# A High-Order Immersed Boundary Technique for Computational Aeroacoustics

Manuel Diaz Escobar, Véronique Fortuné, David Marx, Christian Prax

16ème Congrès Français d'Acoustique
11-15 Avril 2022, Marseille

# A High-order Immersed Boundary Technique for Computational Aeroacoustics

M. A. Diaz [a], V. Fortuné [a], D. Marx [a], C. Prax [a]
[a] Institute Pprime–CNRS, Université de Poitiers et ISAE-ENSMA, B17 -6 rue Marcel Doré, TSA 41105, Cedex 9, 86073 Poitiers, France

Finite difference methods are widely employed for numerical simulation in aeroacoustics, a domain in which high-order, low-dispersive and low-dissipative methods are widespread. However, finite differences are restricted to Cartesian grids. When solids of arbitrary shape are present in the simulation, it is possible to use immersed boundary methods (IBM) to avoid boundary-fitted grids. The order of IBM methods is often restricted to 2, even if several high-order IBM methods have been presented in the literature for aeroacoustics. However, we have found these methods to be rather unstable. In this work we seek for a stable, high-order immersed boundary method, for use in the Navier Stokes equations. The technique treats the immersed boundaries as a sharp interface by enforcing boundary conditions through ghost points. These are computed using characteristics and extrapolation along the normal direction to the interface. The method is tested on convective and diffusive problems, and has been applied to benchmark problems for the linearized Euler equations. High orders of convergence could be observed on model problems, with little change in time-step size, without filtering playing an essential role.

# 1 Introduction

High-order, spectrally optimized, finite difference methods are of common use in acoustic and aeroacoustic computations. When geometries of complex shapes are part of the simulation, the several methods known collectively as immersed boundary (IB) methods are appealing, because they make it possible to keep on using these finite difference methods together with a Cartesian mesh that is easy and cheap to generate. In aeroacoustics, these methods are used both for linearized equations [1, 2] or for nonlinear equations [3]. Their major inconvenient is probably their lack of stability, especially in inviscid or high Reynolds situations when a high order of accuracy is targeted. Our different attempts to use methods of order higher than 2 has often led to stability issues, and the resulting oscillations cannot always be suppressed by using filters. In this paper, we targeted a method not relying on selective filters and being globally 4th order (we do not focus on spectral optimization at this stage).

Several recent works in the literature have tried to obtain techniques that could lead to accurate an stable finite-difference (FD) solvers of PDEs. Recently, Kalili et al. [3] have combined a 6th-order compact scheme with the IB technique of Seo and Mittal [1] and the traditional Image-Point technique to solve directly the *compressible* Navier-Stokes (N-S) equations. The efforts yield a successful third-order numerical solver for the N-S equations. On the other hand, Jianfang et al. [12] have presented a boundary treatment that extends 5th-order WENO [1] schemes functionality on domains where boundary points do not intersect the numerical grid. Although, they also have reported stable solutions only up to third-order for N-S equations [11], the showed that 5th-order solutions for pure hyperbolic PDEs are within reach. In a completely different view, Brady and Livescu [8] chose to avoid the construction of ghost-points and directly tackle the Cut-cell problem by constructing degenerate finite difference schemes using a *truncation error matching* approach. While extremely high-order accuracy on elliptic equations where reported, only 4th-order stable schemes where achieved for pure hyperbolic problems. In a similar fashion, Hosseinverdi and Fasel [6] explored *immersed interface* (II) technique of Wiegmann and Bube [10] to avoid the construction of ghost-points. In their work, unsteady solutions of the *incompressible* N-S equations were achieved. However, only a fourth-order solver was presented. In recent note, the II-technique was associated with 5th-order WENO methods by Fernández-Fidalgo et. al. [9], who reported 5-th order accurate results for pure hyperbolic problems without shocks. All these works propose a different FD-approach that is associated to their custom IB-technique. This results in a rainbow of numerical schemes that obscure the true performance of each individual technique and makes it difficult to compare and improve. To overcome such situation, we choose to reproduce several of IB techniques presented, namely [1, 3, 11, 12], and use them together with explicit central schemes and the classical 4th-order Runge-Kutta to study classical benchmark problems of the literature. The above exercise has led us to compare and produce of a new IB-strategy. This novel strategy is here found to be accurate, stable and have little to no impact on the size of the time step, $\Delta t$. To demonstrate our claim, we use it to obtain 4-th order accurate and stable numerical solutions of hyperbolic and parabolic benchmark problems in Aeroacoustics.

