ISSN 2176-5480

22nd International Congress of Mechanical Engineering (COBEM 2013)
November 3-7, 2013, Ribeirão Preto, SP, Brazil

FAST AND PRECISE SOLUTION FOR THE POISSON EQUATION IN THE PRESENCE OF INTERFACES WITH DISCONTINUITIES

Gabriela Aparecida dos Reis
Italo Valença Mariotti Tasso
Leandro Franco de Souza
José Alberto Cuminato
Department of Applied Mathematics and Statistics, ICMC, USP, São Carlos, SP
gareis@icmc.usp.br, italo@tasso.com.br, lefraso@icmc.usp.br, jacumina@icmc.usp.br

Abstract. *The simulation of free surface flows has attracted much attention in the last years due to the many practical applications in which this type of flow is involved. There are considerable numerical and computational challenges posed by this subject. Besides solving the Navier-Stokes equations, which by itself is a challenge, one must also represent correctly the time varying domain. One way to accomplish this is by means of the Immersed Interface Method (IIM), which allows for the use of high order approximations even in the vicinity of discontinuities. The present work describes the solution of the Poisson equation using a fourth-order version of the IIM. The current IIM methodologies use iterative processes when the jump condition does not have an analytical expression, and a linear system must be solved for every iteration. To overcome that, we propose a version of the IIM that deals with the jump condition implicitly, as part of the linear system solution. The linear system is solved only once, yielding a significant gain in processing time while still maintaining fourth order precision. Furthermore, this processing time is very close to that necessary for solving Poisson's equation without the immersed interface. This version of the IIM is specially suitable for solving the Navier-Stokes equation using the projection method, since Poisson's equation solution consumes most of the processing time.*

Keywords: *Navier-Stokes equation, immersed interface method, Poisson equation, high-order methods.*

1. INTRODUCTION

The study of robust and efficient techniques for the solution of the Poisson equation has always been of great interest. In particular, when solving the Navier-Stokes equation by the projection method (Chorin, 1967, 1968), a Poisson equation needs to be solved at each time step, which consumes most of the processing time. A big challenge that needs to be dealt with is the presence of discontinuities in the domain. From a computational perspective, there are several numerical methods designed for smooth functions that behave poorly on problems with discontinuities, or that don't work at all due to the irregularities present. One way to overcome this problem is the Immersed Interface Method (IIM), which allows for the use of high-order approximations even in the presence of discontinuities.

The IIM (LeVeque and Li, 1994) is second-order accurate and can deal with more general interfaces than the Immersed Boundary Method, IBM, (Peskin, 1972), which is just first-order accurate. The IIM has been considered as an alternative to the traditional numerical approaches used to solve initial or boundary value problems on domains with irregular geometries. A considerable improvement on the IIM has been the explicit jump approach (Wiegmann and Bube, 2000), where the authors made a simple but important remark: finite difference techniques fail when applied to non-smooth functions because the Taylor series expansions, on which they are based, are not valid. In this context, a Taylor series expansion which includes jumps is derived to get second-order accurate approximations. According to (Linnick and Fasel, 2005), the main idea of the IIM is that the finite difference schemes must be corrected on the immersed interfaces to preserve the accuracy of the method. Moreover, according to (Li and Ito, 2006), the IIM requires a priori knowledge of the jump conditions, which can be extracted from physical information or from the governing differential equations.

Given a set of partial differential equations to be solved on a domain containing an interface or discontinuity line, the idea behind the IIM is to represent this interface in such way that the field variables, or their derivatives, are discontinuous across it. To model this discontinuity on the interface, the coefficients of the finite differences are changed and correction terms are introduced on the right-hand side of the equation. These correction terms are calculated based on the jump conditions and their derivatives.

High-order approximations can be obtained with compact finite differences. Although there is an increase in the computational cost, this strategy is preferred according to (Souza *et al.*, 2005) and (Lele, 1992) because it has a stencil with fewer points, smaller error and high resolution.

The method proposed by (Linnick and Fasel, 2005) follows the ideas of (Wiegmann and Bube, 2000), where the authors present high-order IIMs to solve the Navier-Stokes equations on an incompressible fluid. The method has a downside, which is the use of iterative processes when the jump condition does not have an analytical expression. The linear system has to be solved again at each iteration, demanding substantial processing time. Moreover, there is a relaxation parameter that needs to be manually calibrated for each problem. With that in mind, we propose a new version of the IIM which deals with the jump conditions implicitly, directly as part of the linear system. The linear system is solved only once, which yields a meaningful performance gain while ensuring fourth-order accuracy.

