Fast immersed interface Poisson solver for 3D unbounded problems around arbitrary geometries

T. Gillis[a,*], G. Winckelmans[a], P. Chatelain[a]

[a]*Institute of Mechanics, Materials and Civil Engineering, Université catholique de Louvain 1348 Louvain-la-Neuve, Belgium*

Abstract

We present a fast and efficient Fourier-based solver for the Poisson problem around an arbitrary geometry in an unbounded 3D domain. This solver merges two rewarding approaches, the lattice Green's function method and the immersed interface method, using the Sherman-Morrison-Woodbury decomposition formula. The method is intended to be second order up to the boundary. This is verified on two potential flow benchmarks. We also further analyse the iterative process and the convergence behavior of the proposed algorithm. The method is applicable to a wide range of problems involving a Poisson equation around inner bodies, which goes well beyond the present validation on potential flows.

Keywords: Lattice Green's function, Immersed interface method, Unbounded 3D Poisson equation, Sherman-Morrison-Woodbury formula

1. Introduction

Solving the Poisson equation is ubiquitous in many computational disciplines such as electromagnetism, fluid mechanics and theoretical physics. Therefore, the problem has been deeply studied and is a recurrent subject within the literature. Still, two major challenges remain: solving the problem in an unbounded framework, and taking into account the presence of a body within the computational domain. These two remaining challenges are widespread and often coupled in many applications, such as computational fluid dynamics applied to the simulation of flows past bodies, like blades and wings.

One may solve the first challenge, i.e. computing unbounded solutions, using two classes of methods. The first kind, based on James [1] and Lackner [2] ideas, solves two problems to determine coherent outer boundary conditions, as proposed by McCorquodale et al. [3], Marichal et al. [4, 5]. In this work, we use a second type of method based on the convolution of a kernel function. This convolution is performed using fast summation algorithms: Fast Multipole Methods (FMM) [6–11]), Fourier transforms (Hockney-Eastwood technique) [12–15], or a combination of both [16–18] (see [19] for a comparison between the FMM and FFT-based methods). The kernel function is either built through a regularization of the Green's function [11, 13, 14], or by using the lattice Green's function [9, 10, 16–18, 20].

The second challenge, i.e. taking an immersed body and the discontinuities at its interface into account, can be solved using several methods. We will not attempt to provide an exhaustive review of all of them; instead, we highlight some of the prominent techniques. One of the earliest approach is known as the capacitance matrix method. For example, Proskurowski and Widlund [21] and O'Leary and Widlund [22] embed the irregular domain

*Corresponding author
Email address: thomas.gillis@uclouvain.be (Thomas Gillis)

of interest into a regular one and impose the boundary condition using single layer and double layer potentials. The Helmholtz equation is then solved using iterative methods and algebraic decompositions, like the Sherman-Morrison-Woodbury formula (also known as the Woodbury formula), as proposed earlier in a similar context by George [23] and Buzbee et al. [24]. Over the same period of time, Peskin [25] introduced the Immersed Boundary (IB) method, still widely used and renowned for its robustness [18, 26, 27]. It discretizes the additional sources due to the body on several grid points using discrete delta functions, thus mollifying the body interface. Some recent developments propose a harmonic extension in the body to allow the use of a symmetric discrete delta function and therefore increase the order of convergence of the method [28]. However, this harmonic extension requires additional Poisson equation resolutions. Let us also mention the works based on the potential layer theory from Mayo [29], McKenney et al. [30], Mayo and Greenbaum [31] and a recent contribution from Askham and Cerfon [32], which is also based on the potential layer method, but reuses the harmonic expansion proposed earlier in the contexts of the IB technique [28]. In a similar way, the Boundary Element Methods (BEM) [33, 34], relies on panels and FMM acceleration to take the inner obstacle into account. They solve the integral equation associated to the panels and then compute a new Poisson equation including the additional source terms. Finally, let us present the Immersed Interface Method (IIM) [4, 35–38], which uses modified Taylor series to take the inner body into account. The IIM offers a proven high order and sharp treatment of boundaries, does not require the introduction of a mollification or any extension in the body and the introduced perturbation has a tiny support around the boundary. Originally developed in a multidimensional way [35] but subject to stability issues, it has been modified to work with one-dimensional dimensions splitting [36, 37] and recently applied to 2D problems [4, 36].

However, within a regular grid framework, coupling a fast summation based solver and immersed bodies still remains an open challenge. Apart from the fast BEM methods which do not take advantage of the structured grid, none of the above techniques achieves a fast, accurate and grid-based solution of the unbounded Poisson problem around a body. For example, the Miller iteration [39], applied by Marichal et al. [4] to the IIM framework, needs to compute the outer boundary conditions iteratively, together with the inner boundary conditions: this tends to increase dramatically the computational cost. Furthermore, recent fast 3D Poisson solvers, like the FMM-based lattice Green's function convolution [16] or the FTT-based regularized kernel convolution [14], were never coupled with the immersed interface method.

In this work, we consider the inner body as a local perturbation of an unbounded body-free Poisson problem. The Sherman-Morrison-Woodbury (SMW) formula [40, p. 50] efficiently couples the Immersed Interface Method (IIM), used to impose the boundary condition, with a fast Poisson solver. This approach draws inspiration from existing works, such as the capacitance matrix method applied with a direct solver [23, 24] or the work on hybrid grid-particle methods and penalization performed by Chatelin and Poncet [41]. To be consistent in the coupling, we use lattice Green's function (LGF) kernel [16], equivalent to inverting the standard 3-points Laplacian stencil considered in the IIM.

In this paper, we first present the three different technologies (LGF, IIM and SMW) and further explain how to couple them together (Section 2). Secondly, we investigate the rate of convergence on two benchmarked potential flow applications and validate its second order behavior (Section 3). Finally, we close this work by a short conclusion and perspectives for future work.

2. Methodology

We consider the unbounded 3D scalar Poisson problem around a body, Ω_b:

$$\nabla^2 u\left(\mathbf{x}\right) = f\left(\mathbf{x}\right), \quad \mathrm{supp}\left(f\right) \in \Omega \setminus \Omega_b \ , \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^3$, Ω is the computational domain which encapsulates the body and the support of the source term, $f\left(\mathbf{x}\right)$, and the unbounded solution, $u\left(\mathbf{x}\right)$, decays at least as $1/\left|\mathbf{x}\right|$ at infinity. Without loss of generality we assume a Neumann boundary condition on $\partial\Omega_b$. Using a standard 3-point grid discretization for the Laplacian operator, ∇_h^2, we aim at a second order solver, which uses IIM in order to take the inner boundary condition into account.

We decompose the problem (1) into two sub-problems. The first one is the body-free problem, expressed by the unbounded elliptic operator, ∇_h^2. The second one is the boundary problem, the perturbation of this operator by the body contribution, expressed as an additional source term, f_b. In this section we successively address these two problems and the method used to couple them.

2.1. A second order unbounded, body-free, 3D Poisson solver

The straightforward inversion of ∇_h^2 operator is challenging and computationally intensive [1, 2, 39, 42]. For the sake of efficiency, we rather follow an approach of grid-based fast convolution of the Green's function with the source term. Because the IIM tools, discussed in Section 2.2, consist in finite differences modifications, we use the Green's function associated with the standard second order cross stencil ($3 \times 3 \times 3$). As proposed by Liska and Colonius [16], the body-free Poisson equation, expressed in wave space, $(k_x \ ; \ k_y \ ; \ k_z) \in [-\pi/h_x \ ; \ \pi/h_x] \times [-\pi/h_y \ ; \ \pi/h_y] \times [-\pi/h_z \ ; \ \pi/h_z]$, becomes

$$\frac{1}{h^2}\left[4\sin^2\left(k_x h_x/2\right) + 4\sin^2\left(k_y h_y/2\right) + 4\sin^2\left(k_z h_z/2\right)\right]\hat{u} \triangleq \sigma\left(\mathbf{k}h\right)\hat{u} = \hat{f} \ , \tag{2}$$

where $\hat{u}$ and $\hat{f}$ are the Fourier transforms of u and f. In this work, we consider a structured grid with $h = h_x = h_y = h_z$. The associated lattice Green's function (LGF) is given by the unbounded inverse Fourier transform of $1/\left(\sigma\left(\mathbf{k}h\right)\right)$ [16]:

$$\Theta\left(\mathbf{x}_m\right) = \frac{1}{\left(2\pi\right)^3}\iiint_{-\pi/h}^{\pi/h} \frac{h^2 \quad \exp\left[-\mathrm{i}\ \mathbf{x}_m \cdot \mathbf{k}\right]}{4\sin^2\left(k_x h/2\right) + 4\sin^2\left(k_y h/2\right) + 4\sin^2\left(k_z h/2\right)} \quad d\mathbf{k} \triangleq \frac{1}{h}\Theta\left(\mathbf{m}\right) \ , \tag{3}$$

