Copyright © 2011 Tech Science Press FDMP, vol.7, no.3, pp.241-258, 2011

Comparison Between Different Immersed Boundary Conditions for Simulation of Complex Fluid Flows

A. Mark[1 2], **R. Rundqvist**[1] **and F. Edelvik**[1]

Abstract: In the literature immersed boundary methods are employed to simulate complex flows around moving arbitrary bodies without the necessity of remeshing. These methods employ a regular Eulerian mesh to simulate the fluid flow and a Lagrangian representation of the boundary of the bodies. The two representations can be coupled through an immersed boundary condition constraining the fluid to exactly follow the boundary of the bodies (immersed boundaries). Typically such methods suffer from accuracy problems, that arise from spurious mass fluxes over the immersed boundary (IB), pressure boundary conditions or high density ratios. The mirroring IB method Mark (2008); Mark and van Wachem (2008) resolves these problems by ensuring zero mass flux over the IB instead of employing a pressure boundary condition. In this work the mirroring IB method together with a hybrid IB condition are implemented and validated in IBOFLOW. IBOFLOW is an incompressible finite-volume based fluid flow solver. The Navier-Stokes' equations are coupled with the SIMPLEC method by Doormaal and Raithby (1984) and discretized on a Cartesian octree grid that can be dynamically refined and coarsened, enabling grid refinement to follow moving bodies. The variables are stored in a co-located configuration and pressure weighted flux interpolation by Rhie and Chow (1983) is employed to prevent pressure oscillations. In the implemented IB method the immersed bodies are represented by an analytical description or by a triangulation. The method models the presence of the bodies inside the fluid by an implicitly formulated IB condition, which constrains the fluid velocity to the boundary velocity with second-order accuracy. The original mirroring IB condition mirrors the velocity field over the local IB and the hybrid IB condition mirrors and extrapolates the fluid velocity onto the IB. These IB conditions generate a fictitious velocity field inside the bodies, which is excluded in the continuity equation to ensure zero mass flux over the boundary. The fluid flow over an immersed sphere is simulated to validate and compare the different IB conditions. The simulated

[1] FCC, Chalmers Science Park, SE-412 88 Göteborg, Sweden

[2] andreas.mark@fcc.chalmers.se

drag force is compared to experimental findings with excellent agreement and a detailed convergence study of the error of the fluid velocity integrated over the immersed boundary is performed to show the strictly second-order accuracy of the implemented IB conditions. It is shown that the error is reduced with the hybrid IB condition compared to the original mirroring IB condition. In addition, a sedimenting sphere with a moving grid refinement is simulated to validate the hybrid method and show the potential of the dynamic octree grid.

Keywords: immersed boundary condition, mirroring immersed boundary method, implicit method

1 Nomenclature

ρ_f	Fluid density	kg/m^3
ρ_s	Body density	kg/m^3
μ	Dynamic viscosity	Ns/m^2
C_d	Drag coefficient	$-$
f_d	Drag force	N
$\vec{g}$	Gravitation	m^2/s
m_p	Body mass	kg
p	Fluid dynamic pressure	Pa
$\vec{p}$	Grid point position	m
r	Sphere radius	m
t	Time	s
U	Inlet/Free stream velocity	m/s
$\vec{u}$	Fluid velocity	m/s
V	Body volume	m^3
$\vec{v}$	Body velocity	m/s
$\vec{x}$	Body position	m

Abbreviations

DNS	Direct numerical simulations
IB	Immersed Boundary
IBC	Immersed Boundary Condition

2 Introduction

The fluid flows around arbitrary, moving and interacting bodies are both complex and poorly understood. These flows are also the most common in both nature and industrial applications. To gain more knowledge of the complex flow, simulations at small scales are required, where the flow around the particles is completely resolved. To study the real phenomena the equations governing the fluid flow must be solved directly without introducing any extra models. Such simulations are called true direct numerical simulations (DNS). The results obtained from the simulations on a small scale can be used to gain an understanding of the phenomena and to develop new and better large scale methods.

In the literature immersed boundary methods are utilised to simulate complex flows around moving arbitrary bodies without the necessity of remeshing. These methods employ a regular Eulerian mesh to simulate the fluid flow and a Lagrangian representation of the boundary of the bodies (immersed boundary). These two representations can be coupled through Lagrange multipliers (Glowinski, Pan, and Periaux (1994a,b)), explicit forces (Kim and Choi (2000); Kim, Kim, and Choi (2001); Lai and Peskin (2000); Goldstein, Handler, and Sirovich (1993); Lima E Silva, Silveira-Neto, and J.J.R. (2003, 2004); Mohd-Yusof (1997, 1998); Oliveira, A.L.F., and Silveira-Neto (2005)), or by internal boundary conditions (Majumdar, Iaccarino, and Durbin (2001); Mark (2008); Mark and van Wachem (2008)). If the immersed boundary represents an interface between two fluids a discontinuous pressure field is typically present. For such simulations the immersed interface method is more preferable (Leveque and Li (1997)). As a known jump condition can be employed at the immersed boundary, such as surface tension.

