# FREE-SLIP APPROACH OF THE IMMERSED BOUNDARY METHOD

**José Laércio Doricio, josedoricio@yahoo.com.br**
**Paulo Celso Greco Júnior, pgreco@sc.usp.br**
Departamento de Engenharia de Materiais, Aeronáutica e Automobilística
Escola de Engenharia de São Carlos - Universidade de São Paulo
Av. Trabalhador São-Carlense, 400 - Centro - CEP: 13560-970 - São Carlos - São Paulo - Brasil.

***Abstract.*** *The Immersed Boundary Method was proposed by Charles Peskin to solve problems with no-slip boundaries for incompressible flows modelled by Navier-Stokes equations. However, for inviscid compressible flows, modelled by Euler equations, the no-slip condition usually is not employed in Computational Fluid Dynamics (CFD) applications. This work presents a free-slip approach of the Immersed Boundary Method to simulate inviscid compressible flows modelled by the Euler equations. The Finite Differences Method is used, in a structured mesh, to solve the governing equations. The fourth order Runge-Kutta method is employed for time integration, and the second order Steger-Warming method with Min-Mod flux limiter is employed for spatial discretization. The code is verified using the Method of Manufactured Solutions for the Eulerian Domain (briefly shown in this paper), and verified through the reflection problem of oblique shock waves (RPOSW) for the Lagrangian domain. Riemann, Dirichlet, and free-slip immersed boundary conditions were used to simulate the RPOSW at Mach number 2.953.*

***Keywords:*** *Immersed Boundary Method, Free-Slip Boundary Condition, Runge-Kutta, Steger-Warming.*

## 1. INTRODUCTION

The Immersed Boundary Method (IBM) was developed by Peskin (1972) to solve problems involving fluid-structure interaction. In this method, the domain is composed by an Eulerian mesh, used to represent the fluid domain, and a Lagrangian mesh, used to represent the elastic immersed boundary. The interaction between the elastic immersed boundary and the fluid is performed by a Dirac delta function, which is the kernel of the IBM. This approach has been applied in fluid dynamic studies for incompressible fluids by Dillon et al. (1995); Fauci and Peskin (1988); Fogelson and Peskin (1988); Lai and Peskin (2000); McQueen et al. (1982); Meisner et al. (1985); Peskin (1972).

The governing equations, in the IBM, are discretized in Cartesian computational meshes, and this is an advantage of the IBM because this simplifies mesh generation and reduces the complexity of the governing equations. Another advantage of this technique is that the Lagrangian mesh does not need to align with the Eulerian mesh, and this allows to simulate flows with moving immersed boundaries, complex geometries, or topological variations (Ye et al., 1999). A fixed mesh can be used even for complex moving geometries. Mesh refinement will be required only if improvements in a local flow resolution is desired (Linnick and Fasel, 2003).

Studies of aeroelastic instabilities, as flutter, usually give good results when the Euler equations are solved. To be consistent with the inviscid flow assumption, the immersed boundary must be free-slip type. Therefore, a numerical method to simulate compressible flows using the IBM with free-slip boundary is proposed.

## 2. GOVERNING EQUATIONS

Consider compressible homogeneous and inviscid flow in a two-dimensional rectangular domain $\Omega$ with an immersed boundary as a simple closed curve $\Gamma$, represented by $\mathbf{X}(s,t)$, with $0 \leq s \leq L_b$ and with $\mathbf{X}(0,t) = \mathbf{X}(L_b,t)$, where $L_b$ is the length of the $\Gamma$ boundary curve. Consider Lagrangian variables represented by capital letters. The governing equations can be given by:

$$\frac{\partial \mathbf{V}}{\partial t} + \frac{\partial \mathbf{E}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = \mathbf{H}\,, \tag{1}$$

where:

$$\mathbf{V} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho e \end{bmatrix} \quad \mathbf{E} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (\rho e + p)\,u \end{bmatrix} \quad \mathbf{G} = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (\rho e + p)\,v \end{bmatrix} \quad \mathbf{H} = \begin{bmatrix} 0 \\ f_1 \\ f_2 \\ u f_1 + v f_2 \end{bmatrix} \tag{2}$$

$$p = (\gamma - 1)\left(\rho e - \frac{1}{2}\rho\left(u^2 + v^2\right)\right)\,, \tag{3}$$

$$\mathbf{f}(\mathbf{x},t) = \int_0^{L_b} \mathbf{F}(s,t)\delta^2(\mathbf{x} - \mathbf{X}(s,t))ds\,, \tag{4}$$

$$\frac{\partial \mathbf{X}(s,t)}{\partial t} = \mathbf{U}(\mathbf{X}(s,t),t) = \int_{\Omega} \mathbf{u}(\mathbf{x},t)\delta^2(\mathbf{x} - \mathbf{X}(s,t))d\mathbf{x} \,, \tag{5}$$

$$\mathbf{F}(s,t) = \mathbf{S}(\mathbf{X}(s,t),t) \,. \tag{6}$$

In Eq. (1)-(6), $\mathbf{x} = (x, y)$ is the location vector, $\mathbf{u}(\mathbf{x}, t) = (u(\mathbf{x}, t), v(\mathbf{x}, t))$ is the fluid velocity field, $p(\mathbf{x}, t)$ is the pressure field, $\rho(\mathbf{x}, t)$ is the density field and $e(\mathbf{x}, t)$ is the total energy, given by:

$$e = e_i + \frac{1}{2}\left(u^2 + v^2\right) \,, \tag{7}$$

where $e_i$ is the specific internal energy. The force acting on the fluid is given by $\mathbf{f}(\mathbf{x}, t) = (f_1(\mathbf{x}, t), f_2(\mathbf{x}, t))$, while the force acting on the immersed boundary is given by $\mathbf{F}(s, t) = (F_1(s, t), F_2(s, t))$. Equation (3) represents the state equation for pressure considering ideal gas with $\gamma = 1.4$. In Equation (6), $\mathbf{S}(\mathbf{X}(s, t), t)$ expresses the material elasticity, and represents free-slip boundary, differently from the no-slip representation adopted by Griffith and Peskin (2005).

