V International Conference on Computational Methods for Coupled Problems in Science and Engineering
COUPLED PROBLEMS 2013
S. Idelsohn, M. Papadrakakis and B. Schrefler (Eds)

A COMPARATIVE STUDY OF IMMERSED-BOUNDARY INTERPOLATION METHODS FOR A FLOW AROUND A STATIONARY CYLINDER AT LOW REYNOLDS NUMBER

SEYED HOSSEIN MADANI*, JAN WISSINK, HAMID BAHAI

School of Engineering and Design - Brunel University West London
*Islamic Azad University – South Tehran Branch
email: Hossein.Madani@brunel.ac.uk

Key words: Immersed-Boundary, Reconstructive method, Interpolation methods, stationary cylinder, Vortex shedding

ABSTRACT

The accuracy and computational efficiency of various interpolation methods for the implementation of non grid-confirming boundaries is assessed. The aim of the research is to select an interpolation method that is both efficient and sufficiently accurate to be used in the simulation of vortex induced vibration of the flow around a deformable cylinder. Results are presented of an immersed boundary implementation in which the velocities near non-confirming boundaries were interpolated in the normal direction to the walls. The flow field is solved on a Cartesian grid using a finite volume method with a staggered variable arrangement. The Strouhal number and Drag coefficient for various cases are reported. The results show a good agreement with the literature. Also, the drag coefficient and Strouhal number results for five different interpolation methods were compared it was shown that for a stationary cylinder at low Reynolds number, the interpolation method could affect the drag coefficient by a maximum 2% and the Strouhal number by maximum of 3%. In addition, the bi-liner interpolation method took about 2% more computational time per vortex shedding cycle in companion to the other methods.

INTRODUCTION

Obtaining accurate solutions for Fluid-Structure Interaction (FSI) problems is of interest in many engineering and scientific applications. A broad classification of FSI methods is based on the type of mesh employed in the discretisation, where we can differentiate between boundary-conforming and non-boundary-conforming mesh methods [1]. A well-known conforming mesh method is the Arbitrarily Lagrangian-Eulerian method (ALE). For non-conforming mesh methods, usually an immersed boundary method is used and most recent developments in FSI methods are based on this approach. The latter is the subject of this review.

The immersed-boundary (IB) method is a technique for solving flow problems in regions with irregular boundaries using a simple structured grid solver. The term “immersed boundary method” was initially used for a method developed by Peskin [2] which was used to simulate blood flow in a cardiovascular system. It was specifically designed to handle deforming (elastic) boundaries interacting with low Reynolds number flow. The simulation was carried out on a Cartesian grid and at those locations where the boundary did not align with a mesh line the solution algorithm was locally modified to enforce the desired boundary conditions on the flow. More recently, numerous modifications and refinements have been proposed to enhance the accuracy, stability, and application range of the IB method [3].

Depending on the way that the boundary conditions are imposed on the immersed boundary, the IB methods can be generally categorized into continuous and discrete forcing approaches. In the continuous forcing method, a forcing function is applied to the Navier-Stokes equation in order to maintain the boundary condition on the structure (e.g. enforcing a no-slip boundary condition on a stationary body). The most important issue in this method is the definition of the continuous forcing function needed to enforce the correct boundary condition. Several different functions have been developed by Peskin [2], Saiki and Biringen [4], Beyer and Leveque [5], and Lai and Peskin [6], among others. In all cases, a distributed function was used rather than a sharp function because firstly the solid boundaries do not coincide with the Cartesian mesh and, secondly, this way the Gibbs' oscillations phenomenon [7] adjacent to the solid boundaries could be suppressed. Applying a continuous force to enforce the boundary conditions is attractive for elastic boundaries as it has a physical meaning and its implementation is relatively easy. However, the implementation of this method to enforce rigid boundaries is relatively cumbersome due to its definition. Another problem is that by using a smooth function the method cannot sharply represent the immersed boundaries which is not recommended for high Reynolds number flows [3].

Because the Navier-Stokes equations usually cannot be integrated analytically to find the forcing functions, it is usually not possible to derive an analytical forcing function to enforce specific boundary conditions. To tackle this problem, a method has been suggested by Mohd-Yusof [8] and Verzicco [9]. In this method, which is known as Indirect Discrete forcing approach, forcing functions are subtracted from the numerical solution after discretizing the Navier-Stokes equations. The important advantage of this method is that there is no need to define the forcing function parameters prior to solving the Navier-Stokes equations and there is no stability constraint due to using continuous forcing functions (Gibb's oscillation). However, it is still needed to implement the distributed forcing functions which strongly depend on the discretization algorithm. Another division of the discrete forcing approach is Direct Discrete Forcing.