Glowinski, Pan, and Periaux (1994a) was the first to couple the two representations with Lagrange multipliers. The method is implemented in a finite element frame work and by Lagrangian multipliers an immersed boundary condition (IBC) is introduced in the weak formulation of the Navier-Stokes' equations. The resulting method is implicitly formulated and second-order accurate in space. Overall the method performs well but a finite volume implementation is not straight-forward. In the work of Sharma and Patankar (2005); Sharma, Y., and Patankar (2005) the method is implemented in a finite volume framework but the force remains explicitly coupled.

The original immersed boundary method developed by Peskin (1977) couples the representations with a force. In this method a discrete Dirac function is used to distribute a Lagrangian force from the immersed boundary (IB) to the Eulerian grid. The distributed volume force explicitly constrains the fluid to follow the IB. Due to the distribution function the resulting method is only first-order accurate in space. Mohd-Yusof (1997, 1998) developed a momentum forcing method that enforce the fluid velocity at the IB by introducing an explicit force in the momentum equations. The force is applied onto the cells lying inside but close to the IB generating a reversed velocity field over the IB. Resulting problems with mass conservation is solved in Kim and Choi (2000). The explicitly formulated method is second-order accurate in space and commonly used in the literature.

Majumdar, Iaccarino, and Durbin (2001) developed an immersed boundary method, which implicitly constrains the velocity of the fluid at the IB with an immersed boundary condition (IBC). The method is employed for stationary bodies along with a turbulent RANS solver and shows good results. However, the method has potential problems with the weighting coefficients in the boundary condition which may result in oscillations in the resulting solution. Moreover, the method may generate an unphysical mass flux over IB segments.

Mark and van Wachem (2008) developed a stationary immersed boundary method which models the presence of the bodies by an IBC that mirrors the velocity field over the boundary of the body in such a way that the fluid exactly follows the surface of the body. As a result, a fictitious velocity field inside the body is generated, which is excluded in the continuity equation to ensure zero mass flux over the boundary. The method generates no unphysical oscillations around the IB and is second-order accurate in space and implicitly formulated.

In this work the mirroring IB method together with a hybrid immersed boundary condition are implemented and validated. The hybrid IBC mirrors and extrapolates the fluid velocity onto the IB, thus generating additional immersed boundary points where the fluid is strictly constrained to the IB velocity. These additional immersed boundary points reduce the error at the IB. Two different test cases are presented: The fluid flow over a single sphere and simulation of a sedimenting sphere with moving grid refinement. These test cases demonstrate the second-order accuracy of the immersed boundary conditions by integrating the surface error. The test cases also show that the flow solver accurately simulates the resulting drag forces and terminal velocities.

3 The Governing Equations

The flow of an incompressible Newtonian fluid around immersed bodies is governed by the continuity and momentum equations, the Navier-Stokes' equations;

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

$$\rho_f \frac{\partial \vec{u}}{\partial t} + \rho_f \vec{u} \cdot \nabla \vec{u} = -\nabla p + \mu \nabla^2 \vec{u} \tag{2}$$

where ρ_f is the fluid density, $\vec{u}$ is the fluid velocity, p is the dynamic pressure, μ represents the dynamic viscosity of the fluid. To close the governing equations, boundary conditions are employed at the boundaries of the fluid domain.

The movement of a body inside the fluid is governed by Newton's second law or the momentum equations in a Lagrangian framework:

$$m_p \frac{\partial \vec{v}}{\partial t} = \vec{f}_d + (\rho_s - \rho_f) V \vec{g} \tag{3}$$

where m_p is the body mass, $\vec{v}$ is the velocity, ρ_s is the density and V is the volume. Furthermore, $\vec{f}_d$ is the fluid drag force acting on the body, ρ_f is the fluid density, and $\vec{g}$ is the gravitation.

4 IBOFlow

IBOFLOW (Immersed Boundary Octree Flow Solver) is an incompressible finite-volume based fluid flow solver. The velocity and pressure fields are coupled with the SIMPLEC method by Doormaal and Raithby (1984) and discretized on a Cartesian octree grid that can be dynamically refined and coarsened, enabling grid refinement to follow moving bodies with almost no extra computational cost. As the grid is stored as an octree search thread efficient search algorithms can be adopted to find cells and couple the immersed boundaries with the computational grid. The variables are stored in a co-located configuration and pressure weighted flux interpolation by Rhie and Chow (1983) is employed to prevent decoupling of the pressure and velocity field. For the unsteady part of the Navier-Stokes' equations implicit backward Euler time discretization is adopted. The movement of the bodies inside the fluid are discretized by the explicit second order finite difference Crank and Nicolson (1947) time scheme.

5 The Immersed Boundary Methods

This section describes the implementation of the original mirroing immersed boundary method Mark and van Wachem (2008) together with a hybrid immersed boundary condition.

