



City Research Online

City St George's, University of London

Citation: Toja-Silva, F., Favier, J. & Pinelli, A. (2014). Radial basis function (RBF)-based interpolation and spreading for the immersed boundary method. *Computers & Fluids*, 105, pp. 66-75. doi: 10.1016/j.compfluid.2014.09.026

This is the accepted version of the paper.

This version of the publication may differ from the final published version. To cite this item please consult the publisher's version.

Permanent repository link: <https://openaccess.city.ac.uk/id/eprint/6932/>

Link to published version: <https://doi.org/10.1016/j.compfluid.2014.09.026>

Copyright and Reuse: Copyright and Moral Rights remain with the author(s) and/or copyright holders. Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge, unless otherwise indicated, provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way. For full details of reuse please refer to [City Research Online policy](#).

Radial Basis Function (RBF)-based Interpolation and Spreading for the Immersed Boundary Method

Francisco Toja-Silva^{a,b,*}, Julien Favier^c, Alfredo Pinelli^d

^a*Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas (CIEMAT), Av. Complutense 40, 28040 Madrid, Spain.*

^b*Escuela Técnica Superior de Ingenieros Aeronáuticos, Universidad Politécnica de Madrid (UPM), Madrid, Spain.*

^c*Aix Marseille Université, CNRS, Centrale Marseille, M2P2 UMR 7340, France.*

^d*School of Engineering and Mathematical Sciences (SEMS), City University London, EC1V 0HB London, United Kingdom.*

Abstract

Immersed boundary methods are efficient tools of growing interest as they allow to use generic CFD codes to deal with complex, moving and deformable geometries, for a reasonable computational cost compared to classical body-conformal or unstructured mesh approaches. In this work, we propose a new immersed boundary method based on a radial basis functions framework for the spreading-interpolation procedure. The radial basis function approach allows for dealing with a cloud of scattered nodes around the immersed boundary, thus enabling the application of the devised algorithm to any underlying mesh system. The proposed method can also keep into account both Dirichlet and Neumann type conditions. To demonstrate the capabilities of our novel approach, the imposition of Dirichlet boundary conditions on a 2D cylinder geometry in a Navier-Stokes CFD solver, and the imposition of Neumann boundary conditions on an adiabatic wall in an unsteady heat conduction problem are considered. One of the most significant advantage of the proposed method lies in its simplicity given by the algorithmic possibility of carrying out the interpolation and spreading steps all together, in a single step.

Keywords: Immersed-Boundary Method, Interpolation-Spreading, Radial Basis Functions.

*Corresponding author: Tel. +34 649382293 - E-mail frantojasilva@yahoo.es

1. Introduction

When considering complex geometries, immersed boundary (IB) methods constitute an efficient alternative to avoid either difficult or even impossible grid generation procedures when dealing with body-fitted formulations or extra computational costs associated with unstructured grid solvers. Those difficulties associated to classical body fitted or unstructured solvers, become even more severe for moving or deformable boundary, as is the case in fluid-structure interaction problems. Moreover, in those cases, the immersed boundary methods are not restricted to small boundary displacements such as other techniques based on smooth mesh deformation [1]. An alternative to immersed boundary methods is the immersed finite element method (IFEM) proposed by Zhang et al. [2]. This work extends the idea of the discrete Delta functions to unstructured meshes using the idea of Reproducing Kernel Particle Methods (RKPM). Both fluid and solid domains are modeled with finite element methods and the continuity between the fluid and solid sub-domains are enforced via the interpolation of the velocities, and the distribution of the forces with the reproducing kernel particle method (RKPM) Delta function. The immersed boundary method was historically introduced by Peskin [3], in a pioneering work focussed on heart dynamics. Since then it has continuously evolved to tackle numerical simulations of new scientific domains, from biomedical to chemical engineering and aeronautics. The classical approach is to solve the problem equations on a uniform cartesian grid (Eulerian) for both solid and fluid phases. The boundary of the solid is described through a set of markers (Lagrangian) which do not coincide with the fluid mesh points. The communications between fluid and solid is done through volume forces that enforce the no-slip boundary condition and ensure the conservation at the wall of linear momentum, force and torque by means of interpolation-spreading operators (discrete Delta functions in the case of the classical method). Using the same discrete Delta functions for both interpolation and spreading guarantees conservation of energy, along with conservation of force and torque [4].

A review of the different flavours of the method can be found in [5], where they are divided in two groups. The first group of techniques, called “continuous forcing” [3, 6, 7], where the derivation of the body forces is carried out before the discretisation step, have been widely used when dealing with

sharp boundaries and rigid objects [5]. The second group termed as “discrete forcing” methods [8, 12, 13, 14, 15, 16] is based on a set of singular body forces, defined on the Eulerian fluid nodes, to enforce the desired boundary values. This second group of techniques allows to use larger time-steps and, certain formulations, can handle sharp boundaries too [14]. The immersed boundary method that is proposed in the present work, which makes use of radial basis functions (RBFs), can be classified within the second group.

Traditionally, RBFs have been used for scattered data interpolation and their processing, and for function approximation. In the last decades, RBFs have also been employed as basis functions for the solution of PDEs, including fluid dynamics conservation laws (see the seminal paper by Kansa [34]). Applications that are related with the present contribution, include the interpolation of the displacements of boundary nodes of moving meshes to inner unstructured domains [19], data transfer through interfaces of non-matching meshes [20], multivariable interpolation in fluid-solid-interaction problems [21], etc. More recently, RBFs have been also applied in the context of immersed boundary methods. Specific works proposing RBFs for interpolation in fluid-structure interaction applications are briefly reviewed hereafter. Mai-Duy and Tran-Cong [22] introduce one-dimensional integrated RBF networks within a collocation framework to solve PDEs on Cartesian grids. In the proposed formulation, Dirichlet boundary conditions are enforced in a direct way, while Neumann boundary conditions are imposed by means of integration constraints. Fang et al. [23] use Gaussian RBFs to smooth Gibbs oscillations occurring in velocity derivatives near the immersed boundary when spectral-like discretization of the Navier-Stokes equations is used in conjunction with the IB method. Shankar et al. [24] introduce a parametric RBF model to represent the surface of the immersed elastic object. This approach allows for tracking the deformation of the surface efficiently since just a small set of Lagrangian markers is required to that end. Surface boundary conditions are imposed via an interpolation-spreading approach using pseudo delta functions defined on a larger set of immersed surface nodes. Liu et al. [25] use local RBFs (also termed as RBF-FD, i.e., RBF-generated finite differences) to solve the compressible Navier-Stokes equations. The boundary conditions are kept into account by modifying locally the values of the nodes neighbouring the domain limiting surface in such a way that the interpolated values are the desired ones at the boundary. The whole boundary value enforcement procedure is carried out at each Runge-Kutta stage of their time stepping algorithm (a three stage RK method). Thai-Quang et al.

[17] present a direct forcing method based on compact integrated radial basis functions, using a smoothed version of the discrete delta functions to transfer the physical quantities between Eulerian and Lagrangian nodes. Shankar et al. [18] use RBF-based symmetric Hermite interpolation to extend the Augmented Direct Forcing method [16] to handle objects with concavities or objects in close proximity. This modified version of the Augmented Direct Forcing method was used in conjunction with an RBF-FD method for solving reaction-diffusion equations on object surfaces. Their method only requires the coordinates of the scattered nodes representing the surface and its normal-to-the-surface vectors at those locations.

In the present work we propose an original immersed boundary method based on radial basis functions interpolation. To the best of our knowledge, the method differs from previous literature implementations, proving to be easier to implement and more versatile than existing ones. The most significant advantages of the present method are: *i*) it is applicable to any underlying grid system because based on an interpolation-spreading process defined on a scattered cloud of nodes; *ii*) the weights of the RBF are calculated independently from the velocity of the nodes (i.e., for static geometries they only depend on the geometry); and, *iii*) both interpolation and spreading (or convolution) are carried out simultaneously in a single stage.

The paper is organised as follows: firstly, the formulation of the RBFs used within the IB algorithm is presented; next, the numerical algorithm that we envisaged to impose Dirichlet boundary conditions is presented and validated in the context of a 2D Navier-Stokes solver. Afterwards, the imposition of Neumann boundary conditions is discussed and applied in the case of a 2D unsteady heat conduction problem. Finally, conclusions and recommendations for further future works are given.