1. WENO : weighted essentially non-oscillatory methods.

# 2 Methods

## 2.1 Problem Formulation

In this work, we aim to resolve for two-dimensional (2-d) system of first-order PDEs of the form

$$\boldsymbol{q}_t + \mathcal{F}^c_x(\boldsymbol{q}) + \mathcal{G}^c_y(\boldsymbol{q}) = \mathcal{F}^v_x(\boldsymbol{q}) + \mathcal{G}^v_y(\boldsymbol{q}), \quad (x,y) \in \Omega,\ t > 0, \quad (1)$$

subject to

$$\begin{cases} \boldsymbol{q}(x,y,0) = \boldsymbol{q}_0(x,y), & (x,y) \in \Omega, \\ \boldsymbol{q}(x,y,t) = g(t), & (x,y) \in \partial\Omega_{in},\ t > 0, \\ \boldsymbol{q}(x,y,t) = k(t), & (x,y) \in \partial\Omega_{out},\ t > 0, \end{cases} \quad (2)$$

where $\boldsymbol{q}$ denotes the quantities to be evolved, $\mathcal{F}$ and $\mathcal{G}$ denote convective ($c$) and diffusive ($v$) flux functions in the x- and y-directions, respectively. $\Omega$ and $\partial\Omega$ are discretized over a uniform Cartesian mesh with mesh sizes

$$l_x = (b-a), \quad \Delta x = l_x/(N_x - 1),$$
$$l_y = (d-c), \quad \Delta y = l_y/(N_y - 1),$$

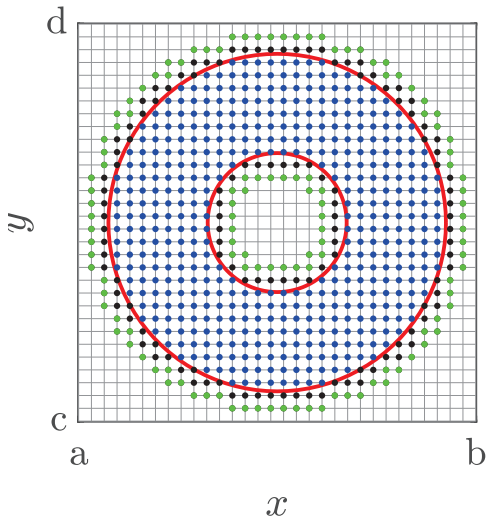



FIGURE 1 – Discrete domain : the blue dots denote points within the domain, $\Omega$, the red lines denote the virtual location of the boundaries, $\partial\Omega_{in}$, $\partial\Omega_{out}$. Lastly, lying outside the domain, the black and green dots denote the first- and second-rows of ghost points, respectively

so that $x_i = a + i\Delta x$, for $i = 0, \ldots, N_x$ and $y_j = c + j\Delta y$, for $j = 0, \ldots, N_y$, as examplified in Fig. 1. For the sake of simplicity, we further assume that $\Delta x = \Delta y = h$.

## 2.2 Space & Time Discretization

Without loss of generality, observe that the governing system (1) can be abstractly written as

$$\boldsymbol{q}_t = \mathcal{L}(\boldsymbol{q}), \tag{3}$$

where $\mathcal{L}$ is a spatial operator involving first-order spatial derivatives. After the spatial operator is discretized, the semi-discrete scheme is written abstractly as the following ordinary differential equation (ODE) system

$$\boldsymbol{q}_t = \mathcal{L}_h(\boldsymbol{q}). \tag{4}$$

Thus, using for example, the classical explicit 4th-order Runge-Kutta scheme, we integrate Eq. (4) in time.

To match our accuracy in time, we constrain ourselves to discretize our flux operators (and any first-order derivative within them) with a simple 4th-order central FD-scheme. For example : the discretazation of the fluxes $\mathcal{F}$ and $\mathcal{G}$ would simply read :

$$\begin{aligned} \mathcal{F}_x(\boldsymbol{q}) &\approx \frac{1}{12h}(\mathcal{F}(\boldsymbol{q}_{i,j-2}) - 8\mathcal{F}(\boldsymbol{q}_{i,j-1}) + 8\mathcal{F}(\boldsymbol{q}_{i,j+1}) - \mathcal{F}(\boldsymbol{q}_{i,j+2})) + O(h^4) \\ \mathcal{G}_y(\boldsymbol{q}) &\approx \frac{1}{12h}(\mathcal{G}(\boldsymbol{q}_{i-2,j}) - 8\mathcal{G}(\boldsymbol{q}_{i-1,j}) + 8\mathcal{G}(\boldsymbol{q}_{i+1,j}) - \mathcal{G}(\boldsymbol{q}_{i+2,j})) + O(h^4) \end{aligned} \tag{5}$$

In other words, we aim to discretize our governing PDE using a uniformly 4th-order scheme in space and time.

## 2.3 Immersed Boundary Technique

Similar to [1, 3, 11, 12], the present IB-technique seeks to avoid the *cut-cell* problem by means of extending each of the $f$-fields, contained within $\boldsymbol{q}$, over points outside the domain, i.e. : the ghost points. This allows us to use the interior scheme directly in the vicinity of the domain boundary. In the following we describe our strategy. For simplicity, here we refer to the points within the domain as *fluid-points* and any point on the boundary of the domain (real or virtual) is called *boundary-point*.