where $\mathbf{m} = \dfrac{\mathbf{x}}{h} = \{m_x \ ; \ m_y \ ; \ m_z\}$ and

$$\Theta\left(\mathbf{m}\right) = \frac{1}{\left(2\pi\right)^3}\iiint_{-\pi}^{\pi} \frac{\exp\left[-\mathrm{i}\ \mathbf{m} \cdot \boldsymbol{\xi}\right]}{4\sin^2\left(\xi_x/2\right) + 4\sin^2\left(\xi_y/2\right) + 4\sin^2\left(\xi_z/2\right)} \quad d\boldsymbol{\xi} \ , \tag{4}$$

with $\boldsymbol{\xi} = \mathbf{k}h$. For small $\left|\mathbf{m}\right|$, we evaluate this integral, while for $\left|\mathbf{m}\right|$ sufficiently large, Eq. (4) is replaced by the following expansion

$$\Theta\left(\mathbf{m}\right) = -\frac{1}{4\pi\left|\mathbf{m}\right|} - \frac{\left[m_1^4 + m_2^4 + m_3^4 - 3m_1^2 m_2^2 - 3m_1^2 m_3^2 - 3m_2^2 m_3^2\right]}{16\pi\left|\mathbf{m}\right|^7} + \ldots + \mathcal{O}\left(1/\left|\mathbf{m}\right|^9\right) \ . \tag{5}$$

A more detailed explanation of the way to compute Eq. (4) and the expansion, is given in Appendix A. Using this LGF, we express the application of the inverse of ∇_h^2 as

$$u = \left[\nabla_h^{-2} f\right]_{i\,j} = \sum_{k\,l} \Theta\left(\mathbf{x}_{k\,l} - \mathbf{x}_{i\,j}\right) f\left(\mathbf{x}_{k\,l}\right) \quad . \tag{6}$$

We use a Hockney-Eastwood FFT-based fast convolution technique [12, 13] to evaluate the unbounded convolution, since it requires the evaluation of the LGF only once on the domain (it only depends of the size of the domain, see Appendix A). We note that a FMM-based approach could also have been used, as proposed in [16].

2.2. A second order immersed interface method

We here follow the approach and notations of Linnick and Fasel [36] and Marichal et al. [4, 5]. Let us first consider a one-dimensional function $u(x) = u^{(0)}(x) \in \mathcal{C}^\infty(\mathbb{R} \setminus \{x_\alpha\})$, as shown on Fig. 1. This function $u(x)$ and its derivatives, $u^{(\cdot)}(x)$, may be discontinous at the interface, $x_\alpha \in [x_i\,;\,x_{i+1}[$. Using a generalized Taylor series, we write a modified finite difference scheme at x_i and x_{i+1} for the second derivative of u, $u^{(2)}$, still valid across the interface (x_α)

$$u^{(2)}(x_i) = \frac{1}{h^2} u(x_{i-1}) - \frac{2}{h^2} u(x_i) + \frac{1}{h^2}\left[u(x_{i+1}) - J_\alpha^+\right] + \mathcal{O}\left(h^2\right) \tag{7}$$

$$u^{(2)}(x_{i+1}) = \frac{1}{h^2}\left[u(x_i) + J_\alpha^-\right] - \frac{2}{h^2} u(x_{i+1}) + \frac{1}{h^2} u(x_{i+2}) + \mathcal{O}\left(h^2\right) \tag{8}$$

where

$$J_\alpha^+ = \left[u^{(0)}\right]_\alpha + \frac{h^+}{1!}\left[u^{(1)}\right]_\alpha + \frac{\left(h^+\right)^2}{2!}\left[u^{(2)}\right]_\alpha + \frac{\left(h^+\right)^3}{3!}\left[u^{(3)}\right]_\alpha \tag{9}$$

$$J_\alpha^- = \left[u^{(0)}\right]_\alpha + \frac{h^-}{1!}\left[u^{(1)}\right]_\alpha + \frac{\left(h^-\right)^2}{2!}\left[u^{(2)}\right]_\alpha + \frac{\left(h^-\right)^3}{3!}\left[u^{(3)}\right]_\alpha \tag{10}$$

with h^+ positive, h^- negative and

$$\left[u^{(k)}\right]_\alpha = u^{(k)}(x_{\alpha^+}) - u^{(k)}(x_{\alpha^-}) \quad . \tag{11}$$

Without loss of generality, let us assume that $x < x_\alpha$ represents Ω_b (see Fig. 1). Inside Ω_b, all derivatives are supposed known since we know the desired solution inside the obstacle; for instance a null field as in Fig. 1. Outside Ω_b, we use the boundary condition (Dirichlet or Neumann) and one-sided interpolation schemes to compute the derivatives $u^{(k)}(x_{\alpha^+})$,

$$u^{(k)}(x_{\alpha^+}) = S_\alpha^k u_\alpha + S_1^k u(x_{i+l+1}) + S_2^k u(x_{i+l+2}) + S_3^k u(x_{i+l+3}) + \mathcal{O}\left(h^{4-k}\right) \quad . \tag{12}$$

which is in agreement with the global accuracy needed by the corrected scheme [4, 36]. The parameter l can, in principle, be chosen arbitrary, however $l = 1$ is used to obtain a well-conditioned system [4, 5]. In the following, we call the points i and $i+1$ discontinuity-affected points, and the points involved in the interpolation stencil, primary

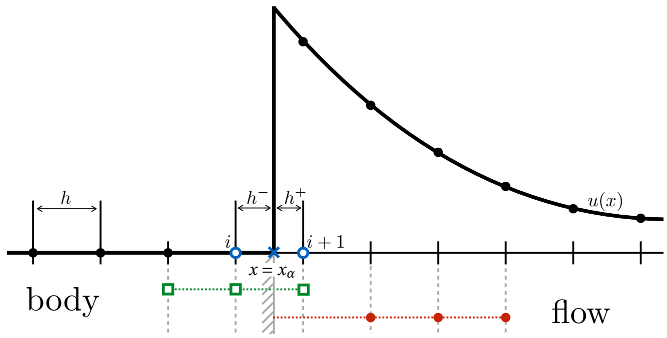

Figure 1: 1D Immersed interface scheme: the standard 3-point stencil (green) crosses the interface and generates a control point (blue cross), discontinuity affected points (blue circles) and primary interpolation points (red dots)

interpolation points. The correction brought by the IIM is then

$$\nabla_h^2 u - \frac{1}{h^2} J_\alpha\left(u\right) = f \qquad \Leftrightarrow \qquad \nabla_h^2 u = f + f_b\left(u\right) \quad , \tag{13}$$

where u is the unknown field, f is the source field and the jump coefficient, $J_\alpha = J_\alpha^+ + J_\alpha^+$, depends on the solution itself, u and can be interpreted as an additional source term, $f_b\left(u\right)$, for the body-free equation. From an operators perspective, one can also identify the IIM as a low-rank perturbation of the body-free Laplacian $\left(\nabla_h^2 \cdot\right)$, since it only affects mesh points in the vicinity of the body interface.

In three dimensions, we follow the dimension splitting approach [4, 5, 36] as operators, such as ∇_h^2, can be handled by correcting the scheme along each direction. It follows that, since the Neumann boundary condition, $\frac{\partial u}{\partial \hat{\mathbf{n}}}\big|_\alpha$, is given along the outgoing normal, the value of the first derivative in the current direction, at the boundary, $u_\alpha^{(1)}$, is not readily available from the boundary condition. To retrieve this value, we use the compatible interpolation scheme, proposed by Marichal et al. [5], and extend it to 3 dimensions.