2. RBF interpolation from an arbitrarily scattered set of nodes

Following the basic idea behind immersed boundary methods, we define a set of *Lagrangian* nodes \mathbf{X}_i , $i = 1 \cdots m$ on the immersed surface Γ surrounded by a cloud of neighbouring nodes (*Eulerian* nodes) \mathbf{x}_k , $k = 1 \cdots n$, belonging to the underlying fluid grid. The key idea of the IB method is to compute the fluid velocity at nodes \mathbf{X}_i via *interpolation* from the \mathbf{x}_k , and thus find the set of localised forces (per unit mass, and unit time) on the interface that restore the desired velocity condition on Γ . Typically, this set of singular forces F_i , defined on Γ , is later on distributed on the fluid nodes via a convolution

(*spreading*) operation, obtaining a set of Eulerian forces f_k , $k = 1 \cdots n$. More details on how this procedure has been adapted to the present algorithm will be illustrated later on. Here, we just highlight that one of the key ingredients of any IB method is the transfer of a force field from the Lagrangian to the Eulerian nodes and viceversa [4, 35]. We will write the relationship between the force defined on the immersed surface and the one acting on the fluid as:

$$F = Wf, \quad \text{with } F = (F_1, F_2, \dots, F_n)^T \quad \text{and } f = (f_1, f_2, \dots, f_m)^T \quad (1)$$

where W is a transformation matrix with m rows and n columns (in general, $n \neq m$) which entries depend on the basis functions used for the interpolation. The W matrix must obey the constraint of having all its column-sum equal to unity to conserve the total force, $\sum_{i=1}^m F_i = \sum_{k=1}^n f_k$, and to guarantee energy conservation (i.e., the work done by the forces in the two grid systems, see [20, 21]). Indeed if $\sum_i W_{i,k} = 1$, we have:

$$\sum_i F_i = \sum_i \sum_k W_{i,k} f_k = \sum_k f_k \sum_i W_{i,k} = \sum_k f_k \quad (2)$$

Thus, the column-sum of W must be equal to unity independently of the interpolatory technique that is used. Next, we assemble matrix W making use of RBFs also introducing the *inverse* operator that transfer the force data from the immersed interface to the fluid grid. We start by writing one of the equations of (1), that provides the interpolated value on a point \mathbf{X}_l of the immersed surface as a function of the n values on the fluid grid:

$$F_l = \sum_{j=1}^n \omega_j f_j, \quad \text{with } F_l = F(\mathbf{X}_l) \quad \text{and } f_j = f(\mathbf{x}_j) \quad (3)$$

where ω_i , $i = 1, \dots, n$ are unknown interpolation weights. To determine the values of the weights, we rewrite (3) as:

$$F_l = \sum_{j=1}^n \underbrace{\left[\sum_{k=1}^n \lambda_{j,k} \phi(\|\mathbf{X}_l - \mathbf{x}_k\|) + \gamma_j \right]}_{\omega_j} f_j, \quad (4)$$

Where $\phi_{j,k} = \phi(\|\mathbf{x}_j - \mathbf{x}_k\|)$ is a radial basis function: a symmetric function which value just depends on the Euclidean distance $\|\mathbf{x}_j - \mathbf{x}_k\|$ between nodes.

To determine the value of the unknown parameters $\lambda_{j,k}$ and γ_j (and thus of ω_j), for each node $\mathbf{x}_j, j = 1, \dots, n$ we impose the cardinality conditions:

$$\sum_{k=1}^n \lambda_{j,k} \phi_{j,k} + \gamma_j = \delta_{j,k} \quad (5)$$

(where $\delta_{j,k}$ is the usual Kroeneker symbol) together with the constraint that the interpolant should exactly represent a constant function $\sum_{k=1}^n \lambda_{j,k} = 0$. Thus, for each node of the support \mathbf{x}_j , we need to solve a linear system of equations to determine the $\lambda_{j,k}, k = 1 \dots n$:

$$\begin{bmatrix} \phi_{1,1} & \cdots & \phi_{1,n} & 1 \\ \cdots & \cdots & \cdots & \cdots \\ \phi_{j,1} & \cdots & \phi_{j,n} & 1 \\ \cdots & \cdots & \cdots & \cdots \\ \phi_{n,1} & \cdots & \phi_{n,n} & 1 \\ 1 & \cdots & 1 & 0 \end{bmatrix} \begin{bmatrix} \lambda_{j,1} \\ \cdots \\ \lambda_{j,j} \\ \cdots \\ \lambda_{j,n} \\ \gamma_j \end{bmatrix} = \begin{bmatrix} 0 \\ \cdots \\ 1 \\ \cdots \\ 0 \\ 0 \end{bmatrix} \quad (6)$$

For the present work we have used the inverse multiquadric radial basis functions (IMQ-RBF):

$$\phi_{j,k} \equiv \phi(r_{j,k}) := \left\{ \frac{1}{\sqrt{1 + (\epsilon r_{j,i})^2}} : j, k \in \mathfrak{S}_l \right\}, \quad r_{j,k} = \|\mathbf{x}_j - \mathbf{x}_k\| \quad (7)$$

where \mathfrak{S}_l denotes the interpolation support of node \mathbf{X}_l (that is discussed in detail later), $\epsilon > 0$ is the shape parameter and $r_{j,k}$ the radial distance between the nodes j and k . Note that $\phi_{i,i} = 1$. The optimum value of the shape parameter ϵ depends on the noise in the data set, and it has to be determined according to each application. In summary, by solving the linear systems formed by Eqs. (6) we determine the weights $\lambda_{j,k}$ that are independent on the location of \mathbf{X}_l . Next, we compute the ω_j by evaluating the expression $\omega_j = \sum_{k=1}^n \lambda_{j,k} \phi(\|\mathbf{X}_l - \mathbf{x}_k\|) + \gamma_j$ to be used in the interpolatory formula (3). By considering all the Lagrangian points \mathbf{X}_l , and thus all the respective Eulerian support \mathfrak{S}_l , we can compute all the weights ω_j that correspond to the entries of matrix W appearing in (1).

An important feature of this interpolant is that the weights $\lambda_{j,k}$ can be also used to approximate the value of the derivative (or of any linear differential operator $\mathcal{L}(f)$) at the Lagrangian node l . Indeed, we have:

$$\mathcal{L}(f)(\mathbf{X}_l) = \sum_{j=1}^n \sum_{k=1}^n [\lambda_{j,k} \mathcal{L}(\phi)(\|\mathbf{X}_l - \mathbf{x}_k\|) + \gamma_j] f_j \quad (8)$$

Where $\mathcal{L}(\phi)(\|\mathbf{X}_l - \mathbf{x}_k\|)$ is the linear differential operator applied to the IMQ-RBF evaluated in \mathbf{X}_l .

In the particular case of the normal derivative to the immersed boundary Γ at point \mathbf{X}_l where the unit normal vector is \vec{n} and the tangent vector is given by $\vec{\tau} = [\tau_x, \tau_y]^T$, one can define the linear differential operator via the projection $P = I - \tau\tau^T$ (where I is the identity matrix). Using P , the approximation to the normal derivative of a function f at $\mathbf{X}_l = (x_l, y_l)$ reads:

$$\mathcal{L}(f)(\mathbf{X}_l) = \vec{\nabla} f \cdot \vec{n} = \sum_{j=1}^n \sum_{k=1}^n P \vec{\nabla} \phi(\|\mathbf{X}_l - \mathbf{x}_k\|) f_j \quad (9)$$

with $\vec{\nabla} \phi(\|\mathbf{X}_l - \mathbf{x}_k\|) = (\partial_x \phi(\|\mathbf{X}_l - \mathbf{x}_k\|), \partial_y \phi(\|\mathbf{X}_l - \mathbf{x}_k\|))^T$ and:

$$\partial_x \phi(\|\mathbf{X}_l - \mathbf{x}_k\|) \equiv -\frac{\epsilon^2(x_l - x_k)}{(1 + (\epsilon r_{l,k})^2)^{3/2}}, \quad (10)$$

$$\partial_y \phi(\|\mathbf{X}_l - \mathbf{x}_k\|) \equiv -\frac{\epsilon^2(y_l - y_k)}{(1 + (\epsilon r_{l,k})^2)^{3/2}}, \quad (11)$$

i.e., the partial x and partial y derivatives of (7). The convergence order of the IMQ-RBF depends on both the smoothness of the target function and the node density of the data sites. For smooth target functions that lie in its native space, the IMQ-RBF can exhibit spectral convergence. For a full discussion, see [26]. In the present case, described in §3.1, the interpolation functions have a first order convergence rate. The order that can be achieved is mainly related with the support used for the interpolation. Figure 1 shows the norm of the interpolation error E_{RBF} at the immersed surface defined by

$$E_{RBF} = \max_l \{U_{RBF}\}, \quad (12)$$

where U_{RBF} refers to the interpolation error of an arbitrary known field at each Lagrangian point l (difference between the interpolated value and the known function at the point). The given results correspond to the method and the support choice described in the next section.

