Contents lists available at ScienceDirect



International Journal of Heat and Mass Transfer

journal homepage: www.elsevier.com/locate/ijhmt

High order ghost-cell immersed boundary method for generalized boundary conditions



Mehrdad Yousefzadeh, Ilenia Battiato*

Energy Resources Engineering, Stanford University, Stanford 94305, United States

ARTICLE INFO

Article history: Received 21 November 2018 Received in revised form 8 March 2019 Accepted 11 March 2019

Keywords: Immersed boundary method Fluid-solid interaction High order discretization Robin boundary condition

ABSTRACT

Flow and reactive transport problems in engineering, medical and environmental applications often involve complex geometries. Grid based methods (e.g. finite volume, finite element, etc.) are a vital tool for studying such problems. Cartesian grids are one of the most attractive options as they possess simple discretization stencils and are usually straightforward to generate at roughly no computational cost. The Immersed Boundary Method, a Cartesian based methodology, maintains most of the useful features of structured grids, while it exhibits a great resilience in dealing with complex geometries. These features make it increasingly more attractive to model transport in evolving porous media as the cost of grid generation reduces greatly. Yet, stability issues due to the geometry of the interpolation stencil combined with limited studies on the implementation of Neumann (constant flux) and linear Robin (e.g. reaction) boundary conditions have significantly limited its applicability to transport in complex topologies. We develop a high-order compact Cartesian model based on ghost cell immersed boundary method for incompressible flow and scalar transport subject to different boundary conditions. The accuracy test shows at least second order of accuracy in L_1, L_2 and L_∞ norms of error. The proposed method is capable of accurately capturing the transport physics near the boundaries for Dirichlet, Neumann and Robin boundary conditions. We tested the method for several transport and flow scenarios, including heat transfer close to an immersed object and mass transport over reactive surfaces.

© 2019 Elsevier Ltd. All rights reserved.

1. Introduction

The ability to handle complex geometries has been a constant challenge for the successful implementation of numerical discretization schemes. Typically, a body fitted mesh (structured or unstructured) is needed to express the boundaries of a body immersed in a fluid domain. Nevertheless, generating a body fitted mesh for arbitrarily complex boundaries, as those typical of porous media systems, is not only non-trivial, but also computationally expensive. Au contraire, Cartesian grid-based methods can serve as a promising alternative to body fitted grid methods. Cartesian grids have two compelling features which render them attractive: first, their generation is performed at relatively no cost; secondly, since PDEs discretization is less challenging, coding and implementation efforts are greatly alleviated.

Immersed Boundary Methods (IBMs), a class of Cartesian grid approaches first introduced by C. Peskin in 1970s [1,2], are based

* Corresponding author. E-mail address: ibattiat@stanford.edu (I. Battiato).

https://doi.org/10.1016/j.ijheatmasstransfer.2019.03.061 0017-9310/© 2019 Elsevier Ltd. All rights reserved. on the idea of adding a force term to the momentum equation in order to mimic the effects that solid boundaries exert on flow. A detailed discussion of IBMs is provided in [3,4]. Since the early work of Peskin, extensions of IBMs have been the subject of many studies. Unlike the flow problems, development of proper IBMs to model transport of a scalar field subject to different boundary conditions is yet an open challenge [5]. Early efforts have been mainly focused on developing accurate IBMs to enforce Dirichlet and Neumann boundary conditions [6–11]. The appropriate formulation of IBMs for more complicated Robin boundary conditions, which frequently arise in conjugate boundary conditions in heat transfer [12] and surface reaction in mass transfer problems [13–16], has been limited to a few works of [17–20].

The ghost cell immersed boundary method (GCIBMs) is a powerful platform to extend IBMs to different flow and transport problems involving undeformable boundaries. The concept of the GCIBMs relies on the early work of Mohd-Yusof [21] as well as [22,23]. In GCIBM, the interface separating the solid body from the liquid is treated sharply and the force is not calculated explicitly. Instead by introducing specific algebraic constraints for the ghost cells, which are the closest solid cells to the liquid cells, the effects of boundary conditions are imposed on the flow [24,25,4]. The algebraic constrains at the ghost cells are obtained by extrapolating the state variable through a polynomial passing through the boundary and fluid points close to the boundary. The fluid point used in the construction of the polynomial is the mirror of the ghost point with respect to the boundary. This ensures that the stability issues caused by the large negative coefficients in the extrapolation formula are avoided. Several interpolating [26,27,19,20,28,18,29–31] and extrapolating formulations [25,4] have been developed to calculate the value at the mirror point and ghost cells, respectively.

Here, we develop and validate a high order ghost cell immersed boundary method that is able to handle Dirichlet, Neumann and Robin boundary conditions. The algebraic equations for the ghost cells ensure that the flow solver senses the existence of the boundary conditions. The interpolation introduced to obtain the mirror point has a compact stencil to ensure local reconstruction of the variable at the ghost cell. Unlike [18–20], the interpolation stencil is compact to avoid unnecessary extension of the discretization stencil. The adopted stencil also removes the arbitrary choice of the distance between the mirror point and the boundary, as in [18,19]. Furthermore, the proposed formulation (i) ensures the formal accuracy of the scheme for different boundary conditions, i.e. the introduction of the IBM treatment does not affect the order of accuracy and, (ii) unlike for [19,18], the immersed boundary formulation is not the limiting factor to achieve higher (i.e. third) order implementation of the equations since the biquadratic interpolation used will maintain third order of accuracy for the ghost points.

The manuscript is organized as follows. In Section 2, we describe the development of the immersed boundary treatment. The proposed GCIBM is validated for flow and heat transfer over a circular cylinder in Section 3 and the method capability to handle different boundary conditions is verified. We conclude and summarize our results in Section 4.

