HIGH-ORDER IMMERSED INTERFACE METHOD USING COMPACT FINITE-DIFFERENCE SCHEMES

Larissa Alves Petri, lariss@gmail.com
Leandro Franco de Souza, lefraso@gmail.com
Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo. São Carlos-SP, Brasil.

Abstract. *The simulation of flow over bodies with complex geometries using high-order approximations is a challenge in computational fluid dynamics. The immersed interface methods are becoming increasingly popular since Cartesian grids can be adopted. Despite obtaining good results, both qualitative and quantitative, most methods rely on low-order, normally first-order. The aim of the present work is to present a high-order immersed interface method for the 2-D incompressible Navier-Stokes equations in stream function-vorticity formulation. The time integration is carried out by a fourth-order explicit Runge-Kutta scheme. Fourth-order compact finite-differences are used for discretize the spatial derivatives. The high formal accuracy at the immersed interface is obtained by corrections to the finite-difference stencils. To validate the method in its application to incompressible flows, a uniform flow past a circular cylinder was used as test case. The results show that higher order can be achieved adopting the correction in the finite-difference stencils.*

Keywords: *high-order immersed interface method, compact finite-differences, incompressible flow.*

1. INTRODUCTION

The Immersed Interface Method (IIM) was first proposed by LeVeque and Li (1994) and it differs from the original Immersed Boundary Method (Peskin, 1972, 1977, 2002; Roma *et al.*, 1999) and from other alternative forms (Goldstein *et al.*, 1993; Lima E Silva *et al.*, 2003) by the use of a discrete delta function. LeVeque and Li (1994) developed a second-order method that allows more general interface conditions.

Considering that a particular set of governing partial differential equations apply throughout the entire domain, the main idea is that the immersed interface represents a singularity and field variables and/or their derivatives will be discontinuous across the immersed interface. To deal with this discontinuity at the interface, the coefficients for the finite differences are made different and the stencil needs one more point to obtain the solution. As well, corrections terms, based on jump conditions, are added on the right-hand side.

A variation of this class of method is the explicit-jump immersed interface method proposed by Wiegmann and Bube (2000). They made the observation that standard finite differences techniques fail when applied to non-smooth functions because underlying Taylor expansions upon which they are based are invalid. In this context, a Taylor expansion including jumps is used to provide second-order approximations. In their method, the formulas for the discontinuities across the interface can be written depending only on solutions values on one side of the interface.

According to Li and Ito (2006) IIM is a sharp interface method in which the discontinuities or the jump conditions are enforced either exactly or approximately. It is crucial for the IIM to have a prior knowledge of the jump conditions either from physical reasoning or from the governing differential equations. Only those grid points near or on the interfaces, which are usually fewer than those regular grid points, need special attention. Away from the interface, standard finite difference method is used in the discretization. The simple data structure of a fixed and uniform grid makes it easy to use the method to solve complicated interface problems with reasonable cost and given accuracy. In their work, while the global errors have second-order of accuracy at all grid points, the local truncation errors may be one order lower at grid points that are near the interface than at regular grid points that are away from the interface.

The key idea of the IIM is that the finite differences schemes at the immersed interface must be corrected in order to maintain the formal accuracy of the underlying numerical scheme (Linnick and Fasel, 2005). In this paper a fourth-order IIM is used to solve incompressible flow. The derivative calculations and the Poisson equation solver are verified to check the accuracy of the IIM. The paper is divided as follows: next section shows the formulation adopted; in Section 3 the numerical method is described; numerical tests are presented in Section 4; in Section 5 the results for 2D simulations are shown; finally conclusions are made in Section 6.

2. FORMULATION

For the numerical solution, the Navier-Stokes equations were written in vorticity-stream function formulation. The vorticity in the spanwise direction, denoted by ω, is:

$$\omega = \frac{\partial u}{\partial y} - \frac{\partial v}{\partial x}, \tag{1}$$

where u and v are the velocities components in the streamwise (x) and wall normal (y) directions, respectively. The stream function ψ is defined such that:

$$u = \frac{\partial \psi}{\partial y}, \tag{2}$$

and

$$v = -\frac{\partial \psi}{\partial x}. \tag{3}$$

The vorticity transport equation can be obtained by applying a rotational of the momentum equation is:

$$\frac{\partial \omega}{\partial t} = -u\frac{\partial \omega}{\partial x} - v\frac{\partial \omega}{\partial y} + \frac{1}{Re}\left(\frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2}\right), \tag{4}$$

where t is the time and the Reynolds number $Re = \dfrac{LU}{\nu}$, with L the channel height, U the max velocity and ν the dynamic viscosity.

