Performance Analysis of a High Order Immersed Interface Method for CFD Applications

***P. C. V. Paino, M. A. F. de Medeiros**

Universidade de São Paulo - Escola de Engenharia de São Carlos, Av. Trabalhador Sancarlense, 400, São Carlos - SP, Brazil

*Corresponding author: paulopaino@usp.br

Abstract

In Computational Fluid Dynamics, the physical representation of immersed objects within the computational domains leads to the loss of validity of the employed Finite Difference Schemes due to Jump Discontinuities. This paper analyses an Immersed Interface Method regarding its performance in High Order Schemes applications in the presence of such conditions. The error decay order of one 1D problem is observed. It is related to the computation of the first two derivatives of the Sin function. The results indicate eventual changes in the decay order of the original Finite Differences Schemes. This behaviour is investigated by a fragmented analysis of the method, which indicates a limitation of one of its numerical sub-steps. Finally, some remarks regarding restrictions to this method's applicability are presented.

Keywords: Immersed Interface Method, Immersed Boundary Method, Compact Schemes.

Introduction

This article is intended to provide outlines and to investigate limitation(s) of the Immersed Interface Method, as proposed by (Linnick and Fasel, 2005) and (Wiegmann and Bube, 2000).

This method has been designed to provide for high-order (4th order and above) flow simulations around complex shape bodies. It's suited for problems such as the evolution of Tollmien-Schlichting waves and other problems that require high near-wall accuracy. One of its key advantages is the possibility of working with fixed, stationary grids, even if the immersed geometry moves within the domain. It's intended to significant reduce the overall computational cost.

This method has been extensively analysed by the authors and, though its mathematical formulation does cope with the necessity of order maintenance, one of its inner steps seem to have restrictions regarding grid refinement. This is investigated through its employment to accomplish a simple calculation, that of the second derivative of a sine function over an uniform grid. The reason behind the selection of the sine function falls beyond the existence of an analytical solution to be compared with the numerical one. It lies on the fact that the sine function has an infinite number of derivatives. This is of particular importance not only in order to compare the final solutions, but also in order to perform individual tests throughout the IIM's substeps. This will become clearer later on.

The Immersed Interface Method

In general, Compact Schemes employed are based upon Taylor Series expansions. If the function is not continuous, these become not valid. Nevertheless, suitable corrections to them can still be applied. The IIM method described below builds upon the variant proposed by (Linnick and Fasel, 2005), which, in its turn, is based upon the original work from (Wiegmann and Bube, 2000). It provides the guidelines to calculate the aforementioned corrections when a function is subjected to Jump Discontinuities such as the one indicated at figure 1. The numerical discretization is performed here through the use of a fourth-order accurate Compact Scheme, in conjunction with the Immersed Interface Method, IIM.

The method is depicted here taking as an example the following approximation of a function's first derivative through the use of a Compact Scheme (second and higher order derivatives follow the same guidelines). Assuming that it consists of a set of equations of the form:

$$\begin{aligned} L_{1,i-1}f'(x(i-1)) + L_{1,i}f'(x(i)) + L_{i+1}f'(x(i+1)) = \\ = R_{1,i-1}f(x(i-1)) + R_{1,i}f(x(i)) + R_{1,i+1}f(x(i+1)) \end{aligned} \tag{1}$$

Where the coefficients $L_{n,i+m}$ e $R_{n,i+m}$ represent Padé Schemes coefficients, with n refering to the

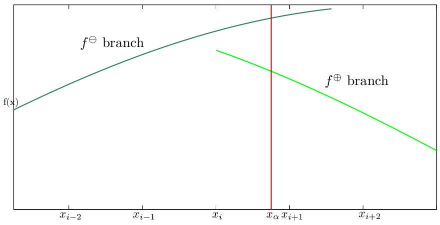


Figure 1. Jump Discontinuity Introduced at the point x_α

n-th order derivative (0 refers to the function value itself), and the letters L and R indicating left and right coefficients, respectively.

These equations are based upon Taylor Series expansions over different grid points. As those expansions presume function continuity, it becomes clear that the presence of Jump Discontinuities requires some kind of correction to be employed in order to maintain their validity.

Two situations might take place according to where the Jumps are placed, relatively to the interface.

Assuming a scheme centered at the point i, the point at which the function needs to be corrected can be either $i+1$ or $i-1$. These situations are respectively represented as $f^n, n = 0, 1, 2, 3, \ldots$ superscripts $^{\ominus}$ and $^{\oplus}$. The negative sign refers to the branch of function downstream the immersed interface, whereas the positive sign refers to the upstream branch.