## 3. NUMERICAL METHOD

The IBM is implemented using the finite differences method for Eulerian and Lagrangian meshes. Consider $\Omega = [0, L] \times [0, L]$ as the flow domain, where $L$ is the domain length. The fluid variables are defined over a $N \times N$ Eulerian mesh with $\mathbf{x} = (x_i, y_j) = (ih, jh)$ for $i, j = 0, 1, ..., N - 1$, where $h = \Delta x = \Delta y = \frac{L}{N}$ is the length of each mesh division. A set of $M$ Lagrangian points defined by $\mathbf{X} = (X_k, Y_k)$ with $k = 0, 1, ..., M - 1$ is used to discretize the immersed boundary, with interval $\Delta s = \frac{L_b}{M}$. The fourth order Runge-Kutta method (Schreier, 1982) is employed for time integration, and the second order Steger-Warming (Steger and Warming, 1981) method with Min-Mod flux limiter is employed for spatial discretization. In the algorithm, each stage of the Runge-Kutta method is represented by $\wp$, and $n + 1 = t_n + \Delta t$ represents the instant of time. Lai and Peskin (2000) describe methods of order 1 and 2. Consider the governing equation (1), that can be written as:

$$\frac{\partial \mathbf{V}}{\partial t} + Pv = 0 \,, \tag{8}$$

where

$$Pv \equiv \frac{\partial \mathbf{E}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} - \mathbf{H} \,. \tag{9}$$

For each time-step $n$, the fourth order Runge-Kutta method is given by:

$$\begin{aligned}
\mathbf{V}^{(0)} &= \mathbf{V}^{(n)} \\
\mathbf{V}^{(1)} &= \mathbf{V}^{(0)} - \frac{\Delta t}{2} Pv^{(0)} \\
\mathbf{V}^{(2)} &= \mathbf{V}^{(0)} - \frac{\Delta t}{2} Pv^{(1)} \\
\mathbf{V}^{(3)} &= \mathbf{V}^{(0)} - \Delta t Pv^{(2)} \\
\mathbf{V}^{(4)} &= \mathbf{V}^{(0)} - \frac{\Delta t}{6}\left(Pv^{(0)} + 2Pv^{(1)} + 2Pv^{(2)} + Pv^{(3)}\right) \\
\mathbf{V}^{(n+1)} &= \mathbf{V}^{(4)}
\end{aligned} \tag{10}$$

where $\mathbf{V}^{(0)}$, $\mathbf{V}^{(1)}$, $\mathbf{V}^{(2)}$, $\mathbf{V}^{(3)}$ and $\mathbf{V}^{(4)}$ are the fluid variables, defined by Eq. (2), in the intermediate stage of the Runge-Kutta method, and $\mathbf{V}^{(n+1)}$ is the fluid variable value in time $t^{(n+1)} = t + \Delta t$. This numerical scheme is stable for Courant-Friedrichs-Lévi (CFL) number of $2\sqrt{2}$. More information about this method can be found in Schreier (1982).

✣ **Preliminary stage of the Runge-Kutta method:**

1. The Lagrangian variables of the immersed boundary are set in $\wp = 0$ with the value in time $t = t_n$:

$$\mathbf{F}^{(0)}(s) = \mathbf{F}^{n}(s) \,,$$

$$\mathbf{U}^{(0)}(s) = \mathbf{U}^{n}(s) \,,$$

$$\mathbf{X}^{(0)}(s) = \mathbf{X}^{n}(s) \,.$$

2. The Eulerian variables of the fluid field are set in $\wp = 0$:

$$\mathbf{f}^{(0)}(\mathbf{x}) = \mathbf{f}^{n}(\mathbf{x}) ,$$

$$\mathbf{V}^{(0)}(\mathbf{x}) = \mathbf{V}^{n}(\mathbf{x}) .$$

After the preliminary stage of the Runge-Kutta method, the intermediate stages are performed for $\wp = 1, 2, 3, 4$.