5.1 Cell types and exterior normal points

The octree grid needs to be connected to the IB. This is done initially and when the IB is moved by determing the cell types and the exterior normal points. First the closet point on the IB from each cell center is determined. For an analytically described IB a bounding box is spanned around the IB with a padding distance and the cells inside the box are found by a bounding box octree search. For the found cells, the closest point on the IB is calculated with an analytical formula specific to the IB description. If the IB is triangulated a triangle search distance kd-tree is constructed and the closest point on the triangulation is simply found by employing

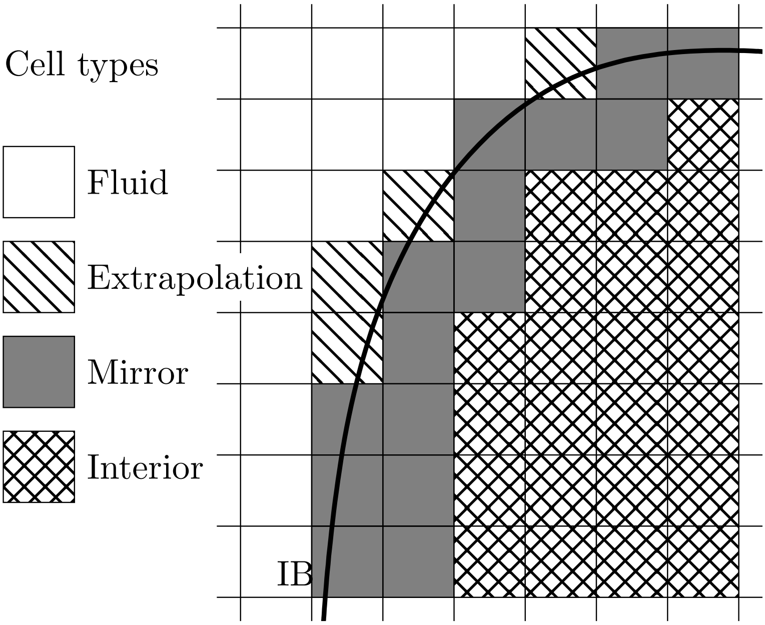

Figure 1: Definitions of the different cell types. The extrapolation cells lie outside and close to the IB and the mirroring cells lie inside and close to the IB. The rest of the cells inside the IB is interior cells and the rest of the cells outside is fluid cells.

a distance search with the kd-tree. From the closest point the minimum distance to the IB is found and the cell type is determined: Cells whose centre lie outside the IB, but with the minimum distance to the IB less than one half cell size, are classified as extrapolation cells. The rest of the cells outside the IB are classified as fluid cells. Cells that lie inside the IB but with a minimum distance to the IB less than one and a half cell size are classified as mirroring cells. The rest of the cells inside the IB are classified as internal cells, see Figure 1 for an example. For mirroring cells the exterior normal point, $\vec{p}_e$, is calculated as;

$$\vec{p}_e = \vec{p}_{mi} + 2.0\,(\vec{p}_{ib} - \vec{p}_{mi}) \tag{4}$$

where $\vec{p}_{mi}$ is the centre of the mirroring cell and $\vec{p}_{ib}$ is the closest point on the local IB, see Figure 2 for a two-dimensional visualization. For extrapolation cells, the extrapolation exterior point, $\vec{p}_e$, is calculated as;

$$\vec{p}_e = \vec{p}_{ib} + 2.0\,(\vec{p}_{ex} - \vec{p}_{ib})\,. \tag{5}$$

Notice that all positions are defined in three dimensions.

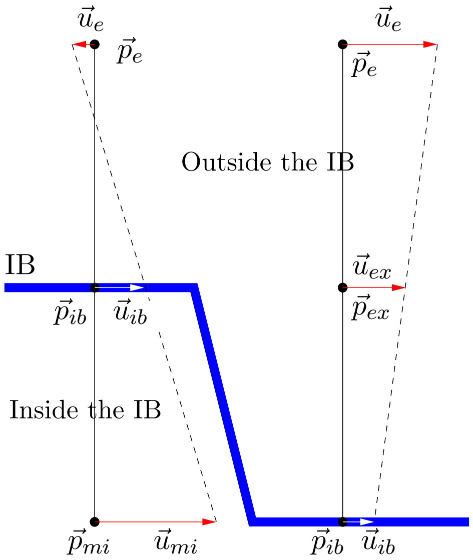

Figure 2: A two-dimensional visualization of the IB, showing a mirroring point, $\vec{p}_{mi}$, lying in the centre of a mirroring cell, an extrapolation point $\vec{p}_{ex}$, lying in the centre of an extrapolation cell, and immersed boundary points, $\vec{p}_{ib}$, lying on the local IB. The mirrored and extrapolated exterior normal points, $\vec{p}_e$, and the respective velocities at the points, $\vec{u}_e$, $\vec{u}_{ib}$, $\vec{u}_{mi}$ and $\vec{u}_{ex}$, are also shown.