The remainder of the article is organized as follows: section 2 introduces the equations on which the IIM is based. Section 3 presents the explicit IIM proposed by (Linnick and Fasel, 2005) and the implicit version proposed by their work. The 2D Poisson equation discretization is also derived. Section 4 brings numerical results. Conclusions are on section 5.

2. THE IMMERSED INTERFACE METHOD

Consider a function $f(x)$ with a discontinuity at $x = x_\alpha$. We would like to use the Taylor series expansion around x_i to approximate $f(x)$ at x_{i+1}. Assuming that $f(x)$ is known at every point of the domain $D = \{x|x_{i-1} \leq x \leq x_{i+1}\}$, except at x_α where there is a jump (discontinuity) on the value of the function or its derivatives. If $x_i < x_\alpha < x_{i+1}$, the standard Taylor series expansion cannot be used to approximate $f(x_{i+1})$, unless a correction term J_α is introduced:

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

where

$$J_\alpha = [f]_\alpha + [f']_\alpha h^+ \frac{1}{2!}[f'']_\alpha (h^+)^2 + \cdots, \tag{2}$$

$h = x_{i+1} - x_i$ and $h^+ = x_{i+1} - x_\alpha$. The term $[f]_\alpha$ represents the jump on the value of f at $x = x_\alpha$, i.e.,

$$[f]_\alpha = \lim_{x\to x_\alpha^+} f(x) - \lim_{x\to x_\alpha^-} f(x), \tag{3}$$

and $[f']_\alpha$, $[f'']_\alpha$, etc. represent the jump on the value of the derivatives of f.

Using the correction term J_α we can modify any finite difference method, and the result will retain the order of accuracy of the original method when the stencil crosses a singularity or jump.

2.1 NUMERICAL METHOD

In the presence of discontinuities, a compact finite difference approximation for the first derivative of f can be written as

$$L^1_{i-1} f^{(1)}_{i-1} + L^1_i f^{(1)}_i + L^1_{i+1} f^{(1)}_{i+1} = R^1_{i-1} f_{i-1} + R^1_i f_i + R^1_{i+1} f_{i+1} + (L^1_I J_{\alpha 1} - R^1_I J_{\alpha 0}), \tag{4}$$

and for the second derivative,

$$L^2_{i-1} f^{(2)}_{i-1} + L^2_i f^{(2)}_i + L^1_{i+1} f^{(2)}_{i+1} = R^2_{i-1} f_{i-1} + R^2_i f_i + R^2_{i+1} f_{i+1} + (L^2_I J_{\alpha 2} - R^2_I J_{\alpha 0}), \tag{5}$$

where L^n_i and R^n_i are the coefficients of the left-hand and right-hand sides of the approximation for the n-th derivative and $J_{\alpha n}$ are the Taylor series expansions of the jumps of $f^{(n)}$ at $x = x_\alpha$.

In these two schemes, $I = i + 1$ if the jump is at $x_i < x_\alpha < x_{i+1}$, $h^+ = x_{i+1} - x_\alpha$ and

$$J_{\alpha 0} = [f^{(0)}]_\alpha + (h^+)[f^{(1)}]_\alpha + \frac{(h^+)^2}{2!}[f^{(2)}]_\alpha + \frac{(h^+)^3}{3!}[f^{(3)}]_\alpha + \frac{(h^+)^4}{4!}[f^{(4)}]_\alpha + \frac{(h^+)^5}{5!}[f^{(5)}]_\alpha, \tag{6}$$

$$J_{\alpha 1} = [f^{(1)}]_\alpha + (h^+)[f^{(2)}]_\alpha + \frac{(h^+)^2}{2!}[f^{(3)}]_\alpha + \frac{(h^+)^3}{3!}[f^{(4)}]_\alpha + \frac{(h^+)^4}{4!}[f^{(5)}]_\alpha, \tag{7}$$

$$J_{\alpha 2} = [f^{(2)}]_\alpha + (h^+)[f^{(3)}]_\alpha + \frac{(h^+)^2}{2!}[f^{(4)}]_\alpha + \frac{(h^+)^3}{3!}[f^{(5)}]_\alpha, \tag{8}$$

