# A Fully Coupled and Implicit Immersed Boundary Method for Multiphase Flows

José E.S. Oliveira, Andreas Mark and Berend G.M. van Wachem

Chalmers University of Technology, Applied Mechanics

412 96 Gothenburg, Sweden

jesoliveira@gmail.com

**Keywords**: immersed boundary method, fully couple solver, flow around a sphere

## Abstract

This paper focusses on the implementation and validation of a distributed force immersed boundary (IB) method and a fully implicit mirroring IB method in the MultiFlow code. MultiFlow is an in-house code, multiblock, multiprocessor, fully coupled solver, develop to solve various models applicable to multiphase flows. Its core is a fully coupled finite volume CFD solver. This code employs momentum weighted interpolation to determine analytical expressions for the cell face velocities which are employed in the multiphase continuity equation in a collocated variable arrangement. A special approach is adopted for the momentum weighted interpolation to handle large source terms, volume fractions, and gradients of these. The resulting linearized equations are solved in a fully coupled manner.

Two different methods are currently implemented, one method is explicit and first order accurate and the other is a no-val and implicit, second order accurate method for rigid bodies. The new method imposes the effect of an immersed body on the flow by directly modifying the Navier-Stokes coefficients obtained from its discretization. The application of the coefficients obtain the desirable boundary condition at the first time-step iteration. Details of implementation and results of numerical simulation of the fluid flow around a stationary sphere using a very simple Cartesian grid are presented. Both methods show excellent results compared to experimental data found in the literature.

## Introduction

Multiphase flow problems span a wide area of industrial processes and the most natural environments. The multiphase flow is often associated to different length scales which results in a difficult set of problems. Although still very much under development, computational fluid dynamics (CFD) has been shown to be a valuable research tool in the multiphase flow area. CFD can be helpful to improve the understanding of the fluid dynamics at particle scale. A deep understanding of the physical phenomena is required to control and consequently optimise the global processes. A number of numerical methods have been developing to simulate multiphase flows. However, many progress is still needed especially an accurate computations of flows interactions with complex solid boundaries.

Different approaches can deal with flow around moving complex geometries, probably the first and most well-know is the Arbitrary Lagrange-Euler formulation presented by Hirt *et al.* (1974) and follow by many other articles. The ALE methods employ a body fitting mesh to model the complex geometries and the movement of the fluid solid interface is handling by a remeshing procedure. In another hand there are non-body fitting meshes: immersed boundary methods Peskin (1977), Cartesian grid method (Melton *et al.* 1995) and the fictitious domain method (Glowinski *et al.* 2001).

The immersed moundary methods have been pioneered by Peskin (1977), and are capable of simulating the flow around a body using a mesh which does not conform to the body geometry. With this methodology, it is in principle possible to determine the flow around any geometry while employing even a simple Cartesian grid. The mesh used to represent the immersed body is completely independent, which allow the body displaces relative to the fixed grid used to discretize the fluid equations. The IB method becomes an efficient alternative to the classical body-fitted/re-meshing methods to handle moving boundaries problems. The immersed boundary method has been successfully applied in many fields. A good review of studies about IB method and its applications can be found in Peskin (2002) and Mittal & Iaccarino (2005).

Since Peskins original work, a number of methods have been proposed aimed at the same objectives. Two of these methods are subject in this work. The first method, called Virtual Physical Model was proposed by Lima e Silva *et al.* (2003). The VPM method is based on the force distribution concept. A force field is calculated over a sequence of Lagrangean points, which represent the interface and then Lagrangian forces are smeared out over the fluid grid. The

force distribute methods are probably the most versatile class of IB methods, with the same formulation is possible simulate problems involving rigid bodies, moving boundaries, elastic membranes and bubbles. However, these methods are explicit and first only order accurate.

Another class of IB methods tries to impose the interface boundary condition avoiding the distribution procedure. This is the aim of a new method develop by Mark & van Wachem (2007) named Mirroring Immersed Boundary Method. The method is implicit and second order accurate. The MIBM method imposes the effect of the immersed body on the flow by directly modifying the Navier-Stokes coefficients obtained from the discretization process, this procedure is very robust because it inherently results in a diagonally dominant system to solve. Therefore, the desirable boundary condition is reached at the first time-step iteration. The method was originally implemented on a staggered scheme in a segregated Navier-Stokes solver employing the SIMPLEC algorithm to the pressure-velocity coupling.

In this work, the IB models was implemented in a fully coupled solver, MultiFlow (2007). The numerical method is based on a finite volume approach on a collocated variable arrangement with a fully coupled pressure-velocity solver. The numerical solution by a fully coupled solver is an attractive and emerging trend in CFD. The main advantage of this approach is an increased robustness of the solution procedure due to the implicit treatment of the pressure-velocity coupling. The benefit of a fully coupled framework to solve the equations governing the flow takes advantage of the strong physical coupling which inherently lies in the Navier-Stokes type of equations. This approach is particularly interesting in problems involving fluid-structure interaction, like vortex-induced vibration, and free falling particles.

Here details of implementation and results of numerical simulation of the fluid flow around a stationary and spinning sphere using a very simple Cartesian grid are presented. Both methods shows accurate results compared against experimental data found in the literature. Moreover, preliminary results of currently under development features of IB methods in MultiFlow are also presented.

