Application of the Immersed Boundary Method for the Simulation of Incompressible Flows in Complex and Moving Geometries

Eike Hylla, Octavian Frederich, Johannes Mauß, and Frank Thiele

Berlin Institute of Technology, Institute of Fluid Mechanics and Engineering Acoustics,
Müller-Breslau-Str. 8, 10623 Berlin, Germany
eike.hylla@cfd.tu-berlin.de
http://www.cfd.tu-berlin.de

Summary

A new variant of the *Immersed Boundary Method* (IBM) has been implemented into an established flow solver. Important aspects of the implementation towards the application of this approach for flow simulations in complex and moving geometries are characterised. Simple validation test cases are addressed first, followed by a moving boundary example and more complex geometries like the Weibel lung model.

1 Introduction

The generation of structured body-fitted computational grids is time-consuming and runs into problems, when the geometry is complex. Moving geometries can be handled by grid deformation algorithms but are limited to small deviations in general. The IBM is a numerical approach, which can avoid these problems. The method was first introduced by Charles Peskin in 1972, for the simulation of blood flow through heart valves [1]. Instead of a body-fitted grid, two grids are used: a Cartesian computational grid and one to represent the surface of the body. Imposition of the boundary conditions at the surface grid, which is *immersed* into the Cartesian grid, is the main issue of the IBM. The essential advantage of this method is the distinction between the computational grid and the surface description. This makes it possible to consider complex and moving or deforming geometries during the simulation. Today, further developments and numerous variations of this method exist. While Peskin [2] applied a *continuous forcing approach* modifying the governing equations to satisfy the boundary conditions, most variations use a *discrete forcing approach* [3], where the boundary conditions are imposed following the discretization on the Cartesian grid. In the framework of the *ghost-cell-method* [4] cells outside of the flow field but next to the surface are used to fulfill the boundary conditions by interpolating appropriate velocities for these ghost-cells.

A. Dillmann et al. (Eds.): Numerical & Experimental Fluid Mechanics VII, NNFM 112, pp. 135–142.
springerlink.com

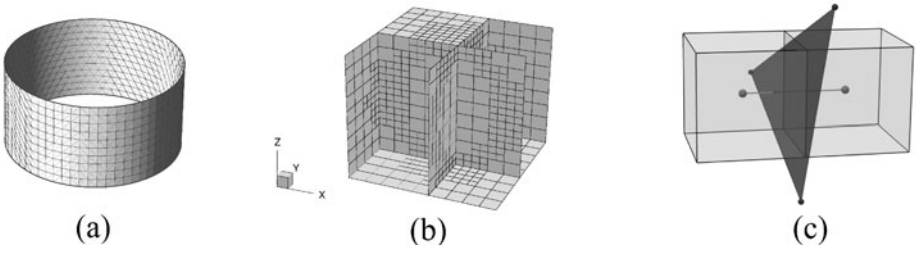

Fig. 1. Surface grid (a), Cartesian domain grid (b), adjacent cells hit by IB triangle (c)

2 Fundamentals of Implementation

A new kind of the ghost-cell-method [5] is implemented into the finite-volume based in-house flow solver [6]. The advantage compared to other established methods is, that the algorithm neglects the calculation of the non-fluid *and* the ghost-cells. Interpolated values of the ghost-cells are shifted to the source term of adjacent inner cells (cf. 2.3). In the following important aspects of the implementation are described.

2.1 Surface and Domain Grids

The surface of the geometry can be easily discretized using interconnected triangles (fig. 1 a). Complex or curved surfaces may require a finer resolution to give an appropriate discretization of the geometry. To distinguish between different kinds of boundary conditions, the associated triangles are grouped separately.

The domain grid (fig. 1 b) has a strict Cartesian 3d structure with orthogonal cells aligned to the Cartesian directions x, y, z. Isotropic grid refinement employing hanging node elements (with a split ratio of 1:2) can be used to effect a higher resolution near the IB. Therefore a mesh generator has been set up, which improves the local mesh resolution by subdividing cells depending on given threshold distances to the surface. The high number of possible refinement levels requires an unstructured data format for the domain grid, leading to the necessity of an unstructured solver for the equation system (see section 2.5). The computational grid passes a preprocessing step, where the decomposition for multiprocessing purpose is done. The shape and structure of the coefficient matrix is detected and stored in *Compressed Row Storage* format [7], which is best suited for sparse matrix handling.