Due to the need for and accurate representation of the boundary layer in high Reynolds number flow, the use of distributed, smooth forcing functions near the immersed boundary is not desirable. In these cases it is recommended to use a sharp interface with a higher local accuracy near the boundary. This goal can be achieved by imposing the boundary conditions directly on the immersed boundary. There are two well-known methods that fit into this category: the Ghost-Cell Finite-Difference Approach and the Cut-Cell Finite-Volume Approach.

In the Ghost-Cell approach the immersed boundary is implemented by the use of ghost cells. Ghost cells are cells inside the solid boundary which have at least one neighbour on the fluid side. The parameters (imaginary velocity and pressure) in the ghost cell (inside the solid) are defined by an interpolation method which implicitly enforces the correct boundary condition for the immersed boundary. Iaccarino and Verzicco [10] showed that a linear interpolation method is acceptable for those cases in which the first points of the interpolation in the fluid are inside the viscous sub layer. Other interpolation methods have been introduced by Ghias [11].

The entire immersed boundary methods discussed so far are not designed to consider the conservations laws near the solid boundary. However, the Cut-Cell method used in combination with a Finite-Volume approach is designed in order to preserve the conservation of momentum and mass near the boundary. In this method the cells, which have been cut by the immersed boundary, are reshaped or absorbed by neighbouring cells in order to form a new trapezoidal control volume cell shape. This method has been used by Mittal [12, 13] to simulate vortex-induced vibration around a stationary and a moving body and for free falling objects. Extending this method to 3D is not straightforward and needs complex polyhedral cells, which complicate the discretization of the Navier-Stokes equations.

As discussed earlier in the discrete forcing approach, the IB is imposed on the domain after the discretization of Navier-Stokes equations. This means that introducing the boundary conditions and forcing functions is not as straightforward as in the continuous forcing approach and depends on the discretization method and its implementation. Also, in discrete forcing approach the definition of the pressure on the boundary is not as straightforward as in the continuous forcing approach and requires special treatment. Advantages of the discrete forcing approach are that the boundary conditions can be introduced sharply without any extra stability constraint, while the fluid and solid domains are clearly separated and the equations that describe the flow are only solved in the fluid domain.

In this paper, we focus on the indirect discrete forcing approach, where the forcing function is not calculated directly and added to the Navier-Stokes equation. We are not intent to use any Cut-Cell or Ghost Cell methods as the applying the Cut-Cell method for fluid-structure interaction problems with moving boundaries takes lots of computational time [14, 15], while the Ghost-Cell approach will create non-physical results when solving the fluid equations in the solid domain.

In the next section, the formulation of the fluid dynamical problem which has been used in the simulation is introduced.

FORMULATION AND NUMERICAL METHODS

The governing equation for an unsteady, incompressible fluid flow in vector form is given by the Navier–Stokes equation which reads:

$$\rho\left(\frac{\partial \mathrm{V}}{\partial \mathrm{t}}+\mathrm{V}.\nabla \mathrm{V}\right)=-\nabla \mathrm{p}+\mu\nabla^2\mathrm{V}+\mathrm{f}, \qquad (1)$$

$$\nabla.\mathrm{V}=0 \qquad (2)$$

where f is the external force on the fluid domain which is used to implement the boundary condition on non-conforming solid boundaries. In this paper this force is not applied directly to the governing equations. Instead, the non-conforming boundary conditions are introduced by interpolating velocities close to the solid boundary.

The incompressible Navier-stokes equations in a 2D Cartesian domain are given by:

$$\frac{\partial \mathrm{u_i}}{\partial \mathrm{t}}+\frac{\partial \mathrm{u_i u_j}}{\partial \mathrm{x_j}}=-\frac{\partial \mathrm{p}}{\partial \mathrm{x_i}}+\frac{1}{\mathrm{Re}}\left(\frac{\partial^2 \mathrm{u_i}}{\partial \mathrm{x_j}\,\partial \mathrm{x_j}}\right) \qquad (3)$$

$$\frac{\partial u_i}{\partial x_i}=0 \qquad (4)$$

where p is the generalised pressure which is defined by the static pressure divided by the density. Hence, to obtain the correct the static pressure we need to multiply p by the density.

A staggered variable arrangement, as introduced by Harlow and Welsh [16], is used to discretize the governing equations on a Cartesian mesh. The continuity equation is enforced by taking the divergence of the momentum equation and using the continuity equation to simplify the results to form a Poisson equation for the pressure field. This equation is solved by strongly implicit procedure (SIP), Stone's method, at every time step [17].

To maintain a consistent implementation, the pressure equation is discretized in a similar way as the momentum equation.

The extent of the computational domain was selected to be relatively large to ensure that the location of the boundaries does not affect the simulation. For this reason the size of the y direction was taken to be 20D and the size of the x direction was taken to be 15D which was deemed to be sufficient to capture the vortex shedding behind the cylinder. The mesh size was chosen was dx=dy=0.05D which was checked in a mesh refinement study.