5.2 The mirroring immersed boundary condition

The velocity of the fluid at the immersed boundary, $\vec{u}_{ib}$, is constrained to the velocity of the immersed boundary itself at $\vec{p}_{ib}$ by an implicit immersed boundary condition. The mirroring IBC interpolates the velocity of the fluid to the fictitious exterior normal point, $\vec{p}_e$, by trilinear interpolation and sets the velocity in the mirror cell to the reversed interpolated velocity, $\vec{u}_e$, plus the double local velocity of the immersed boundary, $\vec{u}_{ib}$. The resulting mirroring immersed boundary condition is as follows;

$$\frac{\vec{u}_{mi} + \vec{u}_e}{2} = \vec{u}_{ib}. \tag{6}$$

5.3 The hybrid immersed boundary condition

The hybrid immersed boundary condition both mirrors and extrapolates the velocity field. For extrapolation cells the hybrid IBC interpolates the velocity of the fluid to the fictitious exterior normal point by trilinear interpolation and sets the velocity in the extrapolation cell to the mean value of the interpolated velocity and the local velocity of the immersed boundary, $\vec{u}_{ib}$. The resulting extrapolation immersed boundary condition is as follows;

$$\frac{\vec{u}_{ib}+\vec{u}_{e}}{2}=\vec{u}_{ex}. \tag{7}$$

5.4 Mass conservation

To ensure that no mass flux over the IB exists, the interpolated internal face velocities are excluded when discretizing the continuity equation. This is sensible since these have no physical meaning. As the velocity field inside the IB is excluded in the continuity equation a pressure field is generated such that all fluid entering a cell have to leave the cell outside the IB. Therefore the velocity field is physically prescribed to be zero over the IB and the normal component of the velocity gradient over the IB becomes zero. As a result an explicit Neumann boundary condition is generated for the velocity. The pressure correction equation will therefore not generate a driving force over the IB in the momentum equation; for this reason it is not necessary to employ any regular (Neumann) boundary condition for the pressure at the IB.

In techniques similar to the mirroring IB method described here, oscillations have occurred in the final solution Majumdar, Iaccarino, and Durbin (2001). We propose that this is due to the presence of an unphysical mass flux and/or an unphysical boundary condition over the IB. To decrease the mass flux over the IB, Majumdar, Iaccarino, and Durbin (2001) apply a Neumann boundary condition for pressure at the IB that results in a zero pressure force over the IB, and thus a decreased mass flux. Due to the flow and other present forces the solution will however still allow for a small mass flux across the boundary.

5.5 Algorithm

The following algorithm is employed to take one fluid time step when immersed boundaries are present:

i. During initialization determine the initial shape, position and velocity of each body.

ii. For the first time step and if an IB is moved connect the octree grid to the IB by setting the cell types: mirror, interior, extrapolation and fluid, see Figure 1. For each mirror or extrapolation cell calculate and store the exterior normal point(s).

iii. Assemble the momentum equations by determining the coefficients for the linearized equations for each cell:

    (a) For all mirror and extrapolation cells the IBC is employed to determine the coefficients of the velocity components.

    (b) If the cell is an interior cell the coefficients of the velocity components are set to match the body velocity, $\vec{u}_{ib}$, by a Dirichlet condition.

    (c) For all other cells, the coefficients are determined by linearizing the Navier-Stokes' equations.

iv. Determine the temporary velocity field by solving the linearized system assembled in (iii).

v. Solve a Poisson equation for pressure correction in the SIMPLEC method, based upon the solution from the previous step. Velocities inside the body are excluded. The pressure and the velocities are corrected with the pressure and velocity correction equations.

vi. Go to step (iii) until the continuity equation is satisfactorily fulfilled.

vii. Calculate by surface integration the fluid surface force for each immersed body.

viii. Calculate the new positions, $\vec{x}$, and velocities, $\vec{v}$, for each IB and continue with the next time step.

6 Results

To verify the accuracy of the different IBCs the flow over a single sphere is simulated and the resulting drag coefficient is compared to analytical or semi empirical data. In addition, a sedimenting sphere is simulated to validate the hybrid method for moving bodies and to show the potential of the dynamic octree grid.

6.1 Flow around a sphere

A sphere with radius 0.5 mm is placed in the middle of a square simulation box with a 10.0 mm side, see Figure 3. The octree grid is refined around the sphere a number of times, see Table 1. The inlet of the simulation box is placed on the x_{min} surface, the outlet on the opposite surface and symmetry boundary conditions are employed for the other surfaces. The inlet velocity is set to 1.0 m/s and the fluid viscosity to 1mPa·s. The Reynolds number is varied by altering the fluid density between 0.01 and 100.0kg/m^3. A time step of 0.1 ms is used in all transient simulations which are run until steady state.