✣ **Intermediate stage of the Runge-Kutta method** ($\wp = 1, 2, 3, 4$)**:**

1. The Lagrangian force $\mathbf{F}^{(\wp+1)}(\ s\ )$ is calculated in the immersed boundary with the configuration given by $\mathbf{X}^{(\wp)}(\ s\ )$ as follows:

$$\mathbf{F}^{(\wp+1)}(\ s\ ) = \mathbf{S}^{(\wp)}(\mathbf{X}^{(\wp)}) , \tag{11}$$

2. The Lagrangian force $\mathbf{F}^{(\wp+1)}(\ s\ )$ is interpolated in the Eulerian field to determine $\mathbf{f}^{(\wp+1)}(\mathbf{x})$:

$$\mathbf{f}^{(\wp+1)}(\mathbf{x}) = \sum_{s} \mathbf{F}^{(\wp+1)}(\ s\ )\delta_h^2(\mathbf{x} - \mathbf{X}^{(\wp)}(\ s\ ))\Delta s , \tag{12}$$

where the delta function is given by:

$$\delta_h^2(\mathbf{x}) = \delta_h(\ x\ )\delta_h(\ y\ ) , \tag{13}$$

with

$$\delta_h(\ r\ ) = \begin{cases} \frac{1}{8h}\left(3 - \frac{2|r|}{h} + \sqrt{1 + \frac{4|r|}{h} - \frac{4r^2}{h^2}}\right) , & |r| \le h , \\ \frac{1}{8h}\left(5 - \frac{2|r|}{h} - \sqrt{-7 + \frac{12|r|}{h} - \frac{4r^2}{h^2}}\right) , & h \le |r| \le 2h , \\ 0 , & 2h \le |r| . \end{cases} \tag{14}$$

3. The Euler equations given by Eqs. (1)-(2) are solved using the force term $\mathbf{f}^{(\wp+1)}(\mathbf{x})$ in the stage $(\wp + 1)$ of the Runge-Kutta method.
4. The Eulerian velocity $\mathbf{u}^{(\wp+1)}(\mathbf{x})$ is interpolated to the Lagrangian points of the immersed boundary:

$$\mathbf{U}^{(\wp+1)}(\ s\ ) = \sum_{\mathbf{x}} \mathbf{u}^{(\wp+1)}(\mathbf{x})\delta_h^2(\mathbf{x} - \mathbf{X}^{(\wp)}(\ s\ ))h^2 . \tag{15}$$

where $\delta_h^2$ is the delta function defined by Eqs. (13)-(14).

5. The Lagrangian points $\mathbf{X}^{(\wp+1)}(\ s\ )$ are updated using the Runge-Kutta method described by Eq. (10):

$$\mathbf{X}^{(\wp+1)}(\ s\ ) = \phi(\mathbf{U}^{(\wp+1)}(\ s\ )) . \tag{16}$$

where $\phi$ represents the Runge-Kutta step of Eq. (10) in the stage $\wp$.

✣ **The fluid variables are updated from time** $t = t_n$ **to** $t = t_n + \Delta t$**:**

1. The Lagrangian variables are updated to time $t = t_n + \Delta t$:

$$\mathbf{F}^{n+1}(\ s\ ) = \mathbf{F}^{(4)}(\ s\ ) ,$$

$$\mathbf{U}^{n+1}(\ s\ ) = \mathbf{U}^{(4)}(\ s\ ) ,$$

$$\mathbf{X}^{n+1}(\ s\ ) = \mathbf{X}^{(4)}(\ s\ ) .$$

2. The Eulerian variables are updated to time $t = t_n + \Delta t$:

$$\mathbf{f}^{n+1}(\mathbf{x}) = \mathbf{f}^{(4)}(\mathbf{x}) ,$$

$$\mathbf{V}^{n+1}(\mathbf{x}) = \mathbf{V}^{(4)}(\mathbf{x}) .$$

In the numerical scheme described above, the points of the elastic boundary must stay close to the original configuration. This can be performed by adequately choosing $\mathbf{S}(\mathbf{X}(s,t),t)$. For example,

$$\mathbf{F}(s,t) = \mathbf{S}(\mathbf{X}(s,t),t) = \kappa\,(\mathbf{X}^e(\ s\ ) - \mathbf{X}(s,t))\ , \tag{17}$$

where $\kappa \gg 1$ is a positive constant. Equation (17) links the immersed boundary points $\mathbf{X}$ to the equilibrium points $\mathbf{X}^e$ by stiff springs. Because of this the elasticity of the boundary depends on the $\kappa$ constant.

Equations (16) and (17) describe the no-slip boundary type because the velocity in $\mathbf{X}(s,t)$ is forced to be close to the velocity of the structure. The free-slip boundary type can be imposed if Eqs. (16) and (17) are modified to:

$$\mathbf{X}^{(\wp+1)}(\ s\ ) = \phi(\mathrm{proj}_{\mathbf{n}(s)}\mathbf{U}^{(\wp+1)}(\ s\ ))\ . \tag{18}$$

$$\mathbf{F}(s,t) = \mathbf{S}(\mathbf{X}(s,t),t) = \kappa\ \mathrm{proj}_{\mathbf{n}(s)}\,(\mathbf{X}^e(\ s\ ) - \mathbf{X}(s,t))\ , \tag{19}$$

where $\mathbf{n}(s)$ is the normal vector of the structure in point $\mathbf{X}(s,t)$.

## 4. CODE VERIFICATION

The numerical implementation of the Immersed Boundary Method was verified using two strategies:

- by the Method of Manufactured Solutions (MMS), following da Silva et al. (2005) and Burg and Murali (2004) for the Eulerian domain, briefly shown in this paper;
- and by the converged solution of the oblique shock-wave reflection problem (RPOSW) for the Lagrangian domain to verify if the free-slip boundary condition is correctly imposed.

