Accuracy Improvement Strategies for the Immersed Layers Method with Finite Volume Discretisation

Miguel Alexandre Santos Sousa
miguel.a.s.sousa@tecnico.ulisboa.pt

Instituto Superior Técnico, Universidade de Lisboa, Portugal

December 2022

Abstract

The Immersed Layers Method developed by Jeff D. Eldredge constitutes a first-order immersed boundary method that allows the imposition of different boundary condition values on each side of the interface, enabling the solving of problems for which previous methods originate incorrect results. In this work, one derives several one-dimensional and two-dimensional accuracy improvement strategies to the Immersed Layers Method, considering non-smooth fields across the interface, when solving the Poisson equation in a finite volume framework. The improvement procedure consists of replacing the original second equation, relying on interpolation with discrete delta functions, by an alternative one. In the one-dimensional case, one derives strategies that perform a polynomial interpolation of the subfields to the interface and strategies expressing the normal derivative jump, with both relying on direct methods and leading to second-order accuracy in the superfluous region. In the two-dimensional case, one derives strategies relying on direct and least squares interpolation methods, to express the normal derivative jump across the interface. For some two-dimensional implementations, superfluous decays very close to second-order were obtained. In a second stage, the numerical field values in the transition region are replaced by estimates computed from the superfluous field and, in some versions, using the information stored in the solid points as well. This final step allows the recovering of interface sharpness and the propagation of the superfluous decays to the entire domain.

Keywords: Immersed Boundary Method, Immersed Layers Method, Discrete Delta Function, Finite Volume Method, Computational Fluid Dynamics

1. Introduction

In nature, most processes are modelled by partial differential equations. However, for complex problems, analytical solutions are usually not possible to express or difficult to obtain. Thus, numerical methods constitute a powerful tool to compute approximate solutions. In Computational Fluid Dynamics (CFD), most problems rely on flows around a solid body, with an interface separating both media. In conventional approaches, the mesh conforms to the body, facilitating the imposition of interface constraints, like the no-slip condition. However, if the body is moving or is allowed to deform, a remeshing process is necessary at each time step. To overcome this costly inconvenient, a class of Immersed Boundary (IB) methods arose, firstly introduced by Peskin [7], in which the interface is allowed to be anywhere relative to the grid. Since then, several IB methods appeared. Mittal and Iccarino [6] divide IB methods into two main families: continuous forcing and discrete forcing methods. As explained in [6], continuous forcing methods discretise a general equation that is valid for all the domain, being easier to implement. However, they lead to lower accuracy for non-smooth fields across the interface. According to Cola *et al.* [1], their reversion to first-order in two dimensions is "a problem that has been known in literature for decades". This problem arises due to the substitution of the Dirac delta function by a discrete analogue, responsible for smearing the interface. On the other hand, discrete forcing methods modify the system of equations in cells near the interface, presenting an accuracy depending on the methods that are locally employed and being able to represent sharp interfaces. In this work, one derives one- and two- dimensional accuracy improvement strategies to the Immersed Layers (IL) Method derived by Eldredge [3]. This method allows the imposition of different constraints at each side of the interface, being able to deal with problems that previous methods not distinguishing both sides of the interface were unable to correctly solve. However, since it relies on Dirac delta functions to perform regularisation and interpolation operations, it is only first-order accurate for non-smooth fields. In this work, one seeks going beyond first-order, pursuing second-order accuracy in one- and two-dimensional problems.

2. Immersed Layers Method

Before deriving the accuracy improvements strategies, a background concerning the original finite differences formulation of the IL method is provided, [3]. Moreover, a finite volume version is also presented as a contribution. The problem to be addressed consists of a computational domain $\Omega \subset \mathbb{R}^\lambda$, with $\lambda = \{1; 2; 3\}$ being the dimension of the multidimensional space, in which two distinct regions exist: an interior solid subdomain Ω^- and an exterior fluid subdomain Ω^+, separated by an interface Γ characterised by an exterior unit normal vector $\mathfrak{n}$. The unknown subfields at each subdomain are respectively named $\phi^-(\mathbf{x})$ and $\phi^+(\mathbf{x})$, with $\mathbf{x} \in \Omega$.

2.1. Indicator Function

An important concept underlying the IL method is the indicator function, $\chi(\mathbf{x}) \in \mathbb{R}$, defined for all the domain, assuming negative values in the solid, positive values in the fluid and being null at the interface. Here, the indicator is defined as the properly signed minimum Euclidean distance to the interface. As such, its gradient is normal to the interface, points out of the solid and has a unitary magnitude. So, when taken at the interface, it corresponds to the exterior unit normal, $\nabla\chi|_\Gamma = \mathfrak{n}$.

2.2. Generalised Functions

The IL method relies on the use of two generalised functions: the Heaviside function, $H(\chi)$, and the Dirac delta function, $\delta(\chi)$. The definition of the Heaviside function is provided in equation 1.

$$H(\chi) = \begin{cases} 1 & , \ \chi > 0 \\ 1/2 & , \ \chi = 0 \\ 0 & , \ \chi < 0 \end{cases} \tag{1}$$

The Dirac delta function is defined in equation 2.

$$\delta(\chi) = \begin{cases} +\infty & , \ \chi = 0 \\ 0 & , \ \chi \neq 0 \end{cases} \tag{2}$$

By definition, $\frac{\partial H}{\partial \chi}(\chi) = \delta(\chi)$. So, the gradient of the Heaviside function is $\nabla H(\pm\chi) = \pm\delta(\chi)\mathfrak{n}$.

The Dirac delta function is generalised to $\mathbb{R}^\lambda$ by taking the product of the scalar function applied to each component, $\delta(\mathbf{x}) = \prod_{i=1}^{\lambda} \delta(x_i)$. This function allows to write volume integrals over Ω into surface integrals over Γ, and vice-versa, [3]. Consider that the interface is parameterised by coordinate(s) $\boldsymbol{\zeta}$, such that $\mathbf{x} = \mathbf{X}(\boldsymbol{\zeta})$. One important result concerns the immersion of a function $F(\boldsymbol{\zeta})$ exclusively defined at the interface into the computational domain, presented in equation 3.

$$F(\boldsymbol{\zeta})\delta\big(\chi(\mathbf{x})\big) = \int_\Gamma F(\boldsymbol{\zeta})\delta\big(\mathbf{x} - \mathbf{X}(\boldsymbol{\zeta})\big)dS(\boldsymbol{\zeta}) \tag{3}$$

On the other hand, considering a function $g(\mathbf{x})$ defined in $\mathbb{R}^\lambda$, its restriction to the interface, with notation $g(\mathbf{x})\delta^T\big(\chi(\mathbf{x})\big) \equiv g\big(\mathbf{X}(\boldsymbol{\zeta})\big)$, is given by equation 4. These derivations are detailed in [3].

$$g(\mathbf{x})\delta^T\big(\chi(\mathbf{x})\big) = \int_{\mathbb{R}^\lambda} g(\mathbf{x})\delta\big(\mathbf{X}(\boldsymbol{\zeta}) - \mathbf{x}\big)d\mathbf{x} \tag{4}$$