2.2 Domain Identification and Marking Algorithm

Before the simulation, the identification and marking of the computational domain is necessary to distinguish between fluid and non-fluid regions. The neighboring cells of element faces to be divided by the IB must therefore be found (fig. 1 c). To achieve that, each domain face has to be checked with every IB triangle for possible intersection points. Those points and hit faces are stored. The areas intersected by the immersed boundary are gradually marked with a painter variable [5].

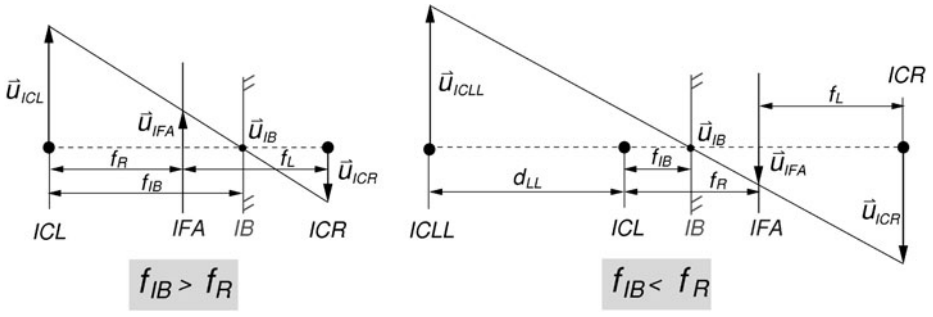

Fig. 2. Different velocity interpolations at IB walls, depending on the intersection point

2.3 Imposition of Boundary Conditions

In the present simulation software, inflow, outflow and wall boundary conditions can be set using IB's. The general approach for all of these boundary conditions is the same. The difference between this method and other established ghost-cell approaches is, that the ghost-cells, which are the non-fluid neighbors of fluid cells, are really not being calculated. Within the flow field discretization for an arbitrary variable Φ leads to an equation for each cell p, including coefficient contributions $a_\square$ from the neighboring cells nb and a source term S_ϕ:

$$\Phi_p a_p + \sum \Phi_{nb}\, a_{nb} = S_\Phi$$

As the calculation of Φ neglects all non-fluid cells, contributions from neighboring ghost-cells are gained by interpolation and shifted to the source term:

$$\Phi_p a_p + \sum \Phi_{nb}^{in}\, a_{nb}^{in} = S_\Phi - \sum \Phi_{nb}^{ghost}\, a_{nb}^{ghost}$$

The no-slip wall is the most important boundary condition for complex geometries and its treatment is discussed in the following. Velocities $\boldsymbol{u}$ of the ghost-cells and the adjoining cell face are linearly interpolated in one direction (see fig. 2) to fulfill the no-slip condition on the IB ($\boldsymbol{u}_{IB} = 0$ in the non-moving case):

$$\left.\begin{aligned} \boldsymbol{u}_{IFA} &= \boldsymbol{u}_{ICL} \cdot \left(1 - \frac{f_R}{f_{IB}}\right) + \boldsymbol{u}_{IB} \cdot \left(\frac{f_R}{f_{IB}}\right) \\ \boldsymbol{u}_{ICR} &= \boldsymbol{u}_{ICL} \cdot \left(1 - \frac{1}{f_{IB}}\right) + \boldsymbol{u}_{IB} \cdot \left(\frac{1}{f_{IB}}\right) \end{aligned}\right\} \quad f_{IB} > f_R$$

$$\left.\begin{aligned} \boldsymbol{u}_{IFA} &= \boldsymbol{u}_{ICLL} \cdot \frac{f_{IB} - f_R}{f_{IB} + d_{LL}} + \boldsymbol{u}_{IB} \cdot \left(1 - \frac{f_{IB} - f_R}{f_{IB} + d_{LL}}\right) \\ \boldsymbol{u}_{ICR} &= \boldsymbol{u}_{ICLL} \cdot \left(\frac{f_{IB} - 1}{f_{IB} + d_{LL}}\right) + \boldsymbol{u}_{IB} \cdot \left(1 - \frac{f_{IB} - 1}{f_{IB} + d_{LL}}\right) \end{aligned}\right\} \quad f_{IB} < f_R$$