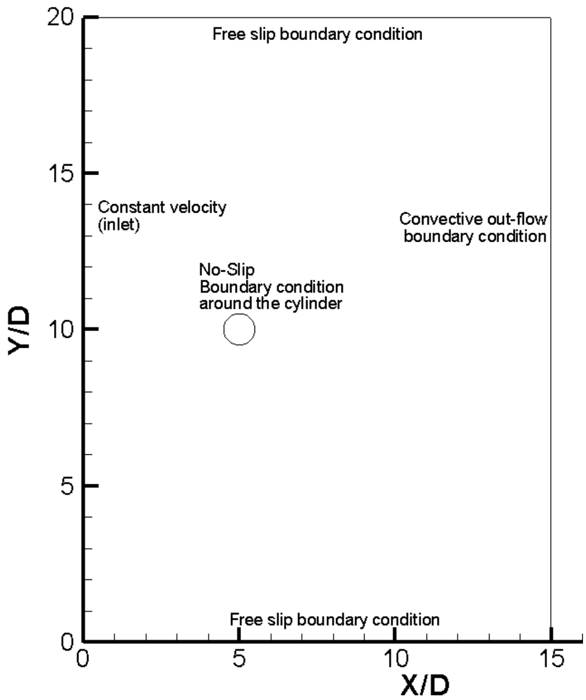

Figure 1: Fluid domain size and boundary conditions.

As the entire domain was meshed using a Cartesian grid, the implementation of the grid-conforming inlet, outlet and side boundary conditions was straightforward, while the boundary conditions along the cylinder were implemented using immersed boundary methods.

In the next section a number of velocity-interpolation methods, both from the literature as well as a novel second-order interpolation method for the implementation of the non-conforming boundary conditions in the fluid domain are introduced and compared. It will be shown that the results show a good agreement with the literature.

INTERPOLATION METHODS

In this section interpolation or reconstruction methods are compared. To enforce boundary conditions using the interpolation method, the forcing function f is not calculated directly. Instead, the flow velocity is interpolated at the interface cells and the forcing term is imposed indirectly to the discrete equations. The interface points are defined as the points in the fluid domain near the solid boundary for which one of the neighbouring points in the discretized equations is inside the solid domain. Therefore, the parameters related to these points cannot be updated through solving the governing equation (Figure 2). Any cells that contain one or more interface points are called interface cells. It is well known that most immersed boundary approaches need some sort of interpolation procedure. In the direct forcing approaches, interpolation is used in order to determine the forcing functions at the interface cells which enforce the correct boundary conditions to the governing equation. In the indirect forcing approach (interpolations approach), each time step the flow parameters in the interface cells are updated by direct interpolation and used as boundary condition for the flow solver. In this review, a number of interpolation procedures which could potentially be used in indirect discrete forcing approaches (interpolation or reconstruction approaches) will be compared.

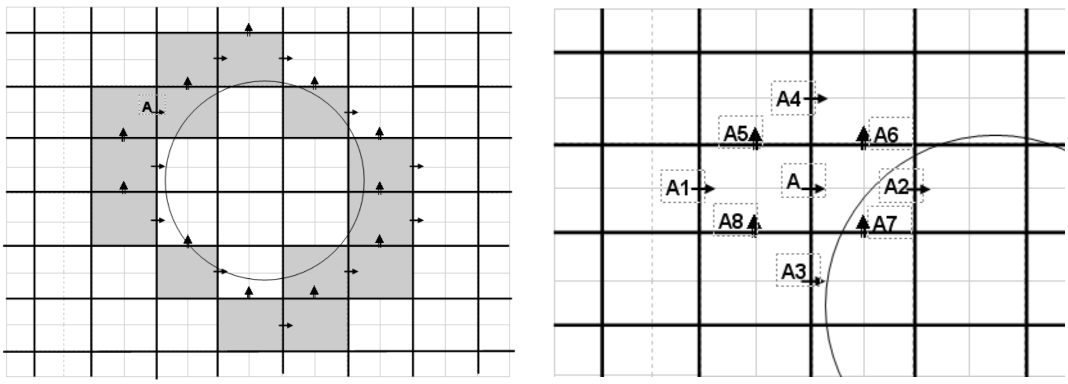

Figure 2: A 2D Cartesian mesh with a solid boundary (circle). Interface points, that require interpolation, are identified by arrows. Points A1 to A8 are all neighbouring points of A. Note that A2 and A7 are inside the solid domain.

Below it will be shown in detail how these interface cells have been treated and how possible problems that may occur near boundaries, like the decoupling of pressure and conservation of mass, may be overcome. To do so, first step is to define the interface cells for the specific geometry, which could be complicated for geometries with unknown functions [10]. Subsequently, it will be ensured that the flow governing equations are not solved inside any interface and solid cells. The most important step in the interpolation methods is to determine the flow parameters in the interface cells adjacent to the solid boundaries which will be used as boundary conditions for the rest of the flow domain that will subsequently be updated by the flow solver. Various interpolation methods have been developed to tackle this problem. In the following part, these methods are categorized and explained in more detail.