The continuity equation is:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0. \tag{5}$$

Using the definitions of stream function Eqs. (2) and (3) and applying them in the vorticity definition Eq. (1) a Poisson-type equation for the ψ can be derived:

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = \omega. \tag{6}$$

Equations (2), (3), (4) and (6) were solved numerically by the schemes described below.

The solution was marched in time according to the following steps:

1. Impose initial conditions for u, v, ω and ψ compatible with each other;
2. Calculate the vorticity from the vorticity transport equation (4), for time $t + dt$;
3. Calculate ψ from the Poisson equation (6);
4. Calculate v velocity from Eq. (3);
5. Calculate u velocity from Eq. (2);
6. Calculate the vorticity generation at the wall for the new velocity distribution;
7. Return to the second step until the desired integration time is reached.

3. NUMERICAL METHOD

The numerical method described here is focused in the calculation of the derivatives in the presence of discontinuities. The IIM consists in the correction of the derivatives calculation by adding jump corrections to the formula. This correction is mandatory at the points were the interface crosses the stencil adopted. The following text explains how to calculate these corrections that are the soul of this method.

Considering f a smooth function in the entire domain, except at point x_α, the Taylor expansion of f at x_{i+1} can be written as

$$f(x_{i+1}) = f(x_i) + hf'(x_i) + \frac{h^2}{2!}f''(x_i) + \ldots + J_\alpha, \tag{7}$$

where

$$J_\alpha = [\![f]\!]_\alpha + h^+[\![f']\!]_\alpha + \frac{(h^+)^2}{2!}[\![f'']\!]_\alpha + \ldots, \tag{8}$$

$h = x_{i+1} - x_i$, $h^+ = x_{i+1} - x_\alpha$, $h^- = x_\alpha - x_i$ and $[\![f]\!]_\alpha$ is the jump of f at x_α, obtained by

$$[\![f]\!]_\alpha = \lim_{x \to x_\alpha^+} f(x) - \lim_{x \to x_\alpha^-} f(x). \tag{9}$$

With these definitions, a compact finite difference approximation for the first derivative of f can be written as

$$L^1_{i-1}f^{(1)}_{i-1} + L^1_i f^{(1)}_i + L^1_{i+1}f^{(1)}_{i+1} = R^1_{i-1}f_{i-1} + R^1_i f_i + R^1_{i+1}f_{i+1} + (L^1_I J_{\alpha 1} - R^1_I J_{\alpha 0}), \tag{10}$$

and so, the second derivative of f can similarly be written as

$$L^2_{i-1}f^{(2)}_{i-1} + L^2_i f^{(2)}_i + L^2_{i+1}f^{(2)}_{i+1} = R^2_{i-1}f_{i-1} + R^2_i f_i + R^2_{i+1}f_{i+1} + (L^2_I J_{\alpha 2} - R^2_I J_{\alpha 0}), \tag{11}$$

where L^n_i and R^n_i are, respectively, the left-hand side and right-hand side coefficients of the n-th derivative approximation and $J_{\alpha i}$ are the Taylor expansions of the jump of $f^{(n)}$ at point $x = x_\alpha$. This jumps can be computed by the following equations, depending on the position of x_α.

If $x_i < x_\alpha < x_{i+1}$, $I = i + 1$

$$J_{\alpha 0} = [\![f^{(0)}]\!]_\alpha + h^+[\![f^{(1)}]\!]_\alpha + \frac{(h^+)^2}{2!}[\![f^{(2)}]\!]_\alpha + \frac{(h^+)^3}{3!}[\![f^{(3)}]\!]_\alpha + \frac{(h^+)^4}{4!}[\![f^{(4)}]\!]_\alpha + \frac{(h^+)^5}{5!}[\![f^{(5)}]\!]_\alpha, \tag{12}$$

$$J_{\alpha 1} = [\![f^{(1)}]\!]_\alpha + h^+[\![f^{(2)}]\!]_\alpha + \frac{(h^+)^2}{2!}[\![f^{(3)}]\!]_\alpha + \frac{(h^+)^3}{3!}[\![f^{(4)}]\!]_\alpha + \frac{(h^+)^4}{4!}[\![f^{(5)}]\!]_\alpha, \tag{13}$$