2. Numerical methodology

2.1. Governing equations

We consider the flow of a Newtonian, incompressible fluid with constant properties around a solid undeformable body. The spatio-temporal distribution of fluid velocity $\mathbf{u}(\mathbf{x},t) = (u_1(\mathbf{x},t), u_2(\mathbf{x},t), u_3(\mathbf{x},t))$ and pressure *p* is governed by the dimensionless incompressible Navier-Stokes and continuity equations

$$\frac{\partial u_i}{\partial t} + \frac{\partial u_i u_j}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{1}{Re} \frac{\partial}{\partial x_j} \left(\frac{\partial u_i}{\partial x_j} \right), \quad \frac{\partial u_i}{\partial x_i} = 0, \quad i, j = \{1, 2, 3\}, \quad (1)$$

where $\mathbf{x} = (x_1, x_2, x_3)$ and *t* are the dimensionless spatial and temporal coordinates, \mathbf{u} and *p* are the dimensionless velocity vector and fluid pressure, respectively, and *Re* is the Reynolds number. In (1), and thereafter, Einstein summation convention is assumed whenever a repeated index appears. Eq. (1) is subject to the no-slip boundary condition at the fluid-solid interface, Γ_{ib} ,

$$u_i = 0, \quad \mathbf{x} \in \Gamma_{ib}.$$
 (2)

The transport of a scalar field ψ , e.g. temperature, concentration, etc., is governed by a dimensionless advection-diffusion equation

$$\frac{\partial \psi}{\partial t} + \frac{\partial (u_j \psi)}{\partial x_j} = D \frac{\partial}{\partial x_j} \left(\frac{\partial \psi}{\partial x_j} \right),\tag{3}$$

where the coefficient $D = (PrRe)^{-1}$ or $D = Pe^{-1}$ in heat transfer and mass transport cases, respectively, with *Pe* and *Pr* the Peclet and Prandtl numbers.

Eq. (3) is subject to the general boundary condition on the solid-fluid interface

$$-n_i \alpha \frac{\partial \psi}{\partial \mathbf{x}_i} = \beta \psi + q, \quad \mathbf{x} \in \Gamma_{ib}, \tag{4}$$

where **n** is the normal vector to the solid surface pointing toward the fluid domain. The coefficients α , β and q allow one to account for different types of boundary conditions: Dirichlet, Neumann or Robin. When $\alpha = 0$ and $\beta = 0$ boundary conditions of the first and second kind, e.g., isothermal or isoflux surfaces, are obtained, respectively. When $\alpha \neq 0$ or $\beta \neq 0$, a boundary condition of the third kind is recovered. The latter is routinely encountered in heat transfer or mass transport problems such as convective or linear surface reaction boundary conditions. The discretization scheme is presented in the following section.

2.2. Finite volume discretization

Eqs. (1) and (3) are discretized using a finite-volume scheme [32,33] on a structured, non-uniform Cartesian grid. In order to avoid the stability problems related to larger time steps, all variables are temporally updated by utilizing a fully implicit backward Euler time integration scheme.

The Navier-Stokes Eq. (1) are solved in the primitive variables (u, p) formulation. The primitive variables are defined on a staggered grid to avoid odd-even decoupling between pressure and velocity [34]. The velocity components are evaluated on the control volume faces, while all other variables (e.g. pressure, temperature, concentration) are located at the cell centers. Hence, three different Cartesian grids are required in a two dimensional problem. The SIMPLE algorithm [35] is adopted for pressure-velocity coupling. SIMPLE is an iterative projection method to obtain the correct pressure after the initial guess of the velocity field, so that velocity and pressure satisfy both the momentum and continuity equations:

$$\frac{u^{n+1} - u^n}{\Delta t} = \{C(u^*) + E\}(u^{n+1}) + Gp^*,\tag{5}$$

where the superscripts correspond to function evaluations at times n + 1 and n, while u^* and p^* are the velocity and pressure values from the previous SIMPLE iteration. In (5), the operators C, E and G are defined as

$$C(u^*) = u_i^* \frac{\delta}{\delta x_i},\tag{6}$$

$$E = \frac{\delta}{\delta \mathbf{x}_i} \left(\frac{\delta}{\delta \mathbf{x}_i} \right),\tag{7}$$

$$G = \frac{\partial}{\partial \mathbf{x}_i}.$$
 (8)

The convection and viscous terms are spatially discretized using third order upwinding (QUICK) [36,37] and second order central difference schemes, respectively. We adopt a similar spatial discretization for the convection and diffusion terms in the scalar transport equation (3).

2.3. Immersed boundary treatment via ghost cells

The present algorithm employs the concept of ghost cells to accurately implement boundary conditions. The GCIB method is based on the preliminary classification of all nodes into fluid, solid and ghost points, whose values are obtained using different strategies outlined below. Fluid cells are all the nodes whose centers are located inside the fluid; solid cells have their centers inside the solids and are not directly adjacent to a fluid cell; ghost cells are the cells inside the solid with at least one neighboring cell inside the fluid [25]. A classification of points among these three classes, after which the cells are flagged, is achieved through a *ray-casting* method based on Jordan curve theorem: since the solid-fluid boundary is a closed and not self-intersecting curve, the method is based on emanating a ray from the origin of the coordinate system (given the origin lies in the fluid domain), or any point inside the fluid and far from the boundary, to all cell centers. If the cell center is in the flow domain, then the ray will intersect the boundary an even number of times. The opposite is true for solid points. The additional case in which the cell center lays on the boundary needs to be explicitly taken into account. Algorithm 1 features the detailed steps of the ray-casting process.

Algorithm 1. Point Identification Algorithm

A mirror point, $X_m = (x_m, y_m)$, is a point that mirrors the ghost point location along the normal line across the solid boundary, identified by **n**, i.e. a ghost point and its corresponding mirror point are equidistant from the boundary. The normal line to the boundary connecting a ghost point and its corresponding mirror point intersects the boundary at the boundary point, $X_{ib} = (x_{ib}, y_{ib})$.

Once the mirror point is found, a unique second order polynomial passing through the mirror point, the ghost cell center and the boundary point can be obtained. Three equations are required to uniquely define the unknown coefficients of the polynomial. We use the values of the state variable and its derivative at the mirror point, as well as the boundary equation at X_{ib} for this purpose.

