n+1} \\ \tilde{f} \end{pmatrix}, \qquad (3.24)$$

the three steps to solve the linear system for axisymmetric case are:

$$(I + \frac{\Delta t}{2}L)\gamma^* = rhs, \qquad (3.25)$$

$$EQ(I + \frac{\Delta t}{2}L)^{-1}\Delta tC^T E^T \tilde{f} = EQ\gamma^* - u_B^{n+1}, \qquad (3.26)$$

$$\gamma^{n+1} = \gamma^* - (I + \frac{\Delta t}{2}L)^{-1}C^T E^T \tilde{f}.$$
(3.27)

To solve the above system, consider sine transform which is denoted as:

$$\hat{\gamma} = S\gamma \quad \leftrightarrow \quad \gamma = S\hat{\gamma},$$

where the circumflex denotes the Fourier coefficients.

In the current simulation, the grid is uniform and Dirichlet boundary condition is used: the velocity normal to the boundary is zero and the vorticity is zero. It is appropriate at the axis of symmetry, r = 0, since  $u_r \mid_{r=0} \equiv 0$  and  $\omega = \frac{\partial u_r}{\partial z} - \frac{\partial u_z}{\partial r} \equiv 0$  resulting from  $\frac{\partial}{\partial r} \equiv 0$ . For the other three boundaries, the error caused by using such kind of simplified far-field boundary conditions in smaller computational domains is minimized by applying a multi-domain technique as mentioned in section 3.3, see Colonius & Taira (2008) for a detailed description and discussion of the technique.

Hence, first type of sine transform in Fortran fast fourier transform library which is odd around k = -1 and odd around k = n is used. The relation of the sine transform pair means that the sine transform can be normalized so that it is identical to its inverse,  $S^{-1} = S$ . So, SS = I.

For a Poisson equation:

$$Lx = b$$

after applying a sine transform, it becomes:

$$SLx = Sb$$

It can be written as:

$$SLSSx = Sb$$

Let  $T \equiv SLS$ ,  $\hat{x} = Sx$ , and  $\hat{b} = Sb$ , then

$$T\hat{x} = \hat{b}$$

Here, T for an axisymmetric system in cylindrical coordinates can be reordered to become a tri-

diagonal matrix

$$\begin{pmatrix}
b_2 & c_2 & & 0 \\
a_3 & b_3 & c_3 \\
& a_4 & b_4 & \ddots \\
& & \ddots & \ddots & c_{N-1} \\
0 & & a_N & b_N
\end{pmatrix}$$

with

$$a_{j} = \frac{j-2}{j-3/2}, \quad j = 3, 4, \cdots, N$$
  

$$b_{j} = -2 + 2\cos\left(\frac{i-1}{M}\pi\right) - \frac{j-1}{j-1/2} - \frac{j-1}{j-3/2}, \quad j = 2, 3, \cdots, N$$
(3.28)

$$c_j = \frac{j}{j - 1/2}, \quad j = 2, 3, \cdots, N - 1$$
 (3.29)

where N and M are the grid size in r direction and z direction, respectively.

Thus, the Poisson equation and the first step (3.25) with this tri-diagonal system can be solved by a routine code.

But, the resulting matrix in the second step (3.26) is not symmetric; hence, the conjugate gradient method cannot be used. Although general minimal residual method(GMRES) method can be applied to solve this non-symmetric system, a large number of iterations are required and the numerical method is not efficient as before.

#### 3.4.2 Non-slip boundary condition for axisymmetric system

A modification of the representational form for the non-slip boundary condition is proposed for solving an axisymmetric system based on the fast immersed boundary projection method. Considering the axisymmetric flow surrounding a moving sphere and a wall in cylindrical coordinates, an additional radial weight factor is inserted into equation (3.3) making new boundary conditions for



Figure 3.10: The configuration of the cylindrical coordinates for the axisymmetric flow surrounding a moving sphere and a wall.

an axisymmetric system in cylindrical coordinates as:

$$r\boldsymbol{u}(\boldsymbol{\xi}(s,t)) = \int_{\boldsymbol{x}} r\boldsymbol{u}(\boldsymbol{x})\delta(\boldsymbol{x}-\boldsymbol{\xi})d\boldsymbol{x} = r_B\boldsymbol{u}_B(\boldsymbol{\xi}(s,t)), \qquad (3.30)$$

where  $r_B$  is defined as shown in figure (3.10).