Jump Correction for the Downstream Branch Based Scheme

Further simplifying the following terms:

$$f^{\ominus,\oplus}(x(i+1)) = f^{\ominus,\oplus}_{i+1} \tag{2}$$

$$f'^{\ominus,\oplus}(x(i+1)) = f'^{\ominus,\oplus}_{i+1} \tag{3}$$

The equation 1 then becomes:

$$L_{1,i-1}f'^{\ominus}_{i-1} + L_{1,i}f'^{\ominus}_{i} + L_{1,i+1}f'^{\ominus}_{i+1} = R_{1,i-1}f^{\ominus}_{i-1} + R_{1,i}f^{\ominus}_{i} + R_{1,i+1}f^{\ominus}_{i+1} \tag{4}$$

In this case, the Jump is introduced in the region $x(i) < x_\alpha < x(i+1)$. This is imposed when boundary and interior conditions are applied to satisfy the presence of a physical object within the domain. With that particular introduction, the scheme, that was entirely based on downstream values, now computes two downstream (at points $i-1$ and i) and one upstream value (at the $i+1$-th grid point). With that in mind, and withouth any correction applied, the equation becomes:

$$L_{1,i-1}f'^{\ominus}_{i-1} + L_{1,i}f'^{\ominus}_{i} + L_{1,i+1}f'^{\oplus}_{i+1} = R_{1,i-1}f^{\ominus}_{i-1} + R_{1,i}f^{\ominus}_{i} + R_{1,i+1}f^{\oplus}_{i+1} \tag{5}$$

At this stage there's the need to introduce the Jump Correction Term, which intends to be a workaround to that problem. Following the definition provided by (Linnick and Fasel, 2005), this term shall, hereinafter, be represented as $J^n_{\alpha,i+m}$. Its meaning shall be elucidated below.

The next challenge is to find expressions for those corrections. Expanding $f(x(i+1))$ through Taylor Series both to the left and to the right side of the interface, and naming these expansions as $f^{\ominus}_{i+1}$ and $f^{\oplus}_{i+1}$, one has:

$$f^{\oplus}_{i+1} = f^{\oplus}_{\alpha} + f^{1\oplus}_{\alpha}dx^+_\alpha + \frac{f^{2\oplus}_{\alpha}(dx^+_\alpha)^2}{2!} + \cdots + \frac{f^{n\oplus}_{\alpha}(dx^+_\alpha)^n}{n!} \tag{6}$$

$$f^{\ominus}_{i+1} = f^{\ominus}_{\alpha} + f^{1\ominus}_{\alpha}dx^+_\alpha + \frac{f^{2\ominus}_{\alpha}(dx^+_\alpha)^2}{2!} + \cdots + \frac{f^{n\ominus}_{\alpha}(dx^+_\alpha)^n}{n!} \tag{7}$$

With:

$$dx^+_\alpha = x(i+1) - x_\alpha \tag{8}$$

$$f^{n\oplus,\ominus}_{\alpha} = \lim_{x \to x_\alpha{}^{+,-}} f^n(x) \tag{9}$$

If one relates $f^{\oplus}_{i+1}$ to $f^{\ominus}_{i+1}$ then it's possible to equal this expression to a term called $J^{0\ominus}_{\alpha,i+m}$, which ultimately results in:

$$J^{0\ominus}_{\alpha,i+1} = f^{\oplus}_{i+1} - f^{\ominus}_{i+1} \tag{10}$$

This becomes, upon manipulation:

$$J^{0\ominus}_{\alpha,i+1} = [f^0_\alpha] + [f^1_\alpha]dx^+_\alpha + [f^2_\alpha]\frac{(dx^+_\alpha)^2}{2!} + \cdots + [f^n_\alpha]\frac{(dx^+_\alpha)^n}{n!} \tag{11}$$

With:

$$[f^n_\alpha] = \lim_{x\to x^+_\alpha} f^n(x) - \lim_{x\to x^-_\alpha} f^n(x)$$

After this procedure, it becomes obvious why the term $J^{0\ominus}_{\alpha,i+1}$ is called Jump Correction Term, and one could proceed similarly to obtain an expression for $J^{1\ominus}_{\alpha,i+m}$:

$$J^{1\ominus}_{\alpha,i+1} = [f^1_\alpha] + [f^2_\alpha]dx^+_\alpha + [f^3_\alpha]\frac{(dx^+_\alpha)^2}{2!} + \cdots + [f^n_\alpha]\frac{(dx^+_\alpha)^{n-1}}{n-1!} \tag{12}$$