2.3. Concept of Masked Variable

The main idea underlying the IL method consists of working with a masked variable, $\bar{\phi}$, that assumes one subfield or the other, depending on the position in which it is evaluated. So, applying the Heaviside function to the indicator, the definition of masked variable $\bar{\phi}$ is obtained according to equation 5.

$$\bar{\phi} = \phi^+ H(\chi) + \phi^- H(-\chi) \tag{5}$$

2.4. Poisson Equation of a Masked Variable

Given the definition of masked variable and the relations derived thus far, the gradient of a masked variable is readily obtained. Then, applying the divergence operator, one reaches the Laplacian of a masked variable, given by equation 6. This relation is the Poisson equation of a masked variable.

$$\begin{aligned} \nabla^2\bar{\phi} =& \overline{\nabla^2\phi} + (\nabla\phi^+ - \nabla\phi^-)\cdot\delta(\chi)\mathfrak{n} + \\ &+ \nabla\cdot\big((\phi^+ - \phi^-)\delta(\chi)\mathfrak{n}\big) \end{aligned} \tag{6}$$

2.5. Implementation of the Poisson Equation

The interface is discretised into p equally-spaced solid points, for which the IL method assumes that both subfield values are known, ϕ^+_{sp} and ϕ^-_{sp}. However, subfield gradients at the solid points are unknown. So, the normal derivative jump across the interface, $\sigma \equiv (\nabla\phi^+_{sp} - \nabla\phi^-_{sp})\cdot\mathfrak{n}$ arises as an unknown at each solid point. To close the system of equations, Eldredge [3] sets equation 7, expressing the interface value by restricting the masked field.

$$\bar{\phi}\delta^T(\chi) = \frac{1}{2}(\phi^+_{sp} + \phi^-_{sp}) \tag{7}$$

The accuracy improvement strategies presented in this work concern the derivation of alternative equations to close the system, instead of this one.

2.6. Domain Discretisation

The computational domain is assumed to present a length $L \in \mathbb{R}$ along each dimension, such that $\Omega = [-L/2; L/2]^\lambda$. The domain discretisation is done according to one of two possible frameworks: Finite Differences (FD) or Finite Volume (FV). In a FD discretisation, cell centroids coincide with the domain boundaries. In FV, the domain boundaries coincide with cell faces. The domain is assumed to be subdivided into n equal cells along each dimension, making a total of $N = n^\lambda$ cells. So, the grid size is $h = L/(n-1)$ in FD and $h = L/n$ in FV.

2.7. Treatment of the System of Equations

In FD, equations 6 and 7 are put together, as in equation 8, being the starting point to discretise.

$$\begin{cases} \nabla^2\bar{\phi} - \sigma\delta(\chi) = \overline{\nabla^2\phi} + \nabla\cdot\big((\phi^+_{sp} - \phi^-_{sp})\delta(\chi)\mathfrak{n}\big) \\ \bar{\phi}\,\delta^T(\chi) = \frac{1}{2}(\phi^+_{sp} + \phi^-_{sp}) \end{cases} \tag{8}$$

On the other hand, in FV, the system of equations is integrated over each cell. Consider that each cell has a volume V, enclosed by a surface $S = \partial V$ with unit exterior normal $\mathbf{n}$. Integrating equation 6 and using the divergence theorem, equation 9 arises.

$$\oiint_{\partial V} \nabla\bar{\phi}\cdot\mathbf{n}\,dS - \iiint_V \sigma\delta(\chi)\,dV = \iiint_V \overline{\nabla^2\phi}\,dV + \\ + \iiint_V \nabla\cdot\big((\phi^+_{sp} - \phi^-_{sp})\delta(\chi)\mathfrak{n}\big)\,dV \tag{9}$$

Notice that the volume integral of the divergence in the right-hand side was not converted into a surface integral, since the vector field over which the divergence applies is not continuously differentiable. Integrating equation 7, equation 10 arises.

$$\iiint_V \bar{\phi}\,\delta^T(\chi)\,dV = \iiint_V \frac{1}{2}(\phi^+_{sp} + \phi^-_{sp})\,dV \tag{10}$$

Equations 9 and 10 constitute the starting point of a FV approach to the IL Poisson equation. Given the targeted accuracy, the integrals are approximated with second order. Equation 9 yields equation 11. The sum over f stands for a sum over the cell faces and subscript C stands for cell centroid.

$$\sum_f \big[(\nabla\bar{\phi})_f\cdot\mathbf{n_f}\big] - \sigma\delta(\chi_C)h^\lambda = \big(\overline{\nabla^2\phi}\big)_C h^\lambda + \\ + \nabla\cdot\big[(\phi^+_{sp} - \phi^-_{sp})\delta(\chi_C)\mathfrak{n}\big]h^\lambda \tag{11}$$

Similarly, equation 10 gives rise to equation 12.

$$\bar{\phi}\,\delta^T(\chi_C)h^\lambda = \frac{1}{2}(\phi^+_{sp} + \phi^-_{sp})h^\lambda \tag{12}$$

Equations 11 and 12 form the second-order FV system of equations used in this work to implement the masked Poisson equation in a FV framework.

2.8. Operators Discretisation

To perform the numerical implementation, the operators appearing in the systems of equations must be discretised. The Dirac delta function is substituted by a discrete delta function, $\delta_h(\mathbf{x})$, mimicking its properties. One important parameter of any discrete delta function is its support width w, indicating that it is not identically null in a region of characteristic length wh around its centre (interval, circle or sphere for $\lambda = 1, 2, 3$, respectively).

In this work, four different discrete delta functions are studied: δ^*_3, δ_{2002}, δ_{2103}, δ_{4206}. The first function, derived in [9], is the one used by Eldredge [3]. The other three functions are given in [5].

Regarding the discretisation of immersion and restriction operations, the integrals are approximated with second-order accuracy ($\overset{O2}{\sim}$). The discretisation of immersion operation appears in equation 13.

$$F(\boldsymbol{\zeta})\delta\big(\chi(\mathbf{x})\big) \overset{O2}{\sim} \sum_{k=1}^{p} F(\boldsymbol{\zeta}_k)\delta\big(\mathbf{x} - \mathbf{X}(\boldsymbol{\zeta}_k)\big)\delta s_k \tag{13}$$

Regarding the restriction, equation 14 arises.

$$g(\mathbf{x})\delta^T\big(\chi(\mathbf{x})\big) \overset{O2}{\sim} \sum_{i=1}^{N} g(\mathbf{x}_i)\delta\big(\mathbf{x}_i - \mathbf{X}(\boldsymbol{\zeta})\big)h^\lambda \tag{14}$$

Finally, all the derivatives arising from the Laplacian, gradient and divergence operators are discretised with second-order schemes.

3. 1D Improvement Strategies

As an alternative to equation 7 set by Eldredge [3] to close the system, two strategies are followed in the one-dimensional (1D) problem: expressing the normal derivative jump, or implementing an equation with interpolation to the interface without using discrete delta functions. Finally, three smearing correction procedures are derived, to recover interface sharpness in the end of the simulation. The 1D problem to be solved is illustrated in figure 1. The parameters $L = 2$ and $x_{sp_2} = 0.5 = -x_{sp_1}$ were set.