## Nomenclature

| | | |
|---|---|---|
| $A$ | projected area | $m^2$ |
| $C_d$ | drag coefficient | $-$ |
| $C_l$ | lift coefficient | $-$ |
| $d$ | diameter | $m$ |
| $F$ | force density | $N\ m^{-3}$ |
| $f$ | force | $N$ |
| $g$ | gravitational constant | $m\ s^{-2}$ |
| $h$ | distance | $m$ |
| $p$ | pressure | $N\ m^{-2}$ |
| $Re$ | Reynolds number | $-$ |
| $u$ | fluid velocity | $m\ s^{-1}$ |
| $U_\infty$ | free stream velocity | $m\ s^{-1}$ |
| $t$ | time | $s$ |
| $x$ | coordinate vector | $m$ |
| $w$ | weight coefficients | $-$ |

*Greek letters*

| | | |
|---|---|---|
| $\gamma$ | velocity ratio | $-$ |
| $\mu$ | dynamic viscosity | $kg\ m^{-1}\ s^{-1}$ |
| $\rho$ | density | $kg\ m^{-3}$ |
| $\tau$ | stress tensor | $N\ m^{-2}$ |
| $\omega$ | angular velocity | $rad\ \ s^{-1}$ |

*Abbreviations*

| | |
|---|---|
| ALE | Arbitrary Lagrange-Euler |
| CFD | Computational Fluid Dynamics |
| IB | Immersed Boundary |
| MIBM | Mirrored Immersed Boundary Method |
| VPM | Virtual Physical Model |

## Single Phase Flow

In single-phase flow is modeled by the Navier-Stokes equations. The continuity equation, Eq. (1), and the momentum equation, Eq. (2), for viscous and incompressible flows can be written as:

$$\frac{\partial u^i}{\partial x^i} = 0, \tag{1}$$

$$\rho\frac{\partial u^j}{\partial t} + \rho\frac{\partial}{\partial x^i}\left(u^i u^j\right) = -\frac{\partial p}{\partial x^j} + \frac{\partial \tau_{ij}}{\partial x^i} - Ru^j - S^j, \tag{2}$$

where source terms are given by $S^j$, which is a general source term and a linearized source term $R$. The $\tau_{ij}$ represents the stress tensor:

$$\tau_{ij} = \mu\left(\frac{\partial u^i}{\partial x^j} + \frac{\partial u^j}{\partial x^i}\right). \tag{3}$$

The cell centered finite volume discretization at $P$ of the velocity terms for one direction of the Eq. (2), is of the form:

$$\left[\frac{\rho V_P}{\Delta t} + a_P^{(u^j)} + V_P\, R_P^{(u^j)}\right] u_P^j = \left[\sum_{nb} a_{nb}^{(u^j)} u_{nb}^j\right]_P - V_P\left[\sum_{nb} b_{nb}^{(j)} P_{nb} + S^{u^j}\right] + \left[\frac{\rho V_P}{\Delta t}\right]_P u_P^{j,O}, \tag{4}$$

where $a$ is a coefficients that combine the convective and shear terms, and $b$ is obtained from the discretization of the pressure gradient.

*Discretization of the Continuity Equation*

Considering the single phase flow continuity equation, Eq. (1), in its discretized form

$$\sum_{f=\text{faces}} M_f = \sum_{f=\text{faces}} u_f^i s_f^i = 0 \tag{5}$$

The coefficients for the continuity equation are determined by the pressure-weighted interpolation Rhie & Chow (1983). The objective is to obtain an expression for the face velocity $u_f^i$ from the averaged the momentum equations. The net force driving the flow in Eq.(4) is given by, and now is redefined as

$$\widetilde{\frac{\partial p}{\partial x^j}} = \left[\frac{\partial p}{\partial x^j} + S^{u^j}\right] \tag{6}$$

It is important to treat the pressure gradient and the sources similar, as their sum is the contribution to accelerating the fluid. With this, Equation 7 becomes

$$\left[1 + \frac{\rho}{\Delta t}\frac{V_P}{a_P^{(u^j)} + V_P\, R_P^{(u^j)}}\right] u_P^j = \left\{\frac{\left[\sum_{nb} a_{nb}^{(u^j)} u_{nb}^j\right]_P}{a_P^{(u^j)} + V_P\, R_P^{(u^j)}}\right\} - V_P \frac{\left[\widetilde{\frac{\partial p}{\partial x^j}}\right]_P}{a_P^{(u^j)} + V_P\, R_P^{(u^j)}} + \left[\frac{\rho}{\Delta t}\right]_P \frac{V_P}{a_P^{(u^j)} + V_P\, R_P^{(u^j)}} u_P^{j,O} \tag{7}$$

Define the follow variables,

$$c = \frac{\rho}{\Delta t} \tag{8}$$

$$d^{(u^j)} = \frac{V_P}{a_P^{(u^j)} + V_P\, R_P^{(u^j)}} \tag{9}$$

$$\widetilde{u^j} = \left\{\frac{\left[\sum_{nb} a_{nb}^{(u^j)} u_{nb}^j\right]_P}{a_P^{(u^j)} + V_P\, R_P^{(u^j)}}\right\} \tag{10}$$

With these abbreviations, Equation 7 for cell $P$, becomes

$$\left[1 + c_p\, d_P^{(u^j)}\right] u_P^j = \widetilde{u^j}_P - d_P^{(u^j)} \left[\widetilde{\frac{\partial p}{\partial x^j}}\right]_P + c_P\, d_P^{(u^j)} u_P^{j,O} \tag{11}$$

A similar equation can be written for all cell centers. Figure 1 shows two non-Cartesian centered cells. To proceed with discretization of the continuity equation, the values of the velocities are require at cell faces or integration points. In the Rhie and Chow approach, an analogous equation can be written down for the velocity at point $e'$.

$$\left[1 + c_{e'}\, d_{e'}^{(u^j)}\right] u_{e'}^j = \widetilde{u^j}_{e'} - d_{e'}^{(u^j)} \left[\widetilde{\frac{\partial p}{\partial x^j}}\right]_{e'} + c_{e'}\, d_{e'}^{(u^j)} u_{e'}^{j,O} \tag{12}$$

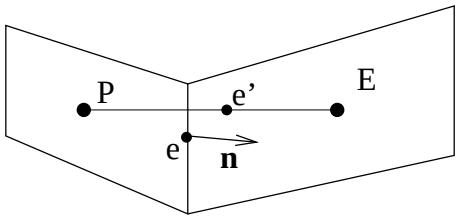


**Figure 1:** A two-dimensional boundary fitted cell ($P$) with its east neighbour ($E$). The center of the face between $P$ and $E$ is shown by point $e$, and the local normal at this point is given by $n^i$.