$$J_{\alpha 2} = [\![f^{(2)}]\!]_\alpha + h^+[\![f^{(3)}]\!]_\alpha + \frac{(h^+)^2}{2!}[\![f^{(4)}]\!]_\alpha + \frac{(h^+)^3}{3!}[\![f^{(5)}]\!]_\alpha, \tag{14}$$

or if $x_{i-1} < x_\alpha < x_i$, $I = i - 1$

$$J_{\alpha 0} = -[\![f^{(0)}]\!]_\alpha + h^-[\![f^{(1)}]\!]_\alpha - \frac{(h^-)^2}{2!}[\![f^{(2)}]\!]_\alpha + \frac{(h^-)^3}{3!}[\![f^{(3)}]\!]_\alpha - \frac{(h^-)^4}{4!}[\![f^{(4)}]\!]_\alpha + \frac{(h^-)^5}{5!}[\![f^{(5)}]\!]_\alpha, \tag{15}$$

$$J_{\alpha 1} = -[\![f^{(1)}]\!]_\alpha + h^-[\![f^{(2)}]\!]_\alpha - \frac{(h^-)^2}{2!}[\![f^{(3)}]\!]_\alpha + \frac{(h^-)^3}{3!}[\![f^{(4)}]\!]_\alpha - \frac{(h^-)^4}{4!}[\![f^{(5)}]\!]_\alpha, \tag{16}$$

$$J_{\alpha 2} = -[\![f^{(2)}]\!]_\alpha + h^-[\![f^{(3)}]\!]_\alpha - \frac{(h^-)^2}{2!}[\![f^{(4)}]\!]_\alpha + \frac{(h^-)^3}{3!}[\![f^{(5)}]\!]_\alpha. \tag{17}$$

The jumps of f can be computed by

$$[\![f^{(n)}]\!]_\alpha = f^{(n)}_+ - f^{(n)}_-, \tag{18}$$

where

$$f^{(n)}_+ = c_{1_{\alpha^+}} f^+_\alpha + c_{1_{i+2}} f_{i+2} + c_{1_{i+3}} f_{i+3} + c_{1_{i+4}} f_{i+4} + c_{1_{i+5}} f_{i+5} + c_{1_{i+6}} f_{i+6}, \tag{19}$$

$$f^{(n)}_- = c_{1_{\alpha^-}} f^-_\alpha + c_{1_{i-1}} f_{i-1} + c_{1_{i-2}} f_{i-2} + c_{1_{i-3}} f_{i-3} + c_{1_{i-4}} f_{i-4} + c_{1_{i-5}} f_{i-5}, \tag{20}$$

in the case that $x_i < x_\alpha < x_{i+1}$.

Coefficients c_n to calculate

$$f^{(n)}_\alpha = c_\alpha f_\alpha + c_i f_i + c_{i+1} f_{i+1} + c_{i+2} f_{i+2} + c_{i+3} f_{i+3} + c_{i+4} f_{i+4} \tag{21}$$

are obtained solving the linear system

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & h_i & h_{i+1} & h_{i+2} & h_{i+3} & h_{i+4} \\ 0 & h_i^2 & h_{i+1}^2 & h_{i+2}^2 & h_{i+3}^2 & h_{i+4}^2 \\ 0 & h_i^3 & h_{i+1}^3 & h_{i+2}^3 & h_{i+3}^3 & h_{i+4}^3 \\ 0 & h_i^4 & h_{i+1}^4 & h_{i+2}^4 & h_{i+3}^4 & h_{i+4}^4 \\ 0 & h_i^5 & h_{i+1}^5 & h_{i+2}^5 & h_{i+3}^5 & h_{i+4}^5 \end{bmatrix} \begin{bmatrix} c_\alpha \\ c_i \\ c_{i+1} \\ c_{i+2} \\ c_{i+3} \\ c_{i+4} \end{bmatrix} = \begin{bmatrix} 1\delta_{n0} \\ 1\delta_{n1} \\ 2!\delta_{n2} \\ 3!\delta_{n3} \\ 4!\delta_{n4} \\ 5!\delta_{n5} \end{bmatrix}, \tag{22}$$