3. Imposition of Dirichlet boundary conditions

A common way of dealing with the incompressible Navier Stokes equations is by employing a continuous projection method (see [9] and [10], for

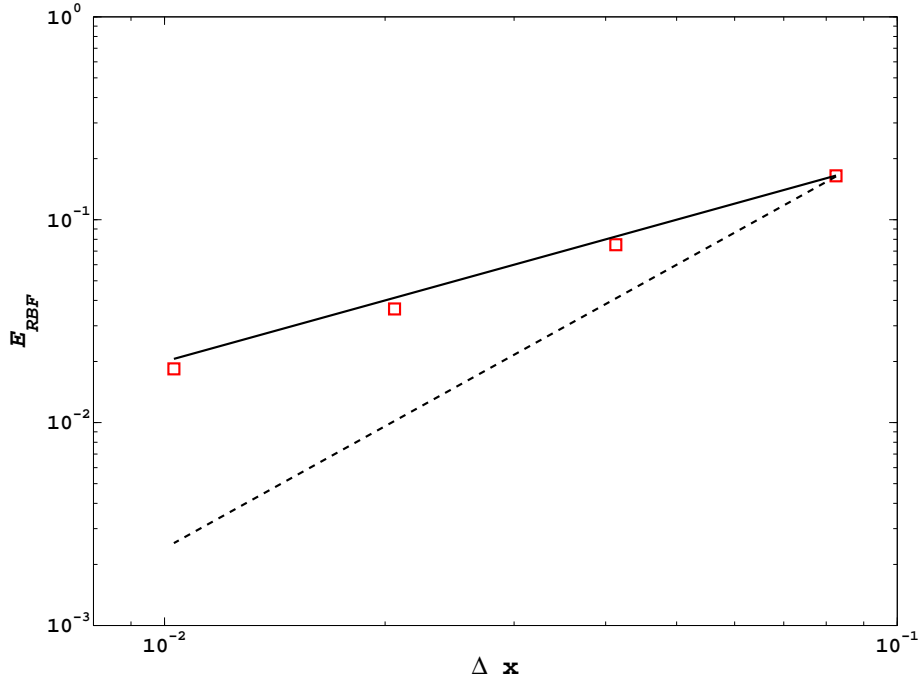


Figure 1: Norm of the interpolation error at the Lagrangian points for the case described in §3.1. Red squares: interpolation error; solid line: Δx (1st order); dashed line: Δx^2 (2nd order).

example). In this framework, a popular time advancement procedure (as it is done in [8] for instance) reads as:

$$\frac{u^* - u^n}{\Delta t} = -N_l(u^n, u^{n-1}) - G\phi^{n-1} + \frac{1}{Re}L(u^*, u^n), \quad (13)$$

for the predicted momentum equation, and

$$L\phi = \frac{1}{\Delta t}Du^*, \quad (14)$$

for the value of the projector (i.e., the pseudo pressure) that enforces the divergence free condition via:

$$u^{n+1} = u^* - \Delta t G\phi^n, \quad (15)$$

In the given, time discretised equations, u^* is the predicted velocity field, u^n is the velocity field obtained at the time-step n , Δt is the time step, N_l , G and D are, respectively, the finite-difference discrete approximation of the non-linear, the gradient and the divergence operators, L is the discrete Laplacian and ϕ is a projection variable (related to the pressure field). Normally, all those operators are defined on a staggered grid system [11].

To impose assigned Dirichlet boundary values on the immersed boundary Γ , the above time sequence is modified by carrying out the time advancement of the predicted momentum equations in two stages: Firstly, a fully explicit prediction step is performed without body force (without keeping into account the Dirichlet values on Γ) by advancing Eq. (13). The predicted velocity field u^* leads to a predicted force field $u^*/\Delta t$ (per unit mass), that is interpolated on the immersed boundary using the procedure described in the previous section. Next, we determine the necessary force per unit mass required to enforce the desired boundary conditions on Γ .

$$F = \frac{U^o}{\Delta t} - \frac{\mathcal{I}(u^*)}{\Delta t}, \quad (16)$$

In (16), U^o is the desired velocity distribution on the immersed boundary, and \mathcal{I} is the interpolator operator (Eulerian mesh to immersed surface). Once the value of the restoring force F on Γ has been determined, we seek corrections to the momentum equation, discretised on the Eulerian mesh, by introducing a set of singular body forces. In other words, we should determine the values of f to be assigned to the interpolating Eulerian nodes to recover the values of F on Γ given by (16):

$$f = R(u^*/\Delta t), \quad (17)$$

where R refers to the interpolation-spreading using a radial basis function. The regularized force f is then added to the right hand side of the momentum equations, and the time advancement of Eq. (13) is repeated:

$$\frac{u^* - u^n}{\Delta t} = -N_l(u^n, u^{n-1}) - G\phi^{n-1} + \frac{1}{Re}L(u^*, u^n) + f. \quad (18)$$

The algorithm completes the time step with the usual solution of the pressure Poisson equation, Eq. (14), and the completion of the projection step (i.e., equation (15)).

3.1. Methodology

For sake of clarity, we consider Γ to be a circle of radius r , centred at (x_c, y_c) . The Eulerian nodes neighbouring the immersed surface belong to an annulus of width $2\Delta r$ defined along the immersed boundary. The nodes enclosed in this region will be the ones involved with the interpolation and spreading operations (see figure 2). We start by tagging the inner nodes,

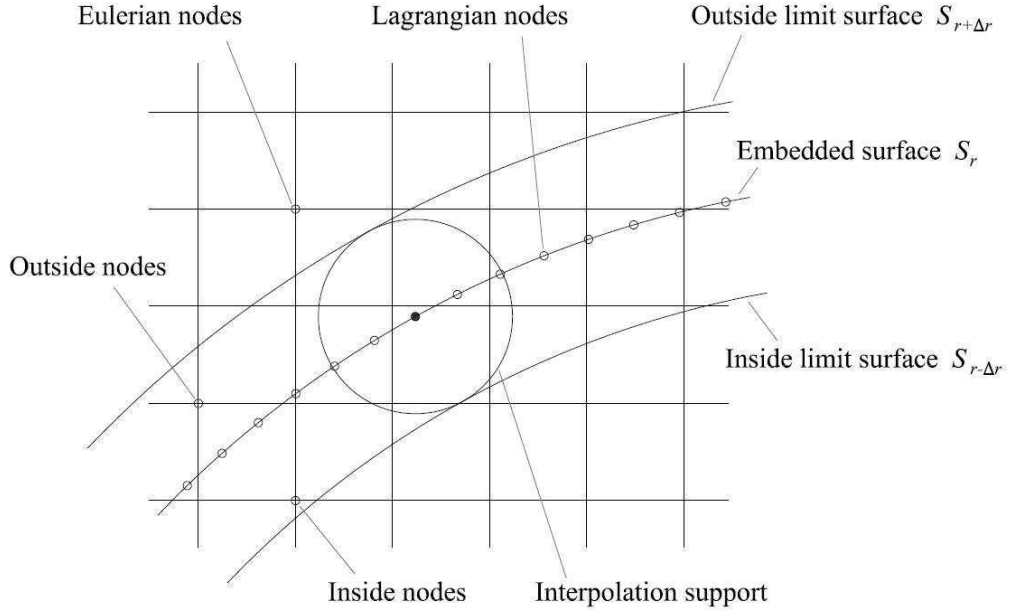


Figure 2: Diagram of the interpolation support.

adjacent to the embedded surface, using the following condition:

$$S_{r-\Delta r} < (x_{in}, y_{in}) < S_r, \quad (19)$$

Similarly, the outer nodes satisfy:

$$S_r < (x_{out}, y_{out}) < S_{r+\Delta r}. \quad (20)$$

where $S_{r+\alpha}$ is the circumference of a circle of radius $r + \alpha$ centred at (x_c, y_c) . For more general geometries, to tag the inner and outer nodes, one could resort to a level set function [28] (in the simple case of a circle given by $h(x, y) = (x - x_c)^2 + (y - y_c)^2 - r^2$) and discriminate the position according to the sign of the function (positive/negative value at point (x, y) corresponds to

an outer/inner location). The number of inner Eulerian nodes will also define the number of Lagrangian nodes used to discretise Γ . The same number will also define the size of the linear system to be solved for the interpolation-spreading process (i.e., n , the size of matrix W of equation (1)). As explained in the following, if the number of Lagrangian nodes is equal to the number of inner Eulerian nodes, the resulting linear system will have the same number of equations and unknowns (square matrix). Other choices are possible if a least square approximation of the boundary values is sought. In this case to improve the accuracy of the method, one should specify a number of Lagrangian nodes larger than the number of Eulerian nodes falling within the internal part of the annulus. The interpolation support is shown in figure 2. A number of numerical experiments has been performed considering different boundary layer thickness (i.e., Reynolds numbers of the flow around the circle) and different mesh sizes. Those numerical experiments suggest that a good compromise between stability and accuracy is attained for a number of Lagrangian nodes taken to be about three times the number of nodes in the inner region. The set of all inner points corresponds to the union of the interpolation supports of each Lagrangian node. The interpolation support for each one of them, is determined using another smaller circle centered on a point belonging to S_r and with a radius of Δr . This circle is thus tangent to both $S_{r+\Delta r}$ and $S_{r-\Delta r}$ (see figure 2). After a parametric study, the value of $\Delta r = \Delta x$ has been found to be robust over variations of the Γ geometry (circular, elliptical, with sharp arcs, etc.). Both the inner Eulerian and Lagrangian nodes that fall within this circle, constitute the inner subset of the support that is used for the interpolation step.