For the sake of clarity, let us assume a local system of axes, defined at the intersection between the grid line and the body's boundary, $\mathbf{x} = \mathbf{x}_\alpha$, as shown in 2D on Fig. 2. We name the current direction, ξ and the 2 transverse directions, η_1 and η_2. The values at grid points aligned with the current direction is given by $u_{\xi,p}$, while values aligned with the transverse directions, $u_{\eta_i,p}$, are not necessarily associated with grid points. The compatible interpolation scheme consists of solving the following interpolation system, expressed at the intersection,

$$\left\{\begin{array}{lll} u_\alpha & = S_\alpha^{\xi,0} \frac{\partial u}{\partial \xi}\big|_\alpha + \sum_{p=1}^{3} S_p^{\xi,0} u_{\xi,l_\xi+p} & \triangleq S_\alpha^{\xi,0} \frac{\partial u}{\partial \xi}\big|_\alpha + S^{\xi,0} \\ u_\alpha & = S_\alpha^{\eta_1,0} \frac{\partial u}{\partial \eta_1}\big|_\alpha + \sum_{p=1}^{3} S_p^{\eta_1,0} u_{\eta_1,l_{\eta_1}+p} & \triangleq S_\alpha^{\eta_1,0} \frac{\partial u}{\partial \eta_1}\big|_\alpha + S^{\eta_1,0} \\ u_\alpha & = S_\alpha^{\eta_2,0} \frac{\partial u}{\partial \eta_2}\big|_\alpha + \sum_{p=1}^{3} S_p^{\eta_2,0} u_{\eta_2,l_{\eta_2}+p} & \triangleq S_\alpha^{\eta_2,0} \frac{\partial u}{\partial \eta_2}\big|_\alpha + S^{\eta_2,0} \\ \frac{\partial u}{\partial \hat{\mathbf{n}}} & = \hat{n}_\xi \frac{\partial u}{\partial \xi} + \hat{n}_{\eta_1} \frac{\partial u}{\partial \eta_1} + \hat{n}_{\eta_2} \frac{\partial u}{\partial \eta_2} & \end{array}\right. , \tag{14}$$

where $S_\cdot^{\xi,0}$, $S_\cdot^{\eta_1,0}$ and $S_\cdot^{\eta_2,0}$ are the interpolation coefficient of Eq. (12) associated to the direction ξ, η_1, η_2, respectively. For conditioning reasons, we use $l_\xi = 1$, $l_{\eta_1} = 0$ and $l_{\eta_2} = 0$, as explained in [5]. Solving this system leads to an expression for $\frac{\partial u}{\partial \xi}$

$$\frac{\partial u}{\partial \xi} = \frac{1}{\mathcal{D}} \left[S_\alpha^{\eta_1,0} S_\alpha^{\eta_2,0} \frac{\partial u}{\partial \hat{\mathbf{n}}} - \left(\hat{n}_{\eta_1} S_\alpha^{\eta_2,0} + \hat{n}_{\eta_2} S_\alpha^{\eta_1,0} \right) S^{\xi,0} + \hat{n}_{\eta_1} S_\alpha^{\eta_2,0} S^{\eta_1,0} + \hat{n}_{\eta_2} S_\alpha^{\eta_1,0} S^{\eta_2,0} \right] \quad , \tag{15}$$

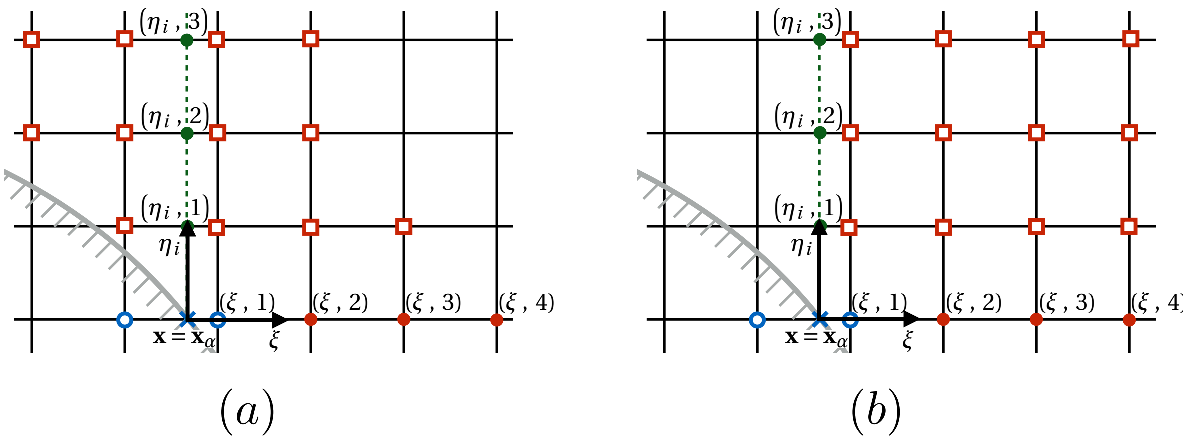

Figure 2: 2D immersed interface compatible scheme: Primary interpolation points (● and ●) and secondary interpolation points (□), for the two approaches (a) and (b). In this work , we use the (a) approach.

where $\mathcal{D}$ is the determinant of the system,

$$\mathcal{D} = \hat{n}_\xi S_\alpha^{\eta_1,0} S_\alpha^{\eta_2,0} + \hat{n}_{\eta_1} S_\alpha^{\xi,0} S_\alpha^{\eta_2,0} + \hat{n}_{\eta_2} S_\alpha^{\xi,0} S_\alpha^{\eta_1,0} \quad . \tag{16}$$

The values aligned with the transverse directions, $u_{\eta_1,p}$ and $u_{\eta_2,p}$, are retrieved by interpolation, using secondary interpolation points (see Fig. 2). The choice of the secondary interpolation points influences the error magnitude and one may consider two approaches. The first one centers the interpolation as much as possible, in order to reduce the error coefficient. The second one selects points away from the wall (where high curvature is likely to be present). This decreases the fourth derivative of the solution and therefore the error itself. In this work, we use the first approach in order to decrease the error coefficient as much as possible.

2.3. The combined problem

We have shown above that the IIM scheme, that accounts for inner boundaries, can be considered as a low rank perturbation of the discretized second order Laplacian operator (see Section 2.2), ∇_h^2. We propose to couple the body-free problem and the IIM perturbation through the Sherman-Morrison-Woodbury (SMW) formula, that precisely allows to exploit the low-dimensionality of the perturbation. The SMW formula [40, 43] expresses the inverse of a low-rank perturbed matrix as the exact following decomposition

$$[A + UV]^{-1} = A^{-1} - A^{-1}U\left[I + VA^{-1}U\right]^{-1} VA^{-1} \quad . \tag{17}$$

The main problem is the body-free unbounded Poisson problem, $A = \nabla_h^2$, while the low rank perturbation, UV, is matched to several steps involved in the IIM technique: $U = E_a C$ and $V = E_i^T$. $E_a \in \mathbb{R}^{n \times k_a}$ is a mapping matrix from the small subspace of discontinuity-affected points (the external vicinity of the body, i.e. the k_a nodes concerned by the IIM corrections) to the global domain of size $n = N_x \times N_y \times N_z$. $E_i \in \mathbb{R}^{n \times k_i}$ is a mapping matrix from the primary and secondary interpolation points to the global space and $C \in \mathbb{R}^{k_a \times k_i}$ is the matrix that transforms the near-wall values into IIM corrections, i.e. the Eqs. (13), (14) and (15) cast as a linear operator.

Applied to our problem, the SMW formula then yields

$$u = \left(\nabla_h^2 + E_a C E_i^T\right)^{-1} f = \nabla_h^{-2} \left\{ I_n - E_a C \left[I_i + E_i^T \nabla_h^{-2} E_a C \right]^{-1} E_i^T \nabla_h^{-2} \right\} f \ , \tag{18}$$

where $I_n \in \mathbb{R}^{n \times n}$ and $I_i \in \mathbb{R}^{k_i \times k_i}$. This allows to identify the boundary problem (and its solution, x_b) as the low-rank system

$$\left[I_i + E_i^T \nabla_h^{-2} E_a C \right] x_b = E_i^T \nabla_h^{-2} f \ , \tag{19}$$

which is solved to compute the additional source term due to the body, $f_b = E_a C x_b$. A GMRes algorithm [44] is typically used to obtain a solution, since it is a non-symmetric problem. Indeed, the IIM operator generates sources, using the interpolation data, at discontinuity-affected points and not at interpolation points.

The SMW decomposition can be exploited towards computational efficiency. Indeed, the resolution of the boundary problem, restricted to the primary and secondary interpolation points, is more affordable and faster to perform than a resolution of the global problem. Nevertheless, it still involves the body-free problem, which is solved using a fast convolution solver (see Section 2.1). Rather than using the full-field solver (FFT) used in this work, one might use a sparse solver (FMM) for additional efficiency (thus avoiding the computation of the unused locations). Finally, it is interesting to observe that, unlike many other methods, the SMW formula works from the perspective of the solution rather than that of the source.

To perform the parallel computations, we used the "PPM" library [45, 46], which provides the MPI communications between subdomains: data movement functions for the body-free FFT based Poisson solver [13], particle to mesh interpolation and ghost particles required by the IIM interpolation.