where $h_i = x_i - x_\alpha$ and δ_{ij} is the Kronecker's delta function

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases} \tag{23}$$

3.1 Spatial discretization

Three-point fourth-order compact finite difference schemes were used to compute numerical approximations to the spatial derivatives. For the first derivative the scheme is given by:

$$\frac{1}{3}f_{i-1}^{(1)} + \frac{4}{3}f_i^{(1)} + \frac{1}{3}f_{i+1}^{(1)} = \frac{-1}{2\Delta x}f_{i-1} + \frac{1}{2\Delta x}f_{i+1} + (L_I^1 J_{\alpha 1} - R_I^1 J_{\alpha 0}), \tag{24}$$

where $L_I^1 = \frac{1}{3}$ and $R_I^1 = \frac{-1}{2\Delta x}$ if $I = i-1$ or $L_I^1 = \frac{1}{3}$ and $R_I^1 = \frac{1}{2\Delta x}$ if $I = i+1$.

The second derivative can be obtained by:

$$\frac{1}{12}f_{i-1}^{(2)} + \frac{10}{12}f_i^{(2)} + \frac{1}{12}f_{i+1}^{(2)} = \frac{1}{\Delta x^2}f_{i-1} - \frac{2}{\Delta x^2}f_i + \frac{1}{\Delta x^2}f_{i+1} + (L_I^2 J_{\alpha 2} - R_I^2 J_{\alpha 0}), \tag{25}$$

where $L_I^2 = \frac{1}{12}$ and $R_I^2 = \frac{1}{\Delta x^2}$ if $I = i-1$ or $I = i+1$.

In both schemes $J_{\alpha i}$ can be calculated by Eq. (12)-(14) or Eq. (15)-(17), depending on the value of I.

In the next section the derivative calculation and Poisson equation solver are verified when discontinuities are present in the domain.

4. NUMERICAL TESTS

This section is divided in two subsections: the first one is devoted to derivative calculations and the second one shows the Poisson equation solutions.

4.1 Derivative calculation

A function $f = \cos(x)$ was used to verify the accuracy of the derivative approximations. The domain considered in this case was the interval $[0, 2\pi]$, with the discontinuity being a circle centered in $x = \pi$ and radius $r = 0.6$. Inside the discontinuity the function and its derivatives are set to be zero. In Fig. 1 are presented exact and approximated solutions to the calculation of first and second derivatives. The continuous lines represent the exact solutions while the dashed ones represent the approximated solutions.

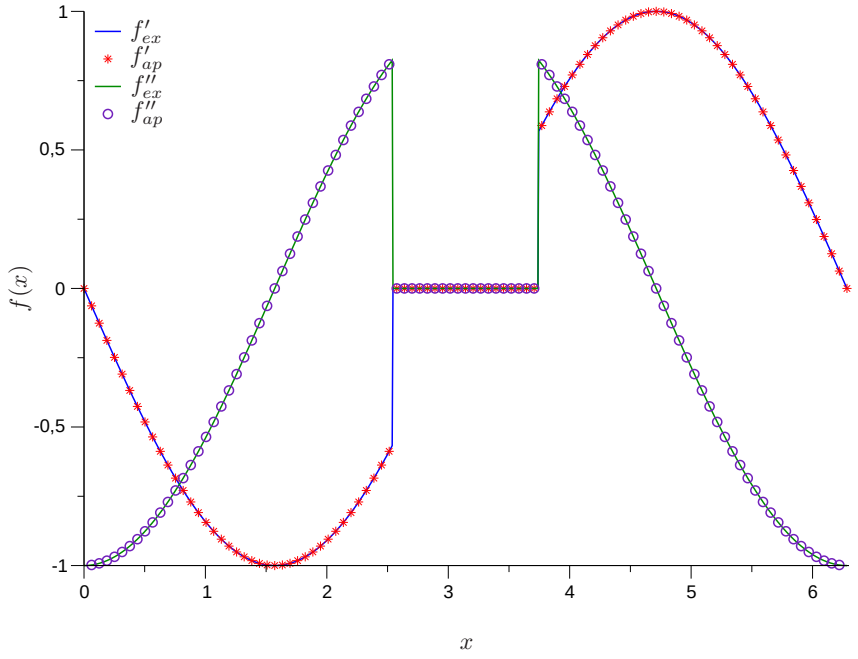

Figure 1. Exact and approximated solutions for first and second derivatives.