Case A: No interpolation

The simplest possible method is to select the interface cells at the solid boundaries and define the solid domain inside those cells. In fact, in this case there is no interpolation and the solid boundary will have a stepwise shape (Figure 3). Also, the boundary itself is somewhat diffused, as in the staggered methods the boundary conditions for the different velocity components are applied at different sides of an element. Fadlun [18] proposed a similar method for imposing forcing functions for immersed boundaries. Here, however, the method is applied directly to define the solid boundaries, while the governing equation will be subsequently solved in the remainder of the computational domain assuming no-slip boundary conditions for the solid boundaries. As interpolation is not needed, this method will be relatively fast while still giving acceptable results. The disadvantage of this method - when used in the calculation of flow around a circular cylinder - is that on course meshes shape and size differences between the cylinder and the solid boundary could affect lift and drag forces.

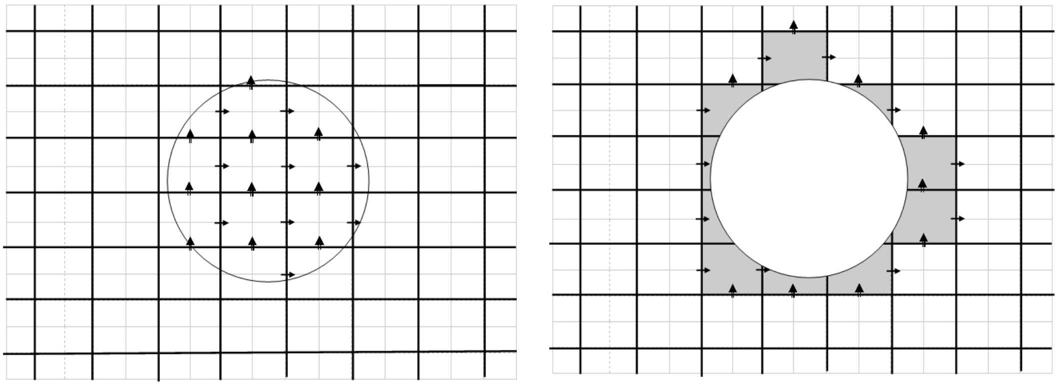

Figure 3: left, the arrows shows velocity inside solid body (solid velocity), assumed zero for a stationary cylinder. Right, hatched cells used to define weighting coefficient.

Case B: Weighting method

This method is similar to the one discussed above. The major difference is that the boundary values for the velocity in those cells that are part fluid and part solid are weighted accordingly. Figure 3 (right) shows the location of these weighted boundary velocities in the cells that are part fluid and part solid. For each of the velocity components a coefficient is determined that corresponds to the ratio of the fluid part of the two adjacent cells to the whole area of the two cells.

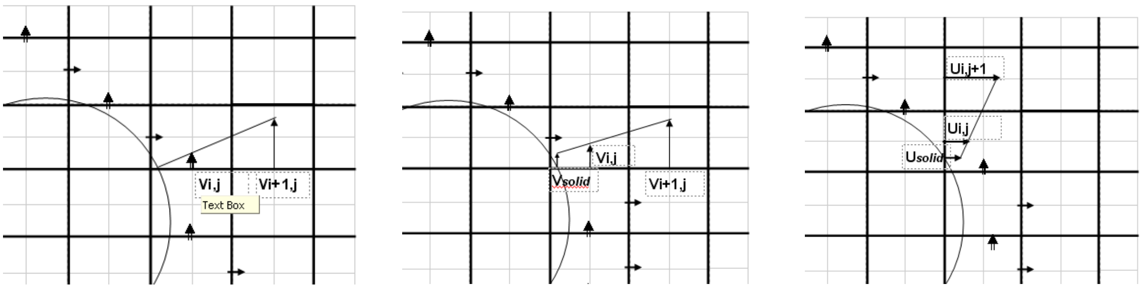

Figure 4: linear interpolation method: interpolating Vi,j for a stationary solid in vertical direction (left), interpolating Vi,j for a moving solid (Vsolid) in vertical direction (middle); interpolating Ui,j for a moving solid (Usolid) in horizontal direction.

Case C: linear interpolation method

The second method is a linear interpolation method where the velocities in the interface cells are calculated by interpolation between the velocity at the solid boundary (applying the no slip condition) and one point in fluid domain. Fadlun [18] suggested using this interpolation method to enforce the boundary condition to the fluid domain in the indirect discrete forcing approaches. In this paper, however, we are only interested in his interpolation method.

Case D: Bilinear interpolation method

Kang [19] presented various methods to interpolate the velocity near the boundary in two directions considering the effect of the pressure near the boundary. As before, only the interpolation is of interest here. At first, in the Standard Reconstruction method (SRM) the two velocities in the horizontal and vertical directions that are closest to the immersed boundary are used to obtain the interpolated velocity at each of the interface points. The resulting interpolation formula for the velocity in horizontal direction has the following form.