The center of the inner cell is named ICL, the outer one ICR, the face neighboring these cells IFA and the position at the immersed surface IB. To avoid the occurrence

of high velocity values at the ghost-cells, which can cause numerical instability, the second interpolation is used if the IB is located very close to the inner cell ICL. The product of velocity and coefficient part of the ghost-cell is then added to the source term. Thereafter the velocity gradient is recalculated, fitting to the appropriate interpolation. The pressure is extrapolated linearly to the IB and the pressure gradient is also recalculated. Finally the mass flux through the involved faces is corrected to be parallel to the IB. The linear interpolation provides first order accuracy at the immersed boundary. To increase accuracy this approach can be replaced by a higher order interpolation.

2.4 Moving Wall Boundary Condition

In most instances a moving boundary condition makes sense only for a no-slip wall. To realize a moving surface, its spatial description has to be known in every time step. Fluid velocities at a moving wall are not zero, they comply with the velocity of the wall itself. This case is considered in the above interpolation: $\boldsymbol{u}_{IB}$ is the velocity vector at the IB, which vanishes in the non-moving case. The velocity of each triangle element of the surface description can be approximated with the displacement $\Delta \boldsymbol{x}_{IB}$ and the time step Δt:

$$\boldsymbol{u}_{IB} \approx \frac{\Delta \boldsymbol{x}_{IB}}{\Delta t}.$$

The identification and marking procedures have to be performed once again, after the IB has changed. Outer cells that become fluid cells during calculation gain their velocity and pressure values with ongoing iteration.

2.5 Solving the Equation System

Discretization and linearization of the governing equations in conjunction with conventionally and IB treated boundary conditions leads to a linear equation system:

$$\mathbf{A} \cdot \mathbf{\Phi} = \mathbf{S}_{\mathbf{\Phi}}$$

This system comprising of an arbitrary and sparse coefficient matrix $\mathbf{A}$, the variable vector $\mathbf{\Phi}$ and a source term $\mathbf{S}_{\mathbf{\Phi}}$ is solved using a BiCGSTAB solver. A range of preconditioners like incomplete *LU* or *Cholesky* decomposition are available to improve the convergence behavior. The equations for the velocities and the SIMPLE pressure correction are solved successively.

3 Results

After implementation of the main functions required for IB treatment, an in-depth validation phase follows. Due to the lack of a turbulence model these test cases have to be of incompressible and laminar nature. First simple test cases are used for validation. Afterwards harmonically oscillating IB's are implemented. Finally, the method is applied to cases of increasingly complex geometry, starting with simple 3d pipes and ending with the Weibel lung model.

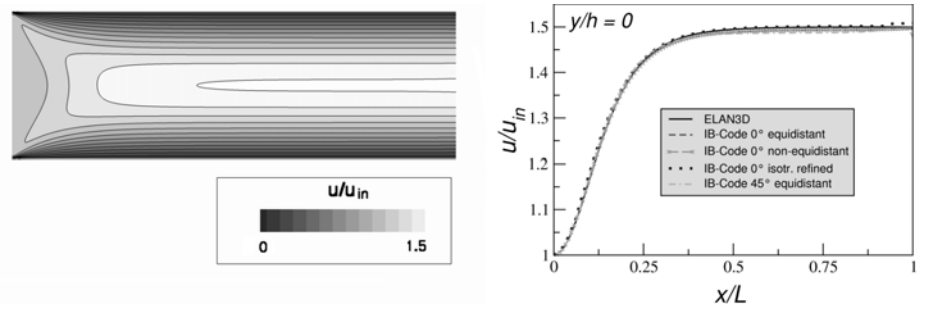

Fig. 3. Developing laminar channel flow at $\mathrm{Re} = 20$. Left: contour plot of u-velocity (only fluid region visible), right: comparison to results of original flow solver