These manipulations lead to the corrected equation of the form:

$$\begin{aligned} L_{1,i-1}f'^{\ominus}_{i-1} + L_{1,i}f'^{\ominus}_i + L_{1,i+1}f'^{\oplus}_{i+1} = \\ = R_{1,i-1}f^{\ominus}_{i-1} + R_{1,i}f^{\ominus}_i + R_{1,i+1}f^{\oplus}_{i+1} - (R_{1,i+1}J^{0\ominus}_{\alpha,i+1} + L_{1,i+1}J^{1\ominus}_{\alpha,i+1}) \end{aligned} \tag{13}$$

Jump Correction for the Upstream Branch Based Scheme

The development for this case follows the same guidelines as those from the previous section. Considering an approximation to the first derivative:

$$L_{1,i-1}f'^{\oplus}_{i-1} + L_{1,i}f'^{\oplus}_i + L_{1,i+1}f'^{\oplus}_{i+1} = R_{1,i-1}f^{\oplus}_{i-1} + R_{1,i}f^{\oplus}_i + R_{1,i+1}f^{\oplus}_{i+1} \tag{14}$$

That becomes:

$$L_{1,i-1}f'^{\ominus}_{i-1} + L_{1,i}f'^{\oplus}_i + L_{1,i+1}f'^{\oplus}_{i+1} = R_{1,i-1}f^{\ominus}_{i-1} + R_{1,i}f^{\oplus}_i + R_{1,i+1}f^{\oplus}_{i+1} \tag{15}$$

Using the following Taylos Series expansions around x_α:

$$f^{\oplus}_{i-1} = f^{\oplus}_\alpha - f^{1\oplus}_\alpha dx^-_\alpha + \frac{f^{2\oplus}_\alpha(dx^-_\alpha)^2}{2!} + \cdots + [-1^{(n)}]\frac{f^{n\oplus}_\alpha(dx^-_\alpha)^{(n)}}{(n)!} \tag{16}$$

$$f^{\ominus}_{i-1} = f^{\ominus}_\alpha - f^{1\ominus}_\alpha dx^-_\alpha + \frac{f^{2\ominus}_\alpha(dx^-_\alpha)^2}{2!} + \cdots + [-1^{(n)}]\frac{f^{n\ominus}_\alpha(dx^-_\alpha)^{(n)}}{(n)!} \tag{17}$$

$$f^{1\oplus}_{i-1} = f^{1\oplus}_\alpha - f^{2\oplus}_\alpha dx^-_\alpha + \frac{f^{3\oplus}_\alpha(dx^-_\alpha)^2}{2!} + \cdots + [-1^{(n-1)}]\frac{f^{n\oplus}_\alpha(dx^-_\alpha)^{(n-1)}}{(n-1)!} \tag{18}$$

$$f^{1\ominus}_{i-1} = f^{1\ominus}_\alpha - f^{2\ominus}_\alpha dx^-_\alpha + \frac{f^{3\ominus}_\alpha(dx^-_\alpha)^2}{2!} + \cdots + [-1^{(n-1)}]\frac{f^{n\ominus}_\alpha(dx^-_\alpha)^{(n-1)}}{(n-1)!} \tag{19}$$

Where:

$$dx^-_\alpha = x_\alpha - x(i-1) \tag{20}$$

$$f^{n\oplus,\ominus}_\alpha = \lim_{x\to {x_\alpha}^{+,-}} f^n(x) \tag{21}$$

And the following definitions:

$$J^{0\oplus}_{\alpha,i-1} = f^{\ominus}_{i-1} - f^{\oplus}_{i-1} \tag{22}$$

$$J^{1\oplus}_{\alpha,i-1} = f'^{\ominus}_{i-1} - f'^{\oplus}_{i-1} \tag{23}$$

The Jump Correction Terms are then written as:

$$J^{0\oplus}_{\alpha,i-1} = -[f^0_\alpha] + [f^1_\alpha]dx^-_\alpha - [f^2_\alpha]\frac{(dx^-_\alpha)^2}{2!} + \cdots + [-1^{(n+1)}][f^n_\alpha]\frac{(dx^-_\alpha)^n}{n!} \tag{24}$$

$$J^{1\oplus}_{\alpha,i-1} = -[f^1_\alpha] + [f^2_\alpha]dx^-_\alpha - [f^3_\alpha]\frac{(dx^-_\alpha)^2}{2!} + \cdots + [-1^{(n)}][f^n_\alpha]\frac{(dx^-_\alpha)^{(n-1)}}{(n-1)!} \tag{25}$$