are the approximations used for a fourth-order accurate method. If the jump is at $x_{i-1} < x_\alpha < x_i$, then $I = i - 1$, $h^- = x_\alpha - x_{i-1}$ and

$$J_{\alpha 0} = -[f^{(0)}]_\alpha + (h^-)[f^{(1)}]_\alpha - \frac{(h^-)^2}{2!}[f^{(2)}]_\alpha + \frac{(h^-)^3}{3!}[f^{(3)}]_\alpha - \frac{(h^-)^4}{4!}[f^{(4)}]_\alpha + \frac{(h^-)^5}{5!}[f^{(5)}]_\alpha, \tag{9}$$

$$J_{\alpha 1} = -[f^{(1)}]_\alpha + (h^-)[f^{(2)}]_\alpha - \frac{(h^-)^2}{2!}[f^{(3)}]_\alpha + \frac{(h^-)^3}{3!}[f^{(4)}]_\alpha - \frac{(h^-)^4}{4!}[f^{(5)}]_\alpha, \tag{10}$$

$$J_{\alpha 2} = -[f^{(2)}]_\alpha + (h^-)[f^{(3)}]_\alpha - \frac{(h^-)^2}{2!}[f^{(4)}]_\alpha + \frac{(h^-)^3}{3!}[f^{(5)}]_\alpha. \tag{11}$$

The jumps are

$$[f^{(n)}]_\alpha = \lim_{x\to x_\alpha^+} f^{(n)}(x) - \lim_{x\to x_\alpha^-} f^{(n)}(x) = f_+^{(n)} - f_-^{(n)}, \tag{12}$$

where

$$f_+^{(n)} = c_{n_{\alpha+}} f_\alpha + c_{n_{i+2}} f_{i+2} + c_{n_{i+3}} f_{i+3} + c_{n_{i+4}} f_{i+4} + c_{n_{i+5}} f_{i+5} + c_{n_{i+6}} f_{i+6}, \tag{13}$$

$$f_-^{(n)} = c_{n_{\alpha-}} f_\alpha + c_{n_{i-1}} f_{i-1} + c_{n_{i-2}} f_{i-2} + c_{n_{i-3}} f_{i-3} + c_{n_{i-4}} f_{i-4} + c_{n_{i-5}} f_{i-5}. \tag{14}$$

Notice that the points x_i and x_{i+1} were excluded to avoid instabilities.

The coefficients c_n of

$$f_\alpha^{(n)} = c_\alpha f_\alpha + c_i f_i + c_{i+1} f_{i+1} + c_{i+2} f_{i+2} + c_{i+3} f_{i+3} + c_{i+4} f_{i+4}, \tag{15}$$

are obtained by solving the linear system:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & h_i & h_{i+1} & h_{i+2} & h_{i+3} & h_{i+4} \\ 0 & h_i^2 & h_{i+1}^2 & h_{i+2}^2 & h_{i+3}^2 & h_{i+4}^2 \\ 0 & h_i^3 & h_{i+1}^3 & h_{i+2}^3 & h_{i+3}^3 & h_{i+4}^3 \\ 0 & h_i^4 & h_{i+1}^4 & h_{i+2}^4 & h_{i+3}^4 & h_{i+4}^4 \\ 0 & h_i^5 & h_{i+1}^5 & h_{i+2}^5 & h_{i+3}^5 & h_{i+4}^5 \end{bmatrix} \begin{bmatrix} c_\alpha \\ c_i \\ c_{i+1} \\ c_{i+2} \\ c_{i+3} \\ c_{i+4} \end{bmatrix} = \begin{bmatrix} 1\delta_{n0} \\ 1\delta_{n1} \\ 2!\delta_{n2} \\ 3!\delta_{n3} \\ 4!\delta_{n4} \\ 5!\delta_{n5} \end{bmatrix} \tag{16}$$

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

$$\delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}. \tag{17}$$

In the absence of discontinuities, fourth-order finite difference schemes, according to (Lele, 1992), for the first and second derivatives are

$$f_{i-1}^{(1)} + 4f_i^{(1)} + f_{i+1}^{(1)} = \frac{3(f_{i+1} - f_{i-1})}{h} + O(h^4), \tag{18}$$

$$f_{i-1}^{(2)} + 10f_i^{(2)} + f_{i+1}^{(2)} = \frac{12(f_{i-1} - 2f_i + f_{i+1})}{h^2} + O(h^4), \tag{19}$$

which provide adequate choices for the coefficients L_i^n and R_i^n in equations (4) and (5).