1: f	unction PointID(X, Y, Γ_{ib})	▷ The type of each grid point
2:	for all $(x_i, y_j) \in \{(X, Y)\}$ do	
3:	$L \leftarrow$ a ray from origin to $\left(x_i, y_j ight)$	
4:	$n \leftarrow$ number of intersections of L with Γ_{ib}	
5:	if <i>n</i> is even then	
6:	$F(i,j) \leftarrow 1$	b flag value 1 is for fluid points
7:	$\left(x_{i},y_{j} ight) \in\left(X_{f},Y_{f} ight)$	
8:	else if	
9:	then $F(i,j) \leftarrow -1$	\triangleright flag value -1 is for solid points
10:	$\left(x_{i}, y_{j}\right) \in \left(X_{s}, Y_{s}\right)$	
11:	for all $(x_i, y_j) \in (X_s, Y_s)$ do	
12:	if $F(i+1,j) + F(i-1,j) + F(i,j+1) + F(i,j-1) \neq -4$ then	
13:	$F(i,j) \leftarrow 0$	b flag value 0 is for ghost points
14:	$\left(\pmb{x}_{i}, \pmb{y}_{j} ight) \in \left(\pmb{X}_{m{g}}, \pmb{Y}_{m{g}} ight)$	
15:	return F	

Once the tagging process is completed (as in Fig. 1), an appropriate formulation for each class of cells is needed. Specifically, each variable in the fluid cells is obtained by solving an algebraic equation obtained by discretizing the governing PDEs. Values in the solid cells are set through an algebraic equation of the type ϕ = constant: this allows one to ease the implementation and complete the final discretization matrix; in alternative, one may choose to exclude the solid cells from the computation since they are completely decoupled from the rest of the domain. Finally, the algebraic equation for the ghost cell values is constructed in such a way that their coupling with the fluid cells will guarantee that the boundary conditions are satisfied. The boundary condition treatment by means of ghost cells is described in detail in the following section.

2.3.1. Ghost-cell immersed boundary formulation

The GCIB method is routinely employed to enforce boundary conditions at solid-liquid interfaces. The successful implicit implementation of the boundary condition in a ghost cell framework requires the formulation of an explicit algebraic equation for the ghost cells. This reconstruction is carried out through a quadratic extrapolating polynomial between the boundary and a so-called *mirror point* inside the fluid. Although this boundary reconstruction is similar to that implemented in [18], its major difference is in the way the state variables and their derivatives values are reconstructed. An alternative approach is to find two mirror points in the normal direction and use them together with the boundary condition to construct the extrapolating polynomial [19,20]. Yet, in the latter scenario, the polynomial approximation of the local behavior of the boundary condition deteriorates when the mirror points are far from the boundary. Consequently, a compact discretization stencil is better achieved by using a single mirror point value as outlined below.

Let ϕ be the state variable to be approximated by the polynomial $P_1(r_n)$, i.e., ϕ represents any of the fluid variables (e.g., velocity, temperature, etc.). We start from the boundary condition (4) for ϕ . Eq. (4) can be also rewritten as

$$-\alpha \frac{\partial \phi}{\partial r_n} = \beta \phi + q, \quad r_n = 0.$$
(9)

The polynomial $P_1(r_n)$ estimates ϕ as $\phi \approx P_1(r_n) + O(r_n^3)$, i.e. $P_1(r_n)$ satisfies

$$P_1(r_n) = a_2 r_n^2 + a_1 r_n + a_0, \tag{10a}$$

$$\frac{dP_1}{dr_n} = 2a_2r_n + a_1. \tag{10b}$$



Fig. 1. A 2D schematic representation of different type of the points in the domain.

The objective is to find the vector of unknown coefficients, $\boldsymbol{a} = [a_2 \ a_1 \ a_0]^T$. This is obtained by solving the following system of equations

$$\begin{pmatrix} 2d & 1 & 0\\ d^2 & d & 1\\ 0 & -\alpha & -\beta \end{pmatrix} \begin{pmatrix} a_2\\ a_1\\ a_0 \end{pmatrix} = \begin{pmatrix} \frac{\partial \phi}{\partial r_n} \\ \phi_m\\ q \end{pmatrix}, \tag{11}$$

where $d = |X_m - X_{ib}|$ is the distance between the mirror point and the boundary point. The first two equations in (11) account for the value of ϕ and its derivative in the normal direction, while the last equation corresponds to the boundary condition at X_{ib} where $r_n = 0$.

For Neumann and Robin boundary conditions, when $\alpha \neq 0$, (11) yields the following explicit expressions for the coefficient vector **a** in terms of ϕ_m and $\frac{\partial \phi}{\partial r_n}$,

$$\begin{pmatrix} a_{2} \\ a_{1} \\ a_{0} \end{pmatrix} = \frac{1}{2 - \xi} \begin{pmatrix} d^{-2} \left[\eta + \xi \phi_{m} + d(1 - \xi) \frac{\partial \phi}{\partial r_{n}} \Big|_{m} \right] \\ d^{-1} \left(-2\eta - 2\xi \phi_{m} + \xi d \frac{\partial \phi}{\partial r_{n}} \Big|_{m} \right) \\ \eta + 2\phi_{m} - d \frac{\partial \phi}{\partial r_{n}} \Big|_{m} \end{pmatrix}$$
(12)

with $\xi = \beta d/\alpha$ and $\eta = qd/\alpha$. Furthermore, to guarantee that the system of equations is not ill-conditioned, the denominator $(2 - \xi)$ should always be positive, i.e. $\xi = \beta d/\alpha < 2$ or $d < 2\alpha/\beta$: this suggests there exists an upper bound for the grid size close to the solid-fluid boundary since the value of *d* is directly related to the local grid size. The variable ξ can be thought of as a numerical Damköhler or Nusselt number.

Once $P_1(r_n)$ is determined, the variable at the ghost cell center can be obtained as follows

$$\phi_g = a_2 r_g^2 - a_1 r_g + a_0, \quad r_g = |X_g - X_{ib}|. \tag{13}$$

The mirror point usually does not coincide with the grid points, and it is interpolated from its surrounding fluid cells. Since the vector $\boldsymbol{a} = [a_2 \ a_1 \ a_0]^T$ is function of the mirror point and its derivative, Eq. (13) provides an algebraic expression for ϕ_g in terms of values of ϕ at the fluid cells.

A bilinear interpolation, that uses four function evaluations at surrounding nodes, is a common practice in the context of the ghost cell method to determine the function value at the mirror point [18,38,19,27,4]. This approach has, however, a number of shortcomings. First of all, if X_g is close to the boundary, it may not be enclosed by four fluid cell centers with one of the four points being the ghost cell itself, see Fig. 3. Using the ghost cell value itself in the interpolation would lead to an ill-posed system of equations, which negatively affects both convergence and stability [27,25]. This problem has been addressed by devising different ad hoc solutions. In [18-20] the mirror point is relocated by increasing d, so it will not be surrounded by the ghost cell value. In [18], the author suggests to set $d = \sqrt{2}\Delta x$, where d is essentially treated as a free parameter: this renders the accuracy of the reconstruction a function of the parameter value. Another method is to use the boundary condition in the bilinear interpolation [27], and to approximate the derivative in (9) with a lower order of accuracy. Additionally, the bilinear interpolation approximates the value of the derivative at the mirror point with first order accuracy: this results in the overall accuracy reduction of the scheme.