Where:

$$[f^n_\alpha] = \lim_{x\to x^+_\alpha} f^n(x) - \lim_{x\to x^-_\alpha} f^n(x)$$

Finally, the corrected first derivative equation:

$$\begin{aligned} &L_{1,i-1}f'^{\ominus}_{i-1} + L_{1,i}f'^{\oplus}_{i} + L_{1,i+1}f'^{\oplus}_{i+1} = \\ &\quad = R_{1,i-1}f^{\ominus}_{i-1} + R_{1,i}f^{\oplus}_{i} + R_{1,i+1}f^{\oplus}_{i+1} - (R_{1,i-1}J^{0\oplus}_{\alpha,i-1} + L_{1,i-1}J^{1\oplus}_{\alpha,i-1}) \end{aligned} \tag{26}$$

Nevertheless, it's yet to be shown how to obtain approximations to those Jump Terms. The only values know at each iteration of the method are function values themselves. So the only option available is to combine them in such a way that all derivatives at the immersed interface can be estimated accordingly. For the case depicted at this subsection, the desired system can be represented by:

$$\begin{aligned} f(x_\alpha) &= c_{\alpha,1}f(x_\alpha) + c_{i+1,1}f(x_{i+1}) + c_{i+2,1}f(x_{i+2}) + \ldots + c_{i+n,1}f(x_{i+n}) \\ f^1(x_\alpha) &= c_{\alpha,2}f(x_\alpha) + c_{i+1,2}f(x_{i+1}) + c_{i+2,2}f(x_{i+2}) + \ldots + c_{i+n,2}f(x_{i+n}) \\ \vdots &= \quad\vdots \quad+\quad \vdots \quad+\quad \vdots \quad+ \vdots + \quad\vdots \\ f^n(x_\alpha) &= c_{\alpha,n}f(x_\alpha) + c_{i+1,n}f(x_{i+1}) + c_{i+2,n}f(x_{i+2}) + \ldots + c_{i+n,n}f(x_{i+n}) \end{aligned} \tag{27}$$

With $f^n(x_\alpha) = f^n_\alpha$ and $f(x_{i+n}) = f^0_{i+n}$, its matrix representation is:

$$\begin{pmatrix} f^0_\alpha \\ f'_\alpha \\ \vdots \\ f^{n-1}_\alpha \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & \ldots & 0 \\ c_{\alpha,2} & c_{i+1,2} & c_{i+2,2} & \cdots & c_{i+n,2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ c_{\alpha,n} & c_{i+1,n} & c_{i+2,n} & \ldots & c_{i+n,n} \end{pmatrix} \times \begin{pmatrix} f^0_\alpha \\ f^0_{i+1} \\ \vdots \\ f^0_{i+n} \end{pmatrix} \tag{28}$$

The Weierstrass's theorem states that if a function $f(x)$ is continuous over a finite interval $a \leq x \leq b$ then it can be approximated as closely as wanted by a power polynomial, provided this polynomial's order is sufficiently large. So, one may want to represent the function values in successive points close to the immersed interface through Taylor Series expansions (according to (Linnick and Fasel, 2005), the fist neighbor point shall be neglected):

$$\begin{pmatrix} f^0_\alpha \\ f^0_{i+1} \\ \vdots \\ f^0_{i+n} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & \ldots & 0 \\ 1 & (dx^+_\alpha + d_x) & \frac{(dx^+_\alpha + d_x)^2}{2!} & \ldots & \frac{(dx^+_\alpha + d_x)^n}{n!} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & (dx^+_\alpha + nd_x) & \frac{(dx^+_\alpha + nd_x)^2}{2!} & \ldots & \frac{(dx^+_\alpha + nd_x)^n}{n!} \end{pmatrix} \times \begin{pmatrix} f^0_\alpha \\ f'_\alpha \\ \vdots \\ f^{n-1}_\alpha \end{pmatrix} \tag{29}$$

Using:

$$[C] = C_{n,n} = \begin{pmatrix} 1 & 0 & 0 & \ldots & 0 \\ c_{\alpha,2} & c_{i+1,2} & c_{i+2,2} & \cdots & c_{i+n,2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ c_{\alpha,n} & c_{i+1,n} & c_{i+2,n} & \ldots & c_{i+n,n} \end{pmatrix} \tag{30}$$

And:

$$[D] = D_{n,n} = \begin{pmatrix} 1 & 0 & 0 & \ldots & 0 \\ 1 & (dx^+_\alpha + d_x) & \frac{(dx^+_\alpha + d_x)^2}{2!} & \ldots & \frac{(dx^+_\alpha + d_x)^n}{n!} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & (dx^+_\alpha + nd_x) & \frac{(dx^+_\alpha + nd_x)^2}{2!} & \ldots & \frac{(dx^+_\alpha + nd_x)^n}{n!} \end{pmatrix} \tag{31}$$