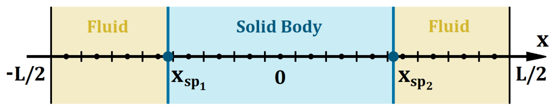

Figure 1: Illustration of the 1D problem.

3.1. Equation for the Normal Derivative Jump

In 1D, the normal derivative jump, σ, is given by equation 15, with superscripts l/r standing for left and right, holding for solid points x_{sp_1} and x_{sp_2}.

$$\sigma = (\nabla\phi^+_{sp} - \nabla\phi^-_{sp})\cdot\mathfrak{n} = \left.\frac{\partial\bar{\phi}}{\partial x}\right|^r_{sp} - \left.\frac{\partial\bar{\phi}}{\partial x}\right|^l_{sp} \tag{15}$$

The first version of this strategy considers a first-order unilateral scheme to compute the derivatives. As already referred, the use of discrete delta functions with non-smooth fields leads to an interface smearing phenomenon, that occurs in the region over which the discrete delta function directly acts, given by $|x - x_{sp}| < wh$. The union of all the regions affected by smearing is defined as the transition region, $\mathbb{T}$. The superfluous region is here defined as its complementary set, that is $\mathbb{S} = \Omega \setminus \mathbb{T}$. Since the transition points are not representative of the desired slope tendencies coming towards the solid points (due to smearing), one should only choose superfluous points to be used in the expressions. Consider $\bar{\phi}_b$ as the grid point immediately before the interface that is out of the transition region. Similarly, consider $\bar{\phi}_a$ as the grid point immediately after the interface that is out of the transition region. Using them together with the solid point analytical subfield values, the expression using first-order unilateral schemes is provided in equation 16.

$$\sigma = \frac{\bar{\phi}_a - \phi^r_{sp}}{x_a - x_{sp}} - \frac{\phi^l_{sp} - \bar{\phi}_b}{x_{sp} - x_b} \tag{16}$$

Hiding the terms multiplying field values in coefficients $c \in \mathbb{R}$ and noting that the subfield values at the solid points are known, equation 17 arises.

$$c_a\bar{\phi}_a + c_b\bar{\phi}_b + (-1)\sigma = -(c^r_{sp}\phi^r_{sp} + c^l_{sp}\phi^l_{sp}) \tag{17}$$

Another version for this strategy was derived using two superfluous points from each subdomain: $\bar{\phi}_{bb}$, $\bar{\phi}_b$, $\bar{\phi}_a$, $\bar{\phi}_{aa}$. Note that $\bar{\phi}_{bb}$ stands for the grid point immediately before $\bar{\phi}_b$, whereas $\bar{\phi}_{aa}$ stands for the grid point immediately after $\bar{\phi}_a$. Performing polynomial expansions for each side of the interface and computing the polynomial coefficients of those expansions by substituting $\bar{\phi}_{bb}$, $\bar{\phi}_b$, $\bar{\phi}^l_{sp}$ in the left expansion and $\bar{\phi}_a$, $\bar{\phi}_{aa}$, $\bar{\phi}^r_{sp}$ in the right expansion, one is able to reach an equation for σ looking like equation 18, wherein terms multiplying field values were again hidden in general $c \in \mathbb{R}$ coefficients.

$$\begin{aligned} c_a\bar{\phi}_a + c_b\bar{\phi}_b + c_{aa}\bar{\phi}_{aa} + c_{bb}\bar{\phi}_{bb} + \\ + (-1)\sigma = -(c^r_{sp}\phi^r_{sp} + c^l_{sp}\phi^l_{sp}) \end{aligned} \tag{18}$$

Summarising, equation 17 presents the first version to compute the normal derivative jump using one superfluous points from each side of the interface, whereas equation 18 presents the second version using two superfluous points from each side.

3.2. Equation Imposing the Interface Value

In this strategy, an equation imposing the field value at the interface is considered, relying on polynomial interpolations. The subfield value at each interface side is computed via a linear combination of superfluous points from the same subdomain. So, one is pursuing equations like 19, where a general set of coefficients $\gamma \in \mathbb{R}$ represents the linear combinations performed for each interface side.

$$\begin{aligned} \frac{1}{2}\Big[\big(\gamma_b\bar{\phi}_b + \gamma_{bb}\bar{\phi}_{bb} + \cdots\big) + \\ + \big(\gamma_a\bar{\phi}_a + \gamma_{aa}\bar{\phi}_{aa} + \cdots\big)\Big] = \frac{1}{2}(\phi^l_{sp} + \phi^r_{sp}) \end{aligned} \tag{19}$$

The computation of the linear combination coefficients $\gamma \in \mathbb{R}$ is performed via polynomial expansions, like the one already described. So, their derivation is omitted. The usual c notation is recovered taking $c_a = \gamma_a/2$, $c_b = \gamma_b/2$, and so on.

In a first version, two superfluous points are used from each side of the interface, as in equation 20.

$$c_b\bar{\phi}_b + c_{bb}\bar{\phi}_{bb} + c_a\bar{\phi}_a + c_{aa}\bar{\phi}_{aa} = \frac{1}{2}(\phi^l_{sp} + \phi^r_{sp}) \tag{20}$$

In a second version, one considers three superfluous points from each side, as in equation 21.

$$\begin{aligned} c_b\bar{\phi}_b + c_{bb}\bar{\phi}_{bb} + c_{bbb}\bar{\phi}_{bbb} + c_a\bar{\phi}_a + \\ + c_{aa}\bar{\phi}_{aa} + c_{aaa}\bar{\phi}_{aaa} = \frac{1}{2}(\phi^l_{sp} + \phi^r_{sp}) \end{aligned} \tag{21}$$

3.3. Smearing Correction

In the end of the numerical simulation, a corrective procedure is implemented, in order to recover interface sharpness. Moreover, this procedures will propagate superfluous error decays to the transition region and, consequently, to the entire domain. To perform the correction, transition values are discarded and new ones are computed. The first corrective procedure (C1S) uses the solid point and the closest superfluous point. To correct the $q-th$ transition value, placed at x_q, equation 22 is applied.

$$\bar{\phi}^{new}_q = \begin{cases} \left(\frac{x_{sp}-x_q}{x_{sp}-x_b}\right)\bar{\phi}_b + \left(\frac{x_q-x_b}{x_{sp}-x_b}\right)\phi^l_{sp} & \Leftarrow x_q < x_{sp} \\ \left(\frac{x_q-x_{sp}}{x_a-x_{sp}}\right)\bar{\phi}_a + \left(\frac{x_a-x_q}{x_a-x_{sp}}\right)\phi^r_{sp} & \Leftarrow x_q > x_{sp} \\ \frac{1}{2}(\phi^l_{sp} + \phi^r_{sp}) & \Leftarrow x_q = x_{sp} \end{cases} \tag{22}$$