3. Validation: 3D potential flows past a body

This section validates our approach and assesses its performance on the solution of the potential flow past a body, Ω_b. As in [47], we use the Helmholtz decomposition of the velocity,

$$\mathbf{u} = \mathbf{U}_\infty + \nabla \times \boldsymbol{\psi} + \nabla \phi \qquad \text{with} \qquad \nabla \cdot \boldsymbol{\psi} = 0 \ , \tag{20}$$

where $\mathbf{U}_\infty = U_\infty \mathbf{e}_z$ represents the free-stream velocity, $\boldsymbol{\psi}$ is the stream function of the velocity induced by the bulk vorticity ($\nabla^2 \boldsymbol{\psi} = -\boldsymbol{\omega}$) and ϕ represents the potential required to satisfy the boundary condition $\mathbf{u} \cdot \hat{\mathbf{n}} = 0$ on $\partial\Omega_b$, where $\hat{\mathbf{n}}$ is the outgoing normal to $\partial\Omega_b$.

To compute the velocity field, one first computes the bulk stream function, neglecting the presence of the body, by solving $\nabla^2 \boldsymbol{\psi} = -\boldsymbol{\omega}$ in the whole computational domain. Then, the body is taken into account by solving the second problem,

$$\nabla^2 \phi = 0 \qquad \frac{\partial \phi}{\partial \hat{\mathbf{n}}} = -\left[\mathbf{U}_\infty + \nabla \times \boldsymbol{\psi} \right] \cdot \hat{\mathbf{n}} \text{ on } \partial\Omega_b \ . \tag{21}$$

In the context of the present potential flows validation, the bulk vorticity is nil, hence $\boldsymbol{\psi} = 0$, without loss of generality. Therefore, we are left with the following problem

$$\nabla^2 \phi = 0 \qquad \text{with} \qquad \frac{\partial \phi}{\partial \hat{\mathbf{n}}} = -\left[\mathbf{U}_\infty\right] \cdot \hat{\mathbf{n}} \text{ on } \partial\Omega_b \ . \tag{22}$$

For our application, we define the local residual for the boundary problem, Eq. (19), as

$$r_b = E_i^T \nabla_h^{-2} f_\infty - \left[I_i + E_i^T \nabla_h^{-2} E_b C\right] x_b \ , \tag{23}$$

where x_b is the solution of the boundary problem, whose support is limited to the vicinity of the body, i.e. the primary and secondary interpolation points. The global residual is then taken, in dimensionless form, as

$$\mathcal{R} = \frac{1}{U_\infty L^{5/2}} \sqrt{\sum_{\mathcal{I}_b} \left|r_b\right|^2 h^3} \ , \tag{24}$$

with L the length of the computational domain and h the spatial step of the mesh. The tolerance on $\mathcal{R}$ is chosen to be 10^{-14} in order to avoid any influence of a non-fully converged solution of the boundary problem.

Furthermore, we define the local error as $e\left(\mathbf{x}\right) = \phi\left(\mathbf{x}\right) - \phi_{exact}\left(\mathbf{x}\right)$, in the bulk, and the associated dimensionless 2-norm and infinite norm, respectively as,

$$E_2 = \frac{1}{U_\infty L^{5/2}} \sqrt{\sum_{\Omega \backslash \Omega_b} e^2\left(\mathbf{x}\right)\ h^3} \qquad \text{and} \qquad E_\infty = \frac{1}{U_\infty L} \max_{\Omega \backslash \Omega_b} \left|e\left(\mathbf{x}\right)\right| \ . \tag{25}$$

We finally define the local convergence orders, r_∞ and r_2 as

$$\left[r_q\right]_i = \log\left(\left[E_q\right]_i / \left[E_q\right]_{i-1}\right) / \log\left(N_{i-1}/N_i\right) \ , \tag{26}$$

with $q = \infty$ and $q = 2$, respectively and i, the resolution level (here, $h_i = \dfrac{L}{N_i}$ and $N_{i+1} = 2N_i$).

3.1. Potential flow past a sphere

We first consider the problem of the potential flow past a sphere. The analytical solution of this flow is given by

$$\phi_{exact} = U_\infty\ z\ \frac{R^3}{2\left|\mathbf{x}\right|^3} \ , \tag{27}$$

where R is the radius. We use a box $[L\ \times\ L\ \times\ L]$, with $L = 2.5\ R$ and N grid points in each direction (in total N^3 unknowns). The global and local convergence slopes are given in Table 1 and Fig. 3, while the solution is visualized in Fig. 4 in terms of streamlines, i.e. tangent lines to $\mathbf{U}_\infty + \nabla\phi$.

Our method achieves the expected second order convergence for both the 2-norm and the infinite norm. We stress that the velocity inside the body remains nil, as imposed by the IIM, without any a posteriori modification. Therefore, there is no need to cancel it and no spurious flow appears across the boundary. Fig. 3 also shows the number of iterations performed while solving the boundary problem using the GMRes algorithm. We observe that increasing the number of points does not require many additional iterations since we already resolve the body

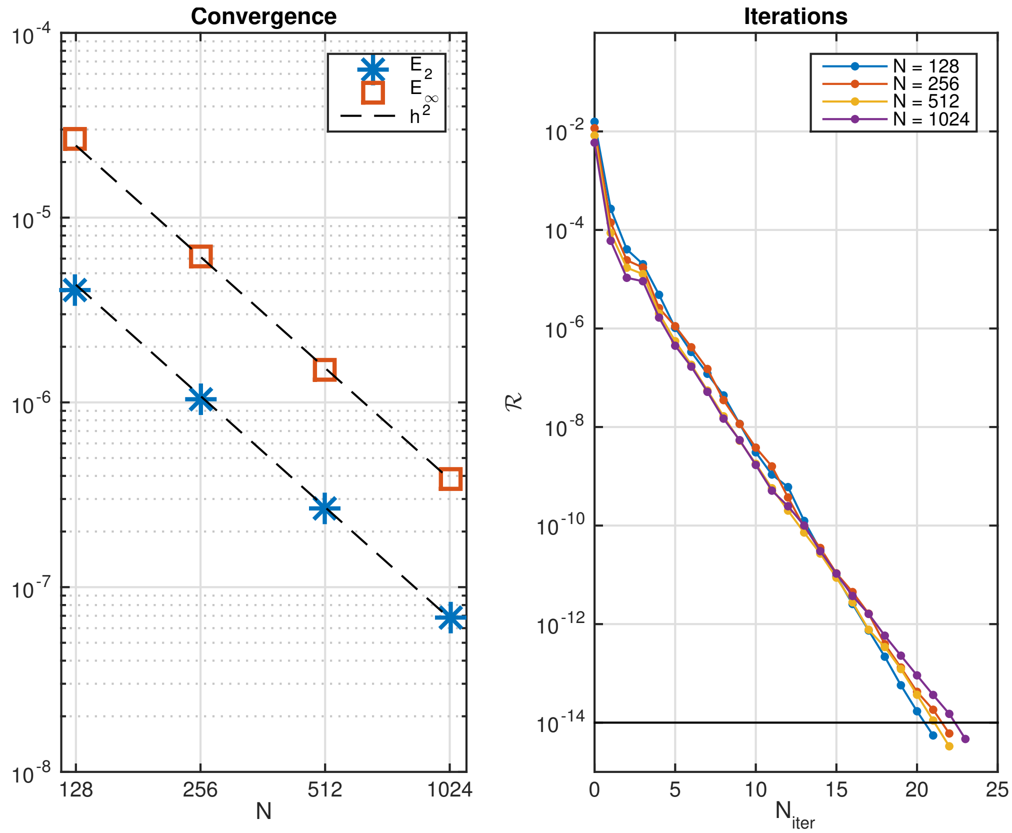

Figure 3: Potential flow past a sphere: Grid convergence of the solution to the analytic expression and number of iteration needed to solve the boundary problem, Eq. (19), up to the chosen tolerance, 10^{-14}.

boundary accurately, even with $N = 128$ ($R/h = [51.2\ ,\ 102.4\ ,\ 204.8\ ,\ 409.6]$ for $N = [128\ ,\ 256\ ,\ 512\ ,\ 1024]$, respectively).

Fig. 5 shows the distribution of the time spent in each part of the key steps: initialization, Poisson equation and other operations (communication and memory movements). We observe that we spend a non-negligible time to initialize the solver. This time is used to detect the intersections with the level set function and pre-compute the interpolation coefficients, i.e. the C matrix from Eq. (18). The remaining time is mostly spent in the body-free Poisson solver, which implies that the time spent in the total algorithm is governed by the choice of this solver. One also notices on Fig. 5 that the percentage of time spent in communication is decreasing with the size of the problem. This is the expected behavior of such a method when one subjects it to a strong scalability test. We use a constant number of CPU's (1024 Intel Haswell's) and increase the computational domain size: each processing entity thus has a decreasing communication load, in relation to the computational work.