It can be observed that the method was capable to identify the discontinuity and do not smear the solution next the interface. Tables 1 and 2 show the convergence of the method in the L_∞ norm, i.e. $\|\cdot\|_\infty = \max_{1\leq i\leq N}[f_i - f(x_i)]$, where N is the number of points, f_i is the approximated solution at point x_i and $f(x_i)$ is the exact solution at point x_i.

The convergence order is given by

$$p = \frac{\log\left(\frac{\|\cdot\|_{\infty_{2\Delta x}}}{\|\cdot\|_{\infty_{\Delta x}}}\right)}{\log(2)}. \tag{26}$$

In particular in Tab. 1, the convergence order is higher than expected.

Table 1. L_∞ norm and convergence order for first derivative.

Δx	$\|\cdot\|_\infty$	p
1.2566×10^{-1}	6.6127×10^{-6}	–
6.2831×10^{-2}	2.0767×10^{-7}	4.9929
3.1415×10^{-2}	6.4977×10^{-9}	4.9982
2.0943×10^{-2}	1.0690×10^{-9}	4.4510

Table 2. L_∞ norm and convergence order for second derivative.

Δx	$\|\cdot\|_\infty$	p
1.2566×10^{-1}	2.7700×10^{-4}	–
6.2831×10^{-2}	1.7416×10^{-5}	3.9914
3.1415×10^{-2}	1.0901×10^{-6}	3.9979
2.0943×10^{-2}	2.1542×10^{-7}	3.9989

As a result, one can see that the IIM achieves fourth-order to calculate the derivatives. Next subsection presents the accuracy of this method in a Poisson equation solver.

4.2 Poisson equation solver

Two cases were considered to verify the accuracy of the Poisson equation solver. The first one is a unidimensional problem, $f_{xx} = s$, with the same domain used in Section 4.1. In this case, it was used a function $f = \cos(x)$, with source term $s = -\cos(x)$. The difference between exact and approximated solutions can be seen in Fig. 2.

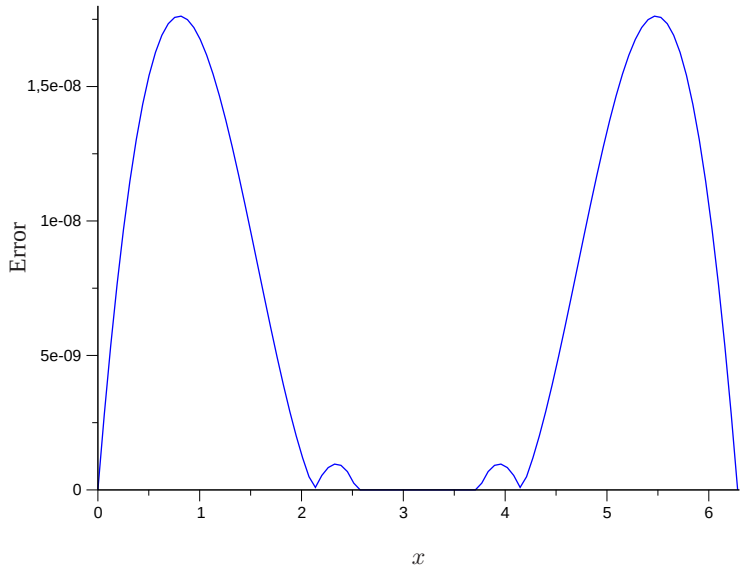

Figure 2. Error for Poisson equation solver.

One can note that the error is greater away from the interface. The convergence order is shown in Tab. 3, where again fourth-order is achieved.

Table 3. L_∞ norm and convergence order for Poisson equation solver.

Δx	$\|\cdot\|_\infty$	p
6.2831×10^{-2}	1.7616×10^{-8}	–
3.1415×10^{-2}	1.1039×10^{-9}	3.9962
1.5707×10^{-2}	6.7536×10^{-11}	4.0308
7.8539×10^{-3}	3.8915×10^{-12}	4.1172

The second case is a two-dimensional problem, $f_{xx} + f_{yy} = s$. The domain is a square $[0, 2\pi] \times [0, 2\pi]$ with a cylinder centered in (π, π) and with radius $r = 0.6$. A function $f = \sin(x)\cos(y)$ was used to verify the accuracy of the Poisson equation solver, with source term $s = -2\sin(x)\cos(y)$.