The second corrective procedure (C2) uses two superfluous points, corresponding to a simple linear extrapolation. Considering that one is correcting the q-th point inside a transition region covering a total of K points, expression 23 is applied.

$$\bar{\phi}^{new}_q = \begin{cases} \bar{\phi}^{new,l}_q = (q+1)\bar{\phi}_b - q\bar{\phi}_{bb} & \Leftarrow x_q < x_{sp} \\ \bar{\phi}^{new,r}_q = \big[(K-q+1)+1\big]\bar{\phi}_a - \\ \qquad -(K-q+1)\bar{\phi}_{aa} & \Leftarrow x_q > x_{sp} \\ \frac{1}{2}(\bar{\phi}^{new,l}_q + \bar{\phi}^{new,r}_q) & \Leftarrow x_q = x_{sp} \end{cases} \tag{23}$$

In the third corrective procedure (C2S), two superfluous points from each side are used, together with the solid point. As such, a corrective polynomial is written for each side, as in equation 24. The computation of the coefficients relies on $\bar{\phi}_a$, $\bar{\phi}_{aa}$, $\bar{\phi}^r_{sp}$ for the right side and on $\bar{\phi}_{bb}$, $\bar{\phi}_b$, $\bar{\phi}^l_{sp}$ for the left.

$$\begin{cases} \phi^l(x) = C^l_0 + C^l_1 x + C^l_2 x^2 \\ \phi^r(x) = C^r_0 + C^r_1 x + C^r_2 x^2 \end{cases} \tag{24}$$

The corrected value is computed by substituting its coordinates in the polynomial, as in equation 25.

$$\bar{\phi}^{new}_q = \begin{cases} \phi^l(x_q) & \Leftarrow x_q < x_{sp} \\ \phi^r(x_q) & \Leftarrow x_q > x_{sp} \\ \frac{1}{2}(\phi^l(x_q) + \phi^r(x_q)) & \Leftarrow x_q = x_{sp} \end{cases} \tag{25}$$

4. 1D Results

Before proceeding with 2D generalisations, the 1D improvement strategies were tested, in order to assess if the formulations lead indeed to second-order accuracy and which one is the best. The function used to compare the strategies is $\phi^-(x) = e^x$ and $\phi^+(x) = 0$, presenting discontinuous value and first derivative across the interface. The 1D corrective procedures are also compared. As long as no correction is used, the transition error does not decay. So, this region is omitted from most tables.

4.1. Original Formulation

The 1D global and superfluous decays for the original formulation in FD are given in table 1 and the equivalent ones for FV are provided in table 2. As the field is non-smooth across the interface, the smearing phenomenon causes the global mean error to be first-order accurate, the global L^2 norm to decay with half order and the global maximum error to not decay (N.D.). All superfluous decays are first-order accurate. Moreover, there is no significant difference between versions IL-FD and IL-FV.

Table 1: Global and superfluous decays, 1D IL-FD.

δ_h	$\bar{\varepsilon}$	$\Vert\varepsilon\Vert_2$	ε_{max}	$\bar{\varepsilon}^{Sup}$	$\Vert\varepsilon\Vert_2^{Sup}$	ε_{max}^{Sup}
δ_3^*	0.9933	0.4988	N.D.	0.9900	0.9909	0.9925
δ_{2002}	0.9963	0.4984	N.D.	0.9945	0.9953	0.9961
δ_{2103}	0.9941	0.4990	N.D.	0.9934	0.9941	0.9930
δ_{4206}	0.9919	0.4988	N.D.	0.9808	0.9846	0.9861

Table 2: Global and superfluous decays, 1D IL-FV.

δ_h	$\bar{\varepsilon}$	$\Vert\varepsilon\Vert_2$	ε_{max}	$\bar{\varepsilon}^{Sup}$	$\Vert\varepsilon\Vert_2^{Sup}$	ε_{max}^{Sup}
δ_3^*	0.9974	0.4998	N.D.	0.9952	0.9956	0.9957
δ_{2002}	0.9990	0.4997	N.D.	0.9976	0.9979	0.9976
δ_{2103}	0.9978	0.4999	N.D.	0.9968	0.9971	0.9958
δ_{4206}	0.9967	0.4999	N.D.	0.9919	0.9926	0.9929

4.2. Equation for the Normal Derivative Jump

The global and superfluous decays achieved using the normal derivative jump equation with one superfluous point, σ-eq (1), are shown in table 3. The corresponding results for two superfluous points, σ-eq (2), appear in table 4. Given that the function is non-smooth across the interface, the global mean error decays with first order, the global L^2 norm decays with half order and the global maximum error does not decay, as well as any transition quantity. However, in the superfluous region, σ equations lead to a second-order decay of all error quantities.

Table 3: Global and superfluous decays for σ-eq (1).

δ_h	$\bar{\varepsilon}$	$\Vert\varepsilon\Vert_2$	ε_{max}	$\bar{\varepsilon}^{Sup}$	$\Vert\varepsilon\Vert_2^{Sup}$	ε_{max}^{Sup}
δ_3^*	1.0003	0.5003	N.D.	1.9964	1.9964	1.9892
δ_{2002}	1.0012	0.5000	N.D.	1.9979	1.9981	1.9937
δ_{2103}	1.0031	0.5004	N.D.	1.9972	1.9973	1.9929
δ_{4206}	1.0059	0.5002	N.D.	1.9951	1.9952	1.9945

Table 4: Global and superfluous decays for σ-eq (2).

δ_h	$\bar{\varepsilon}$	$\Vert\varepsilon\Vert_2$	ε_{max}	$\bar{\varepsilon}^{Sup}$	$\Vert\varepsilon\Vert_2^{Sup}$	ε_{max}^{Sup}
δ_3^*	1.0003	0.5003	N.D.	1.9938	1.9937	2.0092
δ_{2002}	1.0001	0.5000	N.D.	1.9967	1.9969	2.0053
δ_{2103}	1.0004	0.5004	N.D.	1.9948	1.9949	2.0106
δ_{4206}	1.0000	0.5002	N.D.	2.2334	2.1124	2.0915

Notice that, although version σ-eq (1) uses unilateral first-order derivatives, second order was achieved in the superfluous region. As stated by Leveque and Li [4], a first-order error in a set of lower dimension is not able to revert a second-order decay occurring around it. The interface is indeed a set of lower dimension and the derivative expressions only invoke grid points that are superfluous.

4.3. Equation Imposing the Interface Value

The global and superfluous decays achieved using the equation imposing the field value at the interface with two superfluous point, ϕ-eq (2), are presented in table 5. The corresponding results when using three superfluous points, ϕ-eq (3), appear in table 6. Qualitatively, the results using ϕ equations are similar to the ones achieved with σ equations.

Table 5: Global and superfluous decays for ϕ-eq (2).