N	E_∞	r_∞	E_2	r_2	t_i [s]	t_{bf} [s]	t_o [s]
128	$2.68e-05$		$4.07e-06$				
256	$6.22e-06$	2.11	$1.04e-06$	1.96	0.83	4.50	1.63
512	$1.50e-06$	2.05	$2.67e-07$	1.97	4.82	26.81	2.18
1024	$3.84e-07$	1.97	$6.77e-08$	1.98	61.76	239.26	4.86

Table 1: Potential flow past a sphere: convergence rate for e_∞ and e_2 and time spent for initialization (t_i), body-free problem (t_{bf}) and other operations (t_o) in seconds while using 1024 Intel Haswell CPU's

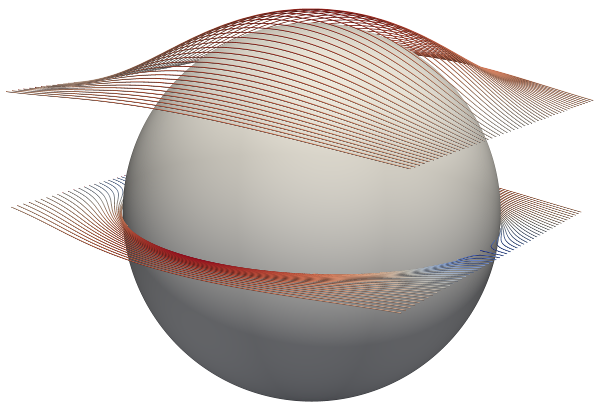

Figure 4: Potential flow past a sphere: Streamlines view colored by the norm of the velocity, $\nabla\phi + \mathbf{U}_\infty$. ($N = 512$)

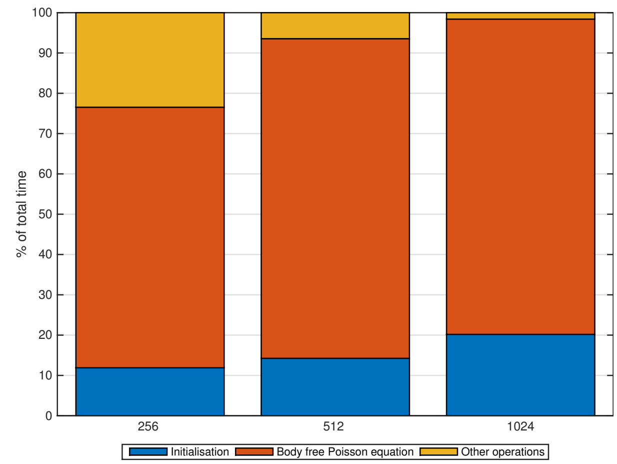

Figure 5: Potential flow past a sphere: Timings for $N = [256\ ,\ 512\ ,\ 1024]$

3.2. Potential flow past an ellipsoid

Following the same approach as before, we now consider the potential flow around an ellipsoid of semi-axes $[a\ ,\ b\ ,\ c]$. The analytical solution of such a problem is given by

$$\phi = \frac{abc\, U_\infty\, z}{3 - abc\, R_D\left(a^2, b^2, c^2\right)} \left[\frac{R_D\left(\epsilon - 1\ ,\ \epsilon - k^2\ ,\ \epsilon\right)}{\left(c^2 - a^2\right)^{3/2}} \right] \ , \tag{28}$$

with ϵ, k and R_D defined in Appendix B, where we present this analytical solution. We choose the computational domain to be $[L \times L \times L]$ with an ellipsoid given by $[a = 0.2L\ ,\ b = 0.3L\ ,\ c = 0.4L]$ and a tolerance for the boundary problem of 10^{-14}. Since $a < b < c$, the smallest radius of curvature, on the surface fo the ellipsoid is $R_{min} = \frac{a^2}{c} = 0.1L$.

The convergence slope of this study case is given in Fig. 6 and Table 2, while the solution is shown in Fig. 7. We conclude that our method achieves again the expected second order convergence both for the 2-norm and infinite norm. The number of iterations to perform in order to converge to the chosen tolerance is almost constant (see Fig. 6). Again, this is due to the fact that the problem is already well resolved ($R_{min}/h = [12.8\ ,\ 25.6\ ,\ 51.2\ ,\ 102.4]$ for $N = [128\ ,\ 256\ ,\ 512\ ,\ 1024]$, respectively). In Section 3.3, we further investigate the evolution of the number of iterations with respect to the resolution of the obstacle's boundary.

Fig. 8 shows the time spent in each part of the key steps. Again, the time spent in the body-free problem is prevailing, which is the expected behavior of the method.

N	E_∞	r_∞	E_2	r_2	t_i [s]	t_{bf} [s]	t_o [s]
128	$9.82e-05$		$4.62e-06$				
256	$1.80e-05$	2.45	$8.86e-07$	2.38	0.81	4.21	1.45
512	$3.40e-06$	2.40	$1.99e-07$	2.15	5.22	24.77	2.07
1024	$8.90e-07$	1.94	$4.96e-08$	2.01	58.65	218.38	5.19

Table 2: Potential flow past an ellipsoid: convergence rate for e_∞ and e_2 and time spent for initialization (t_i), body-free problem (t_{bf}) and other operations (t_o) in seconds while using 1024 Intel Haswell CPU's

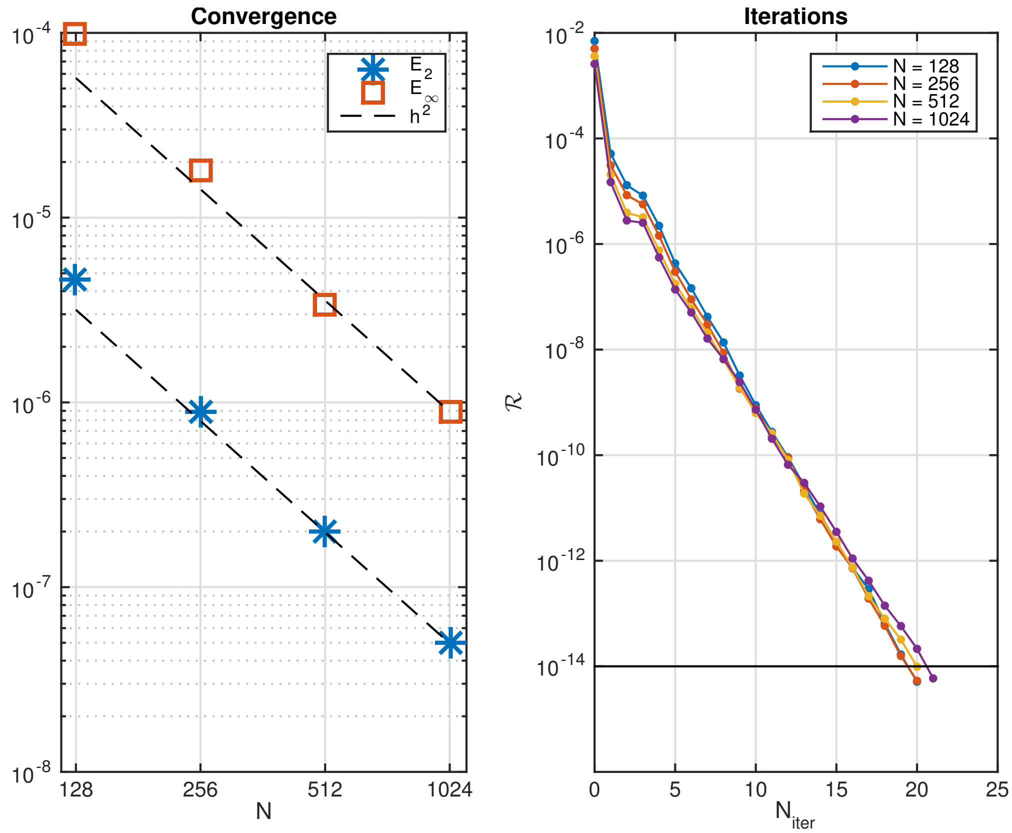

Figure 6: Potential flow past an ellipsoid: Grid convergence of the solution to the analytic expression and number of iteration needed to solve the boundary problem, Eq. (19), up to the chosen tolerance, 10^{-14}.