A higher order interpolation is required to describe the flow variables accurately and to maintain the formal order of accuracy of the method. Therefore, in the following section we introduce a computationally efficient bi-quadratic interpolation to determine ϕ_m , while its uniqueness and existence are studied theoretically.



Fig. 2. The interpolation stencil for the first scenario, where all the enclosing points of the mirror point are fluid type.

2.3.2. Bi-quadratic interpolation

Without loss of generality, we will restrict our analysis to two dimensional domains, although the results presented here can be straightforwardly extended to three dimensions. The mirror point location in two dimensions can be fully characterized by three scenarios represented in Figs. 2-4. These correspond to the case where (I) the ghost cell is not included in the interpolation stencil (Fig. 2), (II) the ghost cell itself is in the interpolation stencil (Fig. 3), and (III) the interpolation stencil includes the ghost cell itself as well as another ghost cell (Fig. 4). Since the inclusion of another ghost cell in the interpolation does not yield to ill-posedness [27], in the following we will focus on excluding the ghost cell value itself. Specifically, we will remove the point located at X_0 , i.e. potentially the ghost point, from the interpolation, irrespective the case of interest. This ensures that for case I, II or III the problem is not ill-posed: it is worth noticing that if the ghost cell is part of the stencil, then it can only be located at X_0 . Fig. 3 shows the geometry of the discretization stencil.

Algorithm 2. Find Bi-quadratic Interpolation Stencil

1: function	⊳ Find
BQSTENCIL $(F, \Gamma_{ib}, X_g, Y_g, X, Y)$	interpolation
、 <u>-</u> -	points
2: for all $\left(x_i, y_j ight) \in \left\{\left(X_g, Y_g ight) ight\}$ do	
3: $\vec{\mathbf{n}} = (n_x, n_y) \leftarrow \text{normal to the } \Gamma_{ib}$	
from (x_i, y_j)	
4: $(x_{ib}, y_{ib}) \leftarrow$ The normal line	
intersection with Γ_{ib}	
5: $(x_M, y_M) \leftarrow (2x_{ib} - x_i, 2y_{ib} - y_i)$	Mirror points
6: We find <i>I</i> and <i>J</i> such that	Mirror point
$x_I < x_M < x_{I+1}$ and $y_J < y_M < y_{J+1}$	direct neighbors
7: if $n_x > 0$ then	
8: $(x_1, x_2, x_5) \leftarrow (x_{I+1}, x_{I+2}, x_{I+1})$	

(continued on next page)



Fig. 3. The interpolation stencil for the second scenario, where the ghost point itself is inside the bilinear interpolation stencil.

```
9:
                (xf_6, xs_6) \leftarrow (x_{I+1}, x_{I+2})
10:
             else if
11:
                then I \leftarrow I + 1
12:
                (x_1, x_2, x_5) \leftarrow (x_{I-1}, x_{I-2}, x_{I-1})
13:
                (xf_6, xs_6) \leftarrow (x_{I-1}, x_{I-2})
14:
             (x_3, x_4) \leftarrow (x_I, x_I)
15:
             if n_v > 0 then
                (y_3, y_4, y_5) \leftarrow (y_{J+1}, y_{J+2}, y_{J+1})
16:
                (yf_6, ys_6) \leftarrow (y_{J+2}, y_{J+1})
17:
18:
             else if
                then J \leftarrow J + 1
19:
                (y_3, y_4, y_5) \leftarrow \left(y_{J-1}, y_{J-2}, y_{J-1}\right)
20:
                (yf_6, ys_6) \leftarrow (y_{J-2}, y_{J-1})
21:
             (y_1, y_2) \leftarrow (y_J, y_J)
22:
             d_f = (fx_6 - x_M, fy_6 - y_M)
23:
24:
             d_s = (sx_6 - x_M, sy_6 - y_M)
             (d_f, d_s) = (\|d_f\|_2, \|d_s\|_2)
25:
26:
             if d_f < d_s then
27:
                (x_6, y_6) \leftarrow (xf_6, yf_6)
28:
             else if d_f \ge d_s then
29:
                (x_6, y_6) \leftarrow (xs_6, ys_6)
30:
         X_i \leftarrow (x_1, x_2, x_3, x_4, x_5, x_6)
31:
         X_i \leftarrow (y_1, y_2, y_3, y_4, y_5, y_6)
32:
         return X_i, Y_i, X_M, Y_M, X_{ib}, Y_{ib}
```

The value $\phi_m \approx P_2(x_m, y_m) + O(\Delta x^3)$ at X_m is calculated through the bi-quadratic polynomial

$$P_2(x,y) = c_{00} + c_{10}x + c_{01}y + c_{11}xy + c_{20}x^2 + c_{02}y^2,$$
(14)

up to the third order of accuracy. The polynomial can be uniquely determined through functional evaluations at six points (in two dimensions). The six points locations (X_i, Y_i) in the biquadratic interpolation are chosen as follows and can be obtained following Algorithm 2. The point X_0 is excluded as shown in Fig. 2; X_1 and X_2 are the two closest points to X_0 in the *x*-direction, while X_3 and X_4 are the two closest points to X_0 in *y*-direction; $X_5 = (x_1, y_3)$, and X_6 needs to be chosen between the two points adjacent to X_5 , whichever is closer to the mirror point. The unknown coefficients of P_2 in Eq. (14) are determined by solving the 6×6 linear system

$$A\boldsymbol{c} = \boldsymbol{\Phi},\tag{15}$$

where

ł

$$\boldsymbol{c} = \begin{bmatrix} c_{00} & c_{10} & c_{01} & c_{11} & c_{20} & c_{02} \end{bmatrix}^{T}$$
(16)
$$\boldsymbol{\Phi} = \begin{bmatrix} c_{00} & c_{01} & c_{11} & c_{20} & c_{02} \end{bmatrix}^{T}$$
(17)

$$A = \begin{pmatrix} 1 & x_1 & y_1 & x_1y_1 & x_1^2 & y_1^2 \\ 1 & x_2 & y_2 & x_2y_2 & x_2^2 & y_2^2 \\ 1 & x_3 & y_3 & x_3y_3 & x_3^2 & y_3^2 \\ 1 & x_4 & y_4 & x_4y_4 & x_4^2 & y_4^2 \\ 1 & x_5 & y_5 & x_5y_5 & x_5^2 & y_5^2 \\ 1 & x_6 & y_6 & x_6y_6 & x_6^2 & y_6^2 \end{pmatrix}.$$
(17)



Fig. 4. The interpolation stencil for the three scenario, where two of the enclosing points of the mirror point have ghost point type.