The Poisson equation was solved including and not including the cylinder into the domain. The error for both cases can be seen in Fig. 3. It shows that the presence of the discontinuity does not increase the error near the interface.

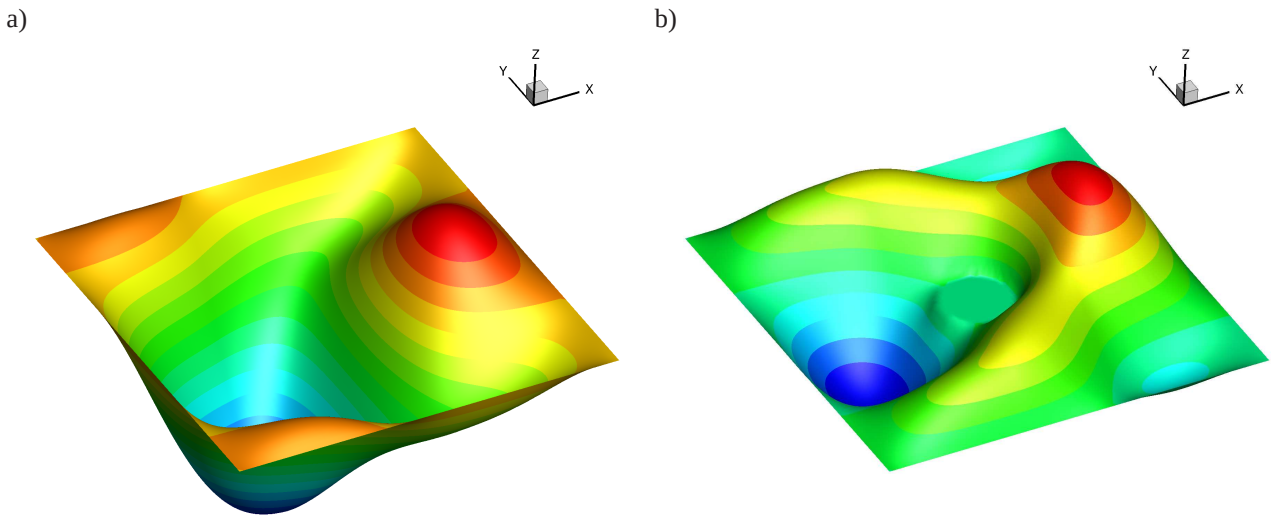

Figure 3. Error for Poisson equation solver: a) without discontinuity; b) with discontinuity.

In the next section are presented the results of a simulation of an incompressible flow using the proposed method.

5. RESULTS

The IIM was applied in the solution of an incompressible flow in stream function-vorticity formulation. A classical fourth-order Runge-Kutta method was used for the time integration. The test case is a Poiseuille flow over a cylinder. The domain is shown in Fig. 4, where $r = 0.15$ is the cylinder radius and $(4.05, 1.05)$ is the center of the cylinder.

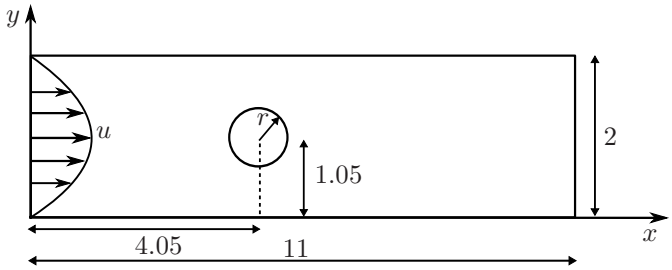

Figure 4. Domain for a Poiseuille flow over a cylinder.

For this simulation a Reynolds number $Re = 25$ was considered. The contour of vorticity near the interface can be seen in Fig. 5 while the velocity field near the interface can be seen in Fig. 6.

This results show that the method could identify the discontinuity in the domain.

6. CONCLUSIONS

In this paper a fourth-order Immersed Interface Method was used to solve incompressible flow. The derivative calculations and the Poisson equation solver were verified to check the accuracy of the method. The results show that the convergence order is not affected by the presence of discontinuities inside the domain. The method was used in 2D simulation of a Poiseuille incompressible flow over a cylinder. The behavior of the method was qualitative checked by verification of velocity and vorticity fields. Further investigations should be carried on to confirm the method accuracy in 2D simulations.

7. ACKNOWLEDGEMENTS

The authors acknowledge the financial support received from FAPESP under grant 2010/00880-2.