The Eulerian domain was verified by the MMS using the following manufactured solutions:

$$\rho(\mathbf{x},0) = \frac{1}{800}\,x^3 + \frac{1}{800}\,y^3 + \frac{3}{4}\ , \tag{20}$$

$$u(\mathbf{x},0) = \frac{1}{3}\sin(\frac{1}{5}\,y) + \frac{1}{3}\sin(\frac{1}{5}\,x) + \frac{1}{2}\ , \tag{21}$$

$$v(\mathbf{x},0) = \frac{1}{4}\cos(\frac{1}{5}\,y) + \frac{1}{4}\cos(\frac{1}{5}\,x) + \frac{4}{7}\ , \tag{22}$$

$$p(\mathbf{x},0) = \frac{1}{7}\sin(\frac{1}{5}\,x) + \frac{1}{7}\,e^{(1/4\,y)}\ . \tag{23}$$

For the RPOSW an oblique shock-wave passing through a stream line (Shapiro, 1953) was considered, as shown in Fig. 1.

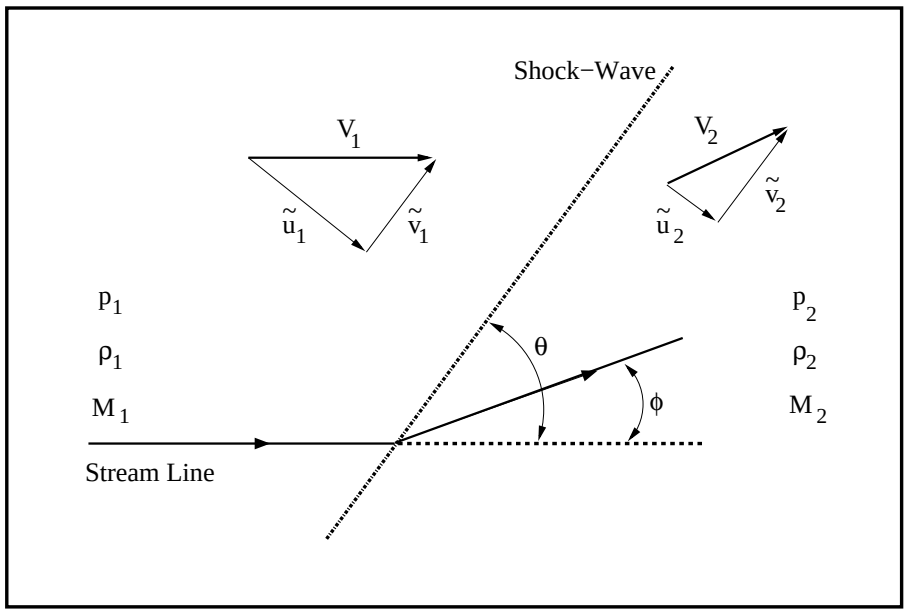


Figure 1. Notation for oblique shock wave problems.

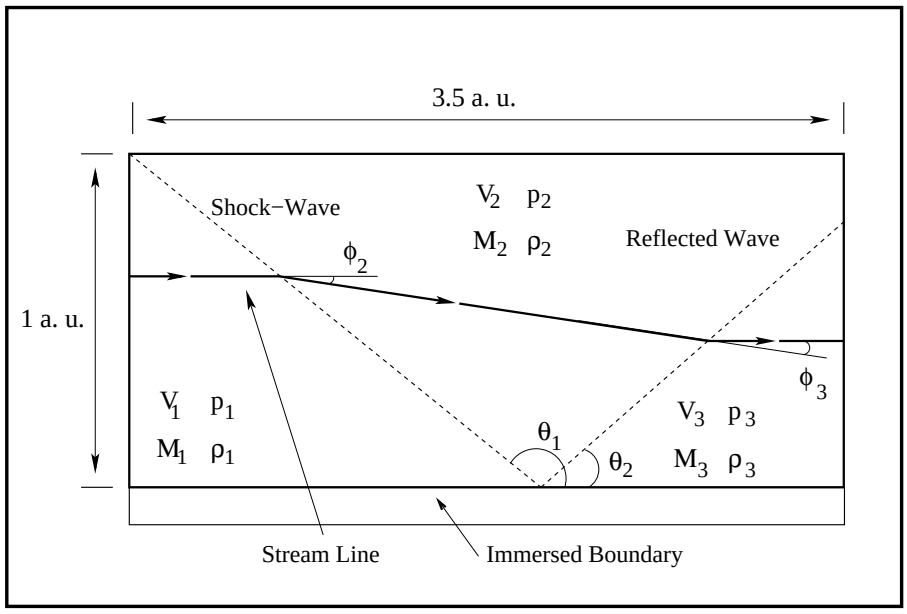


Figure 2. Model for the oblique shock-wave reflection problem.

The following set of equations describe the analytical solution of the oblique shock-waves:

$$\begin{cases} \dfrac{p_2}{p_1} = \dfrac{2\gamma\, M_1^{\,2}\sin^2(\theta) - (\gamma-1)}{\gamma+1}\,, \\ \dfrac{\rho_2}{\rho_1} = \dfrac{(\gamma+1)\, M_1^{\,2}\sin^2(\theta)}{(\gamma-1)M_1^{\,2}\sin^2(\theta)+2}\,, \\ \dfrac{V_2^{\,2}}{V_1^{\,2}} = 1 - 4\dfrac{(M_1^{\,2}\sin^2(\theta)-1)(\gamma\, M_1^{\,2}\sin^2(\theta)+1)}{(\gamma+1)^2 M_1^{\,4}\sin^2(\theta)}\,, \\ \tan\phi = \dfrac{M_1^{\,2}\sin^2(2\theta) - 2\cot(\theta)}{2 + M_1^{\,2}(\gamma+\cos(2\theta))}\,. \end{cases} \tag{24}$$