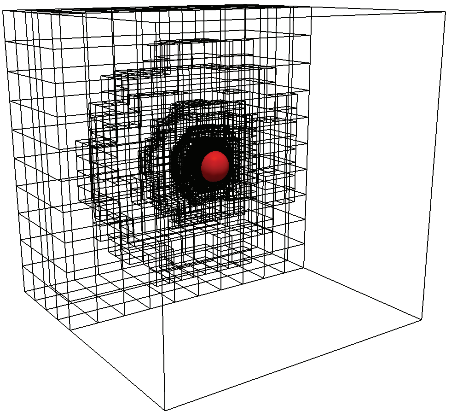

Figure 3: The simulation box with the adaptive octree grid G5 around the immersed sphere.

Table 1: The grids used in the convergence study. The number in the grid name represents the number of times the grid is refined.

Name	Cell size (mm)	Number of cells
G2	1.0 – 0.2500000	5480
G3	1.0 – 0.1250000	10184
G4	1.0 – 0.0625000	21944
G5	1.0 – 0.0312500	58400
G6	1.0 – 0.0156250	190112
G7	1.0 – 0.0078125	690248

In Figure 4 the fluid velocity and fluid pressure are vizualised on a cut through the domain. As seen in the figure the pressure is high on the front side of the sphere and low on the back side. An integration of the pressure generates the pressure part of the drag force acting upon the sphere from the fluid. An integration of the viscous stress tensor gives the viscous part of the drag force.

The drag force acting upon a sphere is calculated as;

$$f_d = \frac{1}{2}\rho_f C_d \pi r^2 U^2 \tag{8}$$

where U is the mean free fluid velocity and r the radius of the sphere. The drag coefficient, C_d, is dependent on the local Reynolds number;

$$Re = \frac{2r\rho_f U}{\mu} \tag{9}$$

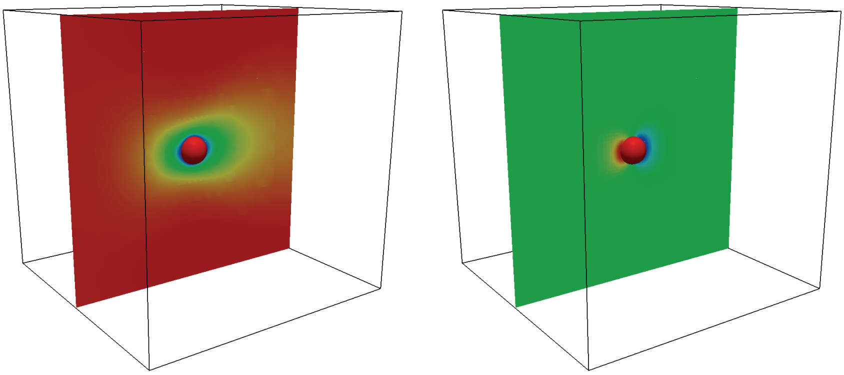

Figure 4: Left: For Reynolds number 1.0 the fluid velocity field is shown on a cut through the fluid domain. Right: The fluid pressure is shown around the sphere. Red indicates high velocity or pressure.

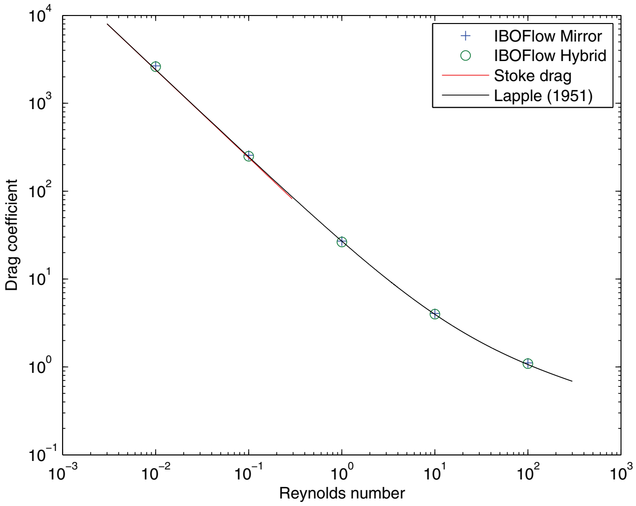

Figure 5: The simulated drag coefficient, C_d, for the two different immersed boundary conditions compared with theoretical findings.

and for $Re < 1000$ it was approximated by Lapple Lapple (1951) to;

$$C_d = 24.0/Re\left(1.0 + 0.125Re^{0.72}\right). \tag{10}$$

In Figure 5 the simulated drag coefficients calculated from the local pressure and