Thus, the discrete form given in equation (3.22) is changed to:

$$E\hat{R}q^{n+1} = ECR(C^T\hat{R}^{-1}CR)^{-1}\gamma^{n+1} = r_B u_B^{n+1};$$
(3.31)

And the corresponding system of algebraic equations is:

$$\begin{pmatrix} (I + \frac{\Delta t}{2}L) & C^T E^T \\ ECR(C^T \hat{R}^{-1} CR)^{-1} & 0 \end{pmatrix} \begin{pmatrix} \gamma^{n+1} \\ \tilde{f} \end{pmatrix} = \begin{pmatrix} rhs \\ r_B u_B^{n+1} \end{pmatrix}$$
(3.32)

After applying the projection approach to decompose the matrix, the coefficient matrix becomes:

$$\begin{pmatrix} (I + \frac{\Delta t}{2}L) & 0 \\ ECR(C^T \hat{R}^{-1}CR)^{-1} & -ECR(C^T \hat{R}^{-1}CR)^{-1}(I + \frac{\Delta t}{2}L)^{-1}C^T E^T \end{pmatrix} \begin{pmatrix} I & (I + \frac{\Delta t}{2}L)^{-1}C^T E^T \\ 0 & I \end{pmatrix}$$
(3.33)

Using the definition for the intermediate variable  $\gamma^*$  given in equation (3.24), the new three steps to solve the system are:

$$\left(\left(I + \frac{\Delta t}{2}L\right)\gamma^* = rhs,\right) \tag{3.34}$$

$$ECR(C^{T}\hat{R}^{-1}CR)^{-1}(I + \frac{\Delta t}{2}L)^{-1}C^{T}E^{T}\tilde{f} = ECR(C^{T}\hat{R}^{-1}CR)^{-1}\gamma^{*} - r_{B}u_{B}^{n+1}\Delta z, \qquad (3.35)$$

$$\gamma^{n+1} = \gamma^* - (I + \frac{\Delta t}{2}L)^{-1}C^T E^T \tilde{f}.$$
(3.36)

The first equation (3.34) is exactly the same as before and can be solved with a routine solver for a tri-diagonal matrix. The current second step, equation (3.35), can be solved with conjugate gradient method since its coefficient matrix is symmetric positive definite as proven below:

$$(C^{T}\hat{R}^{-1}CR)^{-1}(I + \frac{\Delta t}{2}L)^{-1} = [(I + \frac{\Delta t}{2}L)(C^{T}\hat{R}^{-1}CR)]^{-1}$$
  
$$= [C^{T}\hat{R}^{-1}CR + \frac{\Delta t}{2}C^{T}\hat{R}^{-1}CRC^{T}\hat{R}^{-1}CR]^{-1}$$
  
$$= R^{-1}[C^{T}\hat{R}^{-1}C + \frac{\Delta t}{2}C^{T}\hat{R}^{-1}CRC^{T}\hat{R}^{-1}C]^{-1}$$
(3.37)

R and  $\hat{R}$  are positive diagonal matrices, thus  $\left(\hat{R}^{-1}\right)^T=\hat{R}^{-1}.$  Then

$$\left(C^T \hat{R}^{-1} C\right)^T = C^T \hat{R}^{-1} C$$

So,  $C^T \hat{R}^{-1} C R C^T \hat{R}^{-1} C$  must be symmetric positive definite. Thus, the coefficient matrix of (3.35)

must be symmetric and positive definite.

The whole axisymmetric flow field can be solved accurately with high efficiency if the solid boundary velocity is known.

# 3.5 Verification of the modification for axisymmetric system

To validate the cylindrical coordinate modifications, the code was run to solve different problems with and without immersed boundaries in the flow field. The simulated results are compared with theoretical results and the results reported by other researchers.



Figure 3.11: The known vorticity field.