δ_h	$\bar{\varepsilon}$	$\Vert\varepsilon\Vert_2$	ε_{max}	$\bar{\varepsilon}^{Sup}$	$\Vert\varepsilon\Vert_2^{Sup}$	ε_{max}^{Sup}
δ_3^*	1.0042	0.5002	N.D.	1.9961	1.9964	1.9944
δ_{2002}	1.0019	0.5000	N.D.	1.9981	1.9979	1.9967
δ_{2103}	1.0041	0.5003	N.D.	1.9968	1.9964	1.9944
δ_{4206}	1.0071	0.5001	N.D.	1.9944	1.9940	1.9919

Table 6: Global and superfluous decays for ϕ-eq (3).

δ_h	$\bar{\varepsilon}$	$\Vert\varepsilon\Vert_2$	ε_{max}	$\bar{\varepsilon}^{Sup}$	$\Vert\varepsilon\Vert_2^{Sup}$	ε_{max}^{Sup}
δ_3^*	1.0003	0.5003	N.D.	2.0052	2.0098	2.0444
δ_{2002}	1.0001	0.5000	N.D.	2.0036	2.0063	2.0287
δ_{2103}	1.0004	0.5004	N.D.	2.0046	2.0082	2.0385
δ_{4206}	1.0000	0.5002	N.D.	2.6421	2.6208	2.6091

4.4. Comparison between 1D Methods

Given the second-order decay of all superfluous error quantities, strategies σ-eq(1), σ-eq(2), ϕ-eq(2), ϕ-eq(3) are better than the original formulation (IL-FD and IL-FV), that only achieves first order. To determine the best 1D strategy, one should compare the superfluous errors of all strategies for each delta function. Figure 2 exemplifies with the mean superfluous error using δ_{4206}, being σ-eq(2) the strategy with lowest values. Indeed, σ-eq(2) presents the lowest mean value and L^2 norm of the superfluous error for all the other discrete delta functions too. In the maximum superfluous error, for some discrete delta functions, ϕ-eq(3) is slightly better in the finest grids. However, overall for these grids, σ-eq(2) is elected as the best 1D strategy.

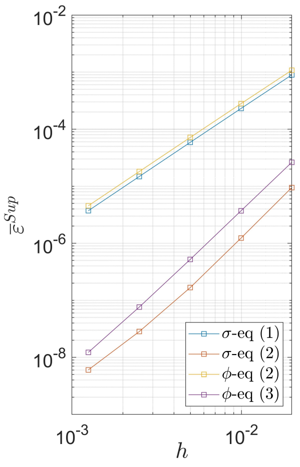

Figure 2: Superfluous mean error decays, for δ_{4206}.

As already evidenced in tables 4 and 6, figure 2 proves that, for these grids, the results with δ_{4206} are still adapting towards a second-order decay, having not yet reached the asymptotic behaviour.

4.5. Smearing Correction

In order to recover interface sharpness and propagate the superfluous decays to the transition region, correction strategies are applied to the numerical solutions coming from the system of equations. Since only transition values are corrected, transition and global decays are the only ones that change.

The corrected results for the best strategy σ-eq(2) are presented in tables 7, 8, 9, when applying procedures C1S, C2, C2S, respectively. The higher than second-order anomalies in δ_{4206} are due to the fact that the superfluous field is not yet fully stabilised.

Table 7: Decays for σ-eq (2) with correction C1S.

δ_h	$\bar{\varepsilon}$	$\|\varepsilon\|_2$	ε_{max}	$\bar{\varepsilon}^{Trans}$	$\|\varepsilon\|_2^{Trans}$	$\varepsilon_{max}^{Trans}$
δ_3^*	2.0120	2.0639	1.9963	1.9986	1.9975	1.9963
δ_{2002}	2.0007	2.0066	1.9983	1.9993	1.9989	1.9983
δ_{2103}	2.0109	2.0513	1.9963	1.9987	1.9974	1.9963
δ_{4206}	2.6650	2.4952	1.9960	2.0001	1.9965	1.9960

Table 8: Decays for σ-eq (2) with correction C2.

δ_h	$\bar{\varepsilon}$	$\|\varepsilon\|_2$	ε_{max}	$\bar{\varepsilon}^{Trans}$	$\|\varepsilon\|_2^{Trans}$	$\varepsilon_{max}^{Trans}$
δ_3^*	2.0794	2.3695	1.9972	1.9990	1.9979	1.9972
δ_{2002}	2.0410	2.2648	1.9983	1.9994	1.9987	1.9983
δ_{2103}	2.0693	2.3421	1.9972	1.9989	1.9979	1.9972
δ_{4206}	2.8649	2.4970	1.9961	1.9996	1.9971	1.9961

Table 9: Decays for σ-eq (2) with correction C2S.

δ_h	$\bar{\varepsilon}$	$\|\varepsilon\|_2$	ε_{max}	$\bar{\varepsilon}^{Trans}$	$\|\varepsilon\|_2^{Trans}$	$\varepsilon_{max}^{Trans}$
δ_3^*	1.9917	1.9923	2.0092	2.0000	2.0025	2.0113
δ_{2002}	1.9952	1.9969	2.0053	2.0008	2.0038	2.0070
δ_{2103}	2.0693	2.3421	1.9972	1.9989	1.9979	1.9972
δ_{4206}	2.2320	2.1118	2.0915	2.9865	2.9867	2.9854

Although all procedures lead to global second-order, one must analyse the values assumed by the global error quantities. Figure 3 presents the comparison between all corrections using δ_{4206}, exemplifying with the mean and maximum error (the conclusions arising from the L^2 norm are similar). Code C0 stands for no correction being used.

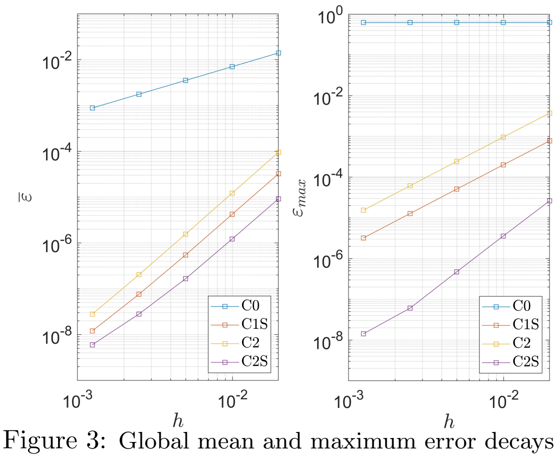

Figure 3: Global mean and maximum error decays using σ-eq(2) with different corrections, for δ_{4206}.

The corrective procedure leading to the lowest errors is C2S. Therefore, summarising all the one-dimensional conclusions, the best strategy consists of using additional equation σ-eq(2) and the best final corrective procedure to implement is C2S.

5. 2D Improvement Strategies

Given the 1D results, the equation expressing the normal derivative jump revealed advantage when compared to the one imposing the interface value. So, in 2D, several approaches to express σ are derived. Figure 4 presents a general 2D problem.