Once the set of all the supports is defined, we proceed to interpolate the fluid velocity onto the Lagrangian nodes using a the compact radial basis function method described in the previous section. Firstly, we evaluate the weights $\omega_j(\mathbf{X}_l)$ using (6) in conjunction with (7). Concerning the value of the parameter ϵ appearing in (7), we have found, via a parametric study, that the value $\epsilon = Re/\Delta x$ is a robust choice over variations of the Reynolds number Re . This point clearly requires further investigation. However, in this work, we simply use this empirically-determined value and found that it performs quite well in the ranger $25 < Re < 250$. During our numerical experiments we have also systematically verified that the column-sum of all weights in matrix W (i.e., equation (1)) is indeed equal to unity.

Having determined the value of the interpolating weights, the fluid velocity (computed using Eq. (13) is estimated on each node \mathbf{X}_l by the Lagrangian

interpolation formula:

$$\frac{u_l}{\Delta t} = \omega_1 \frac{u_1}{\Delta t} + \omega_2 \frac{u_2}{\Delta t} + \dots + \omega_n \frac{u_n}{\Delta t}. \quad (21)$$

As previously mentioned, the fluid velocity is decomposed into an estimated velocity u^* (computed by Eq. 13), and a regularized force f imposing the desired conditions on Γ . In the specific case of homogeneous Dirichlet conditions (no-slip: $U_{\mathbf{x}_l} = 0$ on Γ), the merging of the interpolated values with the unknown values on the support, leads to:

$$0 = \omega_{i1} \left(\frac{u_{i1}^*}{\Delta t} + f_{i1} \right) + \dots + \omega_{in} \left(\frac{u_{in}^*}{\Delta t} + f_{in} \right) + \omega_{o1} \frac{u_{o1}^*}{\Delta t} + \dots + \omega_{on} \frac{u_{on}^*}{\Delta t}. \quad (22)$$

Equation (22) is an overdetermined system of linear equations $Wf = b$, having the values of f at the internal support nodes as unknowns. The matrix W , and the right-hand-side of the system are given by:

$$W = \begin{bmatrix} \omega_{i1(l_1)} & 0 & 0 & \omega_{i4(l_1)} & 0 \\ 0 & \omega_{i2(l_2)} & 0 & \omega_{i4(l_2)} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \omega_{i1(l_{n-1})} & 0 & 0 & 0 & \omega_{i5(l_{n-1})} \\ 0 & 0 & \omega_{i3(l_n)} & 0 & \omega_{i5(l_n)} \end{bmatrix}, \quad (23)$$

and

$$b = \begin{bmatrix} -\sum \omega_i \frac{u_i^*}{\Delta t} |_{l_1} - \sum \omega_o \frac{u_o^*}{\Delta t} |_{l_1} \\ -\sum \omega_i \frac{u_i^*}{\Delta t} |_{l_2} - \sum \omega_o \frac{u_o^*}{\Delta t} |_{l_2} \\ \vdots \\ -\sum \omega_i \frac{u_i^*}{\Delta t} |_{l_{n-1}} - \sum \omega_o \frac{u_o^*}{\Delta t} |_{l_{n-1}} \\ -\sum \omega_i \frac{u_i^*}{\Delta t} |_{l_n} - \sum \omega_o \frac{u_o^*}{\Delta t} |_{l_n} \end{bmatrix} \quad (24)$$

As the matrix W is not necessarily squared, we use a classical least square method to compute the regularized force f_i :

$$W^T W f = W^T b. \quad (25)$$

Before proceeding further, a word of caution is due. We have found that when one or more external Eulerian nodes, belonging to the support, are excessively close to the embedded surface (normal distance less than $0.1\Delta x$), the weights can take on very large values leading to numerical instabilities.

This numerical issue can be fixed by assigning to those nodes the same velocity as the closest ones laying on Γ . In the case of non-regular grids, this normal distance can be computed as $0.1\Delta r$, being Δr the thickness of the interpolation support, which is defined according to the mean distance between the cells close to the embedded surface. Since the boundary is approximated as piecewise linear, the accuracy is hardly affected by dividing a segment into two parts [13]. According to Gibou et al. [27], this approach preserve second order accuracy when solving the Poisson equation in irregular domains.

3.2. Results and discussion

To validate the proposed methodology, we have considered the case of a flow around a circular cylinder of diameter D at two Reynolds numbers, $Re_D = 30$ and $Re_D = 185$. Following the numerical settings of [8], the dimensions of the domain are $L_x = 49D$ in streamwise direction and $L_y = 34D$ in the normal direction. The center of the cylinder is located at $(x, y) = (9D, 17D)$. The mesh spacing is $\Delta x = \Delta y = 0.0576D$ (Cartesian mesh).

The no-slip boundary condition at the cylinder wall are imposed using the proposed algorithm. It is found that the methodology is able to reproduce successfully the characteristics of the flow at both $Re_D = 30$ and $Re_D = 185$. Figures 3 and 4 show the velocity fields and the vorticity contours. There is a qualitative agreement with the literature: at $Re_D = 30$ the flow remains steady with the presence of a recirculating region in the wake; at $Re_D = 185$, periodic shedding vortices are formed downstream of the body (even if this case is nominally a 3D one since an instability in the spanwise direction should already be present, many authors have considered the 2D numerical study of the flow).

A more quantitative analysis is provided in Table 1 that shows comparisons with the literature on the values of the main topological parameters of the wake at $Re = 30$ (see figure 5). Table 2 gives quantitative comparisons for the higher Re , unsteady case considering the values of the Strouhal number and the drag mean coefficient obtained. A general good agreement is obtained, confirming the correct treatment of the boundary condition at the wall using the present method.

Figure 6 shows the norm of the interpolation error E_l at the immersed surface defined by

$$E_l = \max_l \{U_l\}, \quad (26)$$

where U_l refers to the interpolated velocity computed at each Lagrangian

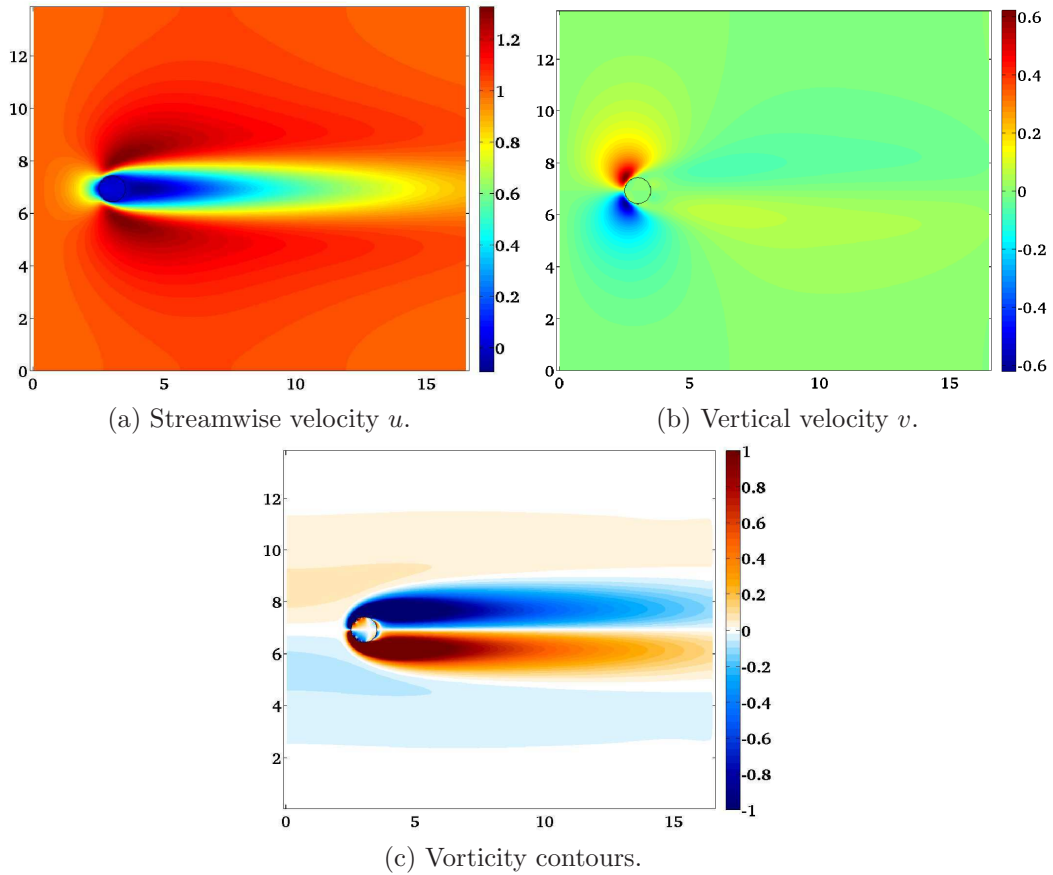


Figure 3: Mean velocity and vorticity contours at $Re_D = 30$.