#### 3.5.1 Verification of the solution for the Poisson-like equation

The first step in the resulting linear system, equation (3.34), is a Poisson-like equation with the Laplacian operator in cylindrical coordinates for axisymmetric system. To verify the solution of this Poisson-like equation, the code was run to solve a streamfunction from a Poisson equation with a known vorticity field. The equation to be solved is:

$$-\nabla^2 \psi = \omega$$

with an arbitrary picked vorticity field:

$$\omega = -8r(1+63r^2+8r^4+8r^2z^2-48zr^2)\exp\left(-4[r^2+(z-3)^2]\right).$$

The theoretical result of the streamfunction can be calculated analytically as:

$$\psi = r^3 \exp\left(-4[r^2 + (z-3)^2]\right).$$

The distribution of the vorticity field is shown in figure (3.11). The numerical result of the streamfunction compares favorably with the analytical result as shown in figure 3.12(b) and 3.12(a).

The computational domain is taken to be  $[0,6] \times [0,3]$ . The  $L^2$  norm of the difference between the numerical and analytical results is calculated and plotted as a function of the Eulerian grid size, N.



Figure 3.12: The streamfunction distribution.



Figure 3.13: The  $L^2$  norm of the difference between the numerical and analytical result of the vorticity field. The solid line is the  $L^2$  norm as a function of the grid size. The dashed line is  $N^{-2}$ .

Figure (3.13) shows that the second order accuracy is achieved in  $L^2$  norm.

Thus, the solution of the Poisson-like equation 3.34 obtained in the same manner can also be trusted with second order accuracy.

#### 3.5.2 Verification of the solution with immersed boundaries

Simulations of flow passing a stationary sphere of unit diameter are performed to verify the solution with immersed boundaries in the flow field. In an axisymmetric domain with initial uniform flow,  $U_{\infty}$ , a set of Lagrangian points along a half circle with its center at (z = 3, r = 0) is introduced to represent the sphere. There are two different configurations for setting the Lagrangian points, as shown in figure (3.14). In figure 3.14(a), there are Lagrangian points on the axisymmetric z axis, and in figure 3.14(b) there is no Lagrangian point on z axis. In those two configurations, the intervals between the Lagrangian points are identical and equal to a cell size in Eulerian grid,  $\Delta s \approx \Delta z = \Delta r$ . Simulations with these two different configurations for Reynolds number Re = 100are both performed and the results are compared in Table (3.1).

The simulated results of wake dimensions and drag coefficient at a steady state for different Reynolds numbers,  $Re = U_{\infty}D/\nu = 100,200$  are compared with the simulated results reported in



Figure 3.14: The configuration of the setting of Lagrangian points.



Figure 3.15: The characteristic dimensions of the wake structure. The parameters, l, a, and b represent the length of the recirculation zone, distance from the sphere to the center of the wake vortex, and the gap between the centers of the wake vortices, respectively.  $\theta$  is the separation angle.

Taira (2008) and Johnson & Patel (1999). The good agreement is shown in Table (3.1). The size of the wake is characterized by the characteristics lengths, l, a, b, and  $\theta$  (non-dimensionalized by the diameter) defined in figure (3.15). The grid resolution and the time step are set to  $\Delta z = \Delta r = 0.02$ and  $\Delta t = 0.005$ , the same as the setting in the other two reports.

According to our simulations, with or without Lagrangian points on the z axis does not make obvious difference. No Lagrangian point on the z axis is used in the following simulation unless otherwise stated.

The characteristics lengths of wake are close to the reported values, except that the separation angle,  $\theta$ , is smaller. This discrepancy comes from the fact that the Lagrangian multipliers adding to the boundary points disturb the development of the boundary layers and the disturbance leads

		1	а	b	$\theta$	$C_D$
Re = 100	with Lagrangian pts on axis	0.867	0.249	0.579	$47.6^{\circ}$	1.10
	w/o Lagrangian pts on axis	0.871	0.249	0.600	$46.8^{\circ}$	1.10
	Johnson & Patel (1999)	0.88	0.25	0.60	$53^{\circ}$	1.10
	Taira (2008)	0.91	0.28	0.59	$50^{\circ}$	1.14
Re = 200	w/o Lagrangian pts on axis	1.400	0.370	0.712	$51.0^{\circ}$	0.78
	Johnson & Patel (1999)	1.46	0.39	0.74	$63^{\circ}$	0.80
	Taira (2008)	1.38	0.37	0.69	$61^{\circ}$	0.82

Table 3.1: Comparison of the simulation results of steady-state wake dimensions and drag coefficient from flow over a sphere.

to a smaller separation angle.