Let $\Pi_k(\mathbb{R}^2)$ denote the space of polynomials of two variables (x, y) of total degree k,

$$P_2(x,y) = \sum_{i=0}^{k} \sum_{j=0}^{k-i} c_{ij} x^i y^j = \sum_{\substack{(0 \le i+j \le k)}} c_{ij} x^i y^j.$$
(19)

The *existence* and *uniqueness* of the polynomial $P_2(x, y)$ depends on the geometry of the points in the interpolation stencil. If there is a *unique* solution to an interpolation problem, then the problem is *poised* [39–41]. In the following, we show that there exists a unique solution to the interpolation problem following the algorithm that we have suggested.

We start with

Theorem 1. Theorem on basis functions of bivariate polynomials: A basis for $\Pi_k(\mathbb{R}^2)$ is the set of functions,

$$(x, y) \mapsto x^{i} y^{j} \qquad (0 \leq i + j \leq k),$$
[42], and
$$(20)$$

Corollary 1.1. *The dimension of* $\Pi_k(\mathbb{R}^2)$ *is*

$$\frac{1}{2}(k+1)(k+2),$$
(21)

[39], i.e. the monomials $x^i y^j$ span the polynomial space $\Pi_k(\mathbb{R}^2)$ [42], whose dimension is (k + 1)(k + 2)/2 [42,39]. Hence, for a bi-quadratic polynomial in two dimensions (k = 2), six points are needed. Full proofs of Theorem 1 and Corollary 1.1 can be found in Chapter 6 of [42]. Furthermore,

Theorem 2. Interpolation by the subspace $\Pi_k(\mathbb{R}^2)$ is possible on a set of points \mathcal{N} with cardinality $\frac{1}{2}(k+1)(k+2)$ if the points lie on lines L_0, \dots, L_k in such a way that (for each i) L_i contains exactly i+1 points,

whose proof is available in [43,42]. Theorem 2 provides a necessary and sufficient condition on the geometry of the \mathcal{N} points so to guarantee that the interpolation is possible, i.e. for any arbitrary set of points, \mathcal{N} , with cardinality 6 (i.e. k = 2), a $P_2(x, y)$ spanned by $\Pi_2(\mathbb{R}^2)$ on \mathcal{N} may not exist, unless the condition stated above is satisfied.

Importantly, the interpolation stencil proposed above satisfies the condition of Theorem 2, since three lines L_0, L_1, L_2 exist, as shown in Fig. 5, i.e. the *geometry* of the six points guarantees the existence of a polynomial $P_2(x, y)$ spanned by the subspace $\Pi_2(\mathbb{R}^2)$. The expression for $P_2(x, y)$ can now be obtained by solving the matrix (15) of size 6×6 . This system can be reduced to two systems of equations of size 3×3 each. Specifically, Eq. (14) can be rewritten as as follows

$$P_2(x,y) = c_{00} + c_{01}y + c_{02}y^2 + x(c_{10} + c_{11}y + c_{20}x)$$
(22)

if
$$L_2$$
 is vertical, or as



Fig. 5. The orientation of the bi-quadratic interpolation stencil with respect to L_0, L_1 and L_2 lines.

Table 1 Comparison between current method and other methods.

Study	Ghost cell	Mirror points	Mirror points interpolation	Pros	Cons
D. Pan (2012) [18]	1D quadratic	1	Bi-linear	(i) 4 points stencil for ghost cell	(i) 1st order of accuracy
K. Luo et al. (2016) [19]	1D quadratic	2	Bi-linear	(i) 2nd order of accuracy	(i) Non-local stencil for ghost cells,(ii) 7–8 point stencil for ghost cells
Current work	1D quadratic	1	Bi-quadratic	 (i) Local construction of ghost cells, (ii) Up to 3rd order of accuracy, (iii) 6 point stencil for ghost cells 	(i) Slightly more computationally expensive than [18]

(23)

(25)

 $P_2(x,y) = c_{00} + c_{10}x + c_{20}x^2 + y(c_{01} + c_{11}x + c_{02}y)$

if L_2 is horizontal. The origin $X_o = (0,0)$ is to be set on either on X_1 or X_3 , respectively. When L_2 is vertical and $X_1 = X_o, x$ is zero at X_1, X_5 and X_6 . Hence, Eq. (22) simplifies to

$$P_2(0,y) = c_{00} + c_{01}y + c_{02}y^2, \tag{24}$$

and the system of Eq. (15) for
$$X_1, X_5, X_6$$
 reduces to

 $B_1 \boldsymbol{c_1} = \Phi_1$

where

 $c_1 = [c_{00} \quad c_{01} \quad c_{02}]^T,$ (26)

$$\Phi_1 = [\phi_1 \ \phi_5 \ \phi_6]^T, \tag{27}$$

$$B_1 = \begin{pmatrix} 1 & y_1 & y_1^2 \\ 1 & y_5 & y_5^2 \\ 1 & y_6 & y_6^2 \end{pmatrix}.$$
 (28)

Table 2

Simulation parameters.

Re	Pr	Δt
20	0.71	4×10^{-6}

Table 3

Boundary conditions.

Boundary	u-velocity	v-velocity	Transfer
Left	$u = u_{\infty}$	u = 0	T = 1
Right	$\frac{\partial u}{\partial x} = 0$	$\frac{\partial v}{\partial x} = 0$	$\frac{\partial T}{\partial x} = 0$
Тор	$u = u_{\infty}$	$\frac{\partial v}{\partial y} = 0$	$\frac{\partial T}{\partial y} = 0$
Bottom	$u = u_{\infty}$	$\frac{\partial v}{\partial y} = 0$	$\frac{\partial T}{\partial y} = 0$



Fig. 6. L_1, L_2 and L_∞ error norm of temperature with respect to grid size for Neumann boundary condition.