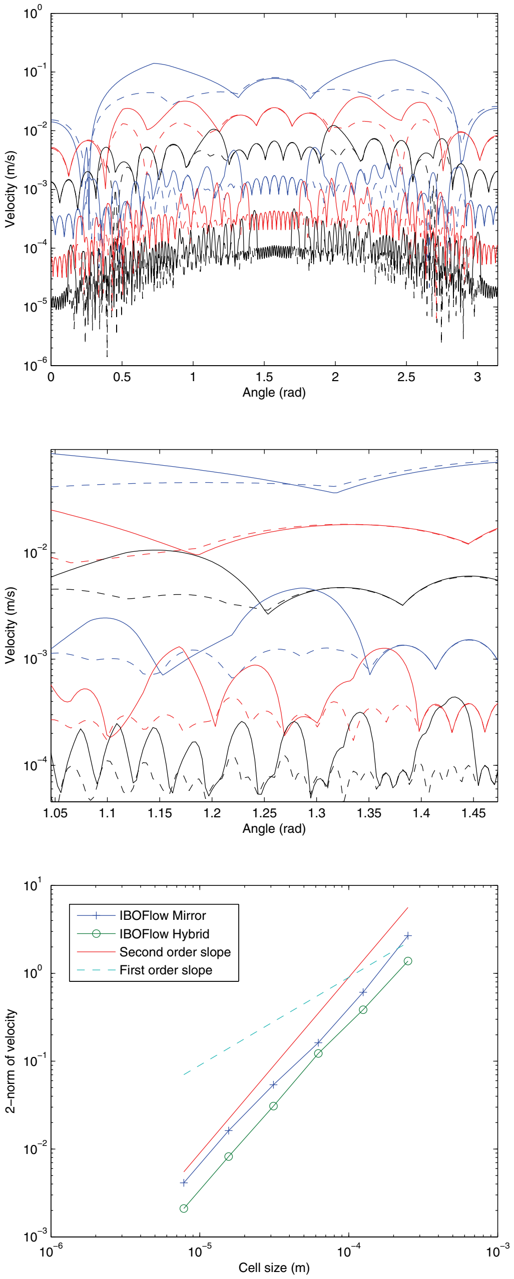

Figure 6: Top left: The velocity error on a half circle bow around the sphere is plotted in log scale as a function of the angle for the six different grids described in Table 1. The error for the mirroring IBC is plotted as solid lines, '–', and the error for the hybrid method is plotted as dashed lines,'- -'. Top right: A zoom of the top figure. Bottom: For the different IBCs the L_2-norm of the error is plotted in log scale as a function of grid size.

velocity gradients are plotted against the Reynolds number and compared to Lapple's results. As seen in the figure both methods accurately simulate the drag coefficient for the specific Reynolds numbers.

To show the order of accuracy of the IBCs a grid convergence study is carried out, evaluating the fluid velocity at the surface of the sphere. In Figure 6 the velocity on a 180 degree arc around the sphere is plotted against the angle for the grids in Table 1. In the figure it is shown that the velocity, which should be zero, decreases with the grid size and that the error between the different methods differs. The mirror IBC has less immersed boundary points than the hybrid IBC and therefore also a larger error. This is particularly visible in the middle part of Figure 6. At each immersed boundary point the error decreases and between the immersed boundary points the error is somewhat larger, therefore the total error of the hybrid method is smaller. To prevent spikes in the error the hybrid IBC is preferable. In the bottom of the figure the L_2-norm of the total velocity error is plotted as a function of the grid size along with a first-order and a second-order curve. From the figure it is concluded that both methods are second-order accurate in space and that the hybrid method generates a smaller error. Therefore only the hybrid method is used in the following simulations.

6.2 Sedimenting sphere

The computational domain in this case is $3.0 \times 3.0 \times 6.0$ mm, with gravity being oriented in the negative z-axis with a magnitude of 10 m/s^2. The sphere with radius 0.2 mm is initially placed at the position $(1.5, 1.5, 5.0)$ mm. The density of the sphere, ρ_s, is altered between 1390 and 5000 kg/m^3 to generate different Reynolds numbers and corresponding terminal velocities. The outlet of the simulation box is placed on the z_{min} surface and all other surfaces are treated with symmetry boundary conditions. The fluid is water with density 1000 kg/m^3 and the viscosity 1.0 mPa·s. The time step is adapted to obtain a CFL number of unity. The computational grid is dynamically refined around the sedimenting sphere, see Table 2, where Grid G2 is refined two times, G3 three times, G4 four times and G5 five times, respectively.

In Figure 7 the simulated fluid velocity field and pressure are shown for grid G3 and particle density 1390 kg/m^3. In the figure the adaptive grid is shown which is moving with the sedimenting sphere.

In Figure 8 the simulated velocity and the normalized drag force (drag coefficient) of the sphere is plotted against the z-position. The figure shows that the forces and velocities are smooth for the finest grids G3, G4 and G5, but for the coarsest grid some oscillations are present. The simulated velocity from grid G2 is lower than for the other grids due to the fact that the boundary layer around the sphere is not

Table 2: The dynamically refined grids employed in the sedimenting sphere simulation.