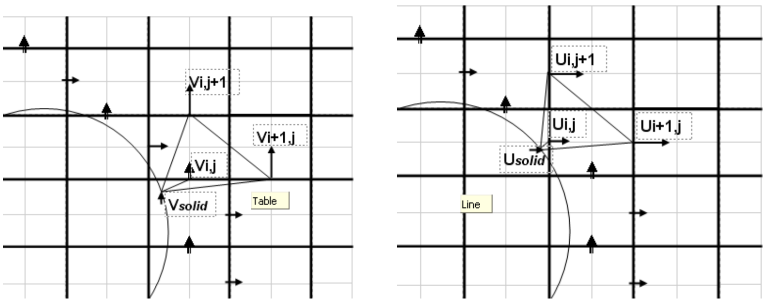

Figure 5:Standard Reconstruction Method (SRM) for velocity in vertical (left) and horizontal (right) direction

$$U_{i,j} = \omega_{i+1,j} U_{i+1,j} + \omega_{i,j+1} U_{i,j+1} + \omega_{solid} U_{solid},$$

where the various (represent the interpolation weights.

Kang [19] has revised the above interpolation and also uses the velocity field in the previous time step to obtain a better interpolation at the next time step and explicitly used the difference between the velocities at two consecutive time steps in the interpolation of the interface velocities. In this case the interpolation formula becomes:

$$\Delta U_{i,j} = \omega_{i+1,j} \Delta U_{i+1,j} + \omega_{i,j+1} \Delta U_{i,j+1} + \omega_{solid} \Delta U_{solid}$$

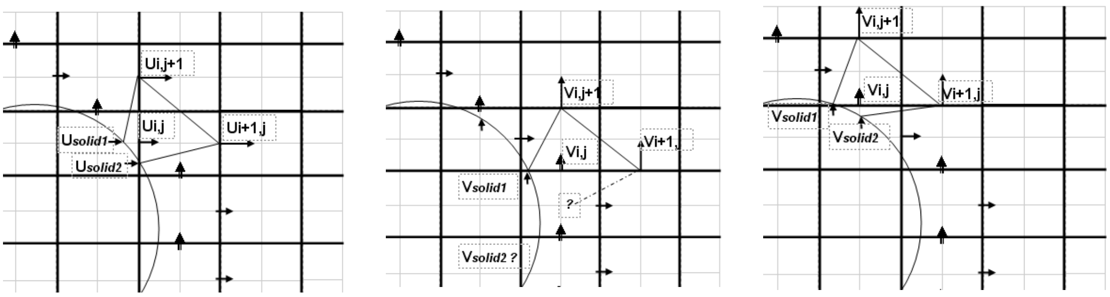

Figure 6: Quadratic interpolation method for an interface velocity in the horizontal (left) and the vertical (right) directions. The middle pane shows that it is not always possible to use this type of quadratic interpolation.

In addition Kang [19] introduced a quadratic interpolation formula to incorporate the local pressure gradient in the velocity interpolation to compensate for a decoupling between the pressure and the velocity near the solid boundary. Figure 6 illustrates their interpolation method that uses four adjacent velocities and enforces the momentum equation by a quadratic interpolation in the two-dimensional case. As we only focus on comparing velocity interpolation methods Kang's method is excluded from the comparison.

Case E: Proposed interpolation method

The bilinear interpolation method proposed in this paper is based on interpolating the boundary velocity values in the direction perpendicular to the solid boundary. In this method, a line from the boundary velocity position is drawn perpendicular to the boundary surface and extended to cut the line between the first two known velocities in the fluid domain (Point A, Figure 7 right). The velocity will be interpolated at the intersection point A. Then, the boundary cell velocity values will be interpolated using the solid boundary velocity (for a stationary cylinder with no-slip conditions this velocity is zero) and the velocity at point A. Figure 7 (left) shows this interpolation for velocities in y direction and Figure 7 (right) shows the interpolation for the velocity in the x direction.

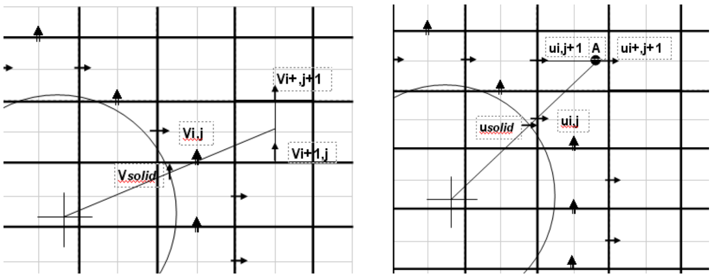

Figure 7: Bilinear proposed interpolation in this study for the cells near the solid boundary in vertical (Left) and horizontal (right) velocity components.