The force exerted by the surrounding liquid on the solid body is calculated based on the boundary forces in the simulation. In equation (3.1), the boundary forces as Lagrangian multipliers are smeared by the discrete delta function to the neighboring Eulerian grids. The resulting body forces are integrated over the whole flow field as:

$$f_z = \sum_{i,j} f_{zi,j} \pi (r_j^2 - r_{j-1}^2) \Delta z$$
(3.38)

where  $f_z$  is the force in z direction. As for an axisymmetric flow, the r direction force is zero. Then the drag coefficient,  $C_D$  can be calculated. As shown in Table (3.1), the results obtained from the axisymmetric simulations with Reynolds number Re = 100,200 are found to be in accord with the values reported by other researchers.

Thus, the evolution of the flow field can be simulated accurately when the position and velocity of the immersed boundaries are prescribed. The force exerted by the liquid flow on the immersed body can be calculated from the simulation.

# 3.6 The evolution of the flow coupled to the motion of the particle

After knowing the effect of liquid force on a moving solid body, the evolution of the flow field and the motion of the sphere are coupled by writing Newton's law in the z direction for a falling sphere as shown in figure (3.10) as:

$$m_p \frac{d\tilde{V}}{d\tilde{t}} = \tilde{f}_{\rm SIM} + m_p g + \tilde{f}_b, \qquad (3.39)$$

where  $\tilde{V}$  is the velocity of the sphere which is considered to be a uniform value for  $\boldsymbol{u}_{\rm B}$  in equation (3.30) over the whole sphere surface;  $m_p$  is the mass of the sphere, g is the gravitational acceleration and  $\tilde{f}_{\rm SIM}$  is the liquid-solid interaction force in vertical direction calculated from the flow solution using equation (3.38);  $\tilde{f}_b$  is the vertical buoyancy force resulting from the hydrostatic pressure gradient which is not included in  $\tilde{f}_{\rm SIM}$ . Rotation of the sphere is not considered in this process.

The position and the velocity of the sphere are the boundary conditions for the Navier-Stokes equations. For the flow field surrounding a falling sphere with diameter D, the Navier-Stokes equations are non-dimensionalized with characteristic length  $L_o = D$  and time  $t_o = \sqrt{D/g}$  so that the non-dimensional parameter, Reynolds number, is defined as  $Re_D = \frac{D\sqrt{Dg}}{\nu}$  where  $\nu$  is the kinematic viscosity of the liquid. After introducing the density ratio,  $\tau = \frac{\rho_P}{\rho_l}$ , where  $\rho_p$  is the density of the particle and  $\rho_l$  is the density of the liquid, the equation of the motion can be non-dimensionalized as:

$$\frac{dV}{dt} = \frac{1}{\tau} (f_{\rm SIM} + \tau - 1)$$
(3.40)

where  $f_{\text{SIM}}$  is the non-dimensional force exerted by the surrounding fluid and can be calculated from  $f_z$ .

The coupled evolution of the flow field and the motion of the sphere are integrated in time using a second order Runge-Kutta method. Starting from *n*th time step with known variables,  $\gamma^n$ ,  $u_B^n$ ,  $x_B^n$  and  $f_z^n$ , the variables at the (n + 1)th time step come be found from the following process.

<sup>1.</sup> Calculate  $f_{\text{SIM}}$  from  $f_z$  at *n*th time step,

2. Solve equation (3.34) for  $\gamma^*$ :

$$(I + \frac{\Delta t}{2}L)\gamma^* = rhs(\gamma^n, x_B^n)$$

3. Calculate  $u_B^{n+\frac{1}{2}}$ ,  $x_B^{n+\frac{1}{2}}$ , and the corresponding operator  $E^{n+\frac{1}{2}}$  and  $E^{T^{n+\frac{1}{2}}}$  as:

$$u_B^{n+\frac{1}{2}} = u_B^n + \frac{\Delta t}{\tau} (f_{\text{SIM}} + \tau - 1)$$
$$x_B^{n+\frac{1}{2}} = x_B^n + \Delta t u_B^n$$

4. Solve equation (3.35) for  $\tilde{f}^*$  and calculate  $f_z^*$  and  $f_{\rm SIM}^*$ :

$$E^{n+\frac{1}{2}}CR(C^T\hat{R}^{-1}CR)^{-1}(I+\frac{\Delta t}{2}L)^{-1}C^TE^{T^{n+\frac{1}{2}}}\tilde{f}^* = E^{n+\frac{1}{2}}CR(C^T\hat{R}^{-1}CR)^{-1}\gamma^* - r_Bu_B^{n+\frac{1}{2}}$$