Figure 2 shows the model for the reflection shock-wave problem. The stream line, in this model, is deflected by $\phi_2$ and $\phi_3$ angles because of the shock-wave presence. The value of $p_2$, $\rho_2$, $V_2$ and $\phi_2$ are calculated using the values of $p_1$, $\rho_1$, $V_1$, $\theta_1$ and Eq. (24). Similarly, $p_3$, $\rho_3$, $V_3$ and $\phi_3$ are calculated using the values of $p_2$, $\rho_2$, $V_2$ and $\theta_2$. The shock wave separates the domain in three regions with fixed properties. In region 1, $p$, $\rho$, $V$ and $M$ are defined by $p_1$, $\rho_1$, $V_1$ and $M_1$. Similarly for region 2 ($p_2$, $\rho_2$, $V_2$, $M_2$ and $\phi_2$) and for region 3 ($p_3$, $\rho_3$, $V_3$, $M_3$ and $\phi_3$) the same definition is applied. Table 1 shows the analytical solution for the fluid variables for each region with $\theta_1 = 151^o$ and $\theta_2 = 23^o$.

Table 1. Analytical solution of the shock-wave reflection problem.

| Property | $V$ | $M$ | $p$ | $\rho$ | $\phi$ |
|---|---|---|---|---|---|
| Region 1 | 2.95342 | 2.95342 | 0.71428 | 1.0 | $0.0^o$ |
| Region 2 | 2.71034 | 2.39999 | 1.58943 | 1.74479 | $-11.37528^o$ |
| Region 3 | 2.43018 | 1.94360 | 3.14003 | 2.81189 | $11.37528^o$ |

## 5. NUMERICAL RESULTS

For the Eulerian domain with MMS, the convergence order was calculated in a square domain of $3\times3$ non-dimensional units using five meshes: $\Delta x = \Delta y = 0.1$, $\Delta x = \Delta y = 0.05$ to $\Delta x = \Delta y = 0.00625$. The result is shown in Fig. 3. The result shown by Fig. 3 demonstrates the second order mesh convergence, accordingly to the theoretical order of the numerical method.

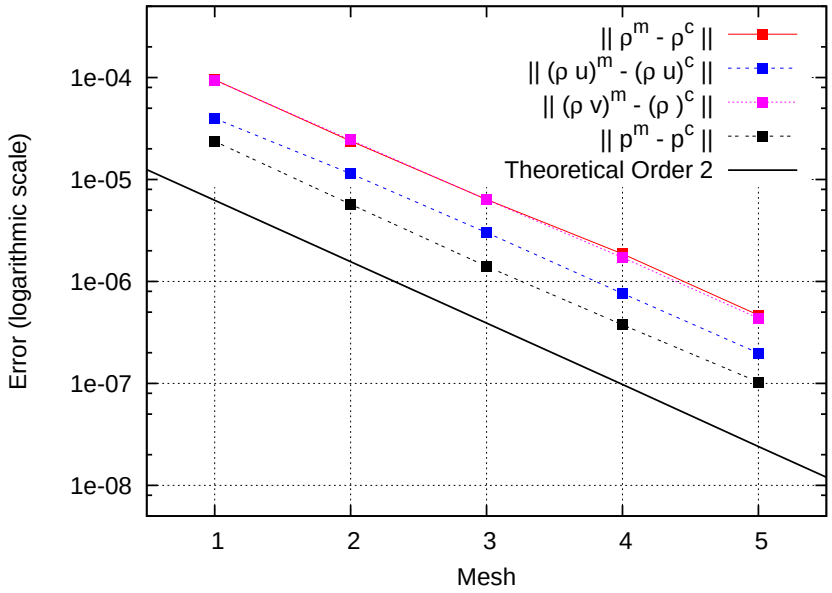


Figure 3. Convergence order compared with the theoretical second order.

The reflection problem was solved using a mesh defined in a computational domain of $3.5 \times 1.0$ non-dimensional units (a. u.), with $\Delta x = \Delta y = 0.0125 a.u.$. In the left and top boundaries, the Dirichlet boundary condition was used according to Tab.1. In the right and bottom boundaries, a non-reflexive boundary condition based on Riemann invariants (Buonomo, 2004) was used. The free-slip immersed boundary was placed above the bottom of domain according to Fig. 2. Figures 4 to 6 show the results for pressure, density and Mach number. The angle formed by the stream line in the region 2 with the direction $x$ was calculated using the local velocity given by $\mathbf{v} = (2.6571, -0.534571)$ at position $\mathbf{x} = (1.30089, 0.750575)$, where the stream line passes. The result is $\phi_2^{conv} = -11.375250128^o$. Comparing the value of this angle with the theoretical value, it follows:

$$\frac{|\,\phi_2^{conv} - \phi_2^{theoretic}\,|}{|\,\phi_2^{theoretic}\,|} = 2.626 \times 10^{-6}\,. \tag{25}$$

The result of Eq. (25) shows that the deflection angle of the stream line is very close to the theoretical value. This shows that the numerical method represents well the theoretical result.