	l/D	a/D	b/D	θ	C_D
Present	1.71	0.56	0.53	47.93	1.78
Pinelli et al. [8]	1.70	0.56	0.52	48.05	1.80
Coutanceau et al. [29]	1.55	0.54	0.54	50.00	-
Tritton [30]	-	-	-	-	1.74

Table 1: Comparison of the main parameters of the wake and the drag coefficient at $Re_D = 30$ with other works and experimental data.

point \mathbf{X}_l after the advancement of the second *corrected* momentum equation. From the figure, it clearly appears that the global method is of order one.

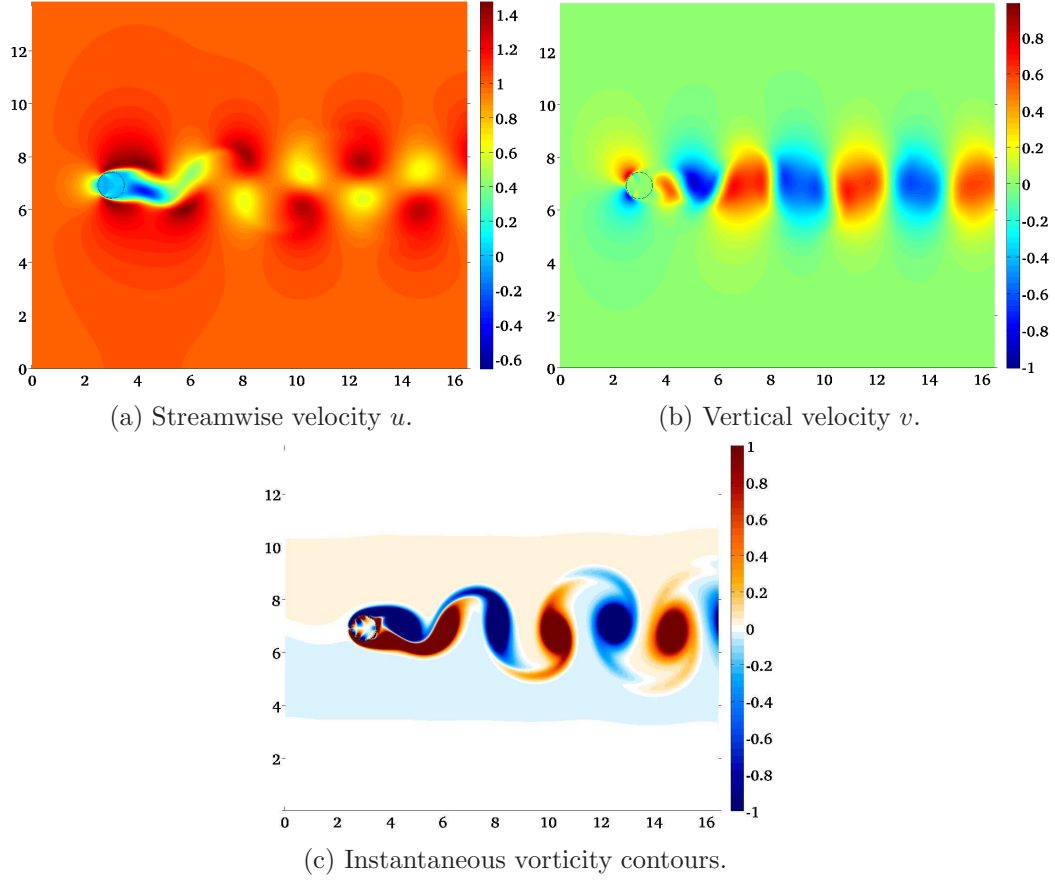


Figure 4: Instantaneous velocity and vorticity contours at $Re_D = 185$.

	St	C_D
Present	0.195	1.31
Pinelli et al. [8]	0.196	1.43
Vanella et al. [31]	-	1.38
Guilmineau et al. [32]	0.195	1.28
Lu et al. [33]	0.195	1.31

Table 2: Comparison of the Strouhal number and the drag coefficient at $Re_D = 185$ with other works and experimental data.

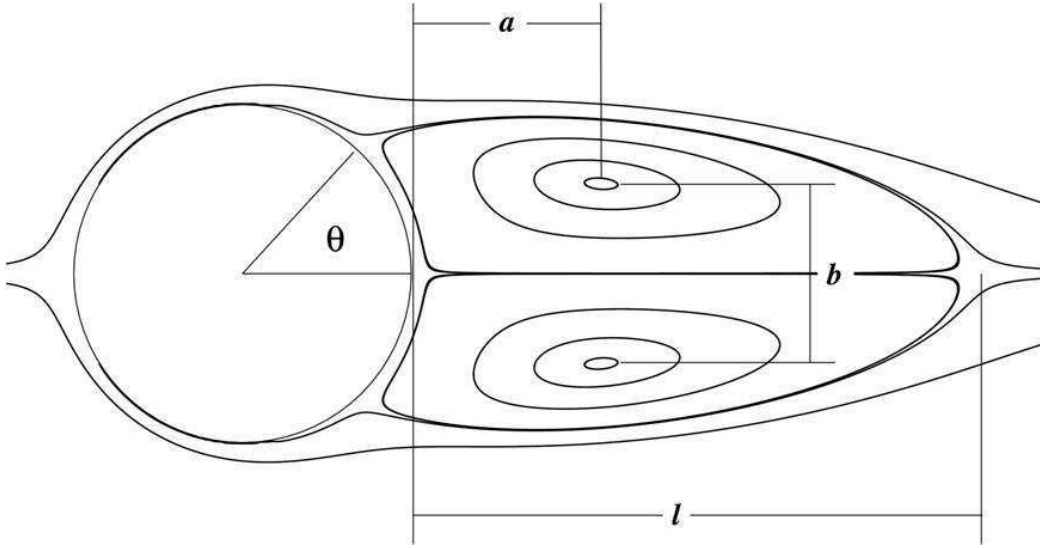


Figure 5: Shape parameters of the wake formed at $Re = 30$ [8].

4. Treatment of Neumann boundary conditions

We have tested the viability of our approach for imposing Neumann conditions by considering a simple 2D heat conduction equation around a thermally insulated cylinder.

Considering constant physical properties, and an implicit time advancing treatment, the governing equation of the heat conduction problem reads as:

$$\frac{T^{n+1} - T^n}{\Delta t} = \alpha \nabla^2 T^{n+1}, \quad (27)$$

where T is the temperature and α the thermal diffusivity, set to a value of $\alpha = 110 \cdot 10^{-6} m^2/s$ (which would correspond to copper thin plate). In what follows, we describe the method by which Neumann conditions $\frac{\partial T}{\partial \bar{n}} = 0$ are imposed at the immersed surface.