We can change between systems:

$$(f) = [D](f_\alpha^n) \tag{32}$$

$$[D]^{-1}(f) = [D]^{-1}[D](f_\alpha^n) \tag{33}$$

Upon close inspection, we conclude that [D] is always invertible. Finally:

$$[C] = [D]^{-1} \tag{34}$$

A brief discussion regarding the order maintenance of these expansions is found in (Linnick and Fasel, 2005). As for the fourth-order Compact Scheme used:

$$\begin{pmatrix} 1 & 11 & 0 & \dots & \dots & \dots \\ 1 & 10 & 1 & \dots & \dots & \dots \\ 0 & 1 & 10 & 1 & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \dots & \dots & 1 & 10 & 1 & 0 \\ \dots & \dots & \dots & 1 & 10 & 1 \\ \dots & \dots & \dots & 0 & 11 & 1 \end{pmatrix} \text{ and } \begin{pmatrix} 39a & -81a & 45a & -3a & \dots & \dots \\ 1b & -2b & 1b & 0 & \dots & \dots \\ 0 & 1b & -2b & 1b & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \dots & \dots & 1b & -2b & 1b & 0 \\ \dots & \dots & 0 & 1b & -2b & 1b \\ \dots & \dots & -3a & 45a & -81a & 39a \end{pmatrix}$$

With $a = 1/3dx^2$ and $b = 12/dx^2$.

Results

Initial implementations of this method by the authors in full fledged CFD codes showed that the decay order of the solutions as the grid was progressively refined were not consistent with the order of the scheme. Considering that the method is modular, the necessity to investigate it in detail has arisen. The reader should observe that this paper does not compare distinct methods. This investigation regards only the decay order test of the Finite Difference when coupled to a particular IIM. The testing approach is only applicable in the following and similarly restrictive contexts. Therefore, it cannot provide ways of improving the method as proposed by (Linnick and Fasel, 2005) for general applications. From this perspective, a computational cost test was not taken into account.

From the mathematical formulation of the method, as presented by (Wiegmann and Bube, 2000), to the numerical application as proposed by (Linnick and Fasel, 2005), the main change regards the Jump Correction Terms. The former presents the mathematical correction needed to maintain a required scheme order. This is accomplished if all function jumps are know at each time step. This is true for functions with analytic solutions. If these jumps are not known, then there will be the necessity to provide approximations to them.

What is presented by (Linnick and Fasel, 2005) is that all Jump Terms can be approximated by linear combinations of the function value themselves. Taking all these considerations into account, a simple test has been conducted to precise the impact of the Jump Correction Terms approximations as presented by (Linnick and Fasel, 2005).

This has been achieved by the the application of the method to the calculation of the first two derivatives of the $sin(x)$ function over a domain length of $L = 10$ (the selected profile is shown in figure . As seen on the figure, the curve is subjected to jump discontinuities at two domain points, namely $x_{\alpha_1} = 1.97875$ and $x_{\alpha_2} = 8.02125$. These points have been purposely selected not to coincide with grid points even for the most refined grid.

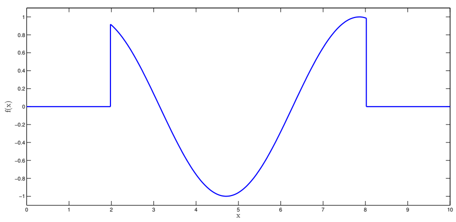


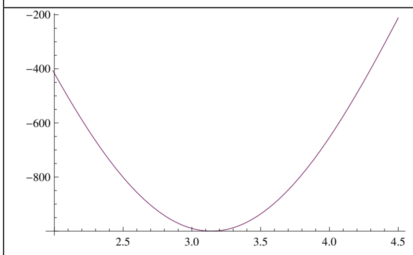
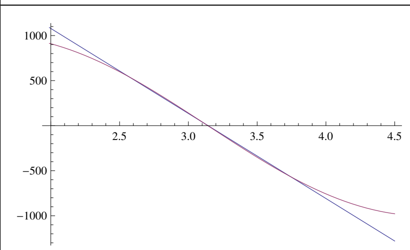
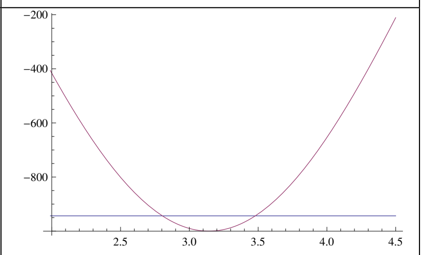
Figure 2. Jump Discontinuity Introduced at the points x_{α_1} and x_{α_2}