Eq. (25)	defines t	the expr	essions	for c_0	$0, C_{01}$	and	<i>c</i> ₀₂ .	The	unknov	٧n
coefficie	nts, c ₁₀ , c	$_{11}$ and c_2	₂₀ , can ł	oe obt	ained	by s	olvii	ng		

$$B_2 \boldsymbol{c_2} = \Phi_2, \tag{29}$$

where

$$\mathbf{c_2} = \left[c_{10} \ c_{11} \ c_{20} \right]^l, \tag{30}$$

$$\Phi_2 = \phi_i - (c_{00} + c_{01}y_i + c_{02}y_i^2), \qquad i \in \{2, 3, 4\}, \tag{31}$$

$$B_2 = \begin{pmatrix} x_2 & x_2y_2 & x_2^2 \\ x_3 & x_3y_3 & x_3^2 \\ x_4 & x_4y_4 & x_4^2 \end{pmatrix}.$$
 (32)

A similar derivation can be followed for the case where L_2 is horizontal. In the following section we describe the numerical implementation of the above scheme.

For the sake of completeness, Table 1 provides a comparison between different ghost cell immersed boundary methods for general boundary conditions. The current method has the advantage of a local discretization and the use of the smallest possible number of stencil points, while maintaining at least second order accuracy for the discretization of all terms. A biguadratic interpolation is slightly more computationally expensive. However, the reduction of this problem to two smaller problems will resolve this issue. Moreover, since the number of ghost points is much smaller than the system size, the increased cost is overall negligible. We emphasize that the proposed method guarantees at most 6 off-diagonal elements in the row corresponding to the ghost cell in 2D, and 10 in 3D. Fewer off-diagonal elements ensure better performance of linear solvers. This represents an improvement over existing methods, e.g. [19], where the number of off-diagonal elements is 8 and 16 in 2D and 3D, respectively. Finally, we want to point out that the point identification and interpolation reconstruction are local problems which can be simply distributed to each worker cell in possible parallel implementation. This suggests that GCIBM does not add any major complexity to the standard finite volume implementation of the PDEs.

3. Numerical validation and results

3.1. Flow and heat transfer past a 2D cylinder

We consider flow and heat transfer over a static 2D circular cylinder with diameter d in a uniform flow with outer velocity



Fig. 7. L_1, L_2 and L_{∞} error norm of temperature with respect to grid size for Robin boundary condition.

 $u_{\infty} = 1$. The center of the cylinder is placed at a distance 5*d* from the inlet (left boundary), 10*d* from the outlet (right boundary) and 5*d* from the top and bottom boundaries. The left boundary is chosen to be far from the cylinder to avoid outlet boundary effects. Eqs. (1)–(3) are subject to appropriate boundary conditions on the domain external boundaries. The boundary conditions for flow and heat transfer as well as the simulation parameters are summarized in the Tables 2 and 3.

Furthermore, on the cylinder walls,

$$-n_i \alpha \frac{\partial \psi}{\partial x_i} = \beta \psi + q \tag{33}$$

with $(\alpha, \beta, q) = (-1, 0, 1)$ or $(\alpha, \beta, q) = (-1, 1, 0)$, for Neumann and Robin boundary conditions, respectively. In the following, we perform first a spatial accuracy test (Section 3.1.1) and then we validate the code against experimental data (Section 3.1.2.1).

Table 4 Flow parameters.

Minimum grid size, Δx Time step. Δt	0.0125 2 5 . 10 ⁻³
α	0
β q	0



Fig. 8. Surface pressure coefficient for Re = 20, compared with results in [44].

3.1.1. Spatial accuracy assessment

In order to ensure that the introduction of the algebraic treatment of the boundary does not alter the formal accuracy of the discretization scheme, special care is taken to retain the formal second order accuracy of the developed ghost cell immersed boundary method.

A sequence of regular Cartesian grids, with grid sizes $\frac{1}{20}$, $\frac{1}{40}$ and $\frac{1}{80}$, is used for the grid convergence test. The converged solutions on the $\frac{1}{80}$ grid is adopted as the reference. Figs. 6 and 7 show the convergence rate versus grid size in L_1 , L_2 and L_∞ norms for Neumann and Robin boundary conditions, respectively. The proposed method shows a second order accurate convergence for both boundary conditions. The line of slope 1 and 2 are also plotted, which correspond to first and second order of accuracy.

3.1.2. Validation against experimental data

3.1.2.1. Flow problem. The Reynolds number and the Prandtl number are set to (Re, Pr) = (20, 0.71). Other parameters in the simulation are summarized in Table 4. The grids used in this simulation are locally refined near the cylinder. We should note that the NS Stokes equation is solved in dimensionless form. The advection time scale is adopted to non-dimentionlize the time.

We start by validating the flow solver by determining the pressure (C_p) around the cylinder in terms of the angle. It is obtained by



Fig. 9. *u*-velocity distribution around a circular cylinder for Re = 200.



Fig. 10. *v*-velocity distribution around a circular cylinder for *Re* = 200.



Fig. 11. Pressure distribution around a circular cylinder for Re = 200.



Fig. 12. Vorticity around a circular cylinder for *Re* = 200.

calculating the pressure and viscous forces around the cylinder as follows

$$C_p = \frac{2(P - P_{\infty})}{u_{\infty}^2},\tag{34}$$

since the IB force is not explicitly calculated. Fig. 8 shows a good agreement between the calculated C_p and published values in the literature [44].

The distribution of the two components of the velocity as well as the pressure, and vorticity over a circular cylinder at Re = 200, are depicted in Figs. 9–12, respectively.

3.1.2.2. Heat transfer problem. Once the flow solution is validated, we proceed with the solution of the heat transfer equation around the cylinder for different boundary conditions. The simulation parameters are summarized in Table 5. In this context, the quantity of interest is the average Nusslet number \overline{Nu} defined as

$$\overline{Nu} = \frac{1}{2\pi r} \sum Nu\Delta S,$$
(35)

where
$$Nu = \left(-\frac{\partial T}{\partial n_{cyl}}\right) \frac{1}{T_{cyl}}$$
 (36)

r is the radius of the cylinder and ΔS is the area of the surface segments. To ensure the proposed GCIBM is able to handle different

	115.				
Case	Re	Pr	α	β	q
Dirichlet	20	0.71	0	-1	1
Neumann	20	0.71	-1	1	0
Robin	20	0.71	-1	1	0

Table 6	
---------	--

Table 5

Co	mparison	of	Nusselt	number	in	different	studies
	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	U 1	1 TUDDUIC	nunber		unicicit	Judico

Study	Re	Pr	Nu (Dirichlet)	Nu (Neumann)	Nu (Robin)
Pan (2012)	20	0.71	2.4553	2.7739	2.7202
Bharti et al. (2007)	20	0.71	2.4653	2.7788	-
Current	20	0.71	2.4614	2.7771	2.7316

boundary conditions, we compare the calculated \overline{Nu} with numerical results available in the literature [45,44,18]. In Table 6, the calculated values of the average Nusselt numbers are compared with previously reported values in the literature [45,18]. Our results are in good agreement with previous published results.