4.1. Methodology

As for the case of Dirichlet conditions, the internal and the external nodes of the Eulerian grid defined by the embedded surface are found using the level set method. The subsequent selection of the support is also performed in the same way as previously described. The choice on the number of Lagrangian

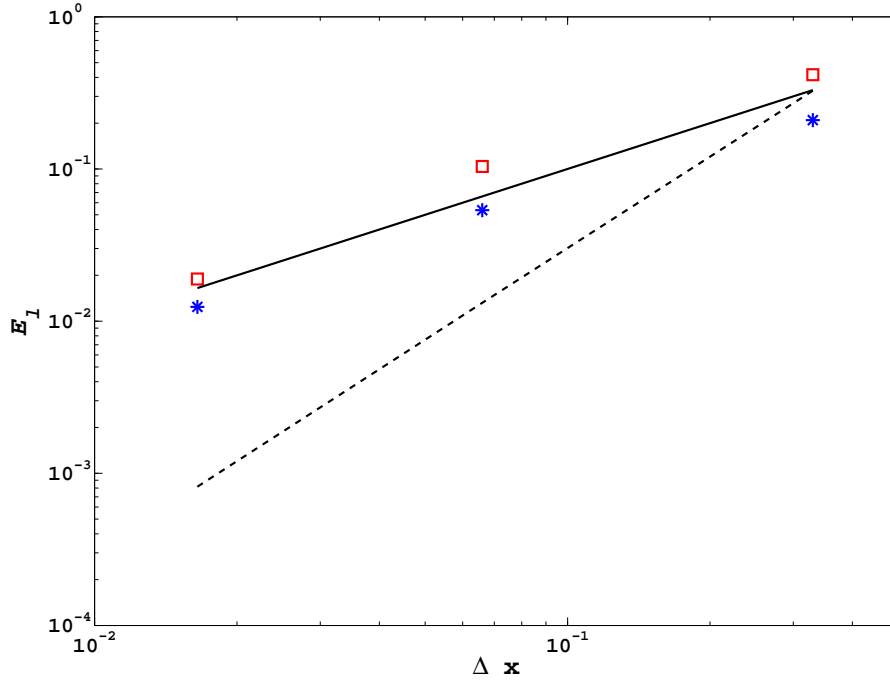


Figure 6: Norm of the interpolation error at the immersed surface for a Dirichlet boundary condition. Red squares: error u ; blue asterisks: error v ; solid line: Δx (1st order); dashed line: Δx^2 (2nd order).

nodes defining the immersed contour is more demanding for the case of Neumann conditions. We have found that the minimal number necessary to maintain the accuracy is about the double than the one required for imposing Dirichlet values.

As already mentioned, the use of radial basis functions for the interpolation of the derivative of the variable over the immersed boundary is a simple variation of the technique used for the interpolation of the values of the variable, thus rendering the method very friendly.

The first step is to evaluate the weights ω_i of the function, applying the following relationship between the Lagrangian node l and each Eulerian node $1 \dots n$ within the interpolation support:

$$\phi_{\bar{n}l,1} = \omega_0 + \omega_1\phi_{1,1} + \omega_2\phi_{1,2} + \dots + \omega_n\phi_{1,n}$$

$$\begin{aligned}
\phi_{\vec{n}l,2} &= \omega_0 + \omega_1\phi_{2,1} + \omega_2\phi_{2,2} + \dots + \omega_n\phi_{2,n} \\
&\vdots \\
\phi_{\vec{n}l,n} &= \omega_0 + \omega_1\phi_{n,1} + \omega_2\phi_{n,2} + \dots + \omega_n\phi_{n,n},
\end{aligned} \tag{28}$$

where $\phi_{i,j}$ is the radial basis function between the node i and the node j , and $\phi_{\vec{n}l,i}$ is the derivative radial basis function between the l Lagrangian node and the i Eulerian node projected on the normal surface direction \vec{n} , as previously defined. By adding the extra condition: $\omega_1 + \omega_2 + \dots + \omega_n = 0$, the following linear system of equations is obtained.

$$\begin{bmatrix} 1 & \phi_{1,2} & \dots & \phi_{1,n} & 1 \\ \phi_{2,1} & 1 & \dots & \phi_{2,n} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \phi_{n,1} & \phi_{n,2} & \dots & 1 & 1 \\ 1 & 1 & \dots & 1 & 0 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_n \\ \omega_0 \end{bmatrix} = \begin{bmatrix} \phi_{\vec{n}l,1} \\ \phi_{\vec{n}l,2} \\ \vdots \\ \phi_{\vec{n}l,n} \\ 0 \end{bmatrix}. \tag{29}$$

By interpolating the value of the temperature normal derivative at each Lagrangian node as:

$$\frac{\partial T}{\partial \vec{n}} = \omega_1 T_1 + \omega_2 T_2 + \dots + \omega_n T_n; \tag{30}$$

and following the same methodology as for the Dirichlet case, we can write the equation:

$$0 = \omega_{i1}(T_{i1}^* + \delta_{i1}) + \dots + \omega_{in}(T_{in}^* + \delta_{in}) + \omega_{o1}T_{o1}^* + \dots + \omega_{on}T_{on}^*, \tag{31}$$

In (31), T_i^* is the temperature on the boundary without having imposed any condition of the surface, and δ_i is the correction term to be applied to the nodes inside the embedded surface.

Using the same algebraic manipulations as for the Dirichlet case, we finally obtain the linear system $W f = b$, with

$$W = \begin{bmatrix} \omega_{i1(l_1)} & 0 & 0 & \omega_{i4(l_1)} & 0 \\ 0 & \omega_{i2(l_2)} & 0 & \omega_{i4(l_2)} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \omega_{i1(l_{n-1})} & 0 & 0 & 0 & \omega_{i5(l_{n-1})} \\ 0 & 0 & \omega_{i3(l_n)} & 0 & \omega_{i5(l_n)} \end{bmatrix} \tag{32}$$

and

$$b = \begin{bmatrix} -\sum \omega_i T_i^*|_{l_1} - \sum \omega_o T_o^*|_{l_1} \\ -\sum \omega_i T_i^*|_{l_2} - \sum \omega_o T_o^*|_{l_2} \\ \vdots \\ -\sum \omega_i T_i^*|_{l_{n-1}} - \sum \omega_o T_o^*|_{l_{n-1}} \\ -\sum \omega_i T_i^*|_{l_n} - \sum \omega_o T_o^*|_{l_n} \end{bmatrix} \quad (33)$$

Again, if W is not a squared matrix, the solution is found by resorting to a least square formulation: $W^T W \delta = W^T b$. We have found that the direct sum of the correction terms δ_i to the estimated temperatures T_i^* induce a strong jump close to the immersed body. To avoid this problem, the corrections δ_i are included in the governing equation of the problem (27) instead. If T_i denotes the final (i.e., that keeps into account the Neumann conditions) temperature field, the values $T_i^* = T_i - \delta_i$ are introduced into the finite difference-discretized governing equation, obtaining:

$$\begin{aligned} a_P(T_{i,j} - \delta_{i,j}) + a_S(T_{i,j-1} - \delta_{i,j-1}) + a_N(T_{i,j+1} - \delta_{i,j+1}) + \\ a_E(T_{i+1,j} - \delta_{i+1,j}) + a_W(T_{i-1,j} - \delta_{i-1,j}) = b, \end{aligned} \quad (34)$$

where a_I and b are the coefficients of the finite difference discretization of the governing equation. Separating the unknowns in Eq. (34) yields:

$$\begin{aligned} a_P T_{i,j} + a_S T_{i,j-1} + a_N T_{i,j+1} + a_E T_{i+1,j} + a_W T_{i-1,j} = \\ b + a_P \delta_{i,j} + a_S \delta_{i,j-1} + a_N \delta_{i,j+1} + a_E \delta_{i+1,j} + a_W \delta_{i-1,j}. \end{aligned} \quad (35)$$

which, finally gives the linear system $A T = b$, where A is the same matrix as the first linear system that must be solved to obtain the estimated temperature field (i.e., standard finite difference Laplacian matrix).

4.2. Results and discussion

Following the procedure described above, we compute the temperature distribution within a section of an infinitely long bar. The boundary on the right (east, E) is a perfectly insulated edge while the other three edges are maintained at a prescribed temperature value: $T_N = 100$, $T_S = 25$ and $T_W = 75$. Figure 7a shows the temperature field obtained for the whole domain without having introduced any immersed boundary. In the same figure, the location of the immersed boundary that has been considered as a validation experiment has been introduced for illustrative purposes. Figure 7b shows the temperature field obtained when the insulated condition is applied

using our immersed boundary method. As expected, the temperature isolines approach the embedded surface in an orthogonal fashion (left hand side of the embedded surface).

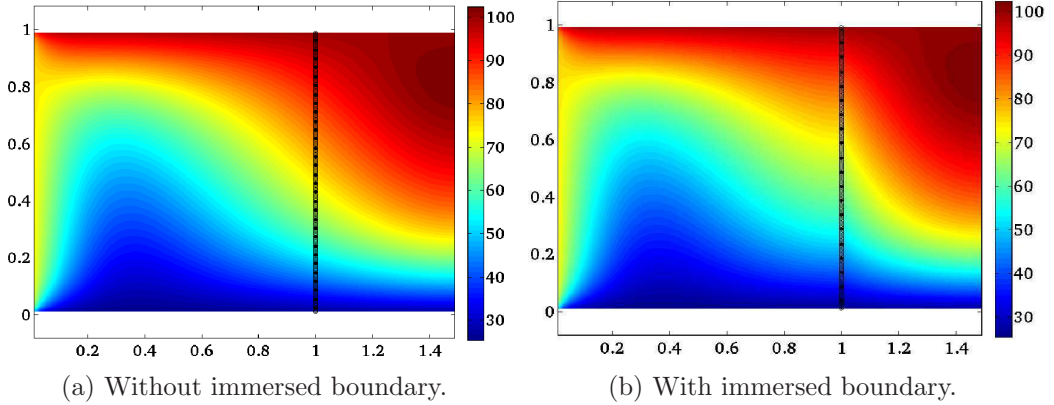


Figure 7: Final temperature field $T(x, y)$. External boundary conditions: $T_N = 100$, $T_S = 25$, $T_W = 75$ and the right hand side wall is considered adiabatic.