3. THE IMMERSED INTERFACE METHOD WITH IMPLICIT JUMP CONDITIONS

Consider the Poisson equation

$$\nabla^2 f = t_f \tag{20}$$

defined on a domain Ω^+, with boundary conditions on $\partial\Omega_0$, which contains an immersed interface $\partial\Omega_i$, as seen in Figure 1. The solution in Ω^- may or may not be of interest and, in this work, it will be considered zero.

For the 1D case, f can be calculated from (5), as

$$R_{i-1}^2 f_{i-1} + R_i^2 f_i + R_{i+1}^2 f_{i+1} = L_{i-1}^2 f_{i-1}^{(2)} + L_i^2 f_i^{(2)} + L_{i+1}^1 f_{i+1}^{(2)} - (L_I^2 J_{\alpha 2} - R_I^2 J_{\alpha 0}). \tag{21}$$

The value of $J_{\alpha n}$ comes from (6)-(11), where the value of f is needed but unknown. A solution is to start with a tentative value for f and improve it iteratively as described in Algorithm 1, where β is a relaxation parameter and ϕ is the signed distance-to-interface function,

$$\phi(x) \begin{cases} < 0, & x \in \Omega^- \\ = 0, & x \in \partial\Omega_i \\ > 0, & x \in \Omega^+ \end{cases}, \tag{22}$$

from which h^+ e h^- can be calculated. This was proposed by (Linnick and Fasel, 2005) and will be referred as *immersed interface method with explicit jump corrections* (IIM-E). As discussed earlier, a disadvantage of this formulation is that it

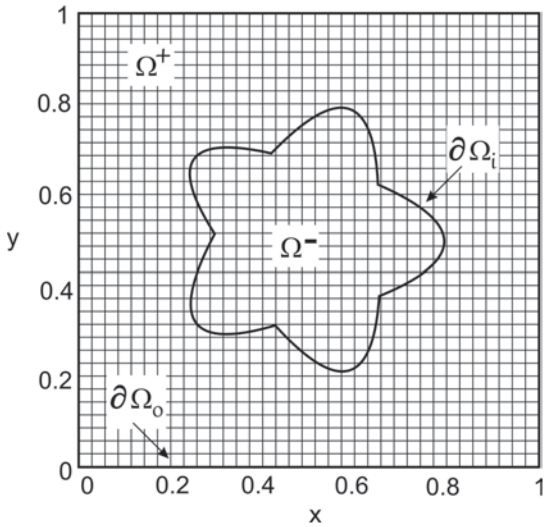

Figure 1. Illustration of the domain for the immersed interface problem, extracted from (Linnick and Fasel, 2005).

Algorithm 1

$f = 0, f_{\text{new}} = 0$
for $i = 0 \rightarrow \text{iter}_{max}$ **do**
poisson_solve($f, t_f, f_{\text{new}}, \phi$)
if $|f - f_{\text{new}}| < \varepsilon$ **then**
break;
else
$f = \beta f_{\text{new}} + (1-\beta) f$
end if
end for

is an iterative procedure. The relaxation parameter β influences the convergence rate and must be calibrated manually for each problem. A bad choice can mean a very slow convergence or no convergence at all. This is even more serious for problems involving free surfaces. Since the interface shape changes, so does the optimal value for β.

As an alternative, we propose calculating the correction $J_{\alpha n}$ implicitly, incorporating it into the linear system. So, equation (21) becomes

$$R^2_{i-1} f_{i-1} + R^2_i f_i + R^2_{i+1} f_{i+1} + (L^2_I J^*_{\alpha 2} - R^2_I J^*_{\alpha 0}) = L^2_{i-1} f^{(2)}_{i-1} + L^2_i f^{(2)}_i + L^1_{i+1} f^{(2)}_{i+1} - (L^2_I J^{**}_{\alpha 2} - R^2_I J^{**}_{\alpha 0}). \quad (23)$$

The jump conditions $J_{\alpha 2}, J_{\alpha 0}$ can be incorporated into the matrix of the linear system since they are approximated as linear functions of the discrete values of f. Their values will be calculated directly, without the need for an iterative procedure. The starred J^*_α are the ones that involve unknown values of f, usually inside the domain, and the double-starred J^{**}_α involve known values of f, usually boundary conditions. The resulting liner system is solved by direct methods with the library PARDISO (Schenk and Gärtner, 2004). Other solving techniques, such as multigrid methods, are being evaluated.