One of the key aspects of this approach is that all Jump Terms could be calculated not only by approximation of the function values themselves, but also by the analytical values of any of their derivatives (in fact, the Jump Correction Terms for the $sin(x)$ function could be calculated based upon polynomials as big as needed, once the function has infinity derivatives). The Jump Terms based upon exact derivatives would lead to the maintenance of the original scheme's order, according to the method introduced by (Wiegmann and Bube, 2000). The results presented here concentrate mainly on the analysis of the differences between these two applications. For the sake of this comparison, from now on a Jump Term will be simply named 'Analytical Jump Term' and the numerical one, 'Numerical Jump Term'. The reader should recall that The Numerical Jump Term, as already seen, is an approximation of the derivative by a linear combination of the function values themselves. One important note about this application is that this procedure has been executed for both the forward and the backward Taylor Series expansions, which are respectively representative of the two discontinuity points, x_{α_1} and x_{α_2}.

The first results regard an extensive check up of all these derivative approximations obtained by the matrix inversion process described. These were executed within the code and then compared to the same procedures executed by the commercial software Wolfram Mathematica. Tables 1 and 2 present results for given combinations of grid points and derivative order, at the point x_{α_1}. In these tables, the 'Calculated value' field refers to the approximation calculated by the authors's code based upon the matrix inversion suggested by (Linnick and Fasel, 2005). The plots show two results from Wolfram Mathematica. The purple line indicates the analytical graphical values of the considered derivative, whereas the blue one indicates a fitted polynomial which best represent the function at the discontinuity point. The start of the x-axis is the point x_{α_1}.

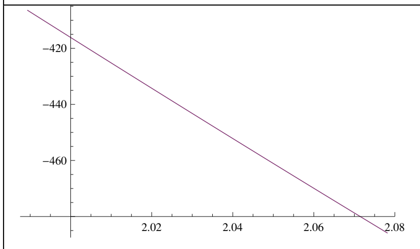
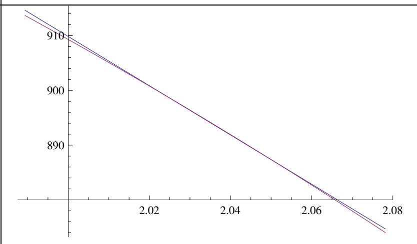
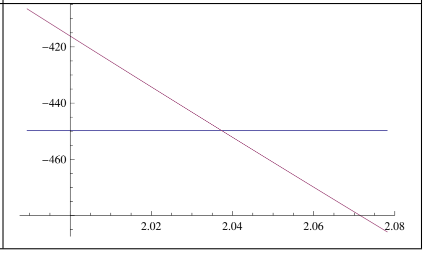
According to the precision order required (discussed in detail in (Linnick and Fasel, 2005)), each higher order derivative approximation can be calculated with a progressively decreasing order. Besides, this precision requirement implies the approximation of a number of derivatives for given scheme and derivative orders. In this case, 4th-order schemes are employed to the calculation of the first two derivatives of a function. Consequently, the Jump Correction Terms must include up to the fifth derivative approximation. The whole set of results has not been shown for the sake of available space. Nevertheless, it's important to note that this has been done for each derivative of each of the tested grids, at both points. The reader should also recall that this procedure should be done from the first to the fifth derivative, which are the required ones for the calculation of up to the second derivative with 4-th order.

Table 1. First, fourth and fifth derivatives at $x_{\alpha_1} = 1.989375$ - 21 points grid

21 points grid		
1st Derivative	4th Derivative	5th Derivative
Calculated value	Calculated value	Calculated value
$-406.8457641830\ 10^{-3}$	1.0887651426	$-943.4528052418\ 10^{-3}$
Analytical value	Analytical value	Analytical value
$-406.4622438469\ 10^{-3}$	$913.6675786778\ 10^{-3}$	$-406.4622438469\ 10^{-3}$
		

The information from these tables show that the fitted polynomial obtained by the Wolfram Mathematica software agrees with the results obtained by the matrix inversion process employed by the authors' code at the jump point $x_{\alpha_1} = 1.989375$, and for all presented derivatives. As the grid is refined the numerical value approaches the analytical value. Of special attention is the behaviour exhibited for the 4th and 5th derivative, depicted by the second and third plots from the left. It shows that the polynomials fitted for all 4th derivatives are of 1st order, and the line for the the 5th derivatives are constant value lines. That is also an indication that each higher derivative is indeed calculated by