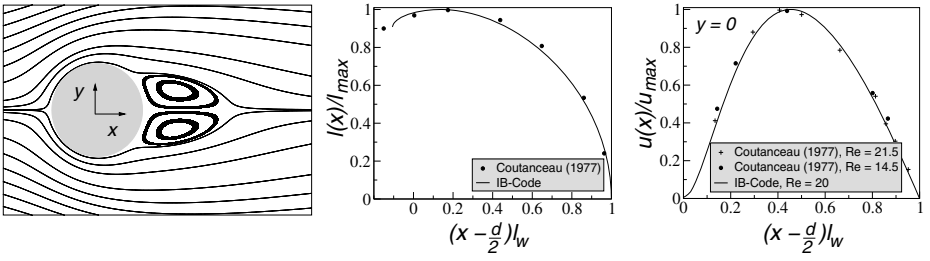

Fig. 4. Laminar cylinder flow at $\mathrm{Re} = 20$ in comparison with experimental data (d: cylinder diameter, l_w: length of the wake, l_{max}: maximum height of the wake)

3.1 Validation Test Cases

Developing flow in a 2d channel is a simple test case to validate IB inflow, outflow and wall boundary conditions. The solution can be directly compared to the results obtained by the original solver. Equidistant, non equidistant and isotropically refined Cartesian grids and the IB under rotation angles of $20\,^\circ$ and $45\,^\circ$ are tested. The velocity of the fully-developed channel flow near the outlet and the increasing u-velocity in the channel centerline can be compared to the analytical solution and the reference simulation (see fig. 3).

Laminar steady ($\mathrm{Re} = 20$) and unsteady ($\mathrm{Re} = 120$) flow past a 2d circular cylinder are test cases, that use the IB's only at the cylinder surface. The laminar wake regime behind the cylinder and the vortex shedding frequency in the unsteady case are adequate features to be compared with experimental results gathered in [8] (see fig. 4).

Both the channel and cylinder flow yield satisfactory results within the modeling scope.

3.2 Oscillating Rectangular Prism

Flow past an oscillating rectangular prism is an appropriate test case to demonstrate the advantage of the IBM. The new position of the body is calculated every time step,

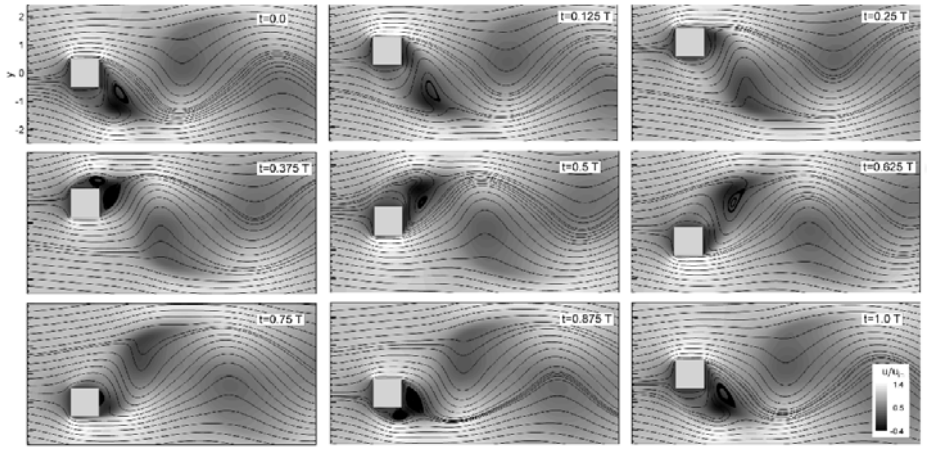

Fig. 5. Flow around an oscillating prism ($\mathrm{Re} = 200$, $A/a = 1$, $\omega/\omega_0 = 0.72$)