This new method is called *immersed interface method with implicit jump corrections* (IIM-I).

3.1 THE $2D$ POISSON EQUATION

The discretization of the Poisson equation (20) in two dimensions is based upon two one-dimension compact finite difference approximations

$$\sum_{i=1}^{3} L^{xx}_i f^{xx}_{ij} = \sum_{i=1}^{3} R^{xx}_i f_{ij}, \quad (24)$$

$$\sum_{j=1}^{3} L^{yy}_j f^{yy}_{ij} = \sum_{j=1}^{3} R^{yy}_j f_{ij}, \quad (25)$$

where $L^{xx}, L^{yy}, R^{xx}, R^{yy}$ indicate the coefficients of the scheme, for example the ones on equation 19, while f^{xx} and f^{yy} are the discrete approximations for the second derivatives in the x and y directions. The discrete Poisson equation is

ISSN 2176-5480

$$f_{ij}^{xx} + f_{ij}^{yy} = t_{f_{ij}}, \quad (26)$$
$$L_i^{xx} L_j^{yy} (f_{ij}^{xx} + f_{ij}^{yy}) = L_i^{xx} L_j^{yy} t_{f_{ij}}, \quad (27)$$
$$L_j^{yy} (L_i^{xx} f_{ij}^{xx}) + L_i^{xx} (L_j^{yy} f_{ij}^{yy}) = L_i^{xx} L_j^{yy} t_{f_{ij}}, \quad (28)$$

where the summation is implied (Einstein notation). Using (24) and (25),

$$L_j^{yy} (R_i^{xx} f_{ij}) + L_i^{xx} (R_j^{yy} f_{ij}) = L_i^{xx} L_j^{yy} t_{f_{ij}}, \quad (29)$$
$$(R_i^{xx} L_j^{yy} + L_i^{xx} R_j^{yy}) f_{ij} = L_i^{xx} L_j^{yy} t_{f_{ij}}. \quad (30)$$

Therefore, when jump corrections are not needed, the $2D$ compact nine-point scheme centered at (i, j) is

$$L_{ij} f_{ij} = R_{ij} t_{f_{ij}}, \quad (31)$$

where

$$L_{ij} = R_i^{xx} L_j^{yy} + L_i^{xx} R_j^{yy}, \quad (32)$$
$$R_{ij} = L_i^{xx} L_j^{yy}. \quad (33)$$

Notice that the $2D$ scheme shown in (31) was obtained from the combination of two $1D$ schemes, one for x and the other for y. As a consequence, when the $2D$ stencil intersects the immersed interface, see Figure 2, the corresponding $2D$ system can be corrected using the $1D$ method, i.e., (24) and (25) become

$$L_i^{xx} f_{ij}^{xx} = R_i^{xx} f_{ij} - J_{\alpha x I}, \quad (34)$$
$$L_j^{yy} f_{ij}^{yy} = R_j^{yy} f_{ij} - J_{\alpha y J}. \quad (35)$$

For convenience, we define

$$J_{\alpha x I} = -(L_I^{xx} J_{\alpha 2x} - R_I^{xx} J_{\alpha 0x}), \quad (36)$$
$$J_{\alpha y J} = -(L_J^{yy} J_{\alpha 2y} - R_J^{yy} J_{\alpha 0y}), \quad (37)$$

and,

$$I = \begin{cases} i-1, & \text{if} \quad x_{i-1} < x_\alpha < x_i \\ i+1, & \text{if} \quad x_i < x_\alpha < x_{i+1} \end{cases}, \quad (38)$$

and similarly for J (J the subscript, not to be confused with the jump J). The resulting discretization for the $2D$ Poisson equation with jump corrections becomes

$$L_{ij} f_{ij} = R_{ij} t_{f_{ij}} + (L_j^{yy} J_{\alpha x I} + L_i^{xx} J_{\alpha y J}). \quad (39)$$