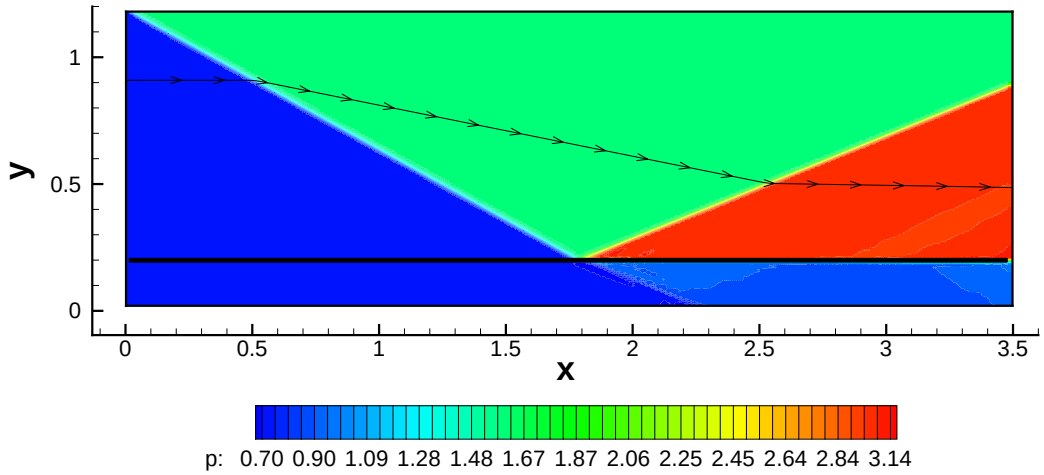


Figure 4. Non-dimensional pressure $p$ and stream line.

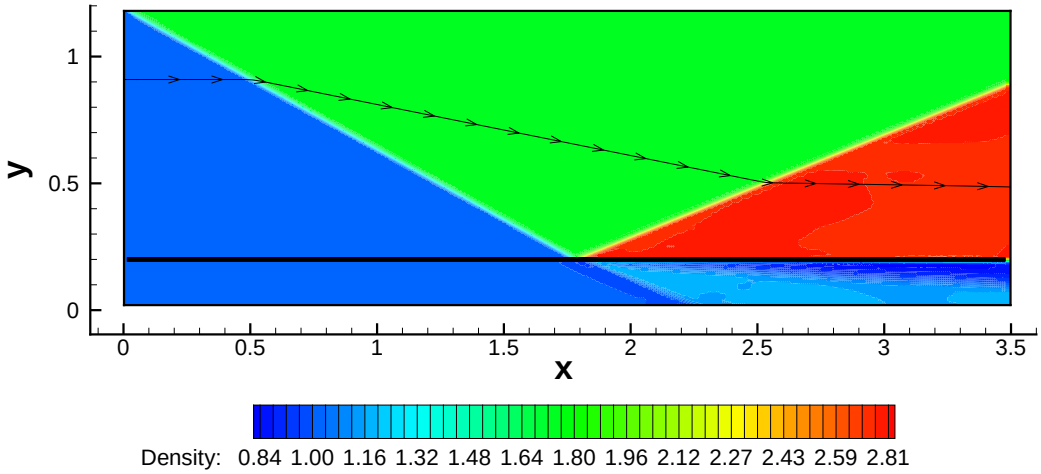


Figure 5. Non-dimensional density $\rho$ and stream line.

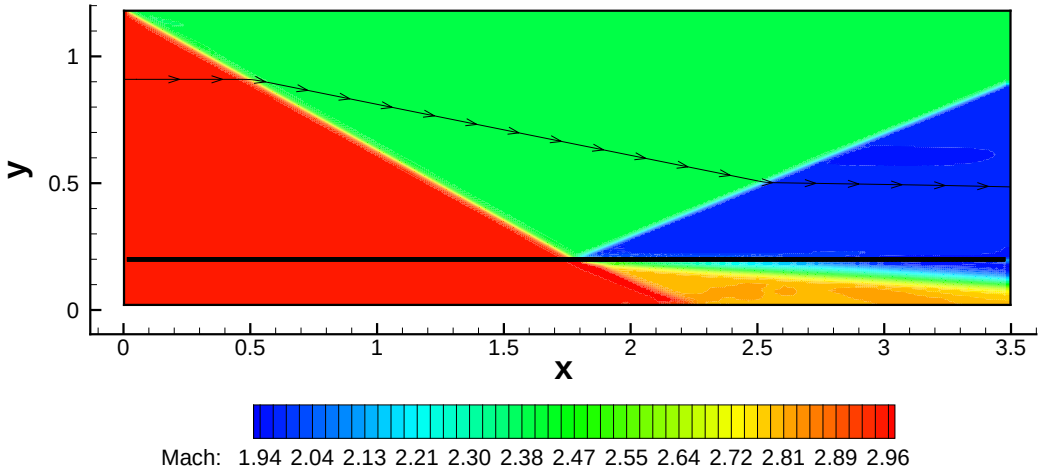


Figure 6. Mach number and stream line.