Table 2. First, fourth and fifth derivatives at $x_{\alpha_1} = 1.989375$ - 641 points grid

641 points grid		
1st Derivative	4th Derivative	5th Derivative
Calculated value	Calculated value	Calculated value
$-406.4622443593\ 10^{-3}$	$914.6222546697\ 10^{-3}$	$-449.7330188751\ 10^{-3}$
Analytical value	Analytical value	Analytical value
$-406.4622438469\ 10^{-3}$	$913.6675786778\ 10^{-3}$	$-406.4622438469\ 10^{-3}$
		

this process with a progressively lower order. Finally, it's also being demonstrated that this process implies a lot different than expected values for the derivatives at that point.

After this discussion, figures 4 and 3 show the results from the grid refinement test. This has begun with a 21 points grid and then has gotten progressively refined to the double of the previous points number, up to a 641 points grid. The errors' standards follow the definition found on (da Silva, 2008). That uses the error norms according to equations 35 and 36. The two plots show the errors L_∞ and L_1 for different solutions, with the index n representing the number of grid points.

$$L_1 = \left[\sum_{n=1}^{j} |f_{ex}(n) - f_{calc}(n)| \right] / j \tag{35}$$

$$L_\infty = max|f_{ex}(n) - f_{calc}(n)| \quad ; \quad n = 1, 2, \ldots, j \tag{36}$$

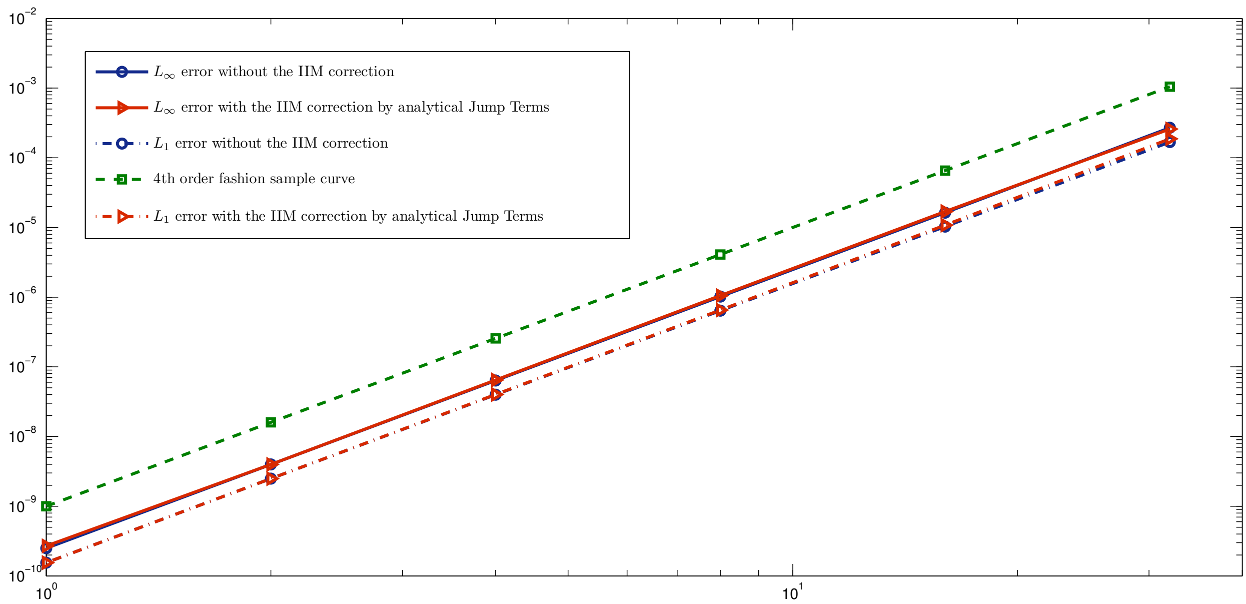


Figure 3. Sin Function - Second Derivative Errors - Analytical Jump Terms (Grid Points Number *vs.*Error)