5.1. Direct Method

One way to compute σ consists of performing a local polynomial expansion around each solid point for each side of the interface, whose expressions are then used to compute each normal derivative. In the Direct Method (DM), the number of polynomial

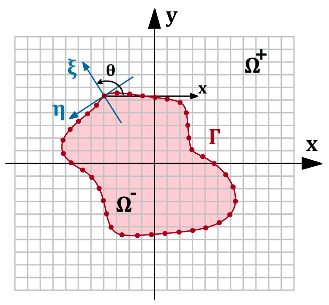

Figure 4: Illustration of a two-dimensional problem.

coefficients is equal to the number of points used to compute them. Given the misalignment between the interface and the Cartesian grid, one works in local coordinates (η, ξ) expressed by equation 26, as in [4]. The solid point has coordinates (x_{sp}, y_{sp}) and θ is measured as represented in figure 4.

$$\begin{cases} \xi = (x - x_{sp})\cos(\theta) + (y - y_{sp})\sin(\theta) \\ \eta = -(x - x_{sp})\sin(\theta) + (y - y_{sp})\cos(\theta) \end{cases} \tag{26}$$

The polynomial expansions with n_C terms have the form given in 27, with $m_\eta, m_\xi \in \mathbb{N}$ as the exponents in the last term to which η and ξ are raised.

$$\begin{aligned} \phi^{\pm}(\eta,\xi) =& C_0^{\pm} + C_1^{\pm}\eta + C_2^{\pm}\xi + C_3^{\pm}\eta\xi + \\ &+ C_4^{\pm}\eta^2 + ... + C_{n_c-1}^{\pm}\eta^{m_\eta}\xi^{m_\xi} \end{aligned} \tag{27}$$

To formally ensure second-order accuracy in the computation of tangential and normal derivatives, one should keep nine terms and compute the coefficients via a block of 3×3 points. However, since the computation of the normal derivative with first order did not prevent second-order 1D results, a version with six terms computed via 3×2 blocks is also tested. The selection of the points used to compute the coefficients is exemplified in figure 5. The first normal level is assumed to be composed of the solid point under study and its previous and next neighbours. The second normal level is composed of points that present a distance from the interface d_Γ such that $\frac{w}{2}h \leq d_\Gamma < (\frac{w}{2}+1)h$. The points in the third normal level obey $(\frac{w}{2}+1)h \leq d_\Gamma < (\frac{w}{2}+2)h$.

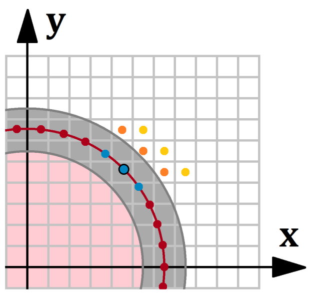

Figure 5: Example of a 3×3 block of points.

Notice that, given the misalignment between the superfluous grid points and the interface solid points, the use of perfect blocks is not possible.

5.2. Least Squares Method

The second way of obtaining σ consisted of performing a polynomial expansion as in equation 27, but computing the coefficients via a Least Squares (LS) method, that is, using a sample of points such that $n_s > n_C$. This overdetermined problem is solved following the work of Diogo [2]. According to his work, to compute first derivatives with second-order accuracy, one formally needs six terms (1, η, ξ, $\eta\xi$, η^2, ξ^2). Though, one will also test the method with the first-order set, with three terms (1, η, ξ).

Gathering the polynomial coefficients in a vector $\mathbf{c}$ and expressing each sample value ϕ_{sa} in the polynomial, one has $\mathbf{Dc} = \boldsymbol{\phi_{sa}}$, with $\mathbf{D}$ relying on the local coordinates of each sample point. The LS method may be formulated with a diagonal weight function matrix $\mathbf{W_{LS}}$ and, following the deductions in [2], the coefficients are given by $\mathbf{c} = \mathbf{P_I}\boldsymbol{\phi_{sa}}$ with $\mathbf{P_I}$being the pseudoinverse matrix with expression $\mathbf{P_I} = (\mathbf{D}^T\mathbf{W_{LS}}\mathbf{D})^{-1}(\mathbf{D}^T\mathbf{W_{LS}})$. In this work, the weights were computed according to the work of Vasconcelos [8], as in equation 28 with $\kappa = 2$.

$$(w_{LS})_{i,i} = \frac{1}{(\sqrt{(\eta_i - \eta_0)^2 + (\xi_i - \xi_0)^2})^{\kappa}} \tag{28}$$

Having the polynomial coefficients computed, the needed derivatives to obtain σ may be expressed.

Since the problem is overdetermined, the constraints will not be exactly fulfilled, since a compromise solution arises. In particular, if the solid point under study is used as sample point, its value predicted by the LS polynomial is different from its known analytical one. Thus, a second variant of the LS method was implemented, in which one imposes the solid point value, $C_0^{\pm} = \phi_0^{\pm}$, by working with relative polynomials $\phi^{\pm}(\eta,\xi) - \phi_0^{\pm}$. Everything deduced remains valid, but now matrix $\mathbf{D}$ has one less row and column, vector $\mathbf{c}$ does not include $C_0^{\pm}$ and the sample vector $\boldsymbol{\phi_{sa}}$ has relative values.

As superfluous sample points, one selects all the ones included in an elliptical cloud centred in the solid point, as illustrated in figure 6 for tangential and normal semiaxes given by $s_\eta = 2h$ and $s_\xi = (\frac{w}{2}+3)h$, respectively. Depending on the case, one may use the solid point (imposed or not) alone, or use its previous and next neighbours too.

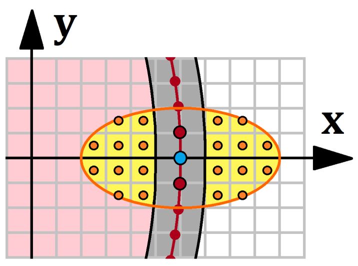

Figure 6: Example of an elliptical cloud.

5.3. 1D-Inspired Method

The last method to be derived aimed to replicate as faithfully as possible the 1D reasoning, named as 1D-Inspired Method (1DIM). In this method, σ is expressed in terms of general points along the normal direction (like if it was a 1D problem) and then, as those points will not be available on the grid, one expresses them in terms of neighbouring grid points. The process is illustrated in figure 7.

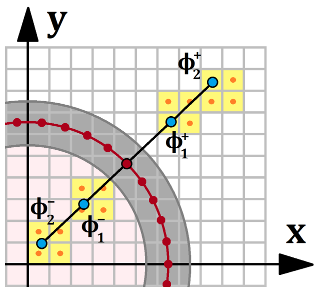

Figure 7: Illustration of the 1D-Inspired Method.

As such, expressing σ in terms of the points along the normal direction is similar to what is performed in 1D (see section 3.1) and expressing those points in terms of grid neighbouring points is simply achieved by performing a local Cartesian polynomial expansion centred at each block. One will test blocks with dimensions 2×2, 3×3 and 4×4.

5.4. Smearing Correction

In the end of the 2D simulation, corrective procedures are also derived, recovering interface sharpness and propagating superfluous decays to the transition region. In the first corrective procedure (C-Poly), a local polynomial expansion is performed around the closest solid point to the transition point under correction. For that, a 3×3 block similar to the one represented in figure 5 for the Direct Method is used to compute the coefficients. Finally, the corrected transition value is obtained by substituting its coordinates in the derived polynomial.