The point $e'$ lies in the center of the line connecting the nodes $P$ and $E$, therefore the weighting coefficient will be exactly $\frac{1}{2}$. An expression for the velocity $\widetilde{u^j}_{e'}$ is obtained by linear interpolation using the velocity expressions for points $E$ and $P$.

$$\widetilde{u^j}_{e'} = \frac{1}{2}\left(\widetilde{u^j}_P + \widetilde{u^j}_E\right) \tag{13}$$

The expression for $\widetilde{u^j}_{e'}$, can be substituted in Eq. (12). The face velocity at point $e$ is determined based upon the face velocity in point $e'$, corrected with the gradient, for instance with a Taylor series:

$$\overline{u}_e^j = \overline{u}_{e'}^j + \overline{\frac{\partial u^j}{\partial x^i}}_{e'} (x_{e'}^i - x_e) \tag{14}$$

Equation (14) is used to close the discretized continuity equation. Now, the continuity equation is dependent upon the pressure field, and a coupled formulation can be given as,

$$\begin{pmatrix} \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ p \end{pmatrix} = \begin{pmatrix} RH_{u_1} \\ RH_{u_2} \\ RH_{u_3} \\ RH_p \end{pmatrix} \tag{15}$$

*Boundary Conditions*

The boundary conditions for the pressure terms, in the momentum equations, can be determined by extrapolation, for a variable $\phi$ the extrapolation can be described as

$$\phi_f = \phi_P + \frac{\partial \phi}{\partial x^i}_P (x_W^i - x_f^i) \tag{16}$$

where $\phi_f$ denotes the value of variable $\phi$ at the wall. The above equation can be directly employed to close the equation for the face velocity.

At boundaries where Dirichlet condition for pressure is applied, velocity is typically extrapolated. In this case, the boundary pressure can be directly specified in both the continuity equation as well as the momentum equation. The velocity at such a boundary can be obtained from equation 7 by inserting the value of pressure at the wall.

*Further Details in MultiFlow*

In MultiFlow, the computational domain is partitioned into subdomains that are called blocks. Blocks are built with locally structured grids that representing specific part of the computational region. The blocks are connected to another blocks by boundary conditions. The multiblock grid is useful to model geometrically complex objects. Even modeling particles and arbitrary bodies immersed in the flow with IB methods, one can gain flexibility to simulate the real world problems using a multiblock mesh to represent the geometrical domain.
The parallel code developed is based upon the numerical libraries of PETSc (Balay *et al.* 2007, 2004, 1997) which provide a large suite of parallel linear and non-linear equation solvers and MPICH (2007) for additional parallel algorithms. In this work, a block-ILU preconditioner is employed with the BiCGSTAB Krylov subspace solver.

## Immersed Boundary Methods

Two different methods are currently implemented in the code, the virtual physical model (VPM), and the mirroring immersed boundary method (MIBM). The two methods are presented in the sections below.

*The Explicit Method: VPM*

The virtual physical model (VPM) was proposed by Lima e Silva *et al.* (2003), it is a distributed force method based on the original IB idea of Peskin (1977). In this method a force field is calculated over a sequence of Lagrangian points, which represent the interface, then the force are added as source term in the Navier-Stokes equations only over the interface.

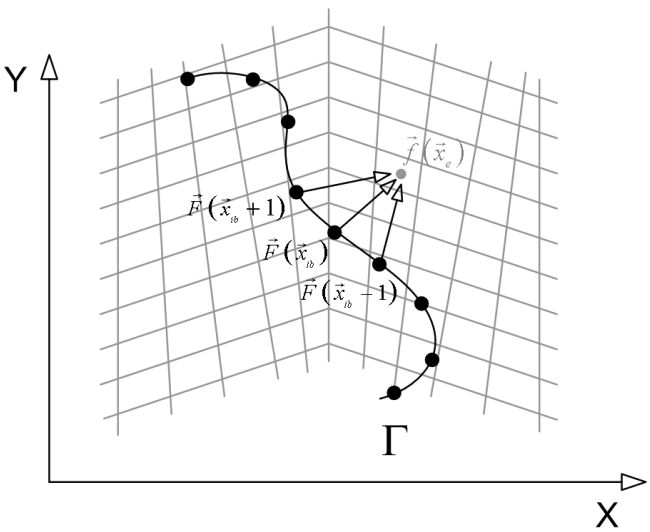


**Figure 2:** A 2D representation of the VPM method showing a distribution of the force field into fluid mesh points.

The VPM method gives dynamical evaluation of the force exerted by the fluid flow over the immersed body. The force density $\vec{F}(\vec{x}_{ib}, t)$ is calculated over the Lagrangian points using the Navier-Stokes terms:

$$\vec{F}^i(\vec{x}_{ib}, t) = \frac{\partial}{\partial t}u^i + \frac{\partial p}{\partial x^i} + \frac{\partial}{\partial x^j}\left(u^i u^j\right) - \frac{\partial \tau_{ij}}{\partial x^j}, \quad (17)$$