### 2.3.1 Ghost Point reconstruction

Extending the field over ghost-points is here performed in a two-steps process :

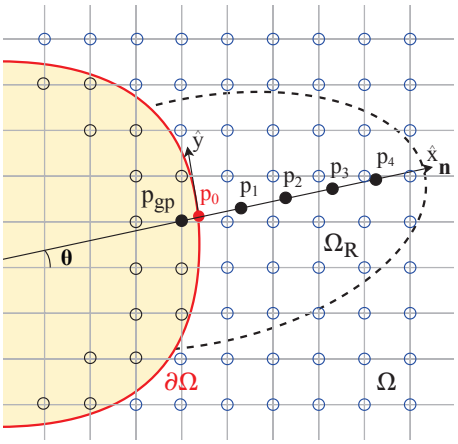



FIGURE 2 – (Step 1) : Interpolate a uniform stencil of points over a sub-region of fluid-points.

**Step 1**. Interpolation : Let us start by considering an outward normal $\boldsymbol{n}$ to $\partial\Omega$ that passes by the ghost point $p_{gp}$ as depicted in Figure (2). Observe that $\boldsymbol{n}$ not only defines the distances to the point, $h_{gp}$, but also defines the location of the virtual boundary-point $p_0$.

On $p_0$, we setup a local coordinate system by

$$\begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} = \boldsymbol{T} \begin{bmatrix} x \\ y \end{bmatrix}, \quad \boldsymbol{T} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}, \tag{6}$$

where $\theta$ is the angle between the outward normal $\boldsymbol{n}$ at the $x$-axis, and $\boldsymbol{T}$ is the rotation matrix. As a result, we can now define points $\{p_i\}_{i=1}^n$ to be uniformly spaced by $h$ on the $\hat{x}$-axis. In Figure 2, the case for $n$=4 is illustrated.

Assume we have the values $f_{ij}$ on the grid points of a function $f(x, y)$ in the interior domain, $\Omega$. Our goal is to interpolate the field values on $\{p_i\}_{i=0}^n$, namely $\{f_i\}_{i=0}^n$.

To this end, we take a stencil $\mathcal{E}$ to initially obtain the high-order 2-d Lagrange approximating polynomial as follows

$$\mathcal{E} = \left\{(x_i, y_j) \in \Omega_R\right\}.$$

where $\Omega_R$ denotes a sub-region of $\Omega$ defined as

$$\Omega_R : \left\{(x, y) \in \Omega, \left(\frac{x'\cos(\theta)+y'\sin(\theta)}{R_{\hat{x}}}\right)^2 + \left(\frac{-x'\sin(\theta)+y'\cos(\theta)}{R_{\hat{y}}}\right)^2 \leq 1\right\},$$

where $x' = (x - x_{p_0})/h$ and $y' = (y - y_{p_0})/h$, denote local coordinates centered on $p_0$-coordinates. $R_{\hat{x}}$ and $R_{\hat{y}}$ denote the

maximum half-widths of the elliptical region [2] in the $\hat{x}$- and $\hat{y}$-direction, respectively. In this work, we set $R_{\hat{x}}$ and $R_{\hat{y}}$ as

$$\begin{cases} R_{\hat{x}} = 2k, \\ R_{\hat{y}} = k/2 + 1, \end{cases} \tag{7}$$

where $k$ denotes the order of the interpolation polynomial to be constructed with all the fluid-points in $\Omega_R$.

We assume that *N-fluid-points*, each with coordinates $(x'_i, y'_j)$, are contained within $\mathcal{E}$. Thus, a 2-d Lagrange polynomial, $P(x', y')$, of $k$-order in $Q^k$ to approximate the values $f_{ij}$ has the form

$$P(x', y') = \sum_{m=0}^{k} \sum_{l=0}^{k} a_{ml} \, (x')^m (y')^l, \quad m + l \leq k, \tag{8}$$

where $a_{ml}$ are constant coefficients associated to each of the terms in the monomial basis $(x')^m (y')^l$. To solve for each $a_{ml}$-coefficient in $P(x', y')$, we require that

$$P(x'_i, y'_j) = f_{ij}, \quad \forall (x'_i, y'_j) \in \mathcal{E}. \tag{9}$$

The above conditions yields an algebraic discrete system of equations of the form $\boldsymbol{f} = \boldsymbol{V}\boldsymbol{a}$, where, $\boldsymbol{V}$ is the Vandermonde matrix. By re-naming the coordinates associated to each $\ell$-mesh vertex in $\mathcal{E}$ as $(x'_\ell, y'_\ell)$, $\boldsymbol{V}$ takes the form :

$$\boldsymbol{V} = \begin{bmatrix} 1 & x'_1 & y'_1 & \cdots & (x'_1)^2 & (y'_1)^2 & \cdots & (x'_1)^k & (y'_1)^k \\ 1 & x'_2 & y'_2 & \cdots & (x'_2)^2 & (y'_2)^2 & \cdots & (x'_2)^k & (y'_2)^k \\ \vdots & \vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots \\ 1 & x'_N & y'_N & \cdots & (x'_N)^2 & (y'_N)^2 & \cdots & (x'_N)^k & (y'_N)^k \end{bmatrix}. \tag{10}$$