RESULTS AND DISCUSSION

The flow around a stationary cylinder at Re=100 has been simulated with different interpolation treatments to represent the immersed boundary. The Strouhal number, drag and lift coefficients for various cases are compared.

The Strouhal number is the non-dimensional frequency of the vortex shedding around the body and is defined by:

$$St = \frac{f_s D}{U_\infty},$$

where is the frequency of the vortex shedding, D is the cylinder diameter and is the far-field velocity.

The drag coefficient on a body in a fluid flow includes both the shear stress and the pressure drag on the solid surface. The dimensionless drag coefficient is defined by:

$$C_D = \frac{F_D}{\frac{1}{2}\rho u_\infty^2 D}$$

The lift force on the cylinder is generated when the vortex shedding starts around the structure. The dimensionless lift coefficient is defined by

$$C_L = \frac{F_L}{\frac{1}{2}\rho u_\infty^2 D}.$$

For any solid body both the pressure distribution and the friction along the solid surface may contribute to the lift and drag forces. In the present study, the pressure at the surface is obtained by taking the wall-nearest pressure values in the flow domain on the outside of the solid body, thereby assuming that the wall normal gradient of the pressure near the surface is negligibly small. The component of the drag and lift forces due to pressure distribution is calculated by integrating the pressure along the solid boundary. On the other hand, the shear-force component of the lift and drag forces is calculated from near the surface of the solid. The tangential velocity near the solid surface is obtained at the wall-nearest point outside of the body and is subsequently used to calculate the wall-shear stress at the cylinder surface.

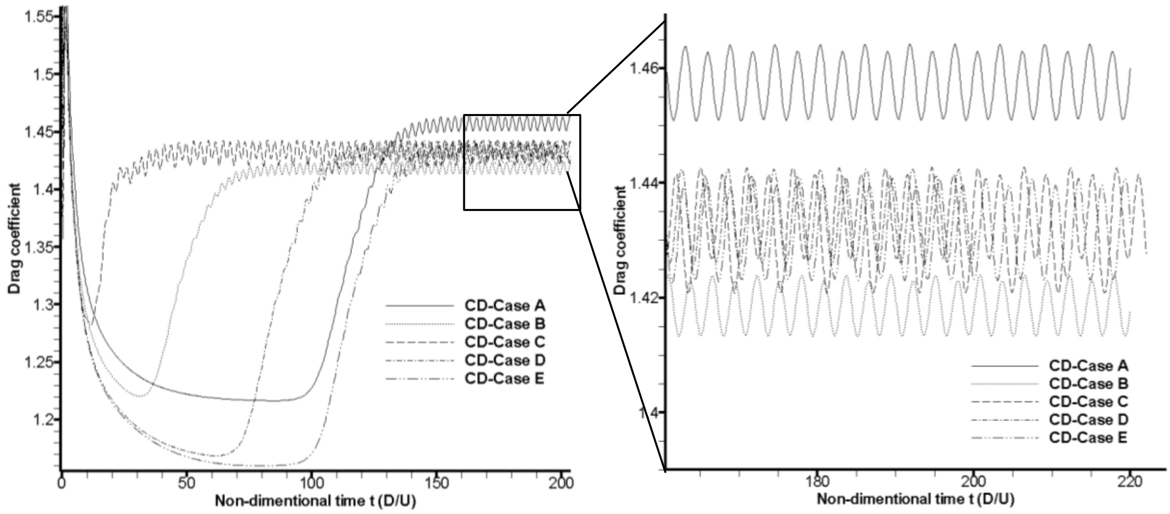

Figure 8: Drag coefficient for the flow around a stationary cylinder at Re=100, Case A, without interpolation; Case B: area weighting method; Case C, Linear interpolation method; Case D, Bilinear interpolation1; Case E, Bilinear interpolation2

Figure 8, shows a comparison of the drag coefficients obtained in calculations of flow over a stationary cylinder at Re=100 using various interpolation methods. It can be seen that in the cases C, D and E, (linear and Bilinear interpolation methods) the results were converging to a value of C_D = 1.43. However, Case A (without interpolation) leads to a higher drag coefficient, CD=1.46 and Case B (weighting method) leads to a lower drag coefficient CD=1.42. Once vortex shedding commenced all simulations were found to run at virtually the same speed (Table 1) showing that the computational effort needed for the interpolation was negligible. However, for a non-stationary cylinder, it is expected that the reguired repeated calculation of interpolation coefficients may lead to a reduction in execution speed.

Table 1: Real computational time, 20 vortex shedding

	Case A	Case B	Case C	Case D	Case E
Real time (s)	3231	3225	3379	4441	3383

Figure 8 (left) shows that Case C (linear interpolation) is the quickest method to develop vortex shedding, which indicates that it the implementation of boundary conditions with linear interpolation causes significant numerical noise. In Case E (proposed bilinear method), on the other hand, the vortex-shedding instability kicks in much later evidencing that the level of numerical noise introduced by this type of interpolation is very small.