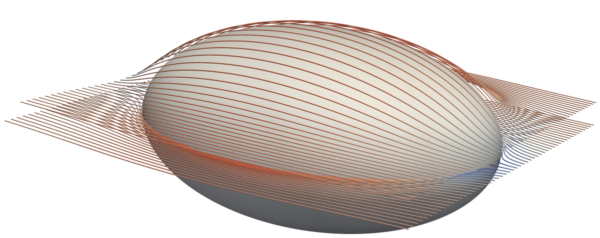

Figure 7: Potential flow past an ellipsoid: Streamlines view colored by the norm of the velocity, $\nabla\phi + \mathbf{U}_\infty$. ($N = 512$)

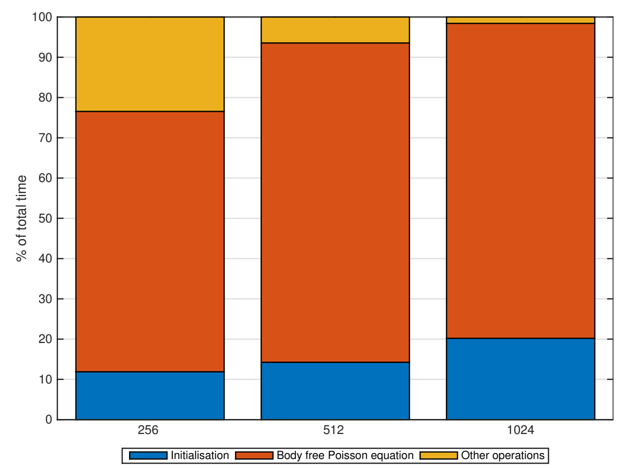

Figure 8: Potential flow past an ellipsoid: Timings for $N = [256\ ,\ 512\ ,\ 1024]$

3.3. Sensitivity analysis: R_{min}/h

The number of iterations and the evolution of the error varies with the number of points used to represent the shape and its finer scales in particular. We now consider an ellipsoid of semi-axes $[a = \beta\ L\ ,\ b = 0.3\ L\ ,\ c = 0.4\ L]$ and keep the computational domain constant. Fig. 9 shows the evolution of the error and the number of iterations to perform ($\text{tol} = 10^{-14}$) for various $R_{min}/h = \frac{\beta^2}{0.4}\frac{L}{h}$, where β varies and h remains constant.

We identify three different behaviors in Fig. 9. When the geometry is over-resolved ($R_{min}/h \geq 30$), the 2-norm exhibits a plateau. However, the decreasing infinite norm shows that an increasing resolution still improves the solution at the tip of the ellipsoid.

For a well-resolved geometry ($30 > R_{min}/h \geq 10$), both errors are decreasing when one increases the resolution Furthermore, the number of iterations to perform stays constant ($\simeq 20$). At coarser resolutions, the tip of the ellipsoid is not well-capturated, this affects the conditioning of the system and the number of iterations requested to converge increases as the resolution decreases.

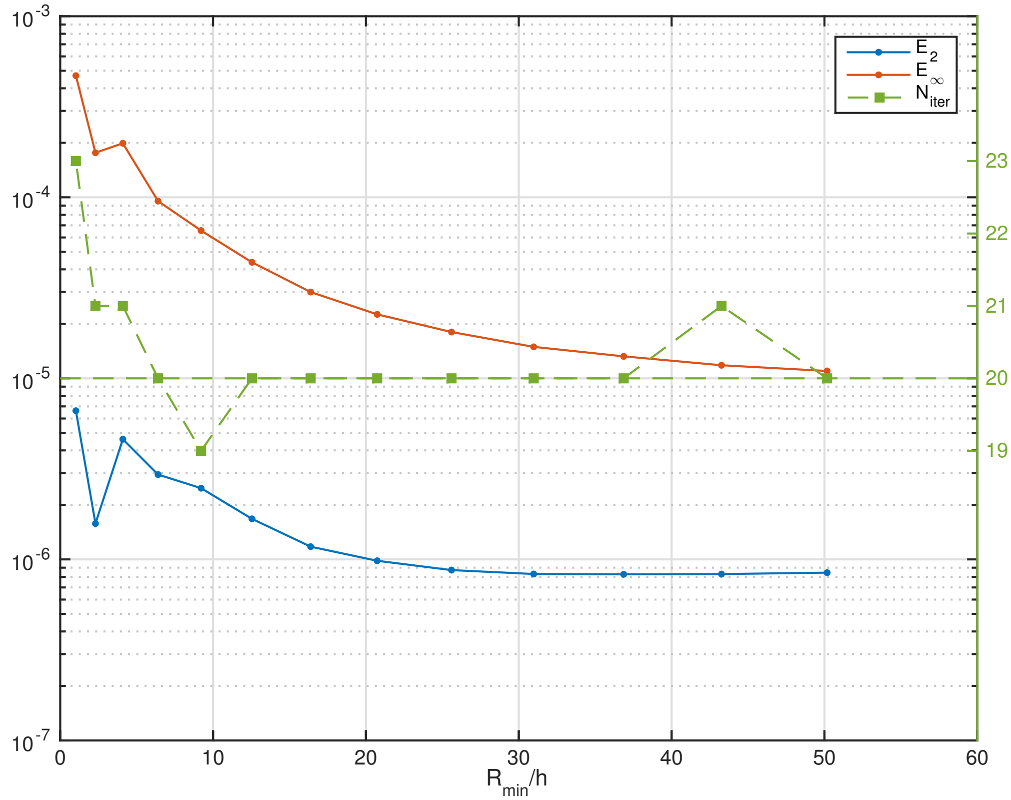

Figure 9: Potential flow past an ellipsoid: Analysis of the sensibility for various R_{min}/h coefficients (with constant h): the 2-norm, infinite norm and the number of iterations required to reach tol $= 10^{-14}$.

3.4. Sensitivity analysis of the boundary problem tolerance

A tolerance parameter has to be chosen for the GMRes iterations when solving the boundary problem. In this section, we study its influence to the expected second order convergence of the solution, as shown on Table 3. It shows that one should iterate enough to solve the boundary problem, Eq. (19), to guarantee a converging solution.

The previous exemples (Section 3.1 and Section 3.2) used a tolerance of 10^{-14}, but a tolerance of 10^{-10} would have been sufficient. We also notice that, at this tolerance, the GMRes convergence is fast and it only costs 13 to 14 iterations to converge a problem of 1024^3 unknowns.

4. Conclusion

In this work, we presented a novel way to use a fast and efficient unbounded elliptic solver in order to take an inner obstacle into account. The method proposed is based on a combination between a lattice Green's function

tol	N	N_{iter}	r_∞	r_2
$1e-06$	512	5	non convergent	non convergent
	1024	5		
$1e-08$	512	9	0.40	1.68
	1024	9		
$1e-10$	512	13	1.99	1.98
	1024	14		
$1e-12$	512	17	1.97	1.98
	1024	18		
$1e-14$	512	22	1.97	1.98
	1024	23		

Table 3: Potential flow past a sphere: Influence of the tolerance in the iterative process with respect to the convergence behavior

convolution carried out by the Hockney-Eastwood method or an FMM-based algorithm and the immersed interface technique. This coupling is performed through the Sherman-Morrison-Woodbury (SMW) formula, which involves a low-rank system to solve, i.e. the boundary problem. This formulation reduces the problem to the vicinity of the body and therefore to a boundary problem to solve instead of a volume problem. Through the SMW formula and the boundary problem, we find the additional source term which, combined with the unbounded source and the application of the body-free Laplacian, produces the correct solution satisfying the boundary condition. Through the SMW formula, our solver preserves the efficiency of the fast convolution-based algorithms and uses the immersed interface method to achieve a second order convergence, up to the boundary.

We demonstrate the second order accuracy of our method on two benchmark cases (the potential flows past a sphere and an ellipsoid). We then study the robustness of our method with respect to the resolution, R_{min}/h and the number of iterations performed within the iterative process for a chosen tolerance. We show that keeping $R_{min}/h \geq 10$ will lead to a well-resolved solution and that the boundary problem solution has to be accurate enough to guarantee the convergence.

This work, so far, is focused on the second order discretization of the Poisson equation. However, it could readily be extended to the Helmholtz problem, since only the Green's function has to be re-computed. Moreover, we stress that our method remains compatible with any solver based on a finite difference stencil, regardless of the body-free Poisson solver (FMM or FFT-based, multigrid approach) or the order of discretization, as long as the IIM perturbation is compatible with the chosen technique. For example, one could extend the method using higher-order stencils, resulting in an widening of the IIM perturbation along the interface. Therefore, this method is applicable to a wide range of problems involving an elliptic equation around inner bodies, and it goes beyond the present applications in computational fluid dynamics. Ongoing work topics include the use of the present technique in a 3D vortex particle-mesh method using the immersed interface approach, as already established in 2D by Marichal et al. [5], and its application to other elliptic solvers such as FMM-based algorithms.