5. Calculate  $u_B^{n+1}$ ,  $x_B^{n+1}$ , and the corresponding operator  $E^{n+1}$  and  $E^{T^{n+1}}$ :

$$u_B^{n+1} = u_B^n + \frac{\Delta t}{\tau} \left(\frac{1}{2}(f_{\text{SIM}} + f_{\text{SIM}}^*) + \tau - 1\right)$$
$$x_B^{n+1} = x_B^n + \Delta t \frac{1}{2}(u_B^n + u_B^{n+\frac{1}{2}})$$

6. Solve equation (3.35) for  $\tilde{f}$  and calculate  $f_z$  at (n+1) th time step:

$$E^{n+1}CR(C^T\hat{R}^{-1}CR)^{-1}(I + \frac{\Delta t}{2}L)^{-1}C^TE^{T^{n+1}}\tilde{f} = E^{n+1}CR(C^T\hat{R}^{-1}CR)^{-1}\gamma^* - r_Bu_B^{n+1}$$

7. Solve equation (3.36) for  $\gamma^{n+1}$ 

$$\gamma^{n+1} = \gamma^* - (I + \frac{\Delta t}{2}L)^{-1}C^T E^T \tilde{f}.$$

Thus, starting from  $t = 0^+$  with an acceleration of the effect of the gravity and buoyancy force,

the particle moves automatically in the flow field. The kinetic behavior of the particle and also the evolution of the flow field introduced by the moving body can both be calculated from the simulation.

### 3.7 Validation of the coupled solution

To validate the coupled algorithm, the simulation results are compared with the experimental results by TenCate *et al.* (2002). In the experiment, a sphere settled toward the bottom of a vessel full of silicon oil. The trajectory and associated flow field are measured from the moment of release until rest at the bottom of the container vessel. The sphere with a diameter D = 15 mm was initially suspended at an initial distance 120 mm. The density of the sphere is  $\rho_p = 1.12 \times 10^3$  kg·m<sup>-3</sup> and the density of the liquid is  $\rho_l = 0.96 \times 10^3$  kg·m<sup>-3</sup>. The dynamic viscosity of the surrounding liquid is  $\mu_f = 0.058$  Pa·s. The Reynolds number based on the ultimate velocity  $V_{\infty}$  is Re = 31.9. Simulations with the above input parameters are performed.

The trajectory and the velocity profiles are compared in figure (3.16). The circles in the figure are digitized from figure 8 in TenCate *et al.* (2002). The comparison shows that the simulation coupling the evolution of the flow field and the motion of the sphere predicts the same motion of the sphere as found in the experiment. Note that the Stokes number at impact St = 4.1 is small and the particle does not rebound from the wall. In the current simulations, similar results are found with two different grid sizes supporting the convergence of the method.

Figure (3.17) shows the comparison of the spatial structure of the flow field at the moment when the sphere is a half sphere diameter from the wall. The velocity magnitude contours show good agreement between the experimental and the simulated flow field.

A time series of the fluid velocity at a particular point in the flow domain are compared in figure (3.18). This point is one diameter from the wall and one diameter from the center of the sphere. The radial and axial velocity components are given on the top and bottom in the figure, respectively. The evolution of the velocity components show that a vortex passes the point and the wake of the sphere follows after the sphere comes to rest. The good agreement indicates that the numerical method accurately captures not only the dynamic behavior of the sphere but also the evolution of



Figure 3.16: Comparison of (a) trajectory and (b) velocity profiles for the experiments of TenCate *et al.* (2002), et al. and the current simulations for a sphere falling on to a surface; the Stokes number based on the settling velocity is 4.1.



Figure 3.17: Comparison of (a) the PIV experiment data from TenCate *et al.* (2002) and (b) the simulated result of flow field when the sphere is a half sphere diameter from the wall. Contour indicates the normalized velocity magnitude.



Figure 3.18: Comparison of time series of the fluid velocity at a point located one diameter from the wall and one diameter from the center of the sphere. The data on the left are the experimental results and the lattice Boltzmann simulation from TenCate *et al.* (2002) and the right are the results from the current simulation.

the flow field.