Name	Cell size (μm)	Mean Number of cells
G2	50.0 – 12.500	10000
G3	50.0 – 6.2500	24000
G4	50.0 – 3.1250	85000
G5	50.0 – 1.5625	290000

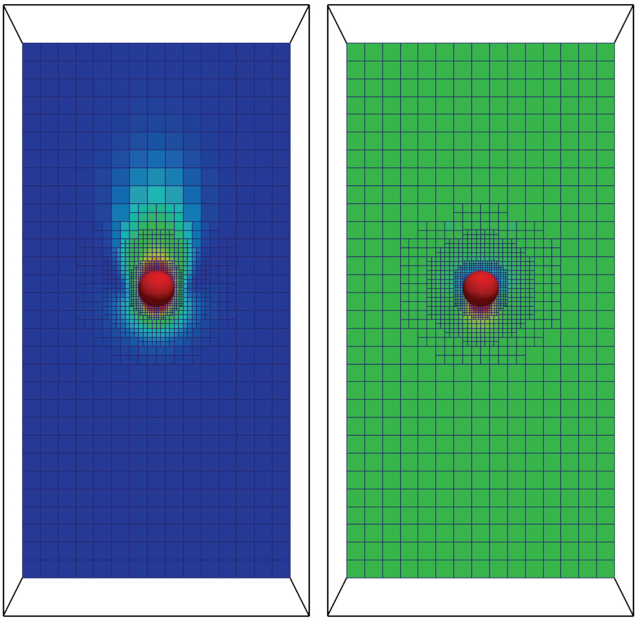

Figure 7: Left: Visualization of the velocity field through the center of the sedimenting sphere with density 1390 kg/m^3. Right: The fluid pressure around the sedimenting sphere. Red indicates high velocity or pressure.

resolved. The other simulated velocities are converging towards a grid independent solution. Furthermore, no oscillations in the velocity or force occur when the sphere enters or leaves a cell. To validate the simulated terminal velocity it is compared to the analytical solution;

$$v_t = \sqrt{\frac{8gr}{3C_d}\frac{\rho_s - \rho_f}{\rho_f}} \tag{11}$$

where C_d is given from Equation (10). In Figure 9 the simulated terminal velocity

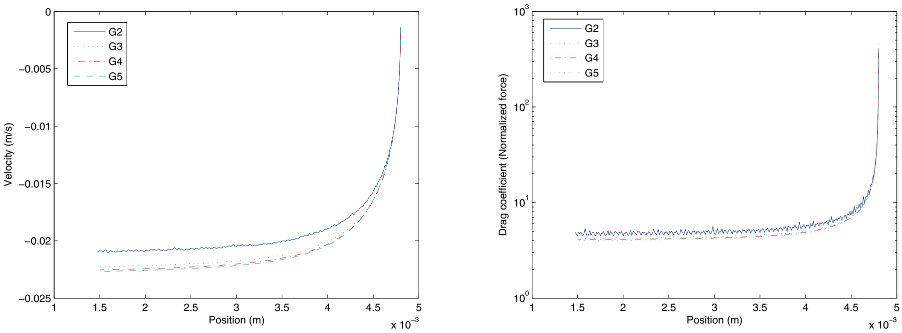

Figure 8: Left: Simulated terminal velocity for the different grids plotted against the vertical position. Right: The drag coefficient plotted against the vertical position.

on grid G3 are compared with the analytical solution for different sphere densities. The simulated terminal velocities agree well with the analytical results with a small deviation for large densities, where the boundary is not completely resolved.

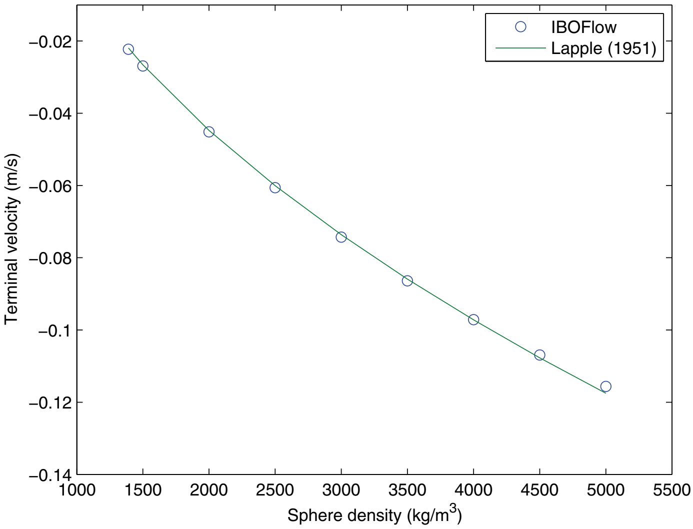

Figure 9: Simulated terminal velocity for different sphere densities compared to the analytical results given by Equation (11).

7 Conclusions

In this work the original mirroring immersed boundary method and a hybrid immersed boundary condition are compared, validated and implemented in IBOFLOW. The hybrid immersed boundary condition both mirrors and extrapolates the velocity field at the IB generating more immersed boundary control points. The fluid flow past a single sphere is simulated and the resulting drag forces are compared and validated against previous results with excellent agreement. Both immersed boundary conditions are shown to be strictly second-order accurate in space, where the hybrid method generates a smaller error due to more immersed boundary points. Finally a sedimenting sphere with a moving grid refinement is accurately simulated with a moving IB that does not generate any oscillations in the solution. Thus, the proposed method can be good alternative when performing DNS simulations on fluid-structure interaction problems.