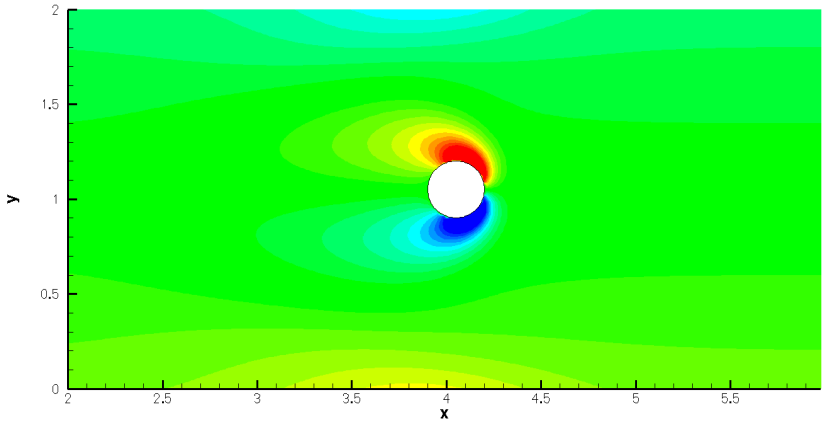

Figure 5. Isovorticity contours near the cylinder.

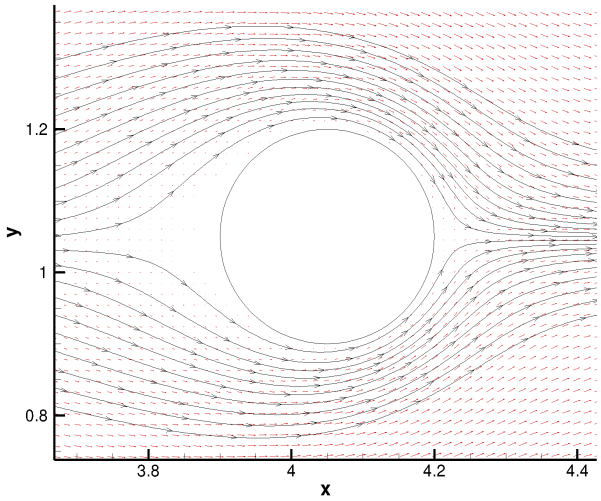

Figure 6. Velocity vectors and streamlines near the cylinder.

8. REFERENCES

Goldstein, D., Handler, R. and Sirovich, L., 1993. "Modeling a no-slip flow boundary with an external force field". *Journal of Computational Physics*, Vol. 105, No. 2, pp. 354–366. ISSN 0021-9991.

LeVeque, R.J. and Li, Z., 1994. "The immersed interface method for elliptic equations with discontinuos coefficients and singular sources". *SIAM Journal on Numerical Analysis*, Vol. 31, pp. 1019–1044.

Li, Z. and Ito, K., 2006. *The Immersed Interface Method - Numerical Solutions of PDEs Involving Interfaces and Irregular Domains*. SIAM.

Lima E Silva, A., Silveira-Neto, A. and Damasceno, J.J.R., 2003. "Numerical simulation of two-dimensional flows over a circular cylinder using the immersed boundary method". *J. Comput. Phys.*, Vol. 189, pp. 351–370. ISSN 0021-9991.

Linnick, M.N. and Fasel, H.F., 2005. "A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains". *Journal of Computational Physics*, Vol. 204, pp. 157–192.

Peskin, C., 1972. "Flow patterns around heart valves: A numerical method". *Journal of Computational Physics*, Vol. 10, pp. 252–271.

Peskin, C., 1977. "Numerical analysis of blood flow in the heart". *Journal of Computational Physics*, Vol. 25, pp. 220–252.

Peskin, C., 2002. "The immersed boundary method". *Acta Numerica*, Vol. 11, pp. 479–517.

Roma, A.M., Peskin, C.S. and Berger, M.J., 1999. "An adaptive version of the immersed boundary method". *Journal of Computational Physics*, Vol. 153, No. 2, pp. 509 – 534. ISSN 0021-9991.

Wiegmann, A. and Bube, K.P., 2000. "The explicit-jump immersed interface method: finite difference methods for PEDs with piecewise smooth solutions". *SIAM Journal on Numerical Analysis*, Vol. 37, pp. 827–862.

9. Responsibility notice

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