followed by the identification and marking procedure. As the surface of the prism coating (length a) is representable by only 8 triangles, identifying the domain is relatively time saving (see section 2.2). The deviation in y-direction of the prism is given by:

$$y(t) = A\sin(\omega t)$$

with the angular frequency $\omega = 2\pi f$ and the amplitude of deviation A . The fluid velocities at the IB can be determined analytically by the first derivative. The undisturbed flow past the non oscillating prism has been calculated for Reynolds numbers 170, 200 and 250. The predicted Strouhal number of $\mathrm{St}_0 = 0.138$ of a non oscillating case coincides with experiments from Okajima [9]. The set of pictures (fig. 5) shows the velocity field of one oscillation period ($\mathrm{Re} = 200$, amplitude: $A/a = 1.0$, frequency ratio: $\mathrm{St}/\mathrm{St}_0 = 0.72$). It can be seen, that the fluid adheres at the surface and follows the prisms movement. At the maximum deviation separation regions evolve behind the prism (fig. 5: $t = 0.375T$ and $t = 0.875T$), which detach downstream with increasing prism velocity. Spectral analysis of the fluctuating values (e.g. the vertical velocity in a fixed monitoring point) leads to the frequencies contained. The dominant part of the spectrum is the prescribed oscillating frequency whereas the natural frequency St_0 is vanishing.

3.3 More Complex Geometries

Turning from simple validation test cases to more complex geometries, 3d pipes and pipe junctions are investigated. One of the principal aims is the application of the IB-based solver to predict flow physics in the bronchial tree of the upper human lung [10]. The surface, which can be gained by medical CT scans is very complex and intensely curved. The lung model (A) of Weibel [11] is an appropriate first test case for such simulations. It is a 4th generation model consisting of joined pipes with different diameters, branching angles of $\Theta = 60°$ and plane angles of $\Phi = 90°$.

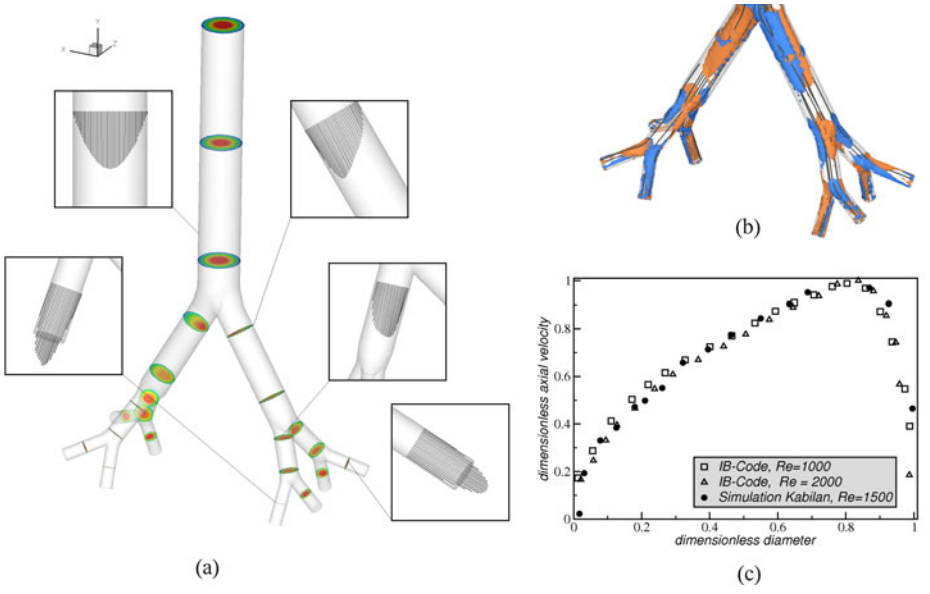

Fig. 6. 4 generation Weibel lung at $\mathrm{Re} = 2000$: Velocity profiles and absolute velocity contour levels (a), helicity isosurfaces explaining counterrotating regions (b), comparison of the velocity profile behind 1. bifurcation with results of Kabilan [13]