Since the problem is expected to be over-determined, the vector of coefficients $\boldsymbol{a}$ is then resolved using a traditional least squares approach, namely :

$$\boldsymbol{a} = [(\boldsymbol{V}^T \boldsymbol{V})^{-1} \boldsymbol{V}^T] \boldsymbol{f}. \tag{LSQ}$$

Therefore, field values at points $\{p_i\}_{i=1}^n$ points with coordinates $(x_i, y_i) \in \Omega_R$ are explicitly *interpolated* as

$$f_i = P(x_i, y_i), \qquad \text{for } i = 1, \ldots, n.$$

Note that the field value at $p_0$ can also be *extrapolated* as $f_0 = P(x_0, y_0)$. However, recall that its final value depends on the boundary condition set the problem formulation. E.g. : following Eq. (2), a *Dirichlet condition* is set on $\partial\Omega_{in}$, therefore $f_0$ is simply

$$f_0 = g(t), \qquad \text{on } \partial\Omega_{in}.$$

**Step 2**. Extrapolation : Having interpolated the $f$-fields on the virtual fluid-points $\{p_j\}_1^n$ and setting our boundary condition at the boundary at $p_0$. Then, the second step is to extrapolate for the ghost-point associated with this configuration construction a 1-d polynomial as illustrated in figures 3a and 3b. Evidently, the polynomial coefficients depend on the nature of the condition set at $p_0$. In this work, the extrapolation of for a single $f$-field, in $\boldsymbol{q}$, on a ghost-point under the requirement of a Dirichlet condition, is computed as

$$f(\hat{x}_{gp}, t) = \frac{g(t) - \sum_{\ell=1}^n c_\ell \cdot f_\ell}{c_0(r)}, \tag{11}$$

where $r = h_{gp}/h$, is the ratio between the ghost point distance, $h_{gp} = p_{gp} - p_0$, and mesh size $h$. $\{c_\ell\}_{\ell=1}^n$ denotes the constant coefficients resolved with a Taylor table approach.

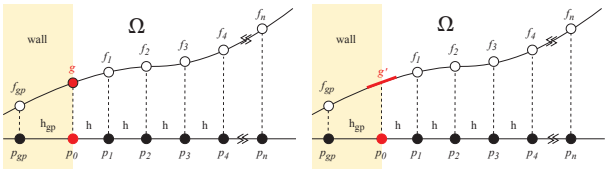
Similarly, a ghost point extrapolation under the requirement of a Neumann boundary condition, is computed as

$$f(\hat{x}_{gp}, t) = \frac{\hat{n} \cdot g'(t) - \sum_{\ell=1}^n \tilde{c}_\ell \cdot f_\ell}{\tilde{c}_0(r)}, \tag{12}$$

where $\hat{n}$ denotes the normal to solid boundary face, $\{\tilde{c}_\ell\}_{\ell=1}^n$ denote constant coefficients resolved with a Taylor table approach.

Lastly, a *free* (or no-) boundary condition means that we look to extrapolate a given $f$-field over some desired ghost points without any assumption. As a result, the ghost points are directly extrapolated using the Lagrange polynomial constructed for stencil points $\{p_j\}_{j=0}^n$ depicted in Figure (2).

**Remark** : Eqs. (11) and (12) are the 1-d equivalent of those presented Seo & Mittal [1].




(a) Dirichlet Boundary (b) Neumann Boundary

FIGURE 3 – (Step 2) Ghost-points extrapolation according to boundary conditions.

### 2.3.2 Characteristics extrapolation

Provided that local coordinate system $(\hat{x}, \hat{y})$ has been established at the virtual (or real) boundary point $p_0$, the rotation matrix $\mathbf{T}$, in Eq. (6), is then used to project the vector-fields contained withing $\boldsymbol{q}$-quantities associated to each point $\{p_i\}_{i=1}^n$. *Example* : assume $\boldsymbol{q} = \{\wp, u, v\}$, then its projection on the local coordinates is denoted $\hat{\boldsymbol{q}} = \{\wp, \hat{u}, \hat{v}\}$, where

$$\begin{bmatrix} \hat{u} \\ \hat{v} \end{bmatrix} = \mathbf{T} \begin{bmatrix} u \\ v \end{bmatrix}.$$

Thus, in this local-coordinates, $\hat{\boldsymbol{q}}$-quantities are now governed by a transformed two-dimensional wave system that reads

$$\hat{\boldsymbol{q}}_t + \mathcal{F}^c_{\hat{x}}(\hat{\boldsymbol{q}}) + \mathcal{G}^c_{\hat{y}}(\hat{\boldsymbol{q}}) = \mathcal{F}^v_{\hat{x}}(\hat{\boldsymbol{q}}) + \mathcal{G}^v_{\hat{y}}(\hat{\boldsymbol{q}}), \quad (\hat{x}, \hat{y}) \in \Omega, \; t > 0. \tag{13}$$