Acknowledgements

We would like to acknowledge the helpful discussion with Y. Marichal (WaPT, Belgium, formerly at UCL). T. Gillis is supported by a FRIA grant from the Fonds de la Recherche Scientifique de Belgique (F.R.S.-FNRS). The present research also benefited from computational resources made available on the Tier-1 supercomputer of the

Federation Wallonie-Bruxelles, infrastructure funded by the Walloon Region under the grant agreement $n°1117545$.

Appendix A. Computation of the lattice Green function

As proposed by Liska and Colonius [16], for small $\left|\mathbf{m}\right|$, we do not compute the highly oscillating integral form Eq. (4). Rather, we evaluate the following equivalent integral

$$\Theta\left(\mathbf{m}\right) = -\int_0^{\infty} e^{-6t} I_{m_1}\left(2t\right) I_{m_2}\left(2t\right) I_{m_3}\left(2t\right) dt \quad , \tag{A.1}$$

where $I_n\left(\mathbf{x}\right)$ is the modified Bessel function of the first kind of order n. We use the asymptotic expansions of the I_n for large argument for the part $t \in [\alpha; \infty[$, with $\alpha = 75$. For larger $\left|\mathbf{m}\right|$, we use an expansion to avoid heavy computations. The way to derive the expansion for large $\mathbf{m}$'s is given by Martinsson and Rodin [20]. We here present the expansion as used in our method:

$$\begin{aligned}
\Theta\left(\mathbf{m}\right) = \; & \frac{1}{4\pi|\mathbf{m}|} \\
& + \frac{1}{16\pi|\mathbf{m}|^7} \Big[\quad m_1^4 + m_2^4 + m_3^4 - 3m_1^2m_2^2 - 3m_1^2m_3^2 - 3m_2^2m_3^2 \Big] \\
& + \frac{1}{128\pi|\mathbf{m}|^{13}} \Big[\quad 23 \left(m_1^8 + m_2^8 + m_3^8\right) \\
& \qquad -244 \left(m_1^6\left(m_2^2 + m_3^2\right) + m_2^6\left(m_1^2 + m_3^2\right) + m_3^6\left(m_1^2 + m_2^2\right)\right) \\
& \qquad -228 \; m_1^2m_2^2m_3^2\left|\mathbf{m}\right|^2 + 621\left(m_1^4m_2^4 + m_2^4m_3^4 + m_1^4m_3^4\right)\Big] \\
& + \frac{1}{2048\pi|\mathbf{m}|^{19}} \Big[\quad 2588 \left(m_1^{12} + m_2^{12} + m_3^{12}\right) \\
& \qquad -65676 \left(m_1^{10}m_2^2 + m_1^2m_2^{10} + m_1^{10}m_3^2 + m_1^2m_3^{10} + m_2^{10}m_3^2 + m_2^2m_3^{10}\right) \\
& \qquad +426144 \left(m_1^8m_2^4 + m_1^4m_2^8 + m_1^8m_3^4 + m_1^4m_3^8 + m_2^8m_3^4 + m_2^4m_3^8\right) \\
& \qquad -712884 \left(m_1^6m_2^6 + m_1^6m_3^6 + m_2^6m_3^6\right) \\
& \qquad -62892 \left(m_1^8m_2^2m_3^2 + m_1^2m_2^8m_3^2 + m_1^2m_2^2m_3^8\right) \\
& \qquad -297876 \left(m_1^6m_2^4m_3^2 + m_1^6m_2^2m_3^4 + m_1^4m_2^6m_3^2 \right. \\
& \qquad \qquad \left. + m_1^2m_2^6m_3^4 + m_1^2m_2^4m_3^6 + m_1^4m_2^2m_3^6\right) \\
& \qquad +2507340 \left(m_1^4m_2^4m_3^4\right)\Big] + \mathcal{O}\left(\frac{1}{|\mathbf{m}|^9}\right) \; .
\end{aligned} \tag{A.2}$$

We chose to use the expansion as soon as one component of $\mathbf{m}$ is greater than 32; i.e. we use the integral formulation for $\mathbf{m}$ if $m_1, m_2, m_3 < 32$.

Appendix B. Computation of the analytical solution of a potential flow around an ellipsoid

The analytical expression of the potential flow past an ellispoid is given by [48, page 152]. We have that the solution of the potential flow aligned with z, around an ellipsoid,

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1 \quad , \tag{B.1}$$

with $a < b < c$, is given by

$$\phi = \frac{abc}{2 - \gamma_0} \, U_\infty z \int_\lambda^\infty \left(a^2 + \lambda\right)^{-1/2} \left(b^2 + \lambda\right)^{-1/2} \left(c^2 + \lambda\right)^{-3/2} d\lambda \quad , \tag{B.2}$$

with

$$\gamma_0 = abc \int_0^\infty \left(a^2 + \lambda\right)^{-1/2} \left(b^2 + \lambda\right)^{-1/2} \left(c^2 + \lambda\right)^{-3/2} d\lambda = abc \, \frac{2}{3} \, R_D \left(a^2, b^2, c^2\right) \quad , \tag{B.3}$$

and where λ is the higher absolute value root of the third order polynomial

$$\frac{x^2}{a^2 + \theta} + \frac{y^2}{b^2 + \theta} + \frac{z^2}{c^2 + \theta} = 1 \quad . \tag{B.4}$$

In other words, for every point in space, one has to find λ by finding the roots of

$$\begin{aligned}
&\theta^3 \\
&+\theta^2 \, \left(a^2 + b^2 + c^2 - x^2 - y^2 - z^2\right) \\
&+\theta \, \left(a^2 b^2 + b^2 c^2 + c^2 a^2 - x^2 \left(b^2 + c^2\right) - y^2 \left(a^2 + c^2\right) - z^2 \left(a^2 + b^2\right)\right) \\
&+ \left(a^2 b^2 c^2 - x^2 b^2 c^2 - y^2 a^2 c^2 - z^2 a^2 b^2\right) = 0
\end{aligned} \quad , \tag{B.5}$$

which can be solved using exact formula [49, §1.11(iii)]. The remaining integral, I,

$$I = \int_\lambda^\infty \left(a^2 + \lambda\right)^{-1/2} \left(b^2 + \lambda\right)^{-1/2} \left(c^2 + \lambda\right)^{-3/2} d\lambda \quad . \tag{B.6}$$

can also been expressed as (see [50])

$$I = \frac{2}{\left(c^2 - a^2\right)^{3/2}} \, D \left(\zeta \, , \; k\right) \quad , \tag{B.7}$$

where D is the Legendre integral defined in [49, §19.2(ii)] and

$$\sin\zeta = \eta = \sqrt{\frac{c^2 - a^2}{c^2 + \lambda}} \qquad\qquad k = \sqrt{\frac{c^2 - b^2}{c^2 - a^2}} \quad . \tag{B.8}$$

Using the definition of R_D, given in [49, §19.25(i)], we finally get

$$I = \frac{2/3}{\left(c^2 - a^2\right)^{3/2}} \, R_D \left(\epsilon - 1 \, , \; \epsilon - k^2 \, , \; \epsilon\right) \quad , \tag{B.9}$$

with $\epsilon = \dfrac{1}{\sin^2\zeta} = \dfrac{1}{\eta^2}$. Finally, the potential is expressed as

$$\phi = \frac{abc\, U_\infty\, z}{3 - abc\, R_D\left(a^2, b^2, c^2\right)} \left[\frac{R_D\left(\epsilon - 1\ ,\ \epsilon - k^2\ ,\ \epsilon\right)}{\left(c^2 - a^2\right)^{3/2}}\right] \quad , \tag{B.10}$$

which is evaluated using [51].

References

[1] R. A. James, Solution of Poisson's equation for isolated source distributions, Journal of Computational Physics 25 (2) (1977) 71–93.

[2] K. Lackner, Computation of ideal MHD equilibria, Computer Physics Communications 12 (1) (1976) 33 – 44.

[3] P. McCorquodale, P. Colella, G. T. Balls, S. B. Baden, A Local Corrections Algorithm for Solving Poisson's Equation in Three Dimensions, Communications in Applied Mathematics and Computational Science 2 (1) (2007) 57–81.

[4] Y. Marichal, P. Chatelain, G. Winckelmans, An immersed interface solver for the 2-D unbounded Poisson equation and its application to potential flow, Computers & Fluids 96 (2014) 76–86.

[5] Y. Marichal, P. Chatelain, G. Winckelmans, Immersed interface interpolation schemes for particle–mesh methods, Journal of Computational Physics 326 (2016) 947 – 972.