3.2. Flow and transport over an array of circular cylinders in a planar fracture

We now apply the proposed GCIBM to a more realistic scenario. Heterogeneous surface reaction in an idealized porous medium is computed as an example of multiple application of the proposed GCIBM. Simulation of flow and transport in porous media is challenging since the geometry of the porous structure is often complex and calls for a numerical scheme that is capable of effectively handle complex geometries. Furthermore, chemical (and physical) heterogeneity of the pore-structure, e.g. the presence of solid grains with different surface reactivity, often calls for the formulation of multiscale numerical algorithms which cou-



Fig. 13. Steady state solution of the flow regimes over an array of cylinders for Re = 1, v-velocity, u-velocity, pressure and streamlines (top to bottom).



Fig. 14. Concentration of the 2D solution at times t = 1, t = 3, t = 7, t = 10, and t = 13 (top to bottom).

Table 7 Flow parameters.

10
1
1
0.1
2

ple precesses at different scales [46,16,14,13]. In such cases, the primary challenge is to determine coupling conditions between subdomains. In this context, the Immersed Boundary Method is a desirable scheme, as the coupling is much more effortless in Cartesian grids.

We consider flow and solute transport in a channel occupied by an array of cylinders. The solute undergoes a linear surface reaction on the grain surface. This problem set up is a classical benchmark often used to test both the capabilities of (i) mass transport solvers and (ii) hybrid schemes [13,47]. The Reynolds number is set to 1, which results in a Stokes flow around the cylinders as shown in Fig. 13. The proposed GCIBM is very efficient in simulating Stokes flows. The calculated flow field is used to model transport of a solute subject to heterogeneous reactions on the surface of the cylinders. Two of the cylinders have faster reaction rates, characterized by Damköhler number Da_2 , compared to the remaining cylinders, whose Damköhler number is Da_1 . The concentration distribution is shown in Fig. 14 for different temporal snapshots. The simulation parameters are listed in Table 7. A comparison with the results by [13, Fig. 14] show a good agreement, and demonstrate the algorithm capability to properly handle reactive transport in complex topologies.

4. Conclusions

We introduced a high-order compact immersed boundary method for different boundary conditions, based on the ghost cell formulation. The primary advantage of the proposed method over existing ones is a more accurate interpolation, employed to approximate mirror points. Also, we theoretically show the existence and uniqueness of such an interpolation using approximation theory. The proposed method uses local grid points to reconstruct the ghost cell values, and is therefore suitable for problems involving surface phenomena. The present GCIBM shows second order accuracy for three types of boundary conditions: Dirichlet, Neuman and Robin. The method has been successfully validated for the classical heat transfer problem around a circular cylinder. Finally, we demonstrate the method can be successfully employed to model flow and reactive transport in complex porous media: specifically we investigate the classical benchmark of flow past an array of reactive cylinders in a long channel. Further improvements, such as extension to three dimensional geometries and inclusion of non-circular solid bodies, are straightforward.

Data availability

The code is available at https://github.com/mehrdadyo/IBM-Code.

Conflict of interest

The authors declared that there is no conflict of interest.

Acknowledgments

Full support by the Department of Energy under the Early Career award DE-SC0014227 'Multiscale dynamics of reactive fronts in the subsurface' is gratefully acknowledged.

References

- C.S. Peskin, Flow patterns around heart valves: a numerical method, J. Comput. Phys. 10 (2) (1972) 252–271.
- [2] C.S. Peskin, Numerical analysis of blood flow in the heart, J. Comput. Phys. 25 (3) (1977) 220-252.
- [3] C.S. Peskin, The immersed boundary method, Acta Numer. 11 (2002) 479–517.
- [4] R. Mittal, G. Iaccarino, Immersed boundary methods, Annu. Rev. Fluid Mech. 37 (2005) 239–261.
- [5] J. Kim, H. Choi, An immersed-boundary finite-volume method for simulation of heat transfer in complex geometries, KSME Int. J. 18 (6) (2004) 1026–1035.
- [6] N. Zhang, Z. Zheng, S. Eckels, Study of heat-transfer on the surface of a circular cylinder in flow using an immersed-boundary method, Int. J. Heat Fluid Flow 29 (6) (2008) 1558–1566.
- [7] A. Mark, E. Svenning, F. Edelvik, An immersed boundary method for simulation of flow with heat transfer, Int. J. Heat Mass Transfer 56 (1) (2013) 424–435.
- [8] Z. Wang, J. Fan, K. Luo, K. Cen, Immersed boundary method for the simulation of flows with heat transfer, Int. J. Heat Mass Transfer 52 (19) (2009) 4510– 4518.
- [9] W. Ren, C. Shu, W. Yang, An efficient immersed boundary method for thermal flow problems with heat flux boundary conditions, Int. J. Heat Mass Transfer 64 (2013) 694–705.
- [10] D. Pan, An immersed boundary method on unstructured Cartesian meshes for incompressible flows with heat transfer, Numer. Heat Transfer, Part B: Fundam. 49 (3) (2006) 277–297.
- [11] J. Pacheco, A. Pacheco-Vega, T. Rodić, R. Peck, Numerical simulations of heat transfer, fluid flow problems using an immersed-boundary finite-volume method on nonstaggered grids, Numer. Heat Transfer, Part B: Fundam. 48 (1) (2005) 1–24.