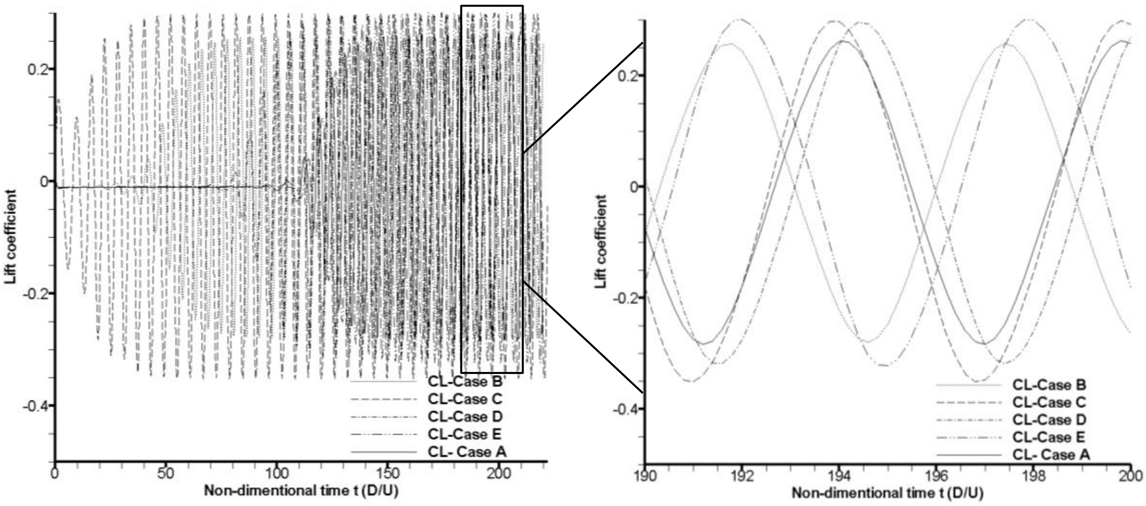

Figure 9: Lift coefficient for the flow around a stationary cylinder at Re=100, Case A, without interpolation; Case B: area weighting method; Case C, Linear interpolation method; Case D, Bilinear interpolation1; Case E, Bilinear interpolation2

Figure 9, shows the comparison of lift coefficients for the various interpolation cases. It can be seen from the figure that like the drag coefficient, the lift coefficient for the linear and bilinear cases are nearly the same (Case C, D and E) CL=0.33. However, the Case B (related to weighting area method) shows lower values for the lift (CL=0.27). In Case A (without interpolation), the lift due to shear stress is out of range, but the lift due to pressure is acceptable. The reason for the unacceptable results for the lift due to shear is that the velocity for case A was selected out of the boundary layer (the aim was to choose similar conditions for all cases).

Table 2: Strouhal number, lift and Drag coefficient for the flow around a stationary cylinder and Re=100.

simulation methods	Strouhal Number	Drag Coefficient	Lift coefficient
Case A	0.174	1.46	0.26
Case B	0.175	1.42	0.27
Case C	0.169	1.432	0.325
Case D	0.169	1.434	0.305
Case E	0.168	1.432	0.31
Park [22] fitted method	0.165	1.33	0.33
Williamson(exp.)[20]	0.166	….	….
Kim [22]	0.165	1.33	0.32
Roshko (exp.)[20]	0.164	….	….
Lai and Peskin [6]	0.165	1.4473	0.3299
Choi [7]	….	1.351	0.315
Corbalan & Souza [24]	….	1.44	0.31

Table 2 shows the comparison of the Strouhal number, lift and drag coefficient for various methods; from the experimental methods (Roshko and Willamsion reported by [20]) to the body fitted mesh [21] and immersed boundary methods [22, 6, 23, 24], for the flow around a stationary cylinder at Re=100. It can be seen that the Strouhal number varies between 0.16 and 0.18; the Drag coefficient between 1.33 and 1.4473 and the lift coefficient between 0.31 and 0.33.

CONCLUSION

The objective of the present study is to compare the accuracy and expenses of different IB interpolation methods and select the most accurate and least expensive method for future use in simulations of flow around a deformable cylinder. A finite-volume method on a Cartesian grid with a staggered variable arrangement has been used. In this IB implementation the velocities near non-confirming boundaries were interpolated in the normal direction to walls, thereby considering the curvature of the geometry. The Strouhal number and Drag coefficient for different cases are reported. The results show a good agreement with the literature for most of the interpolation methods for the stationary cylinder. The drag coefficient and Strouhal number results for five different interpolation methods were compared it was shown that for a stationary cylinder at low Reynolds number, the interpolation method could affect the drag coefficient by a maximum 2% and the Strouhal number by maximum of 3%. In addition, the bi-liner interpolation method took about 2% more computational time per vortex shedding cycle in companion to the other methods.