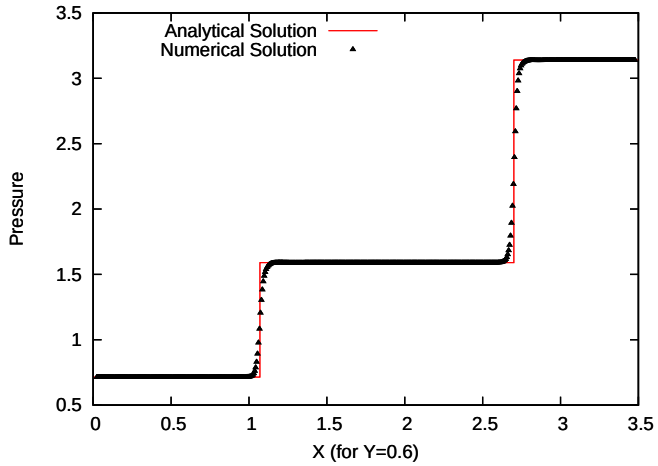


Figure 7. Comparison of the analytical with the numerical pressure at Y=0.6

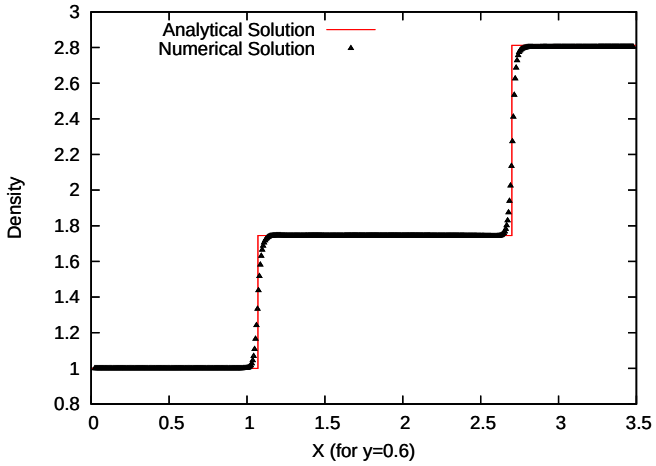


Figure 8. Comparison of the analytical with the numerical density at Y=0.6

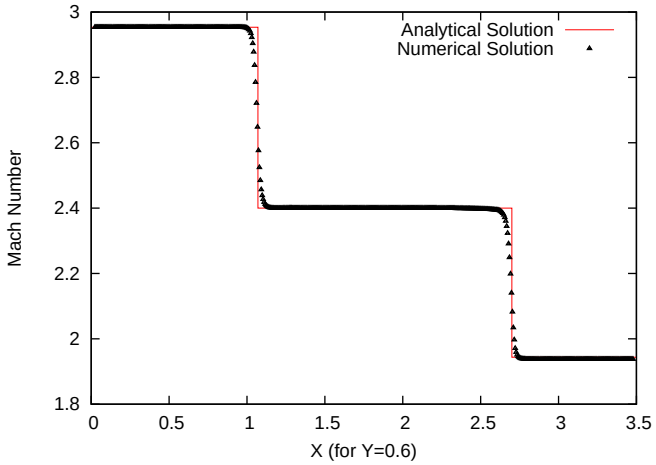


Figure 9. Comparison of the analytical with the numerical Mach number at Y=0.6.

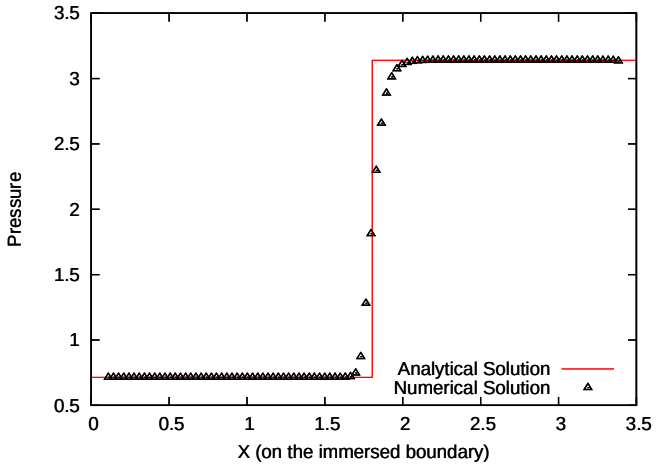


Figure 10. Comparison of the analytical with the numerical density on the immersed boundary.

Figures 7 to 9 show the comparison between the analytical solution with the converged numerical solution for $p$, $\rho$ and $M$ using a horizontal line along the domain at $y = 0.6$, and Fig. 10 to 12 on the immersed boundary. This results show that the numerical method was capable of reproducing the discontinuity generated by the shock waves. It is important to note in Fig. 10 to 12 that the free-slip condition is satisfied. However, there is more dissipation of the shock wave on the immersed boundary, this occur because the delta function interpolator is first order in space. The IBM was implemented using $C^{++}$ programming language for Ubuntu linux system, and the code was executed in a Pentium 4 2.40GHz computer

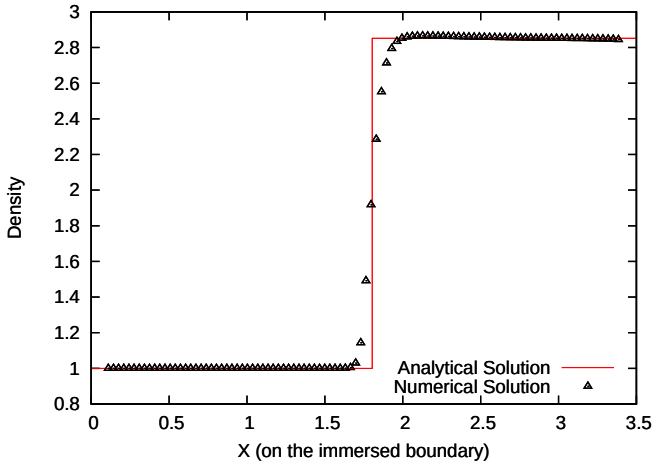


Figure 11. Comparison of the analytical with the numerical density on the immersed boundary.