Without loss of generality, one can re-express the governing equations (1) in quasilinear form as

$$\hat{\boldsymbol{q}}_t + \boldsymbol{A}(\hat{\boldsymbol{q}})\hat{\boldsymbol{q}}_{\hat{x}} + \mathcal{G}^c_{\hat{y}}(\hat{\boldsymbol{q}}) = \mathcal{F}^v_{\hat{x}}(\hat{\boldsymbol{q}}) + \mathcal{G}^v_{\hat{y}}(\hat{\boldsymbol{q}}),$$

2. Note that for the case $R_{\hat{x}}$=$R_{\hat{y}}$=$R$, the method recovers the approach traditionally used by [10, 1, 3] and [12].

where, $\boldsymbol{A}(\hat{\boldsymbol{q}})=[\mathcal{F}^c_{\hat{\boldsymbol{q}}}(\hat{\boldsymbol{q}})]$ so that $\boldsymbol{A}(\hat{\boldsymbol{q}}) = \mathbf{L}\boldsymbol{\Lambda}\mathbf{R}$, where $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_n)$ are eigenvalues that denote principals speed of propagations in the flux function and $\mathbf{L}$ and $\mathbf{R}$ are the right and left eigenvectors. This allows us to reformulate Eq. (13) into characteristics-form in the $\hat{x}$-direction, namely

$$\mathbf{v}_t + \boldsymbol{\Lambda}\mathbf{v}_{\hat{x}} = -\mathbf{L}\mathcal{G}^c_{\hat{y}}(\hat{\boldsymbol{q}}) + \mathbf{L}\mathcal{F}^v_{\hat{x}}(\hat{\boldsymbol{q}}) + \mathbf{L}\mathcal{G}^v_{\hat{y}}(\hat{\boldsymbol{q}}), \qquad (14)$$

where $\mathbf{v} = \mathbf{L}\hat{\boldsymbol{q}}$. In this work, we do not use any slope-limiting or weighted non-oscillatory technique, thus we only deal with subsonic scenarios. Therefore, building ghost-points using characteristic boundary conditions is a two-step procedure :

**Step 1**. Define $\hat{\boldsymbol{q}}_{p_0}$ : Often than not, $\mathbf{L} : \mathbf{L}(\boldsymbol{q})$ and $\mathbf{R} : \mathbf{R}(\boldsymbol{q})$. In this work, we use $\mathbf{L}(\hat{\boldsymbol{q}}_{p_0})$ and $\mathbf{R}(\hat{\boldsymbol{q}}_{p_0})$, where $\hat{\boldsymbol{q}}_{p_0}$ is determined using the conditions set of $\partial\Omega$ and it is complemented with the extrapolation of the remaining undefined quantities. $\hat{\boldsymbol{q}}_{p_0}$ also allows us to resolve for $\lambda_1, \ldots, \lambda_n$, and therefore, distinguish the incoming and outgoing characteristics.

**Step 2**. Define $\mathbf{v}_{p_0}$ : We start by projecting the $\boldsymbol{q}$-quantities associated to points $\{p_i\}_{i=1}^n$ into the characteristic space by doing

$$\{\mathbf{v}_{p_i}\}_{i=1}^n = \mathbf{L}\{\hat{\boldsymbol{q}}_{p_i}\}_{i=1}^n.$$

Defining $\mathbf{v}_{p_0}$, now requires that we identify within relations $\mathbf{v} = \mathbf{L}\hat{\boldsymbol{q}}$ and/or $\hat{\boldsymbol{q}} = \mathbf{R}\mathbf{v}$ those that allows us to define the incoming components in $\mathbf{v}_{p_0}$ while the outgoing ones are simply extrapolated using information within $\Omega$.

*Example* : consider the case of a non-slip isothermal-wall boundary condition on the 2-d compressible Navier-Stokes Eqs. [13]. Thus, we aim to impose $(\varrho\hat{u})_{wall}$=$(\varrho\hat{v})_{wall}$=0 and $T_{wall}$. Let us further assume that acoustic waves (or the flow) are moving towards to wall, i.e. $\hat{u} < 0$. As a result, $\lambda_4 = \hat{u} + c$ indicates that $v_4$ is the only incoming characteristic. Thus we resolve for $v_4$ at $p_0$ as

$$v_4|_{p_0} = l_{41}q_1 + l_{42}q_2 + l_{43}q_3 + l_{44}q_4,$$

where : $q_1 = \varrho^{ext}_{p_0}$, $q_2$=$(\varrho\hat{u})|_{p_0}$=0, $q_3$=$(\varrho\hat{v})|_{p_0}$=0, $q_4$=$\varrho^{ext}T_{wall}/\gamma$ and $l_{ij}$ denotes components of $\mathbf{L}$. Therefore, provided that $\mathbf{v}_{p_0}$ is defined, we simply use Eqs. (11) and (12) to extrapolate $\boldsymbol{v}_{gp}$. These quantities are then projected back into our local physical space as