The different terms on the right hand side of Eq. (17) are referred as: acceleration force, pressure force, inertial force and viscous force. To evaluate the force terms in Eq. (17) the pressure and the velocity fields must be known previously. These fields are calculated on the Eulerian grid while the force terms must be calculated over the interface. The coupling between the two meshes is made by the follow equation:

$$\vec{f}(\vec{x}_e) = \int_\Gamma \vec{F}(\vec{x}_{ib},\ t)\ \delta(\vec{x}_e - \vec{x}_{ib})\ d\vec{x}_{ib}, \quad (18)$$

where $\vec{F}$ is the Lagrangian force density placed on $\vec{x}_{ib}$ points of the Lagrangian control-volume $\Gamma$ and $\delta(\vec{x}_e - \vec{x}_{ib})$ is the Dirac delta function.

*The Implicit Method: MIBM*

A new method called mirroring immersed boundary method (MIBM) develop by Mark & van Wachem (2007) was also implemented in our code. The new method imposes the effect of the immersed body on the flow by directly modifying the Navier-Stokes coefficients obtained from the discretization process, which impose the desirable boundary condition at the first time-step iteration.

In the MIBM method, the interface $\Gamma$ is used to map the fluid mesh defining a three subsets of points (see Figure 3): $\vec{x}_i$ points inside the interface, $\vec{x}_{iib}$ points inside and near the IB interface and $\vec{x}_e$ points outside the interface.

The method directly mirrors the external velocity over the interface $\Gamma$. The internal immersed-boundary velocities, $\vec{u}_{iib}$, are set in the opposite direction of the exterior normal velocities, $\vec{u}_e$, plus the immersed-boundary velocity, $\vec{u}_{ib}$, such that the velocity constraint is enforced at the interface $\Gamma$:

$$\vec{u}_{iib} + \vec{u}_e = \vec{u}_{ib}. \quad (19)$$

This is done by setting a fictitious exterior normal point, as depicted in Figure (4). The mirror point is defined as the $\vec{x}_{iib}$ mirrored along the normal of its closest interface point. Hence the smallest distance between the interface and the fictitious point $\vec{x}_e$ are the same.

$$\vec{x}_e = \vec{x}_{iib} + 2d\vec{n} \quad (20)$$

If the exterior normal point coincides with a discrete velocity point, the Dirichlet condition for that IIB velocity is trivial. More generally, the exterior normal point lies between the discrete velocity points and therefore the velocity needs to be implicitly interpolated using the surround points. The IIB velocity can then be set to the reversed interpolated velocity plus the boundary velocity. The interior velocities are set to

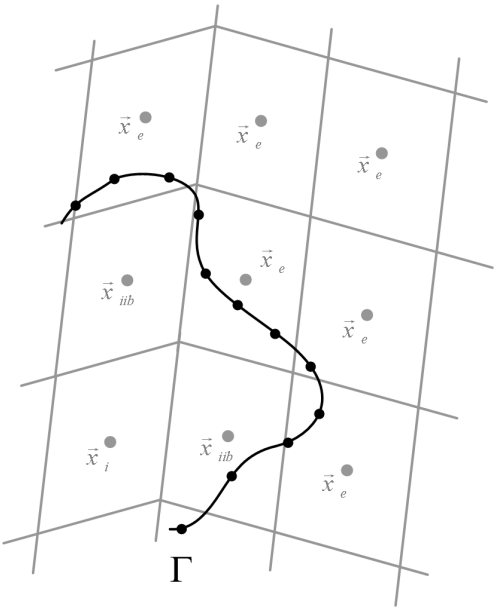


**Figure 3:** Mapping of the cell centered fluid cells, MIBM method.

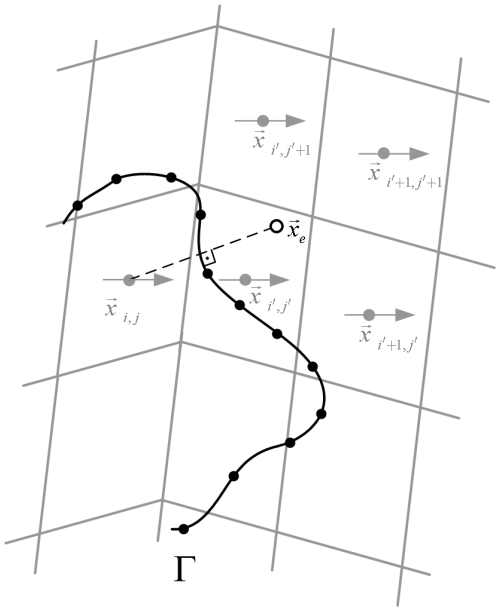


**Figure 4:** A 2D representation of the MIBM method, the exterior normal point $\vec{p}_e$ with its surrounding interpolation points.

the boundary (IB) velocity.

In both implemented methods, interpolation of various variables in the three dimensional discretizes spaces is frequently required. In this working an inversed weighted interpolation is used. The inverse distance weighted methods are based on the assumption that the interpolating points should be influenced most by the nearby points and less by the more distant points. Several options are available for inverse distance weighted interpolation; the simplest form is referred to as Shepard's method. The equation used is as follows:

$$w_i = \frac{h_i^{-2}}{\sum_{j=1}^{n} h_i^{-2}} \tag{21}$$

where $h_i$ is the distance from the scatter point to the interpolation point.