In a second corrective procedure (C-Cart), simple polynomial extrapolations along each Cartesian direction are performed, as illustrated in figure 8.

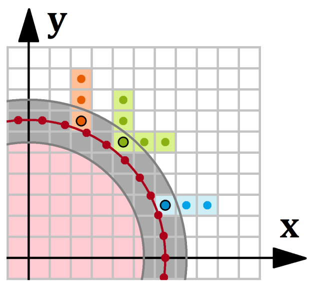

Figure 8: Illustration of the Cartesian correction.

In this strategy, the closest superfluous points along x and y are determined. If one of them is closer than the other, the extrapolation along that direction is taken. If they are equidistant from the transition point, both extrapolations are performed and their average is computed. If a transition point is above the interface, the average between an interior and an exterior procedure is taken.

6. 2D Results

In this section, the 2D results are presented. In that regard, the discontinuous function of discontinuous normal derivative to be considered was characterised by $\phi^-(\mathbf{x}) = e^x \cos(y)$ and $\phi^+(\mathbf{x}) = 0$. The solid body was particularised into a circle of radius $R = 0.5$ and $L = 2$. The decays were computed for grids $N = \{50^2; 200^2\}$, with an intermediate one $N = 100^2$ in the decay graphs to confirm the existence of a straight tendency (asymptotic decay).

6.1. Original Formulation

The decays concerning the original formulation implemented in finite differences (IL-FD) and finite volume (IL-FV) are presented in tables 10 and 11. The results are similar to the ones observed in the 1D original formulation, with transition errors not decaying, nor the maximum global error, whereas the global mean error decays with first order and the global L^2 norm presents half-order accuracy.

Table 10: Global and superfluous decay, 2D IL-FD.

δ_h	$\bar{\varepsilon}$	$\|\varepsilon\|_2$	ε_{max}	$\bar{\varepsilon}^{Sup}$	$\|\varepsilon\|_2^{Sup}$	ε_{max}^{Sup}
δ_3^*	0.9452	0.4769	N.D.	0.9128	0.9257	0.9209
δ_{2002}	0.9859	0.4903	N.D.	0.9925	0.9721	0.3453
δ_{2103}	0.9998	0.4774	N.D.	1.1460	1.0648	0.2919
δ_{4206}	0.9905	0.4899	N.D.	1.0446	1.0173	0.7524

Table 11: Global and superfluous decay, 2D IL-FV.

δ_h	$\bar{\varepsilon}$	$\|\varepsilon\|_2$	ε_{max}	$\bar{\varepsilon}^{Sup}$	$\|\varepsilon\|_2^{Sup}$	ε_{max}^{Sup}
δ_3^*	0.9841	0.5254	N.D.	0.9345	0.9338	0.8581
δ_{2002}	0.9683	0.5212	N.D.	0.9350	0.9112	0.1647
δ_{2103}	0.9862	0.5256	N.D.	0.9592	0.9452	0.3645
δ_{4206}	0.9844	0.5159	N.D.	0.9434	0.9317	0.5081

The mean value and L^2 norm of the superfluous error present first-order decays for all discrete delta functions. The result highlighted in blue in table 10 is the one corresponding to the first-order superfluous L^2 norm decay reported by Eldredge [3]. In 2D, the maximum superfluous decay suffers a penalisation compared to the 1D case. Delta functions δ_{2002} and δ_{2103} are the most affected, followed by δ_{4206}, and δ_3^* is the one with the smallest changes. Except for δ_{2103}, the FV version leads to further penalties.

6.2. Direct Method

Concerning the DM, the superfluous decays using the 3×2 (DM-32) and using 3×3 (DM-33) blocks of points are provided in table 12. As in 1D, as long as no corrective procedure is applied, transition error quantities do not decay. As such, they are omitted.

Table 12: Superfluous decay for DM-32 and DM-33.

		$\bar{\varepsilon}^{Sup}$	$\lVert\varepsilon\rVert_2^{Sup}$	ε_{max}^{Sup}
DM-32	δ_3^*	1.8625	1.8154	0.9521
	δ_{2002}	0.3482	0.4517	N.D.
	δ_{2103}	0.6462	0.6484	N.D.
	δ_{4206}	1.7519	1.5627	0.7658
DM-33	δ_3^*	1.8348	1.6889	0.3650
	δ_{2002}	0.4156	0.4793	N.D.
	δ_{2103}	1.6121	0.9798	N.D.
	δ_{4206}	1.1392	0.9834	0.2984

The best results for the DM are achieved for DM-32 with δ_3^*. Now, δ_{2002} and δ_{2103} do not even decay for the maximum superfluous error. This is not acceptable, since one is looking for good superfluous decays. Until now, all the 2D results concerning δ_{2002} and δ_{2103} were the worst ones and this tendency occurs for all the upcoming strategies as well. So, from now on, these functions are dropped.

6.3. Least Squares Method

In LS method, an extensive study was made concerning the several free parameters. Only the best results are presented. One tested elliptical clouds with $s_\eta = Zh$ and $s_\xi = (\frac{w}{2} + Z)h$, for $Z = \{3; 4; 5\}$.

When using three polynomial terms, the best result was obtained using weights, imposing the solid point, using the previous and next solid neighbours and using an elliptical cloud with $Z = 3$. One calls this strategy LSM-Y. When using six polynomial terms, the best result arose for not using weights, not imposing the solid point, using the previous and next neighbours and having an elliptical cloud with $Z = 5$. One calls this strategy LSM-O.

Table 13: Superfluous decays for LSM-Y/O.

		$\bar{\varepsilon}^{Sup}$	$\lVert\varepsilon\rVert_2^{Sup}$	ε_{max}^{Sup}
LSM-Y	δ_3^*	1.8680	1.8376	1.1217
	δ_{4206}	1.6750	1.5641	0.8708
LSM-O	δ_3^*	1.9294	1.8006	0.4564
	δ_{4206}	1.6834	1.1833	0.4657

6.4. 1D-Inspired Method

The superfluous decays observed for the 1DIM considering 2×2 (1DIM-2), 3×3 (1DIM-3) and 4×4 (1DIM-4) blocks are presented in table 14.

The increase in the dimension of the blocks led overall to an increase of the decays, with the superfluous mean value and L^2 norm reaching second-order decay values for 1DIM-4 when using δ_3^*. However, the decay in the superfluous maximum error was always found to be lower than first-order.

Table 14: Superfluous decay of method 1DIM.

		$\bar{\varepsilon}^{Sup}$	$\lVert\varepsilon\rVert_2^{Sup}$	ε_{max}^{Sup}
1DIM-2	δ_3^*	1.9407	1.8079	0.4490
	δ_{4206}	1.7698	1.2110	0.4791
1DIM-3	δ_3^*	1.9861	1.8702	0.5407
	δ_{4206}	1.8072	1.2638	0.5117
1DIM-4	δ_3^*	2.0107	1.9010	0.5891
	δ_{4206}	1.7982	1.1292	0.4644

6.5. Comparison between 2D Methods

At this point, one should compare the results of the best version of each method among themselves and with the original formulation, in order to determine the best overall strategy. In that regard, table 15 gathers the superfluous decays of all the best implementations. Note that all of them concerned δ_3^*.