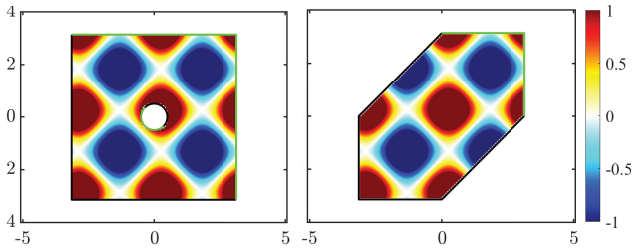
$$\hat{\boldsymbol{q}}_{gp} = \mathbf{R}\mathbf{v}_{gp}.$$

Lastly, we project back the vector-fields within $\hat{\boldsymbol{q}}_{gp}$ to arrive $\boldsymbol{q}_{gp}$ in $(x, y)$-coordinates. Following our initial example, this is simply done by doing :

$$\begin{bmatrix} u \\ v \end{bmatrix}_{gp} = \mathbf{T}^\intercal \begin{bmatrix} \hat{u} \\ \hat{v} \end{bmatrix}_{gp},$$

given the fact that $\mathbf{T}^{-1} = \mathbf{T}^\intercal$.

**Remark** : Provided that the normal associated to a given ghost-point is unique, then present method would always leads to a unique solution of $\boldsymbol{q}_{gp}$.



(a) Ω case (1) (b) Ω case (2)

FIGURE 4 – Numerical solution of the two-dimensional linear advection equation domains with embedded boundaries.

# 3 Numerical Experiments

## 3.1 2-d Advection Equation

This is a linear scalar problem proposed in [6]. Because it is a linear problem, it is very easy to 4th-order accuracy of the method. We skip-it to save space in this presentation. However, by solving for a longer time, we use it showcase the stability of the present boundary technique.

This problem is solved in two difference cases. In both cases, we solve for a 2-d linear advection equation problem of the form

$$q_t + a\,q_x + b\,q_y = 0, \quad \text{for } (x, y) \in [-3.5, 3.5]^2 \cap \Omega \qquad (15)$$

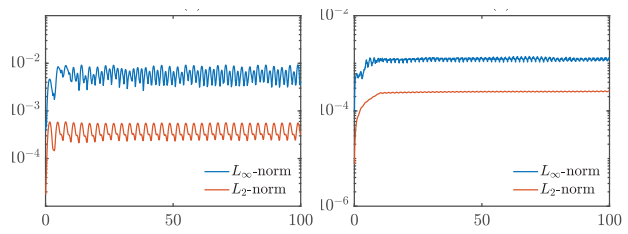
where $q$ is scalar quantity and $\Omega$, for case (1), is defined as $\Omega = \{(x, y) : \sqrt{x^2 + y^2} \leq 0.5 \cap -\pi \leq (x, y) \leq \pi\}$ as shown in Fig. 4a. For case (2), $\Omega$ is the polygonal region defined by the vertex points :

$$(x_i, y_i) = \{(-\pi, -\pi), (0, -\pi), (\pi, 0), (\pi, \pi), (-\pi, 0), (0, -\pi)\}, \qquad (16)$$

as illustrated in Fig. 4b. In both cases, it is assumed that $a = b = 1$. Therefore, $\partial\Omega$ is each case is composed of two types of boundary conditions, namely : time-dependent boundary conditions (black segments) and free-boundary conditions (green segments). The black segments and the initial condition, are here specified using the exact solution :

$$q^{exact}(x, y, t) = sin(2(x - a\,t) + sin(2(y - b\,t))). \qquad (17)$$

For case (1), is solved on 109×109 equidistant Cartesian grid. In case (2) a 105×105. This is deliberately done to




(a) Ω case (1) (b) Ω case (2)

FIGURE 5 – Illustration of the asymptotic stability of proposed IB technique in $L_2$ and $L_\infty$.

ensure that no boundary point coincides with any real grid point. The interpolation and extrapolation polynomials are set to be order k=3. The CFL number is set to $CFL = 0.9$ in both cases. Figures 5a and 5b show the evolution of the error, $\varepsilon = q - q^{exact}$, measured in $L_2$- and $L_\infty$-norm over time $t$=[0,100]. In both cases, accurate and stable time evolution are achieved. Moreover, unlike in [6], we didn't have to resource to a small CFL condition to obtain long term stable solutions.

## 3.2 Pulse diffraction by a cylinder using the 2-d Linearized Euler Equations

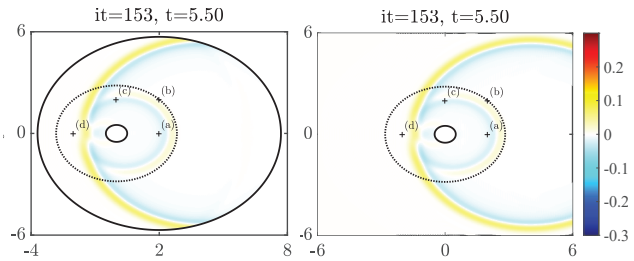



FIGURE 6 – Numerical solution of an acoustic pulse being reflected by a rigid cylinder using the 2-d LEE model.