To complete the validation, the same problem has been tackled using a body conforming mesh. Figure 8 shows a longitudinal section in x direction at the center of the domain, where it can be seen that the results of both methodologies, body conformal (blue line) and immersed boundary using radial basis function (black line), fully agree.

Furthermore, figure 9 gives the values of the normal derivative of the temperature field just outside the vertical wall, using finite differences. The values of this parameter approach to zero (black squares) in front of the values without considering the adiabatic embedded surface (red circles).

Figure 10 shows the norm of the interpolation error E_d at the immersed surface defined by

$$E_d = \max_l U_d, \quad (36)$$

where U_d refers to the interpolated temperature derivative from the external side computed at each Lagrangian point l . It results a decreasing of order Δx (1st order).

The method was also checked for other geometries as, for example, a circular adiabatic surface (see figure 11).

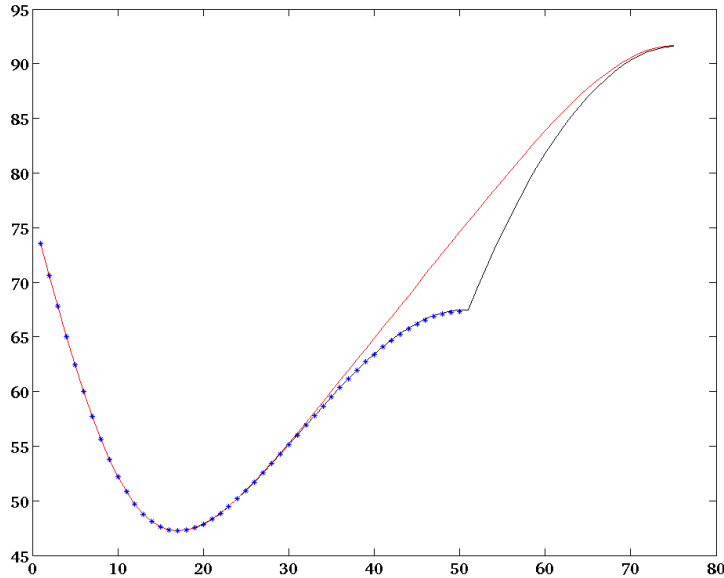


Figure 8: Streamwise section of the temperature field at the center of the domain. Red line: without considering the adiabatic embedded surface. Black line: considering the adiabatic embedded surface. Blue line: body conformal case.

5. Conclusions

In this work, we have presented a new interpolation and spreading procedure based on radial basis functions. The procedure is easy to implement and allows the imposition of both Dirichlet and Neumann boundary conditions. Both interpolation and spreading tasks are carried out together within the same stage, which yields a straightforward implementation. As the radial basis functions require a scattered cloud of points, another advantage is that the method works for any type of mesh, even unstructured.

To validate the method, a Dirichlet boundary condition has been imposed on a 2D cylinder geometry in a Navier-Stokes CFD solver, and a Neumann boundary condition has been imposed in an adiabatic embedded surface in an unsteady heat conduction problem. The obtained results agree with literature results.

The convergence rate of this method is one. This value is expected to be

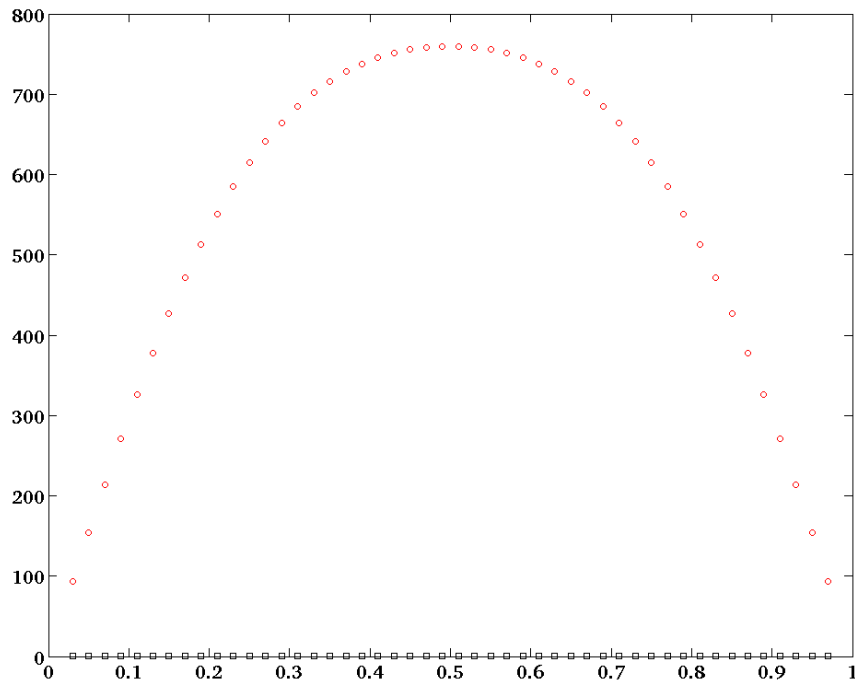


Figure 9: Normal derivative of the closest temperature field outside the vertical wall, in front of the height. Red circles: without considering the adiabatic embedded surface. Black squares: considering the adiabatic embedded surface.

increased by future works on the interpolation support. This new interpolation support should use more external nodes, since the first order accuracy derives from the order of the interpolation process. The most important challenge is to define a simple rule to select the appropriated nodes that will be able to deal with any surface shape.

A first version of the method is presented in this work, showing to be robust over variations of geometry and Reynolds number (staying in laminar flows). As deeper improvements of the method, further investigations on the size and shape of the interpolation support are expected, as well as the use of other kinds of RBFs and other ways to determine the shape parameter. Finding the optimal number of Lagrangian nodes for different geometries

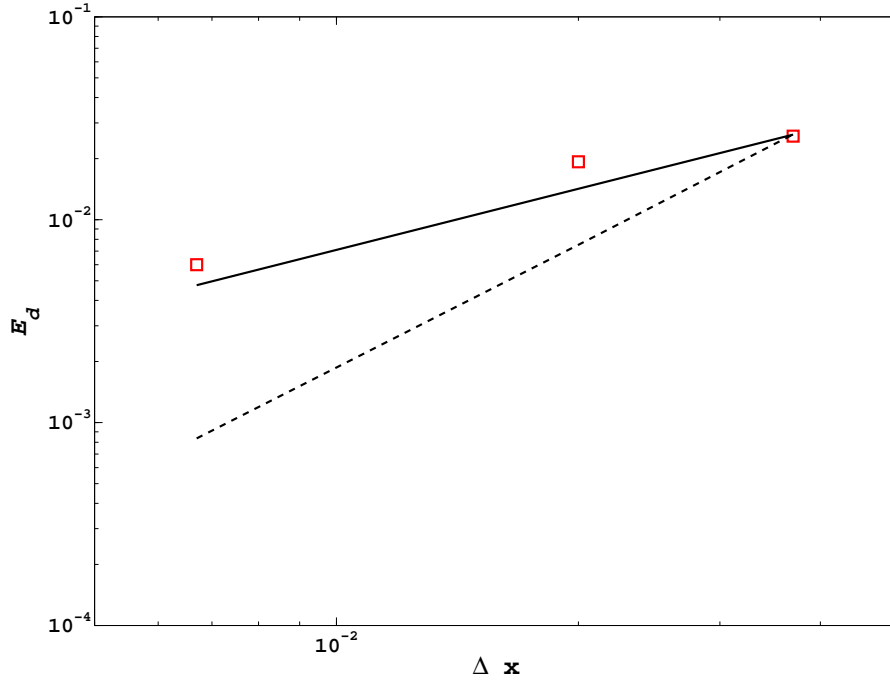


Figure 10: Norm of the interpolation error of the derivative at the immersed surface. Red squares: derivative error; solid line: Δx (1st order); dashed line: Δx^2 (2nd order).

is also a remaining issue, which is also encountered in other methods in literature.

Short-term applications of the work also includes the imposition of Neumann boundary conditions on the Poisson equation in a CFD code, and the use of the present method to implement wind turbine models such as actuator disc and actuator line ones.

6. Acknowledgments

The authors want to acknowledge the detailed and constructive suggestions of the reviewers, which significantly improved the quality of this work.