The inverse weighted distance is forced to have a minimum number of the data points, however the choice of the maximum number of data points is arbitrary. In this work a fixed amount of eight points ($n = 8$), that define an interpolation box surround the interpolation point, is employed.

*Calculation of the Forces*

For the MIBM method the total force consists of two contributions, i.e. the pressure force and the viscous force. These forces are integrated over the interface the interface Γ.

$$F_i = \int_\Gamma \left(-p\delta_{ij} + \tau_{ij}\right) n_j \, \mathrm{dS} \tag{22}$$

In order to proceed with the numerical calculation of the forces acting at the body, the interface Γ must be discretized.

In this work, the interface is discretized using triangles generated by employing the `gts` library (GTS 2007). The discretization of the integral is determined for each triangle and summed up to get the total surface force acting on the body. In order to discretize the force terms needs to be defined a set of auxiliary points, as depicted in Figures (5) and (6).

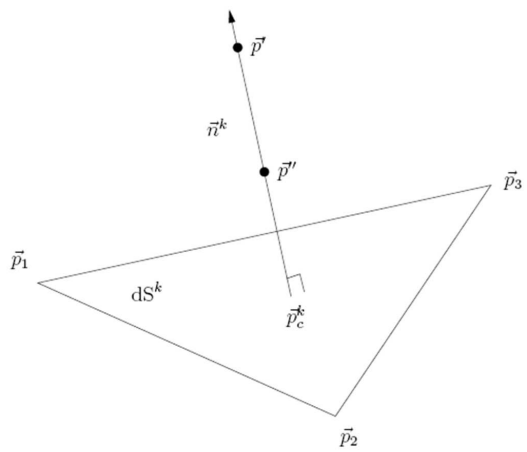


**Figure 5:** Triangle $k$ in the triangulation of the immersed-boundary interface Γ with the exterior points, $\vec{p}'$ and $\vec{p}''$.

The evaluation of the pressure force is done with the help of two auxiliary points $\vec{p}'$ and $\vec{p}''$ which lie on the normal of the triangle. The pressure is interpolated onto the auxiliary points and then extrapolated onto the centre of the triangle,

$$P_c = 2P'' - P' \tag{23}$$

To evaluate the viscous part, the tensor $\tau_{ij}$ is need at the

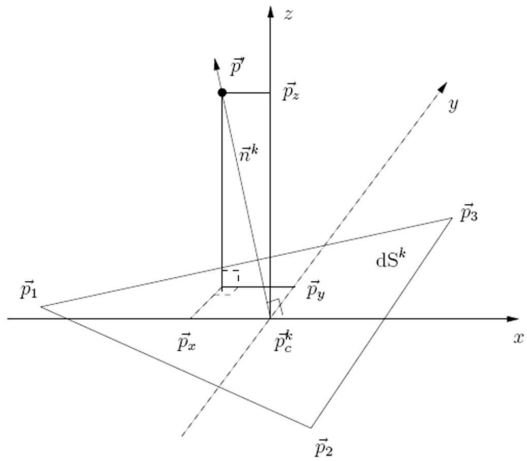


**Figure 6:** Triangle $k$ with the projection of $\vec{p}_p$ onto the coordinate axises ($\vec{p}_x$,$\vec{p}_y$,$\vec{p}_z$).

center of each triangle. This is done by introducing three new points, $\vec{p_x}'$, $\vec{p_y}'$ and $\vec{p_z}'$ which are the projections of auxiliary point $\vec{p}'$ onto the Cartesian axes. The fluid velocities are interpolated onto these points. Details of the discretization process can be found in Mark & van Wachem (2007).

In the VPM method the forces are explicit calculated using the fluid properties of the previous time step. Besides the pressure and viscous force, the VPM method needs additional force terms as shown in Equation (17). In this work, for the sake of simplicity, the additional terms are combine into a single term called *forcing term.* This term is responsible to assure the non-slip condition at the interface, it is aproximated by $(\vec{u}_\Gamma - \vec{u}_{\vec{p}_c})/\Delta t$. The reader interested in more detail of the discretization should refer to Lima e Silva *et al.* (2003).

## Results and Discussion

*Stationary Sphere*

The computational domain is a unitary cube discretized by $60 \times 60 \times 60$ volumes. The sphere shell generated has $1280$ trigangles. The sphere center is placed at the center of computational domain and it has a diameter $D = 0.1m$. The boundary condition for the fluid velocity components on the side walls is considered free-slip and Neumann condition on the outlet and the inlet has a constant profile $v = U_\infty$. The pressure is set to zero in the outlet and Neumann conditions for all other faces. The time step employed is $10^{-3}$ for explicit method and $10^{-2}$ for the fully implicit method.

The drag coefficient on a sphere immersed in a crossflow is defined as,

$$C_d = \frac{2 f_d}{\rho_f \mu A U_\infty^2} \tag{24}$$

where $f_d$ represents the drag force, $A$ the projected frontal area and $U_\infty$ the free stream velocity of the fluid in the direction of the drag force.

The drag coefficient has a well known relationship with the Reynolds number based on the sphere diameter, see Tab. (1). This relationship can be used to validate the implementations of the IB methods.

**Table 1:** The drag on a sphere immersed in a fluid.

| References | Range | Correlation for $C_d$ |
|---|---|---|
| Stokes | $\mathrm{Re} \ll 1$ | $\frac{24}{\mathrm{Re}}$ |
| Oseen (1910) | $\mathrm{Re} < 1$ | $\frac{24}{\mathrm{Re}}(1 + \frac{3}{16}\mathrm{Re})$ |
| White (1991) | $\mathrm{Re} < 2 \times 10^5$ | $\frac{24}{\mathrm{Re}} + \frac{6}{1+\sqrt{Re}} + 0.4)$ |

Figure (7) shows the numerical results for the drag coefficient, it can be seen that the two methods achieved an accurate prediction for the drag force up to Reynolds $10$. However, in order to obtain the same accuracy, the explicit method requires a time step $10$ times smaller. The implicit method is more robust and provides a precise solution of the viscous and pressure force, the results is shown in Fig. (8).