- [12] C.G. Bell, H. Byrne, J. Whiteley, S. Waters, Heat or mass transfer at low Péclet number for Brinkman and Darcy flow round a sphere, Int. J. Heat Mass Transfer 68 (2014) 247–258.
- [13] M. Yousefzadeh, I. Battiato, Physics-based hybrid method for multiscale transport in porous media, J. Comput. Phys. 344 (2017) 320–338.
- [14] I. Battiato, D.M. Tartakovsky, A.M. Tartakovsky, T.D. Scheibe, Hybrid models of reactive transport in porous and fractured media, Adv. Water Resour. 34 (9) (2011) 1140–1150.
- [15] E.M. Ryan, A.M. Tartakovsky, C. Amon, A novel method for modeling Neumann and robin boundary conditions in smoothed particle hydrodynamics, Comput. Phys. Commun. 181 (12) (2010) 2008–2023.
- [16] A.M. Tartakovsky, D.M. Tartakovsky, T.D. Scheibe, P. Meakin, Hybrid simulations of reaction-diffusion systems in porous media, SIAM J. Sci. Comput. 30 (6) (2008) 2799–2816.
- [17] A. Pacheco-Vega, J.R. Pacheco, T. Rodić, A general scheme for the boundary conditions in convective and diffusive heat transfer with immersed boundary methods, J. Heat Transfer 129 (11) (2007) 1506–1516.
- [18] D. Pan, A general boundary condition treatment in immersed boundary methods for incompressible Navier-Stokes equations with heat transfer, Numer. Heat Transfer, Part B: Fundam. 61 (4) (2012) 279–297.
- [19] K. Luo, Z. Zhuang, J. Fan, N.E.L. Haugen, A ghost-cell immersed boundary method for simulations of heat transfer in compressible flows under different boundary conditions, Int. J. Heat Mass Transfer 92 (2016) 708–717.
- [20] K. Luo, C. Mao, Z. Zhuang, J. Fan, N.E.L. Haugen, A ghost-cell immersed boundary method for the simulations of heat transfer in compressible flows under different boundary conditions part-ii: complex geometries, Int. J. Heat Mass Transfer 104 (2017) 98–111.
- [21] J. Mohd-Yusof, Combined immersed-boundary/B-spline methods for simulations of flow in complex geometries, Annual Research Briefs, Center for Turbulence Research, Stanford University, 1997, 317–327.
- [22] E. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof, Combined immersedboundary finite-difference methods for three-dimensional complex flow simulations, J. Comput. Phys. 161 (1) (2000) 35–60.
- [23] E. Balaras, Modeling complex boundaries using an external force field on fixed Cartesian grids in large-eddy simulations, Comput. Fluids 33 (3) (2004) 375– 404.
- [24] S. Majumdar, G. Iaccarino, P. Durbin, RANS solvers with adaptive structured boundary non-conforming grids, Annual Research Briefs, Center for Turbulence Research, Stanford University, 2001, 353–466.
- [25] Y.-H. Tseng, J.H. Ferziger, A ghost-cell immersed boundary method for flow in complex geometry, J. Comput. Phys. 192 (2) (2003) 593–623.
- [26] R. Ghias, R. Mittal, T.S. Lund, A non-body conformal grid method for simulation of compressible flows with complex immersed boundaries, AIAA paper 80 (2004) (2004) 6.
- [27] R. Ghias, R. Mittal, H. Dong, A sharp interface immersed boundary method for compressible viscous flows, J. Comput. Phys. 225 (1) (2007) 528– 553.
- [28] J. Xia, K. Luo, J. Fan, A ghost-cell based high-order immersed boundary method for inter-phase heat transfer simulation, Int. J. Heat Mass Transfer 75 (2014) 302–312.
- [29] S. Das, N.G. Deen, J. Kuipers, Immersed boundary method (ibm) based direct numerical simulation of open-cell solid foams: hydrodynamics, AIChE J. 63 (3) (2017) 1152–1173.
- [30] H. Tavassoli, E. Peters, J. Kuipers, Direct numerical simulation of nonisothermal flow through dense bidisperse random arrays of spheres, Powder Technol. 314 (2017) 291–298.
- [31] J. Derksen, G. Reynolds, A. Crampton, Z. Huang, J. Booth, Simulations of dissolution of spherical particles in laminar shear flow, Chem. Eng. Res. Des. 93 (2015) 66–78.
- [32] J.H. Ferziger, M. Peric, Computational Methods for Fluid Dynamics, Springer Science & Business Media, 2012.
- [33] H.K. Versteeg, W. Malalasekera, An Introduction to Computational Fluid Dynamics: The Finite Volume Method, Pearson Education, 2007.
- [34] F.H. Harlow, J.E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, Phys. Fluids 8 (12) (1965) 2182– 2189.
- [35] J. Van Doormaal, G. Raithby, Enhancements of the simple method for predicting incompressible fluid flows, Numer. Heat Transfer 7 (2) (1984) 147–163.
- [36] B.P. Leonard, A stable and accurate convective modelling procedure based on quadratic upstream interpolation, Comput. Methods Appl. Mech. Eng. 19 (1) (1979) 59–98.
- [37] T. Hayase, J. Humphrey, R. Greif, A consistently formulated quick scheme for fast and stable convergence using finite-volume iterative calculation procedures, J. Comput. Phys. 98 (1) (1992) 108–118.
- [38] R. Mittal, H. Dong, M. Bozkurttas, F. Najjar, A. Vargas, A. von Loebbecke, A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries, J. Comput. Phys. 227 (10) (2008) 4825–4852.
- [39] B. Bojanov, Y. Xu, On polynomial interpolation of two variables, J. Approx. Theory 120 (2) (2003) 267–282.
- [40] M. Gasca, T. Sauer, Polynomial interpolation in several variables, Adv. Comput. Math. 12 (4) (2000) 377–410.

- [41] T. Sauer, Y. Xu, On multivariate lagrange interpolation, Math. Comput. 64 (211)
- (1995) 1147–1170.
 [42] D.R. Kincaid, E.W. Cheney, Numerical Analysis: Mathematics of Scientific Computing, vol. 2, American Mathematical Soc., 2002.
- [43] M. Gasca, J.I. Maeztu, On lagrange and hermite interpolation in r k, Numer. Math. 39 (1) (1982) 1-14.
- [44] B. Fornberg, A numerical study of steady viscous flow past a circular cylinder, J. Fluid Mech. 98 (4) (1980) 819–855.
- [45] R.P. Bharti, R. Chhabra, V. Eswaran, A numerical study of the steady forced convection heat transfer from an unconfined circular cylinder, Heat Mass Transfer 43 (7) (2007) 639-648.
- [46] F.J. Alexander, A.L. Garcia, D.M. Tartakovsky, Noise in algorithm refinement
- [47] C. Soulaine, S. Roman, A. Kovscek, H.A. Tchelepi, Mineral dissolution and wormholing from a pore-scale perspective, J. Fluid Mech. 827 (2017) 457–483.