Table 15: Comparison between formulations, with δ_3^*.

	$\bar{\varepsilon}^{Sup}$	$\lVert\varepsilon\rVert_2^{Sup}$	ε_{max}^{Sup}
IL-FD	0.9128	0.9257	0.9209
IL-FV	0.9345	0.9338	0.8581
DM-32	1.8625	1.8154	0.9521
LSM-Y	1.8680	1.8376	1.1217
LSM-O	1.9294	1.8006	0.4564
1DIM-4	2.0107	1.9010	0.5891

Analysing table 15, one concludes that only two strategies performed better than the original formulation (in blue): DM-32 and LSM-Y. All the others, despite the high decays in the mean and L^2 norm, present a lower than first-order maximum superfluous error decay, performing worse than the original formulation. Besides comparing the error decays, one should also compare their values. This comparison is made in figure 9 for the mean and maximum superfluous errors (the L^2 norm graph is qualitatively identical to the mean). As IL-FD and IL-FV present superimposed lines, only IL-FV appears.

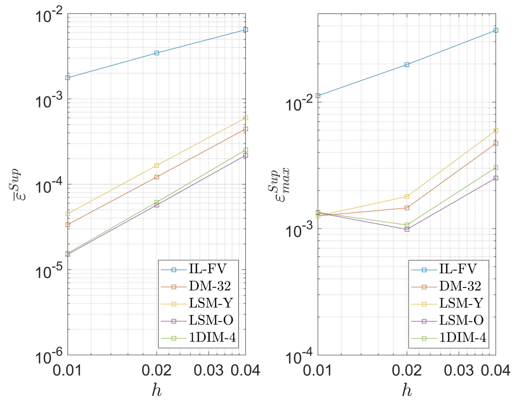

Figure 9: General comparison, using δ_3^*.

Figure 9 shows that the original formulation (blue line for IL-FV and IL-FD) is the one with the highest error. For the superfluous mean, the derived strategy with the highest error is LSM-Y, followed by DM-32, 1DIM-4 and LSM-O. The order inverts for the finest grid in the maximum superfluous error. If one intends to work exclusively in this range of grids, probably one should use DM-32. Though, given the higher decays of LSM-Y, for finest grids it is predicted to surpass DM-32. Overall, given the higher than first-order decay of the maximum superfluous error, strategy LSM-Y is elected the best.

6.6. Smearing Correction

In order to test the derived 2D corrective procedures, C-Poly and C-Cart were applied to strategy LSM-Y using δ_3^*. After the correction, the global and transition decays are summarised in table 16.

Table 16: Corrected decays with strategy LSM-Y.

	$\bar{\varepsilon}$	$\|\varepsilon\|_2$	ε_{max}	$\bar{\varepsilon}^{Trans}$	$\|\varepsilon\|_2^{Trans}$	$\varepsilon_{max}^{Trans}$
C-Poly	2.0787	2.0388	0.8219	1.7809	1.6942	0.8219
C-Cart	2.1131	2.0986	1.3739	1.7649	1.7248	1.3739

According to figure 9, the maximum superfluous error is not decaying as a straight line, but rather oscillates around a straight tendency. So, as verified in the 1D results (recall tables 7 to 9), the decays obtained in the corrected transition region, and consequently in the entire domain, may occur to be higher than the ones verified in the superfluous region that is not completely stabilised. Figure 10 compares the global mean and maximum errors originated by both correction procedures. The global L^2 norm behaves like the mean error.

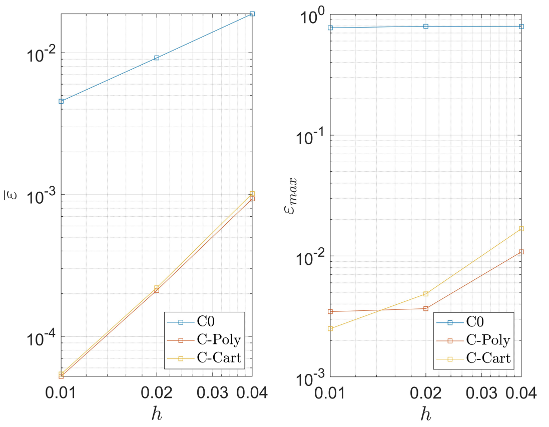

Figure 10: 2D Corrections with LSM-Y and δ_3^*.

The error values presented by both corrective procedures are very close to each other. For example, there is a 4% difference concerning the central point in the mean error graph. Concerning the global maximum error, none procedure dominates all the time and finest grids are needed to check if they keep oscillating and changing relative position.

7. Conclusions

All derived 1D strategies led to second order in the superfluous region, later extended to the transition region by the corrective procedures, regardless of the discrete delta function. In 2D, only two strategies were able to be overall better than the original formulation (DM-32 and LSM-Y) and both with δ_3^*, highlighting its advantageous properties. The best 2D strategy is able to achieve superfluous mean and L^2 norm decays around 1.8 and a superfluous maximum error decay around 1.1.

Acknowledgements

The Author thanks *Fundação para a Ciência e Tecnologia* (FCT), for the financial support given to the research project HIBforMBP, identified as PTDC/EME-EME/32315/2017.

References

[1] V. Cola, S. Cuomo, and G. Severino. Remarks on the numerical approximation of Dirac delta functions. *Results in Applied Mathematics*, 12:100200, Nov. 2021.

[2] F. Diogo. A very high-order finite volume technique for convection-diffusion problems on unstructured grids. Master's thesis, Instituto Superior Técnico, 2019.

[3] J. Eldredge. A method of immersed layers on Cartesian grids, with application to incompressible flows. *Journal of Computational Physics*, 448:110716, Jan. 2022.

[4] R. Leveque and Z. Li. The Immersed Interface Method for elliptic equations with discontinuous coefficients and singular sources. *SIAM Journal on Numerical Analysis*, 31(4):1019–1044, Aug. 1994.

[5] Y. Liu and Y. Mori. Properties of discrete delta functions and local convergence of the immersed boundary method. *SIAM Journal on Numerical Analysis*, 50(6):2986–3015, Aug. 2012.

[6] R. Mittal and G. Iccarino. Immersed Boundary Methods. *Annual Review of Fluid Mechanics*, 37:239–261, 2005.

[7] C. Peskin. The Immersed Boundary Method. *Acta Numerica*, 11:479–517, Jan. 2002.

[8] A. Vasconcelos. A very high-order finite volume method based on weighted least squares for the solution of poisson equation on unstructured grids. Master's thesis, Instituto Superior Técnico, 2017.

[9] X. Yang, X. Zhang, Z. Li, and G. He. A smoothing technique for discrete delta functions with application to immersed boundary method in moving boundary simulations. *Journal of Computational Physics*, 228(20):7821–7836, Nov. 2009.