[6] L. Greengard, A rapid evaluation of potential fields in particle systems, MIT Press, 1988.

[7] J. K. Salmon, M. S. Warren, Fast Parallel Tree Codes for Gravitational and Fluid Dynamical N-Body Problems, The International Journal of Supercomputer Applications and High Performance Computing 8 (2) (1994) 129–142.

[8] R. Beatson, L. Greengard, A short course on fast multipole methods, in: Wavelets, Multilevel Methods and Elliptic PDEs, Oxford University Press, 1–37, 1997.

[9] A. Gillman, P. Martinsson, Fast and accurate numerical methods for solving elliptic difference equations defined on lattices, Journal of Computational Physics 229 (24) (2010) 9026 – 9041.

[10] A. Gillman, P. Martinsson, A fast solver for Poisson problems on infinite regular lattices, Journal of Computational and Applied Mathematics 258 (2014) 42 – 56.

[11] F. Vico, L. Greengard, M. Ferrando, Fast convolution with free-space Green's functions, Journal of Computational Physics 323 (2016) 191 – 203.

[12] R. Hockney, J. Eastwood, Computer Simulation using Particles, Taylor & Francis, Inc. Bristol, PA, USA, 1988.

[13] P. Chatelain, P. Koumoutsakos, A Fourier-based elliptic solver for vortical flows with periodic and unbounded directions, Journal of Computational Physics 229 (7) (2010) 2425–2431.

[14] M. Hejlesen, J. Rasmussen, P. Chatelain, J. Walther, A high order solver for the unbounded Poisson equation, Journal of Computational Physics 252 (2013) 458–467.

[15] M. M. Hejlesen, J. H. Walther, A multiresolution method for solving the Poisson equation using high order regularization, Journal of Computational Physics 326 (2016) 188 – 196.

[16] S. Liska, T. Colonius, A parallel fast multipole method for elliptic difference equations, Journal of Computational Physics 278 (2014) 76 – 91.

[17] S. Liska, T. Colonius, A fast lattice Green's function method for solving viscous incompressible flows on unbounded domains, Journal of Computational Physics 316 (2016) 360 – 384.

[18] S. Liska, T. Colonius, A fast immersed boundary method for external incompressible viscous flows using lattice Green's functions, Journal of Computational Physics 331 (2017) 257 – 279.

[19] A. Gholami, D. Malhotra, H. Sundar, G. Biros, FFT, FMM, or Multigrid? A comparative Study of State-Of-the-Art Poisson Solvers for Uniform and Nonuniform Grids in the Unit Cube, SIAM Journal on Scientific Computing 38 (3) (2016) C280–C306.

[20] P.-G. Martinsson, G. J. Rodin, Asymptotic expansion of lattice Green's function, Proceedings: Mathematical, Physical and Engineering Sciences 458 (2027) (2002) 2609–2622.

[21] W. Proskurowski, O. Widlund, Numerical Solution of Helmholtz's Equation by Implicit Capacitance Matrix Methods, ACM Trans. Math. Softw. 5 (1) (1979) 36–49.

[22] D. P. O'Leary, O. Widlund, Capacitance Matrix Methods for the Helmholtz Equation on General Three-Dimensional Regions, Mathematics of Computation 33 (147) (1979) 849–879.

[23] J. A. George, The Use of Direct Methods for the Solution of the Discrete Poisson Equation on Non-rectangular Regions, Tech. Rep., Stanford, CA, USA, 1970.

[24] B. L. Buzbee, F. W. Dorr, J. A. George, G. H. Golub, The Direct Solution of the Discrete Poisson Equation on Irregular Regions, SIAM Journal on Numerical Analysis 8 (4) (1971) 722–736.

[25] C. S. Peskin, Numerical analysis of blood flow in the heart, Journal of Computational Physics 25 (3) (1977) 220–252.

[26] K. Taira, T. Colonius, The immersed boundary method: A projection approach, Journal of Computational Physics 225 (2) (2007) 2118–2137.

[27] T. Colonius, K. Taira, A fast immersed boundary method using a nullspace approach and multi-domain far-field boundary conditions, Computer Methods in Applied Mechanics and Engineering 197 (25-28) (2008) 2131–2146.

[28] D. B. Stein, R. D. Guy, B. Thomases, Immersed boundary smooth extension: A high-order method for solving {PDE} on arbitrary smooth domains using Fourier spectral methods, Journal of Computational Physics 304 (2016) 252 – 274.

[29] A. Mayo, The rapid evaluation of volume integrals of potential theory on general regions, Journal of Computational Physics 100 (2) (1992) 236 – 245.

[30] A. McKenney, L. Greengard, A. Mayo, A fast Poisson solver for complex geometries, Journal of Computational Physics 118 (1995) 348–355.

[31] A. Mayo, A. Greenbaum, Fourth order accurate evaluation of integrals in potential theory on exterior 3D regions, Journal of Computational Physics 220 (2) (2007) 900 – 914.

[32] T. Askham, A. Cerfon, An adaptive fast multipole accelerated Poisson solver for complex geometries, Journal of Computational Physics 344 (2017) 1 – 22.

[33] P. Ploumhans, G. S. Winckelmans, Vortex Methods for High-Resolution Simulations of Viscous Flow Past Bluff Bodies of General Geometry, Journal of Computational Physics 165 (2) (2000) 354–406.

[34] P. Ploumhans, G. Winckelmans, Vortex Methods for Direct Numerical Simulation of Three-Dimensional Bluff Body Flows: Application to the Sphere at Re= 300, 500, and 1000, Journal of Computational Physics 178 (2) (2002) 427–463.

[35] R. J. Leveque, Z. L. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, SIAM Journal on Numerical Analysis 31 (4) (1994) 1019–1044.

[36] M. N. Linnick, H. F. Fasel, A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains, Journal of Computational Physics 204 (2005) 157–192.

[37] Z. Li, I. Kazufumi, The immersed interface method: Numerical solutions of PDEs involving interfaces and irregular domains, SIAM, 2006.

[38] S. Hosseinverdi, H. F. Fasel, Very High-Order Accurate Sharp Immersed Interface Method: Application to Direct Numerical Simulations of Incompressible Flows, American Institute of Aeronautics and Astronautics, 2017.

[39] G. Miller, An iterative boundary potential method for the infinite domain Poisson problem with interior Dirichlet boundaries, Journal of Computational Physics 227 (16) (2008) 7917 – 7928.

[40] G. H. Golub, C. F. Van Loan, Matrix Computation, The Johns Hopkins University Press, 1996.

[41] R. Chatelin, P. Poncet, Hybrid grid-particle methods and Penalization: A Sherman-Morrison-Woodbury approach to compute 3D viscous flows using {FFT}, J. Comput. Phys. 269 (0) (2014) 314 – 328.

[42] Y. Marichal, P. Chatelain, G. Winckelmans, An immersed interface vortex particle-mesh solver, in: Bulletin of the American Physical Society, vol. 59, American Physical Society, 2014.

[43] W. W. Hager, Updating the Inverse of a Matrix, SIAM Review 31 (2) (1989) 221–239.

[44] Y. Saad, Iterative Methods for Sparse Linear Systems, Society for Industrial and Applied Mathematics, 2003.

[45] I. F. Sbalzarini, J. H. Walther, M. Bergdorf, S. E. Hieber, E. M. Kotsalis, P. Koumoutsakos, PPM A highly efficient parallel particle mesh library for the simulation of continuum systems, J. Comput. Phys. 215 (2006) 566–588.

[46] P. Chatelain, A. Curioni, M. Bergdorf, D. Rossinelli, W. Andreoni, P. Koumoutsakos, Billion vortex particle Direct Numerical Simulations of aircraft wakes, Computer Methods in Applied Mechanics and Engineering 197 (13) (2008) 1296–1304.

[47] P. Poncet, Analysis of an immersed boundary method for three-dimensional flows in vorticity formulation, J. Comput. Phys. 228 (2009) 7268–7288.

[48] S. H. Lamb, Hydrodynamics, Dover, New York, 1993.

[49] DLMF, NIST Digital Library of Mathematical Functions, http://dlmf.nist.gov/, Release 1.0.14 of 2016-12-21, f. W. J. Olver, A. B. Olde Daalhuis, D. W. Lozier, B. I. Schneider, R. F. Boisvert, C. W. Clark, B. R. Miller and B. V. Saunders, eds., 2016.

[50] T. Funada, Funada's file of potential functions for ellipsoids, 2006.

[51] B. C. Carlson, E. M. Notis, Algorithm 577: Algorithms for Incomplete Elliptic Integrals [S21], ACM Trans. Math. Softw. 7 (3) (1981) 398–403.