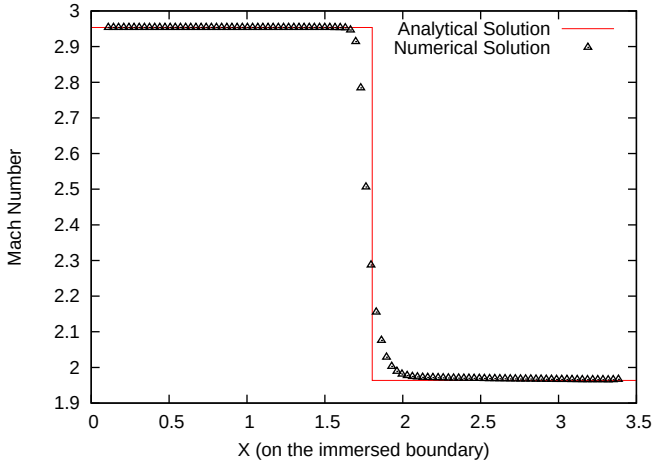


Figure 12. Comparison of the analytical with the numerical Mach number on the immersed boundary.

with 256Mb of RAM. The computational time necessary to achieve convergence for the shock wave problem was 0.72 hour.

## 6. CONCLUDING REMARKS

A numerical method for free-slip Immersed Boundary Method is proposed in this work. The numerical method was verified by the Method of Manufactured Solutions for the Eulerian domain (da Silva et al., 2005), and verified by the reflection problem of oblique shock waves for the Lagrangian domain, and the results showed to be in close agreement with the analytical results. However, extension and verification of the method for complex geometries as circular cylinder or airfoil profiles must be performed. That is a topic for future work.

## 7. ACKNOWLEDGEMENTS

The authors are thankful for the financial support of CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) 141051/2006-0.

## 8. REFERENCES

Buonomo, C. A. 2004. Técnica de Fronteiras Imersas com Formulação Viscosa e Compressível. Master's thesis, Instituto Tecnológico de Aeronáutica.

Burg, C. O. E. and Murali, V. K. 2004. Efficient Code Verification using the Residual Formulation of the Method of Manufactured Solutions. pages 1–13. 34th AIAA Fluid Dynamics Conference.

da Silva, H. G., de Medeiros, M. A. F., and de Souza, L. F. 2005. A Verification Test for a Direct Numerical Simulation Code that uses a High Order Discretization Scheme. 18th International Congress of Mechanical Engineering.

Dillon, R., Fauci, L. J., and Gaver, D. 1995. A Microscale Model of Bacterial Swimming, Chemotaxis and Substrate

Transport. J. Theor. Biol., 177:325–340.
Fauci, L. and Peskin, C. S. 1988. Computational Model of Aquatic Animal Locomotion. J. Comput. Phys., 77:85–108.
Fogelson, A. L. and Peskin, C. S. 1988. A Fast Numerical Method for Solving the Three-Dimensional Stokes' Equations in the Presence of Suspended Particles. J. Comput. Phys., 79:50–69.
Griffith, B. E. and Peskin, C. S. 2005. On the Order of Accuracy of the Immersed Boundary Method: High Order Convergence Rates for Sufficiently Smooth Problems. Journal of Computational Physics, 208:75–105.
Lai, M.-C. and Peskin, C. S. 2000. An Immersed Boundary Method with Formal Second-Order Accuracy and Reduced Numerical Viscosity. Journal of Computational Physics, 160:705–719.
Linnick, M. N. and Fasel, H. F. 2003. A High-Order Immersed Boundary Method for Unsteady Incompressible Flow Calculations. AIAA 2003-1124.
McQueen, D. M., Peskin, C. S., and Yellin, E. L. 1982. Fluid Dynamics of the Mitral Valve: Physiological Aspects of a Mathematical Model. Amer. J. Physiol., 242:H1095–H1110.
Meisner, J. S., McQueen, D. M., Ishida, Y., Vetter, H. O., Bortolotti, U., Strom, J. A., Frater, R. W. M., Peskin, C. S., and Yellin, E. L. 1985. Effects of Timing of Atrial Systole on LV Filling and Mitral Valve Closure: Computer and Dog Studies. Amer. J. Physiol., 249:H604–H619.
Peskin, C. S. 1972. Flow Patterns around Heart Valves: A digital computer method to solve the equations of motion. PhD thesis, Albert Eistein College of Medicine.
Schreier, S. 1982. Compressible Flow. John Willey & Sons.
Shapiro, A. H. 1953. The Dynamics and Thermodynamics of Compressible Fluid Flow. Wiley.
Steger, J. L. and Warming, R. F. 1981. Flux Vector Splitting of the Inviscid Gasdynamic Equations with Application to Finite-Difference Methods. Journal of Computational Physics, 40(2):263–293.
Ye, T., Mittal, R., Udaykumar, H. S., and Shyy, W. 1999. Incompressible Flows with Complex Immersed Boundaries. Journal of Computational Physics, 156:209–240.

## 9. Responsibility notice

The authors are the only responsible for the printed material included in this paper.