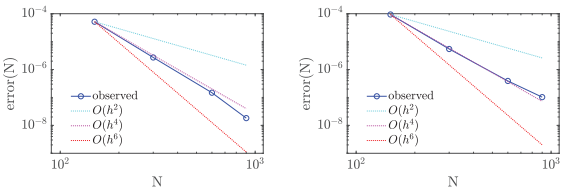
Pulse diffraction by a cylinder is a classical test in acoustics used for example by Seo & Mittal [1] to compute the order of accuracy of their IB-method for the linearized Euler equations (LEE). We use it here to compare and test for the accuracy and the stability of our method.

The full statement of the problem and exact solution of the pressure field, $\wp^{exact}$, can be found in [4]. Here, for the sake of space we limit our description to the most important aspects.

In this example we resolve the acoustic scattering of an Gaussian pulse by a circular cylinder of diameter $D$=1, in normalized units. Similarly to the previous experiment, this problem is solved for two cases : for case (1), $\partial\Omega_{outer}$ is designed as an embedded circle of radius 4.0 centered at (0,2), as depicted in Fig. (6a). Here, an outflow is constructed with characteristic variables. For case (2), the $\partial\Omega_{outer}$ is a perfectly square centered about the cylinder and the outflow boundary is imposed using Tam's radiation boundary condition [5]. In both cases, the cylinder becomes our $\partial\Omega_{inner}$ and its reflective boundary condition is set by imposing that the normal pressure gradient around be zero (as proposed in [1]).

Both cases are run using $\Delta x$=$\Delta y$=12/{150, 300, 600, 900} from time $0 < t \leq 10$ under a CFL=0.9 . At time 5.5, we compare the $\wp$-field within the circular region around the reflective cylinder, as depicted in Figs. 6a and 6b. The error, $\varepsilon = \wp - \wp^{exact}$ is captured and the $L_2$-norm values of each case reported in Figs. 7a and 7b.

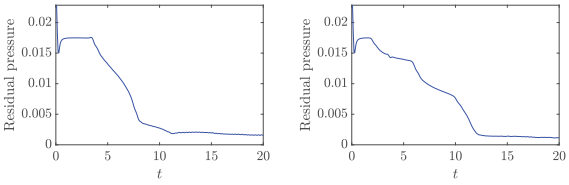
The $L_2$-norm indicate that a 4th-order accuracy has been achieved.




(a) Ω case (1) (b) Ω case (2)

FIGURE 7 – Observed order-of-accuracy for the numerical solution of a pulse reflected by a rigid cylinder.

To probe the stability of this problem, we choose the grid case $\Delta x$=$\Delta y$=150 and run this simulation up to time $t$=20, i.e. way far after the reflected waves have left the domain. See Figs. 8a and 8b. For both cases, we observe that after the waves leave the domain, the minimal value of the pressure field is reached and no artifact is reflected back into the domain.




(a) Ω case (1) (b) Ω case (2)

FIGURE 8 – Asymptotic behavior of the pressure residual in the domain.

In other words, we have observed that accurate and stable solutions using the LEE system can be achieved with the present boundary technique. Moreover, unlike the solution presented in [1], no degradation of the OOA is observed after the wave is reflected, and no selective (low-pass) filter is used.

## 3.3 Flow around a cylinder at Re=20 using the 2-d Navier-Stokes Equations

We aim to use our IB-method to simulate directly the formation of acoustic waves occurring on compressible-viscous flows. To do that, we develop a 2-d numerical solver for the compressible Navier-Stokes [13] using a normalization based on free-stream values.

To validate our solver, we choose to resolve the classical solution of a flow around a infinitely long circular cylinder placed in a uniform free-stream. Here, in particular, we study the a flow for a Reynolds number of $Re$=$\frac{\varrho_\infty u_\infty D}{\mu}$=20, which corresponds to a general convection-diffusion case. A free-stream Mach number of $Mach$=$\frac{u_\infty}{c_0}$=0.03 is set in order be able to compare with numerical solutions of incompressible N-S solutions.

The Cylinder is placed at the origin of the domain, so that the distance from the cylinder to the lower and upper

free-stream boundaries is set to $13D$; and the inflow and outflow are located at $5D$ and $15D$ respectively. This problem is resolved on a stretched mesh whose grid spacing in the vicinity of the cylinder is approximately $\Delta x{=}\Delta y{=}D/40$.

In this example, the domain's inlet, outlet and the isothermal-wall boundary condition on the cylinder are imposed using the *characteristic extrapolation* described in section 2.3.2. We run our simulation with a CFL=0.9, which corresponds to a time step of size $\Delta t{=}0.004$. The Simulation is run until $t{=}1000$. The Vortex field and the eddy obtained from this configuration is shown in Fig. 9. At the final time step, the drag coefficient, $C_D$, is measured by interpolating the values of the surface pressure and viscous stress on the cylinder's body. At this grid resolution, it is found that $C_D{=}2.12$. Comparison with the results summarized in [14, Table 1] indicates that the present numerical result is in reasonable agreement with previously reported results.