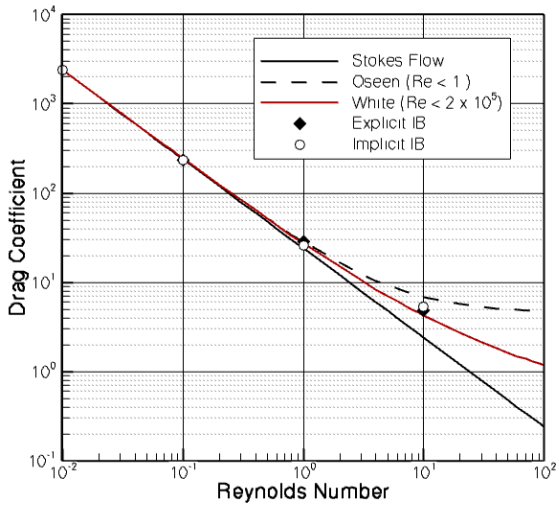


**Figure 7:** Simulated drag coeficient.

A simulation of a flow around a sphere at $Re$ number $10$ is visualized in Figure (9), the surface of sphere is colored with the pressure distribution. The final steady state solution of the velocity field and the flow streamlines can be visualized in a cut plane in the center of the domain.

Both methods have been previously validated for high Reynolds numbers as shown by Mark & van Wachem (2007) and Oliveira *et al.* (2005).

*Rotating Sphere*

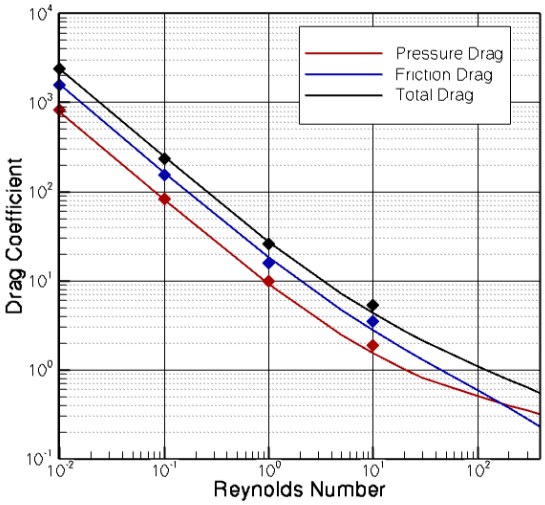


**Figure 8:** Viscous and pressure contribution for drag coefficient, calculated by MIB method.

In order to illustrate the imposition of a different boundary condition at the interface, a rotating sphere immersed in a uniform crossflow is considered. The lift force develop due to the rotation of the sphere are show in Fig. (10) is called Magnus effect. The sphere rotation produce an asymmetry in the velocity field causing a pressure differential between the two sides of the sphere.

The rotational speed of the sphere is usually represented by a ratio of the tangential velocity of the sphere to the fluid stream velocity

$$\gamma = \frac{d\omega}{2U_\infty} \tag{25}$$

A theoretical solution for $Re_D < 1$ has been developed by Rubinow & Keller (1961) and shows that the lift force, for a rotating sphere, can be expressed in terms of the rotational speed ($\gamma$),

$$C_l = 2\gamma \tag{26}$$

The lift force coefficient calculated numerically is given by

$$C_l = \frac{2f_l}{\rho_f \mu A U_\infty^2} \tag{27}$$

Figure (11) shows a comparison between lift coefficients predicted by numerical simulation using the MIBM method, for $Re_D = 0.5$ and the theoretical data. These predictions agree well with theory. This confirms a good accuracy of the method to impose a desirable boundary condition at the interface.

The flow pattern is shown in the Figure (12), can be observed the good achieved of the no-slip boundary condition on the surface sphere. The velocity of the fluid near the sphere is gradually modified by the influence of the rotation of the sphere, producing a strong flow asymmetry.

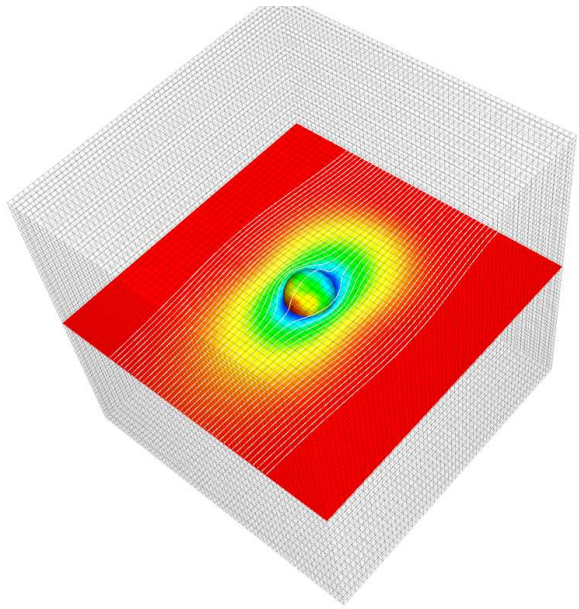

**Figure 9:** Simulation of the flow around a sphere, $Re_D = 10$, using a simple cartesian grid.

*Computational Performance*

The implementation of the IB methods implies an extra computational cost, to perform all interpolation procedures and setup the appropriated IB coefficients. The impact of this implementation depends, of course, of the refinement of the triangulation in the interface. Considering a fluid grid of $40 \times 40 \times 40$ at Reynolds 1, the extra cost to insert a immersed sphere with $1280$ triangles is about $13\%$, including here the calculus of the forces in each time step. The cost only due to setting the IB coefficients is less than $4\%$ for this simulation.