A laminar velocity profile, which can be obtained by the law of Hagen-Poiseuille, is set to the inlet. Mass flux at the outlets is constant, derived from the respective outlet area. The Reynolds number is determined by the diameter of the first generation tube, kinematic viscosity and the mean inflow velocity. Simulations with increasing Reynolds numbers $\mathrm{Re} = 10$, 100, 1000 and 2000 are carried out. Since the whole geometry has a symmetric shape, the flow is also symmetrical. The eight outlet branches can be divided into two groups: four branches that point to the outside and four that point to the inside. Velocity profiles between these two groups of outlet branches differ from each other, which was also found by [12]. Axial velocity profiles and contour plots of the absolute velocity in several stages and planes of interest are shown in figure 6 a. The inlet velocity profile is preserved up to the first bifurcation, where the mean flow is superposed by Dean flow effects, which can be illustrated by helicity contour plots (see fig. 6). The axial velocity profile behind the first bifurcation fits well with the CFD results from Kabilan et al. [13] at Reynolds number of 1500 (see fig. 6 b).

4 Conclusion and Outlook

The IBM was successfully implemented into the established in-house flow solver. All validation studies deliver satisfactory results and coincide well with experiments. The

flow prediction in the Weibel lung model yields promising results in terms of simulating the flow in a realistic CT based lung.

Further work will be carried out to make the solver more applicable to a wider range of problems: turbulence modeling, calculation of integral forces. To improve convergence or to refine the grid at a moving boundary, *h*-adaptive grid techniques are conceivable.

Acknowledgments

The authors would like to thank S. Kabilan, Department of Bioengineering, University of Washington providing the Weibel lung geometry.

References

[1] Peskin, C.S.: Flow patterns around heart valves: a numerical method. Journal of Computational Physics 10, 252–271 (1972)
[2] Peskin, C.S.: The immersed boundary method. Acta Numerica, pp. 479–512. Cambridge University Press, Cambridge (2002)
[3] Mittal, R., Iaccarino, G.: Immersed boundary methods. Annual Review of Fluid Mechanics 37, 239–261 (2005)
[4] Tseng, Y., Ferziger, J.H.: A ghost-cell immersed boundary method for flow in complex geometry. Journal of Computational Phyics 192(2), 593–623 (2003)
[5] Hylla, E.: Validierung und Erweiterung eines numerischen Verfahrens zur Simulation von inkompressiblen Strömungen mittels der Immersed Boundary Methode. Diploma Thesis, ISTA TU-Berlin (2008)
[6] Xue, L.: Entwicklung eines effizienten parallelen Lösungsalgorithmus zur dreidimensionalen Simulation komplexer turbulenter Strömungen. PhD Thesis, TU-Berlin (1998)
[7] Smailbegovic, F.S., Gaydadjiev, G.N., Vassiliadis, S.: Sparse Matrix Storage Format. In: Proceedings of the 16th Annual Workshop on Circuits, Systems and Signal Processing, pp. 445–448 (2005)
[8] Zdravkovich, M.M.: Flow Around Circular Cylinders. Fundamentals, vol. 1. Oxford Science Publications (1997)
[9] Okajima, A.: Strouhal numbers of rectangular cylinders. Journal of Fluid Mechanics 123, 379–389 (1982)
[10] Frederich, O., Amtsfeld, P., Hylla, E., Thiele, F., Puderbach, M., Kauczor, H.-U., Wegener, I., Meinzer, H.-P.: Numerical Simulation and Analysis of the Flow in Central Airways. In: Proceedings of STAB 2008. NNFM. Springer, Heidelberg (to be published, 2009)
[11] Weibel, E.R.: Morphometry of the Human Lung. Springer, Berlin (1963)
[12] Nowak, N., Kakade, P., Annapragada, A.: Computational Fluid Dynamics Simulation of Airflow and Aerosol Deposition in Human Lungs. Annals of Biomedical Engineering 31, 374–390 (2002)
[13] Kabilan, S., Lin, C., Hoffman, E.: Characteristics of airflow in a CT-based ovine lung: a numerical study. Journal of Applied Physiology 102, 1469–1482 (2007)