An important application of the IBOFLOW software is for simulation of paint and surface treatment processes in automotive paintshops. These processes are very energy intensive and the increased demand on sustainability makes it necessary to further develop and optimize them. This research contributes to sustainable production by providing simulation tools that can be used by the automotive industry to reduce the time required for introduction of new car models, reduce the environmental impact and increase quality.

Acknowledgement: This work was supported in part by the Swedish Governmental Agency for Innovation Systems, VINNOVA, through the FFI Sustainable Production Technology program, and in part by the Sustainable Production Initiative and the Production Area of Advance at Chalmers. The support is gratefully acknowledged.

References

Crank, J.; Nicolson, P. (1947): A practical method for numerical evaluation of solutions of partial differential equations of the heat conduction type. *Proc. Camb. Phil. Soc.*, vol. 43, pp. 50–67.

Doormaal, J. V.; Raithby, G. (1984): Enhancements of the simple method for predicting incompressible fluid flows. *Num. Heat Transfer*, vol. 7, pp. 147–163.

Glowinski, R.; Pan, T.-W.; Periaux, J. (1994): A fictitious domain method for Dirichlet problem and applications. *Comp. Meth. Appl. Mech. Eng.*, vol. 111, pp. pp. 283–303.

Glowinski, R.; Pan, T.-W.; Periaux, J. (1994): A fictitious domain method for external incompressible viscous flow modeled by Navier-Stokes equations. *Comp. Meth. Appl. Mech. Eng.*, vol. 112, pp. pp. 133–148.

Goldstein, D.; Handler, R.; Sirovich, L. (1993): Modeling a non-slip flow boundary with an external force field. *J. Comput. Phys.*, vol. 105, pp. 354–366.

Kim, D.; Choi, H. (2000): A second-order time-accurate finite volume method for unsteady incompressible flow on hybrid unstructured grids. *J. Comp. Phys.*, vol. 162, pp. pp. 411–428.

Kim, J.; Kim, D.; Choi, H. (2001): An immersed-boundary finite-volume method for simulations of flow in complex geometries. *J. Comp. Phys.*, vol. 171, pp. pp. 132–150.

Lai, M.-C.; Peskin, C. (2000): An immersed boundary method with formal second-order accuracy and reduced numerical viscosity. *J. Comp. Phys.*, vol. 160, pp. 705–719.

Lapple, C. E. (1951): Particle dynamics. *Ver. Deut. Ing.*, pp. Eng. Res. Lab., E.I. DuPont De Nemours and Co, Wilmington, Delaware.

Leveque, R.; Li, Z. (1997): Immersed interface methods for stokes flow with elastic boundaries or surface tension. *SIAM journal on scientific computing*, vol. 18, pp. 709–735.

Lima E Silva, A.; Silveira-Neto, A.; J.J.R., D. (2003): Numerical simulation of two-dimensional flows over a circular cylinder using the immersed boundary method. *J. Comp. Phys.*, vol. 189, pp. 351–370.

Lima E Silva, A.; Silveira-Neto, A.; J.J.R., D. (2004): Numerical simulation of two-dimensional complex flows over bluff bodies using the immersed boundary method. Technical report, School of Mechanical Egineering, Federal University of Uberlandia, Brazil, 2004.

Majumdar, S.; Iaccarino, G.; Durbin, P. (2001): Rans solvers with adaptive structured boundary non-conforming grids. Technical report, Center for Turbulence Research, Annual Research Briefs, 2001.

Mark, A. (2008): *The Mirroring Immersed Boundary Method.* PhD thesis, Chalmers University of Thechnology, Göteborg, 2008.

Mark, A.; van Wachem, B. (2008): Derivation and valdiation of a novel implicit second-order accurate immersed boundary method. *J. Comput. Phys.*, vol. 227, pp. 6660–6680.

Mohd-Yusof, J. (1997): Combined immersed-boundary/b-spline mehtods for simulations of flow in comlex geometries. Technical report, Center for Turbulence Research, Annual Research Briefs, 1997.

Mohd-Yusof, J. (1998): Development of immersed boundary methods for complex geometries. Technical report, Center for Turbulence Research, Annual Research Briefs, 1998.

Oliveira, J.; A.L.F., L.; Silveira-Neto, A. (2005): Numerical simulation of high reynolds number flows over circular cylinders using the immersed boundary method with turbulence modelling. Technical report, Mechanical Engineering College, 2005.

Peskin, C. (1977): Numerical analysis of blood flow in the heart. *J. Comp. Phys.*, vol. 25, pp. 220–252.

Rhie, C.; Chow, W. (1983): Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA J1*, vol. 21, pp. 1527–1532.

Sharma, N.; Patankar, N. (2005): A fast computation technique for the direct numerical simulation of rigid particulate flows. *J. Comp. Phys.*, vol. 205, pp. 439–457.

Sharma, N.; Y., C.; Patankar, N. (2005): A distributed lagrange multiplier based computational method for the simulation of particulate-Stokes flows. *Comput. Methods Appl. Mech. Engrg.*, vol. 194, pp. 4716–4730.