The performance of the MIBM method in the fully coupled solver was also compared against the segregated solver. MultiFlow was $9\%$ higher than the segregate solver, regards the computational time. The elapsed computer time for $100$ times steps to simulate $0.1s$ physical time, carried out on an AMD64 Athlon with 2 GHz was about $34\ min$. The simulations was performed using, in both codes, the time step restriction imposed by the segregated solver. In fact the coupled can converge employing higher times steps, therefore the global performance of the couple solver is superior without time steps restrictions.

*Future Developments*

This section are showing preliminary results of the currently under development features in MultiFlow.

In Figure (13), simulation of a falling disk is visualized. Initially, the disc and fluid are at rest. The density ratio between the body and the fluid is $\rho_d/\rho_f = 2$ and the disc begins to accelerate by action of the gravity. The picture is showing the falling disc at different times, colored by the pressure field, the cutting plane shown the modulus of the

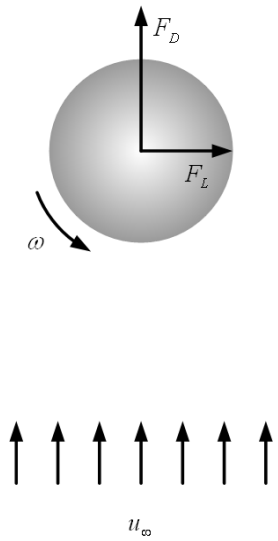


**Figure 10:** Magnus force (lift) due to sphere rotation.

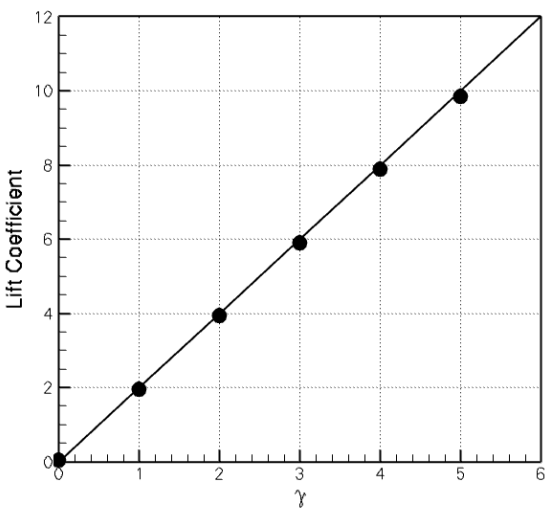


**Figure 11:** Comparison between theoretical and simulated results for $C_l$ at $Re_D = 0.5$.

instantaneous fluid velocity.

A simulation of flow around two spheres is shown in Figure (14). This simulation was performed in a cylindrical domain discretized with a multi-block structured mesh. The figure shows the multi-block mesh employed and the fluid velocity vectors along the domain in a center plane.

## Conlusions

In this paper we describe the derivation, implementation and validation of two immersed boundary (IB) methods in a fully coupled in-house code MultiFlow. The two different methods are the virtual physical model (VPM) and the mirrored immersed boundary method (MIBM). It is shown that both methods successfully simulate the fluid flow at low

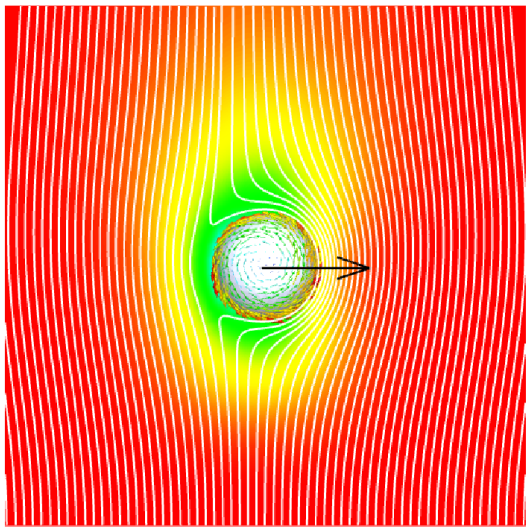

**Figure 12:** Final stead steady solution of a rotating sphere in a crossflow at $Re_D = 0.5$.

Reynolds number around a sphere; the results are showing good coherence with the expected physical phenomena and literature data. Due to the fully coupled framework, very low computational cost is required to simulate the cases.

The VPM is and explicit and first order accurate method employing a force distributed around the immersed boundary. It requires relatively smaller times steps and a fine distribution of the control points on the immersed boundary to achieve good accuracy. The MIBM is fully implicit and second order accurate in space and therefore more robust and stable, with a lower computational cost. The MIBM method has geometric restrictions and it is able to simulate rigid and close bodies. The both methods confers a good flexibility to MultiFlow, the code is able to simulate a wide range of fluid-structure problems.

To show the potential of the fully implicit MIBM method in the fully coupled flowsolver,the flow around a falling non-spherical body and two spherical particles in a fluid are resolved using a multiblock, non-Cartesian mesh.

## Acknowledgements

This work has been financially supported by the Bo Rydin Research Foundation and the Swedish Research Council (VR).

## References

van Wachem, B. & Gopala, V.R. & Oliveira, J.E.S. & Mark, A. & Benavides, A. & Hemph, R., MultiFlow - A Coupled Multiphase Flowsolver, `www.multiflow.org`, (2007).