REFERENCES

[1] Hou G, Wang J. and Layton A, Numerical Methods for Fluid-Structure Interaction – A Review, Commun. Comput. Phys doi:10.4208/, Vol.12,No. 2, pp. 337-377(2012).
[2] Peskin C.S., Flow patterns around heart valves: A numerical method, Journal Comput. Phys. 10,252 (1972).
[3] Mittal, R., and Iaccarino, G., “Immersed Boundary Methods,” Annual Review of Fluid Mechanics, Vol. 37, 2005, pp. 239–261. doi:10.1146/annurev.fluid.37.061903.175743
[4] Saiki EM, Biringen S.. Numerical simulation of a cylinder in uniform flow: application of a virtual boundary method. Journal of Computational Phyics 123:450–65(1996).
[5] Beyer RP, Leveque RJ., Analysis of a one dimensional model for the immersed boundary method. SIAM Journal Number Annual 29:332–641992.
[6] Lai MC, Peskin CS., An immersed boundary method with formal second-order accuracy and reduced numerical viscosity, Journal of Computational. Phys. 160:705–19 (2000).
[7] Briscolini M. and Santangelo P., Development of the mask method for incompressible unsteady flows, J. Computational Physics 84-57,1988.
[8] Mohd-Yusof J., Combined immersed boundaries/B-Spline Methods for simulations of flows in complex geometries, CTR Annual Research Briefs, NASA AMES/Stanford University, 1997.
[9] verzicco R.m Mohd-Yusof J., Orlandi P. & Haworth D., Large eddy simulation in complex geometric configurations using boundary body forces, AIAA J. 38(3), 427 (2000).

[10] Iaccarino G, Verzicco R. Immersed boundary technique for turbulent flow simulations. Appl. Mech. Rev. 56:331–47(2003).
[11] Ghias R, Mittal R, Lund TS. A non-body conformal grid method for simulation of compressible flows with complex immersed boundaries. AIAA Pap. 2004-0080 (2004).
[12] Mittal R, Bonilla C, Udaykumar HS. Cartesian grid methods for simulating flows with moving boundaries. In Computational Methods and Experimental Measurements- XI, ed. CA Brebbia, GM Carlomagno, P Anagnostopoulous(2003).
[13] Mittal R, Seshadri V, Udaykumar HS. Flutter, tumble and vortex induced autorotation. Theor. Comput. Fluid Dyn. 17(3):165– 70(2004).
[14] Udaykumar HS, Mittal R, ShyyW. Computation of solid-liquid phase fronts in the sharp interface limit on fixed grids. J. Comput. Phys. 153:534–74(1999).
[15] Udaykumar HS, Mittal R, Rampunggoon P, Khanna A.. A sharp interface Cartesian grid method for simulating flows with complex moving boundaries. J. Comput. Phys. 174:345–80(2001).
[16] Harlow F.H. and welsh J.E., Numerical Calculation of time-dependent viscous incomperissible Flow of Fluid with Free surface, The Physics of Fluids, Vol. 8, No.12, 1965
[17] Ferziger J.H and Peric M, Computational methods for fluid dynamics, 3rd edition, Springer, 2002.
[18] Fadlun, E. A., Verzicco, R., Orlandi, P., and Mohd-Yusof, J., "Combined Immersed-Boundary Finite Difference Methods for Three- Dimensional Complex Flow Simulations," Journal of Computational Physics, Vol. 161, No. 1, pp. 35–60 (2000).
[19] Kang S., Iaccarino G., Moin P., Accurate immersed-boundary reconstructions for viscous flow simulations, AIAA J. 47 (7) (2009) 1750–1760.
[20] Liu C., Zheng X., Sung C.H., Preconditioned multigrid methods for unsteady incompressible flows, journal of computational physics 139,359(1998).
[21] Park J., Kwon K., and Choi H., Numerical solutions of flow past a circular cylinder at Reynolds number up to 160, KSME international Journal, vol. 12, No. 6, 1998, pp.1200-1205.
[22] Kim J., Kim D., Choi H., and immersed boundary finite volume method for simulations of flow in complex geometries, journal of computational physics, 171 (2001) 132.
[23] Choi J.I, Oberoi R.C., Edwards J.R., Rosati J. A., An immersed boundary method for complex incompressible flows, Journal of Computational Physics, Volume 224, Issue 2, 10 June 2007, Pages 757-784, ISSN 0021-9991, 10.1016/j.jcp.2006.10.032.
[24] Corbalan Gois E.R. and De Souza L.F., An Eulerian immersed boundary methos for flos simulations over stationary and moving rigid bodies, Journal of the Braz. Soc, Of Mech. Sci & Eng, special issue 2010, vol. XXXII, No. 5/477.