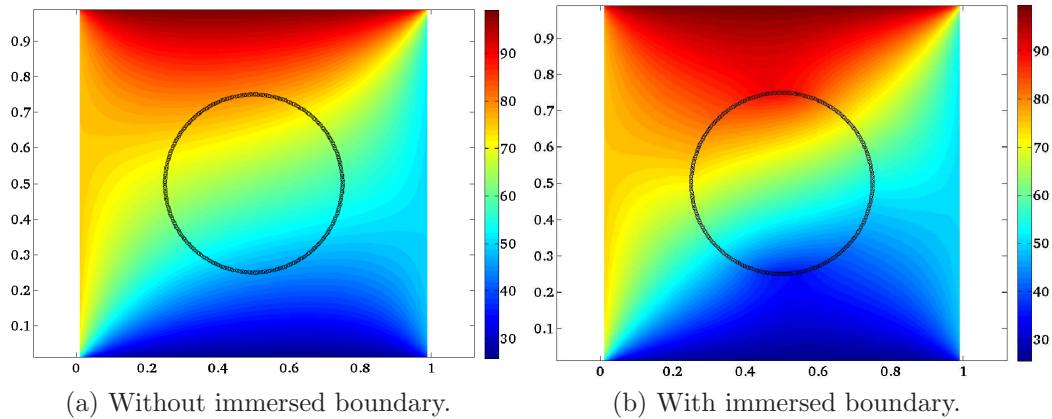


Figure 11: Temperature field $T(x, y)$. External boundary conditions: $T_N = 100$, $T_S = 25$, $T_W = 75$ and $T_E = 50$.

References

- [1] L. Ji, K. Sreenivas, D. G. Hyams, R. V. Wilson. A parallel universal mesh deformation scheme for hydrodynamic applications. 28th Symposium on Naval Hydrodynamics, Pasadena (USA), 12-17 September 2010.
- [2] L. Zhang, A. Gerstenberger, X. Wang, W. K. Liu. Immersed finite element method. *Computer Methods in Applied Mechanics and Engineering*, 193 (2004): 2051-2067.
- [3] C. S. Peskin. Flow patterns around heart valves: a numerical method. *Journal of Computational Physics*, 10 (1972): 252-271.
- [4] C. S. Peskin. The immersed boundary method. *Acta Numerica*, 11 (2002): 479-517.
- [5] R. Mittal, G. Iaccarino. Immersed boundary method. *Annual Review of Fluid Mechanics*, 37 (2005): 239-261.
- [6] Z. Li, M. Lai. The immersed interface method for the Navier-Stokes equations with singular forces. *Journal of Computational Physics*, 171 (2001): 822-842.
- [7] K. Taira, T. Colonius. The immersed boundary method: a projection approach. *Journal of Computational Physics*, 10 (2007): 252-271.

- [8] A. Pinelli, I. Z. Naqavi, U. Piomelli, J. Favier. Immersed-boundary methods for general finite-difference and finite-volume Navier-Stokes solvers. *Journal of Computational Physics*, 229 (2010): 9073-9091.
- [9] J. van Kan. A second-order accurate pressure-correction scheme for viscous incompressible flow. *SIAM Journal on Scientific and Statistical Computing*, 7 (1986): 870-891.
- [10] D. L. Brown, R. Cortez, M. L. Minion. Accurate projection methods for the incompressible Navier-Stokes equations. *Journal of Computational Physics*, 168 (2001): 464-499.
- [11] J.H. Ferziger, M. Peric. *Computational methods for fluid dynamics*. 3rd Ed. (2002) Springer.
- [12] E. A. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof. Combined immersed-boundary finite-difference methods for three dimensional complex flow simulations. *Journal of Computational Physics*, 161 (2000): 35-60.
- [13] Y. Tseng, J. H. Ferziger. A ghost-cell immersed boundary method for flow in complex geometry. *Journal of Computational Physics*, 192 (2003): 593-623.
- [14] R. Mittal, H. Dong, M. Bozkurttas, F. M. Najjar, A. Vargas, A. von Loebbecke. A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries. *Journal of Computational Physics*, 227 (2008): 4825-4852.
- [15] J. Lee, D. You. An implicit ghost-cell immersed boundary method for simulations of moving body problems with control of spurious force oscillations. *Journal of Computational Physics*, accepted manuscript (2012), doi: <http://dx.doi.org/10.1016/j.jcp.2012.08.044>
- [16] L. Yao, A. L. Fogelson. Simulations of chemical transport and reaction in a suspension of cells I: an augmented forcing point method for the stationary case. *International Journal for Numerical Methods in Fluids*, 69 (2012): 1736-1752.

- [17] N. Thai-Quang, N. Mai-Duy, C. D. Tran, T. Tran-Cong. A direct forcing immersed boundary method employed with compact integrated RBF approximations for heat transfer and fluid flow problems. *Computer Modeling in Engineering and Sciences*, 96 (2013): 49-90.
- [18] V. Shankar, G. B. Wright, A. L. Fogelson, R. M. Kirby. A radial basis function (RBF)-finite difference method for the simulation of reaction-diffusion equations on stationary platelets within the augmented forcing method. *International Journal for Numerical Methods in Fluids*, 75 (2014): 1-22.
- [19] A. de Boer, M. van der Schoot, H. Bijl. Mesh deformation based on radial basis function interpolation. *Computers and Structures*, 85 (2007): 784-795.
- [20] A. de Boer, A. van Zuijlen, H. Bijl. Review of coupling methods for non-matching meshes. *Computer Methods in Applied Mechanics and Engineering*, 196 (2007): 1515-1525.
- [21] A. Beckert, H. Wendland. Multivariate interpolation for fluid-structure-interaction problems using radial basis functions. *Aerospace Science and Technology*, 5 (2001): 125-134.
- [22] N. Mai-Duy, T. Tran-Cong. A Cartesian-grid collocation method based on radial-basis-function networks for solving PDEs in irregular domains. *Numerical Methods for Partial Differential Equations*, Wiley Online Library, 23 (2007): 1192-1210.
- [23] J. Fang, M. Diebold, C. Higgins, M. B. Parlange. Towards oscillation-free implementation of the immersed boundary method with spectral-like methods. *Journal of Computational Physics*, 230 (2011): 8179-8191.
- [24] V. Shankar, G. B. Wright, R. M. Kirby, A. L. Fogelson. Augmenting the immersed boundary method with radial basis functions (RBFs) for the modeling of platelets in hemodynamic flows. *arXiv preprint* (2013), arXiv: 1304.7479.
- [25] J. Liu, N. Zhao, O. Hu, M. Goman, X. K. Li. A new immersed boundary method for compressible Navier-Stokes equations. *International Journal of Computational Fluid Dynamics*, Taylor and Francis, 27 (2013): 151-163.

- [26] H. Wendland. Scattered data approximation. Cambridge monographs on applied and computational mathematics. Cambridge: Cambridge University Press, 2005.
- [27] F. Gibou, R. P. Fedkiw, L. T. Cheng, M. Kang. A second-order-accurate symmetric discretization of the Poisson equation on irregular domains. *Journal of Computational Physics*, 176 (2002): 205-227.
- [28] Z. Li, K. Ito. The immersed interface method. Numerical solutions of PDEs involving interfaces and irregular domains. Philadelphia: Society for Industrial and Applied Mathematics, 2006.
- [29] M. Coutanceau, R. Bouard. Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation. Part 1. Steady flow. *Journal of Fluid Mechanics*, 79 (2) (1977): 231-256.
- [30] D. J. Triton. Experiments on the flow past a circular cylinder at low Reynolds numbers. *Journal of Fluid Mechanics*, 6 (1959): 547-567.
- [31] M. Vanella, E. Balaras. A moving-least-squares reconstruction for embedded-boundary formulations. *Journal of Computational Physics*, 228 (18) (2009): 6617-6628.
- [32] E. Guilmineau, P. Queutey. A numerical simulation of vortex shedding from an oscillating circular cylinder. *Journal of Fluids and Structures*, 16 (6) (2002): 773-794.
- [33] X. Y. Lu, C. Dalton. Calculation of the timing of vortex formation from an oscillating cylinder. *Journal of Fluids and Structures*, 10 (5) (1996): 527-541.
- [34] E. J. Kansa. Multiquadrics-A scattered data approximation scheme with applications to computational fluid-dynamicsII solutions to parabolic, hyperbolic and elliptic partial differential equations. *Computers & Mathematics with Applications*, 19 (1990): 147-161.
- [35] J. Favier, A. Revell, A. Pinelli. A Lattice Boltzmann Immersed Boundary method to simulate the fluid interaction with moving and slender flexible objects. *Journal of Computational Physics*, 261 (2014): 145-161.

- [36] V. Casalegno, P. Vavassori, M. Valle, M. Ferraris, M. Salvo, G. Pintsuk. Measurement of thermal properties of a ceramic/metal joint by laser flash method. *Journal of Nuclear Materials*, 407 (2010): 83-87.