4. RESULTS

Consider the $2D$ Poisson equation

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = t_f, \quad (40)$$

defined over a unity square, with source term (t_f) and exact solution $f(x, y)$ given by

$$t_f = \begin{cases} 0 & \text{if} \quad (x-0.5)^2 + (y-0.5)^2 \le 0.05123^2 \\ -4\pi^2 \sin(2\pi x)\cos(2\pi y) & \text{otherwise} \end{cases}, \quad (41)$$
$$f(x, y) = 0.5 \sin(2\pi x) \cos(2\pi y), \quad (42)$$

The numerical solution and the corresponding error using the IIM-I are shown in Figure 3. Notice that the largest error is not near the discontinuity. Table 1 shows the error and order of convergence for the IIM-I. The results for the IIM-E

ISSN 2176-5480

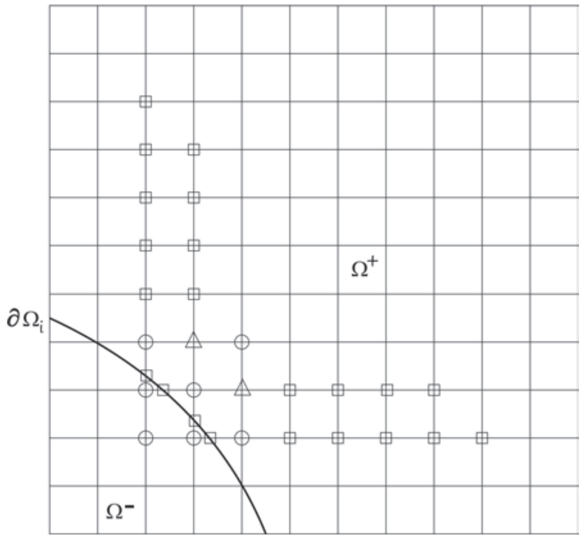

Figure 2. Stencil intersected by the immersed interface, extracted from (Linnick and Fasel, 2005). In the figure, (◯) represents the points of the stencil from (31) and (□) the correction terms. The points where (◯) and (□) overlap are represented by (△).

were omitted because they were the same as the IIM-I. The solution is calculated for different grids and the corresponding error is calculated using the norms

$$L_1 = \frac{1}{N}\left(\sum_{i=1}^{N} E_i\right), \qquad L_2 = \sqrt{\frac{1}{N}\left(\sum_{i=1}^{N} E_i^2\right)}, \qquad L_\infty = \max_i\,(E_i), \tag{43}$$

where $E_i = |f_{\text{numerical}} - f_{\text{exact}}|$ and N is the number of unknowns. The order is calculated from

$$p = \frac{\log\left(\frac{error(h_1)}{error(h_2)}\right)}{\log\left(\frac{h_1}{h_2}\right)}. \tag{44}$$

Table 1 clearly shows a fourth-order convergence rate, as expected.

Table 1. Error and order of convergence for the IIM-I and IIM-E.

grid	L_1 norm		L_2 norm		L_∞ norm	
	error	order	error	order	error	order
25×25	2.207e-06	-	3.539e-06	-	1.498e-05	-
50×50	1.003e-07	4.459	1.426e-07	4.632	4.096e-07	5.192
100×100	5.921e-09	4.083	8.307e-09	4.102	2.460e-08	4.057
200×200	3.675e-10	4.010	5.127e-10	4.018	1.529e-09	4.007
400×400	2.304e-11	3.995	3.206e-11	3.999	9.541e-11	4.003

In order to evaluate the additional computational cost introduced by the immersed interface, equation (40) was also solved without the interface. Table 2 shows the wall clock time, in seconds, for different grid sizes using the IIM-I, IIM-E and without the interface. The simulations were carried out on a dual Intel Xeon E5690 (3.46 GHz, 12MB cache each) with 32GB RAM. Each processor has six physical cores for a total of 24 logical cores due to Hyper-threading. Parallelism was provided by OpenMP (Intel C++ compiler 2013) on a Linux-based OS. The linear solver PARDISO provided by the Intel MKL was used. Our method, IIM-I, outperforms the original IIM-E in all cases tested.