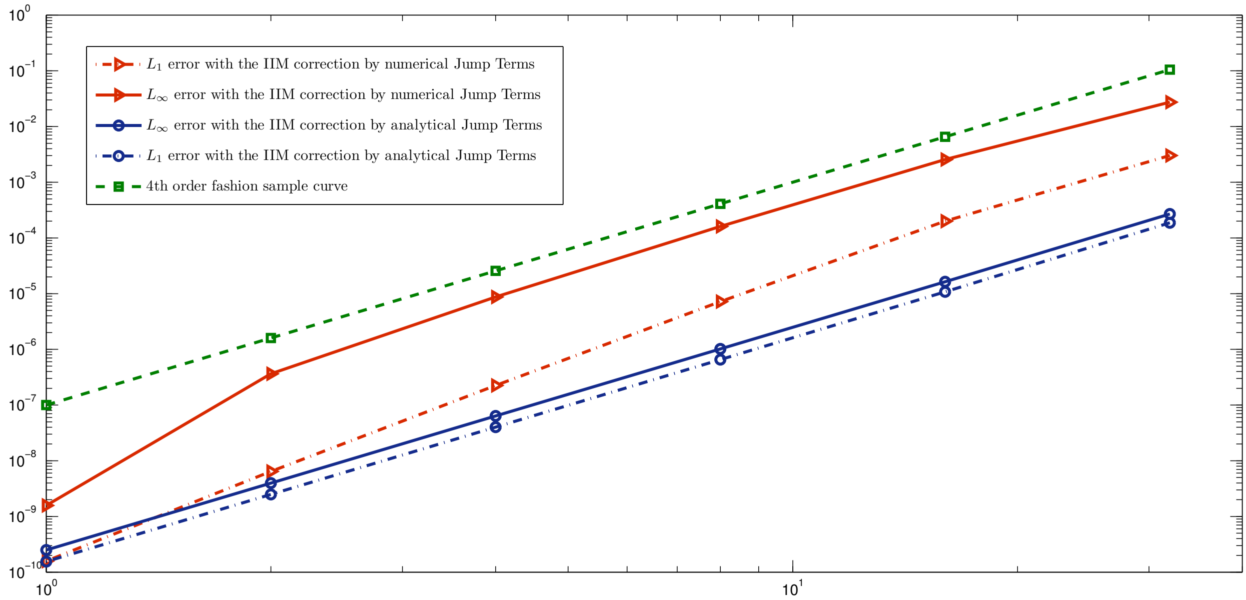


Figure 4. Sin Function - Second Derivative Errors - Numerical Jump Terms (Grid Points Number *vs.* Error)

First, the plot from figure 3 shows the comparison between the results for the IIM with Analytical Jump Terms and the solution if no Jump Discontinuities were imposed (which implies that the function would be continuous for any x from 0 to 10). Comparing the error norms from both solutions to the green 4-th order sample curve it's shown that the decay order of the scheme is indeed maintained by the method. It is important to notice that the IIM based upon Analytical Jump Terms not only does that, decaying to the same order than the original scheme, but also agrees with it when it comes to absolute values. On the other hand, a direct comparison between the Analytical and the Numerical IIM (showed in figure 4), shows that the error norms of the IIM based upon the Numerical Jump Terms are not only considerably higher and inconsistent in terms of decay order, but they also start to match the Analytical IIM values only for the most refined grid, with 641 points. This behaviour rises the question of how effective is the approximation of those derivatives at points x_{α_n} by the linear combination of the function values.

Closer inspection of the equations applied to those approximations shows that the method (as proposed by (Linnick and Fasel, 2005)) requires the same amount of neighbour points to be computed for each more refined grid, but one can immediately point out that their positions along the coordinate axis change as the grid becomes more refined. Compared to the domain length, this direct translates into a collection of points progressively more collapsed as we change from one grid to the next. This means that the portion of the curve that is actually being computed by this process, and therefore its shape, also changes from one grid to another.

All these discussion and results show that the method proposed by (Linnick and Fasel, 2005) is not capable of always maintaining the scheme's order in the same manner as the original mathematical formulation by (Wiegmann and Bube, 2000) does. This can have an ultimate effect on the grid refinement, requiring a great amount of points to reach a desirable error magnitude and diminishing the advantage of employing a High Order Scheme.

Acknowledgements

The authors would like to thank CNPQ (Conselho Nacional de Desenvolvimento Científico e Tecnológico), which has fomented this paper's related research in the form of a scholarship.

References

Wiegmann A., Bube K. P. (2000), The Explicit-Jump Immersed Interface Method: Finite Difference Methods for PDE's with Piecewise Smooth Solutions, *SIAM Journal on Numerical Analysis*, v.37 n.3, pp.827-862.

Linnick M. K. and Fasel H. F. (2005), A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains, *Journal of Computational Physics*, v.204, issue 1, pp.157-192.

Da Silva H. G. (2008), Regime Não-linear de Trens de Ondas Modulados na Direção Transversal em um Escoamento de Poiseuille Plano, Tese de Doutorado, *Escola de Engenharia de São Carlos*.

Kopal Z. (1961), Numerical Analysis, 2nd Edition, *Chapman & Hall*, London.