GTS - GNU Triangulated Surface Library, `gts.sourceforge.net`, (2007).

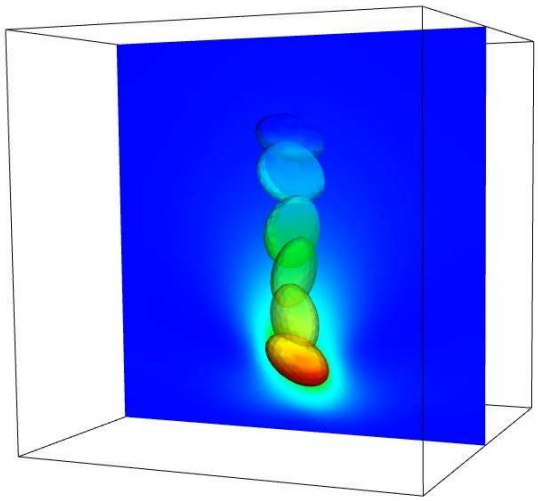

**Figure 13:** A transient simulation of a falling disc in a resting fluid.

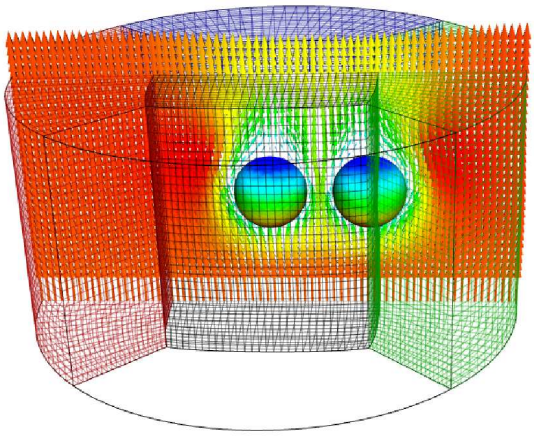

**Figure 14:** A simulation of two spheres in a non-Cartesian multiblock mesh.

The MPI message passing interface, `www-unix.mcs.anl.gov/mpi/mpich/`, (2007).

Mark, A. & van Wachem, B., A Novel Implicit Second-Order Accurate Finite-Volume Immersed Boundary Method, Submmited to Journal of Computational Physics, (2007).

Hirt, C.W. & Amsden, A.A & Cook J.L., An arbitrary Lagrangian-Eulerian computing method for all speeds. Journal of Computational Physics, Volume 14, 227–253. (1974)

Glowinski, R. & Pan, T.W & Hesla, T.I. & Joseph. D.D. & Periaux, J., A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: application to particulate flow, Journal of Computational Physics, Volume 169, 363–426. (2001)

Melton, J.E. & Berger, M.J. & Aftosmis, M.J. & Wong, M.D., 3D Applications of a Cartesian grid Euler method. AIAA Paper 95-0853, (1995).

Mittal, R. & Iaccarino, G., Immersed boundary methods, Ann. Rev. Fluid Mech, Volume 37, 239–261 (2005).

Peskin, C., The immersed boundary method. Acta Numerica, 1–39 (2002).

Lima e Silva, A.L.F., Silveira-Neto, A. and Damasceno, J.J.R., Numerical simulation of two-dimensional flows over a circular cylinder using the immersed boundary method, Journal of Computational Physics, Volume 189, 351–370 (2003).

Peskin, C. S., Numerical Analysis of Blood Flow in the Heart, Journal of Computational Physics, Volume 25, 220–250 (1977).

Balay, S. & Buschelman, K. & Eijkhout, V. & Gropp, W.D. & Kaushik, D. & Kneple. M.G. & McInnes, L.C. & Smith, B.F. & Zhang, H., PETSc Users Manual, ANL-95/11 - Revision 2.1.5, Argonne National Laboratory, (2006).

Balay, S. & Buschelman, K. & Eijkhout, V. & Gropp, W.D. & Kaushik, D. & Kneple. M.G. & McInnes, L.C. & Smith, B.F. & Zhang, H., PETSc Users Manual, ANL-95/11 - Revision 2.1.5, Argonne National Laboratory, (2004).

Balay, S. & Buschelman, K. & Eijkhout, V. & Gropp, W.D. & Kaushik, D. & Kneple. M.G. & McInnes, L.C. & Smith, B.F. & Zhang, H., PETSc Web page, http://www.mcs.anl.gov/petsc, (2007).

Balay, S. & Gropp, W.D. & McInnes, L.C. & Smith, B.F., Efficient Management of Parallelism in Object Oriented Numerical Software Libraries, pp. 163–202, Birkhäuser Press, (1997).

Oseen, C.W., Über die Stokes' sche Formel und über eine verwandte Aufgabe in der Hydrodynamik, Ark. Math. Astronom. Fys, Volume 6, 29 (1910).

Oliveira, J.E.S., & Lima e Silva, A. & Silveira-Neto, A., Numerical simulation of the high reynolds number flow over circular cylinders using the immersed boudary method, In 18th International Congress of Mechanical Engineering, Rio de Janeiro, Brazil (2005).

Rhie, C.M. & Chow, W.L., Numerical study of the turbulent flow past an airfoil with trailing edge separation, AIAA J1, Volume 21, 1527–1532 (1983).

Rubinow, S.I. & Keller, J.B., The transverse force on a spinning sphere moving in a viscous fluid, Journal of Fluid Mechanics, Volume 11, 447–459 (1961).

White, F.M., Viscous fluid flow, 2nd Ed., McGraw-Hill, New York, (1991).