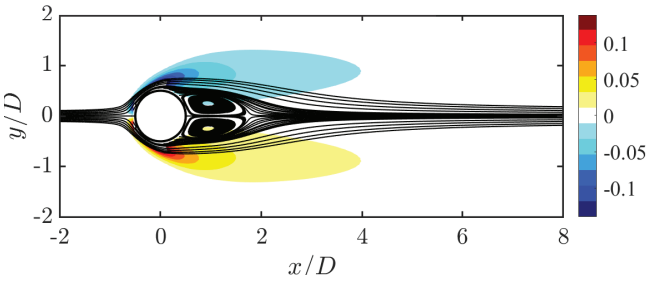



FIGURE 9 – Vorticity field obtained from the flow over a cylinder at Mach 0.03 with a Reynolds number of 20.

## 4 Conclusion

A new immersed boundary technique for solving hyperbolic and parabolic problems has been presented. This technique is found to yield a sharp and stable 4th-order finite-difference numerical approach that does not rely on artifacts such as filters, WENO interpolation strategies or jump correction techniques.

As the present IB-technique relies on traditional interpolation approaches, we believe its extensions into 3-d problems is fairly straight forward, and its adoption in existing numerical solvers based on compact schemes could be transparent.

Lastly, it is hoped that this new technique would enable the study of aeroacoustics wave generation and propagation with 4th-order resolution on vectorized CPUs and GPUs. Extensions for even higher orders-of-accuracy are currently under investigation.

## Acknowledgements

We are indebted to the CPER Feder, programme Transport, for financial support. The first author wishes to express his gratitude to Dr. Oleg Kravchenko and Prof. Hugo Alfredo Carrillo Serrano for fruitful discussions on central finite-differences methods.

## Références

[1] J.-H. Seo and R. Mittal, A high-order immersed boundary method for acoustic wave scattering and low-Mach number flow-induced sound in complex geometries. *Journal of computational physics*, **230.4**, 1000-1019 (2011).

[2] Y. Fukushima, T. Misaka, S. Obayashi, D. Sasaki, K. Nakahashi, Wavenumber Optimized Immersed Boundary Method for Aeroacoustic Analysis Based on Cartesian Mesh, AIAA Journal, **54**, 2988 :3001 (2016).

[3] M. E. Khalili, M. Larsson and B. Müller, High-order ghost-point immersed boundary method for viscous compressible flows based on summation-by-parts operators. *International Journal for Numerical Methods in Fluids*, **89.7**, 256-282 (2019).

[4] C. K. Tam, & J. C. Hardin, Second computational aeroacoustics (CAA) workshop on benchmark problems. *Second Computational Aeroacoustics (CAA)*, Workshop on Benchmark Problems, No. NAS 1.55 : 3352 (1997).

[5] C. K. Tam, and J. C. Webb. Dispersion-relation-preserving finite difference schemes for computational acoustics. *Journal of computational physics*, **107.2**, 262-281 (1993).

[6] S. Hosseinverdi, and H. F. Fasel, Very high-order accurate sharp immersed interface method : application to direct numerical simulations of incompressible flows. *23rd AIAA Computational Fluid Dynamics Conference*, 2017-3624 (2017).

[7] B. Sjogreen and N. A. Petersson. A Cartesian embedded boundary method for hyperbolic conservation laws. *Communications in Computational Physics*, **2.6**, 1199-1219 2, no. UCRL-JRNL-226709 (2006).

[8] P. T. Brady and D. Livescu, Foundations for high-order, conservative cut-cell methods : Stable discretizations on degenerate meshes. *Journal of Computational Physics*, **426**, 109794 (2021).

[9] J. Fernández-Fidalgo, S. Clain, L. Ramírez, I. Colominas and X, Nogueira, Very high-order method on immersed curved domains for finite difference schemes with regular Cartesian grids. *Computer Methods in Applied Mechanics and Engineering*, **360**, 112782 (2020).

[10] A. Wiegmann and K. P. Bube, The explicit-jump immersed interface method : finite difference methods for PDEs with piecewise smooth solutions. *SIAM Journal on Numerical Analysis* textbf37.3, 827-862 (2000).

[11] J. Lu, J. Fang, S.Tan, C.-W. Shu and M. Zhang, Inverse Lax–Wendroff procedure for numerical boundary conditions of convection–diffusion equations. *Journal of Computational Physics*, **317**, 276-300 (2016).

[12] J. Lu, C.-W. Shu, S. Tan and M. Zhang, An inverse Lax-Wendroff procedure for hyperbolic conservation laws with changing wind direction on the boundary. *Journal of Computational Physics*, **426**, 109940 (2021).

[13] E. F. Toro, Riemann solvers and numerical methods for fluid dynamics : a practical introduction. *Springer Science & Business Media*, (2013).

[14] F. De Vanna, F. Picano, and E. Benini. A sharp-interface immersed boundary method for moving objects in compressible viscous flows. *Computers & Fluids*, **201**, 104415 (2020).