5. CONCLUSIONS

A modification of the IIM-E from (Linnick and Fasel, 2005) has been presented. The new method, IIM-I, reduced considerably the processing time when compared to the original method for all cases tested, while maintaining the same accuracy. For the $2D$ test problem performed, the speedup was around 300% for the finer grids. This is very relevant

ISSN 2176-5480

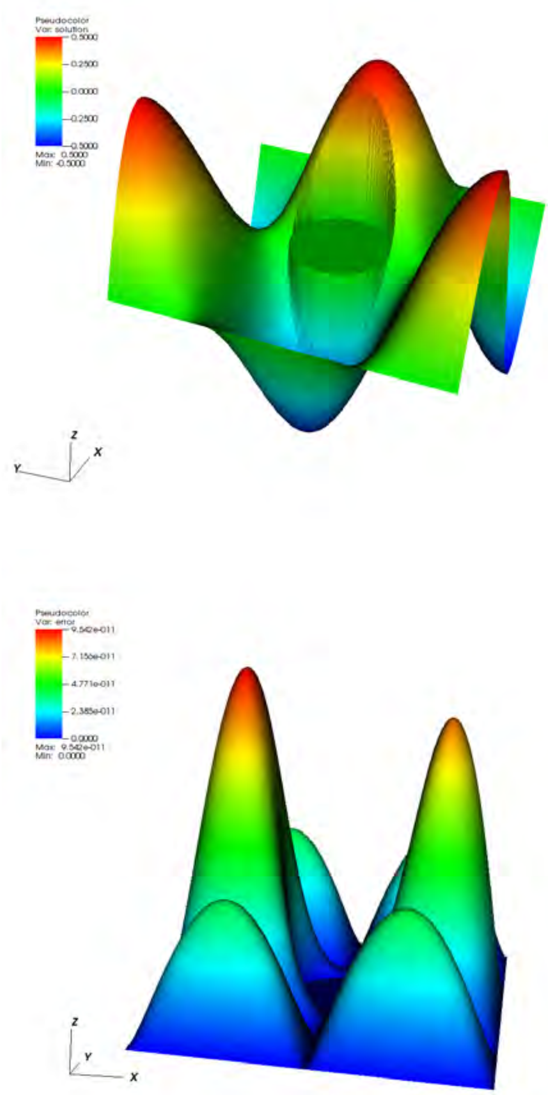

Figure 3. Solution for the Poisson equation (40) using the IIM-I. On the top, the numerical solution, and below, the corresponding error for a 400×400 grid.

Table 2. Processing time.

grid	no interface	IIM-E	IIM-I	speedup
25×25	0.003145s	0.256026s	0.005499s	4655%
50×50	0.008611s	0.171169s	0.011719s	1460%
100×100	0.039583s	0.323859s	0.041598s	778%
200×200	0.16142s	0.84582s	0.159381s	530%
400×400	0.652636s	3.29273s	1.16988s	281%
800×800	2.95684s	15.7838s	4.9852s	316%

when solving the Navier-Stokes with the projection method, since the Poisson equation consumes most of the processing time.

Another advantage is that our method is fully automatic and does not require the manual calibration of the relaxation parameter. This is very relevant for problems involving free surfaces, since the interface is constantly changing shape, and a different shape means a different optimal value for the relaxation parameter.

6. ACKNOWLEDGEMENTS

Financial support provided by FAPESP, process no. 2012/04471-5, is acknowledged.

7. REFERENCES

Chorin, A., 1967. "A numerical method for solving incompressible viscous flow problems". *J. Comp. Physics*, Vol. 2, pp. 12–26.

Chorin, A., 1968. "Numerical solution of the Navier-Stokes". *Mathematics of Computation*, Vol. 2, pp. 745–762.

Lele, S.K., 1992. "Compact finite difference schemes with spectral-like resolution". *Jorunal of Computational Physics*, Vol. 103, pp. 16–42.

LeVeque, R.J. and Li, Z., 1994. "The immersed interface method for elliptic equations with discontinuous coefficients and sigular 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*.

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

Schenk, O. and Gärtner, K., 2004. "Solving unsymmetric sparse systems of linear equations with PARDISO". *Future Generation Computer Systems*, Vol. 20, No. 3, pp. 475–487.

Souza, L.F., Mendonça, M.T. and Medeiros, M.A.F., 2005. "The advantages of using high-order finite differences schemes in laminar–turbulent transition studies". *International journal for numerical methods in fluids*, Vol. 48, No. 5, pp. 565–582.

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

8. RESPONSIBILITY NOTICE

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