



Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp


Modified immersed boundary method for flows over randomly rough surfaces

 Chunsong Kwon ^b, Daniel M. Tartakovsky ^{a,*}
^a Department of Energy Resources Engineering, Stanford University, 367 Panama Street, Stanford, CA 94305, USA

^b Mechanical and Aerospace Engineering Department, University of California, 9500 Gilman Drive, San Diego, CA 92093, USA


ARTICLE INFO

Article history:

Received 7 May 2019

Received in revised form 3 November 2019

Accepted 15 December 2019

Available online 20 December 2019

Keywords:

Immersed boundary method

Direct interaction force

Euler-Lagrange coupling

Finite-difference method

Navier-Stokes equations

ABSTRACT

Many phenomena, ranging from biology to electronics, involve flow over rough or irregular surfaces. We treat such surfaces as random fields and use an immersed boundary method (IBM) with discrete (random) forcing to solve resulting stochastic flow problems. Our approach relies on the Uhlmann formulation of the fluid-solid interaction force; computational savings stem from the modification of the time advancement scheme that obviates the need to solve the Poisson equation for pressure at each sub-step. We start by testing the proposed algorithm on two classical benchmark problems. The first deals with the Wannier problem of Stokesian flow around a cylinder in the vicinity of a moving plate. The second problem considers steady-state and transient flows over a stationary circular cylinder. Our simulation results show that our algorithm achieves second-order temporal accuracy. It is faster than the original IBM, while yielding consistent estimates of such quantities of interest as the drag and lift coefficients, the length of a recirculation zone in a cylinder's wake, and the Strouhal number. Then we use the proposed IBM algorithm to model flow over cylinders whose surface is either (deterministically) corrugated or (randomly) rough.

© 2019 Elsevier Inc. All rights reserved.

1. Introduction

A plethora of natural and artificial phenomena, ranging from biology [1] and hydrogeology [2] to electronics [3], involve flow over rough or irregular surfaces. Modeling this process is complicated by both the lack of sufficient characterization of surface variability and the discrepancy of scales between this variability and bulk flow. The former challenge can be addressed by treating rough surfaces as random fields [4–6], whose ensemble statistics are inferred from spatially distributed observations by invoking ergodicity [7,8]. Numerical methods for solving the resulting problems on random simulation domains (e.g., random domain mappings [4,5] or smoothed profile methods [6]) rely on finite-term approximations (e.g., via the Karhunen-Loève expansion) of random surfaces. This limits the applicability of such solutions to well-correlated surfaces, which might or might not correspond to reality.

Immersed boundary methods (IBM) [9,10] discretize a (rough) surface into a set of points that do not necessarily coincide with the Cartesian grid, and represent the effects of this surface on the fluid with an appropriate spatially distributed force. Treating such surfaces as random fields renders this force stochastic. Since its inception, various modifications and flavors of the IBM have been proposed [11–13]. These can be subdivided into two broad categories depending on how the force

* Corresponding author.

E-mail address: tartakovsky@stanford.edu (D.M. Tartakovsky).

is implemented in governing (usually, Navier-Stokes) equations. The first category is referred to as a “continuous forcing method” because it incorporates a forcing term into continuous governing equations, before their discretization. It is well suited for flows in domains with immersed elastic boundaries [9,12,14,15] and, consequently, is often deployed in biological applications [16–18]. At the same time, such an approach has well-known limitations in handling rigid boundaries [11,14].

In the second class of IBM implementations, which is called a “discrete forcing method”, a discrete forcing is added, either explicitly or implicitly, to the discretized governing equations [19–24]. Since the choice of a forcing scheme is highly dependent on the spatial discretization, its implementation is not straightforward compared to the continuous forcing method. However, a key advantage of this approach is that the discrete forcing allows for sharp representation of an immersed boundary.

We adopt the discrete forcing strategy and, more specifically, its implementation based on a fractional method with a three-step Runge-Kutta scheme for time advancement [23]. The most time-consuming part of this approach is to solve, for every sub-step, a Poisson equation for fluid pressure. Our first goal is to reduce the computational cost of this method by designing a new time-advancement strategy that ameliorates the need to solve this equation. Then, we use this accelerated IBM to solve flow problems on random domains representing, e.g., rough solid surfaces.

In Section 2, we present a standard discrete-forcing IBM based on a fractional method with a three-step Runge-Kutta scheme for time advancement [23]. Our approach to accelerate this method is introduced in Section 3. Section 4 illustrates its performance on a series of numerical experiments dealing with Wannier flow, flows around a stationary circular cylinder and cylinders with sinusoidal and randomly rough surfaces. Major conclusions drawn from our study are summarized in Section 6.

2. Standard IBM implementation

We consider two-dimensional flow of an incompressible fluid, of density ρ and kinematic viscosity ν , past a solid body with surface Γ , in the flow domain $\Omega \in \mathbb{R}^2$. At every space-time point (\mathbf{x}, t) , velocity $\mathbf{u}(\mathbf{x}, t)$ and (normalized with ρ) fluid pressure $p(\mathbf{x}, t)$ are described by a combination of the Navier-Stokes equations

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla) \mathbf{u} + \nu \nabla^2 \mathbf{u} - \nabla p + \mathbf{f} \quad (1)$$

and a continuity equation

$$\nabla \cdot \mathbf{u} = 0, \quad (2)$$

where \mathbf{f} is a volumetric force. These equations are discretized using a staggered marker-and-cell (MAC) grid [25], so that fluid variables $\mathbf{u}(\mathbf{x}, t)$ and $p(\mathbf{x}, t)$ are defined at different locations. The nonlinear convective term, $N(\mathbf{u}) \equiv (\mathbf{u} \cdot \nabla) \mathbf{u}$, is approximated with a three-step Runge-Kutta scheme. The viscous term, $L(\mathbf{u}) \equiv \nabla^2 \mathbf{u}$, is handled with an implicit Crank-Nicholson scheme, which eliminates the viscous stability constraint. A conventional fractional-step method is employed for time integration, resulting in the overall second-order temporal accuracy. All the spatial derivatives are approximated by a second-order central finite-difference method. The time stepping from current time step \mathbf{u}^n to next time step \mathbf{u}^{n+1} , carried out in three sub-steps of size Δt each, replaces (1) with

$$\frac{\tilde{\mathbf{u}}^{n,k} - \mathbf{u}^{n,k-1}}{\Delta t} = \nu(\alpha_k + \beta_k)L(\mathbf{u}^{n,k-1}) - \gamma_k N(\mathbf{u}^{n,k-1}) - \zeta_k N(\mathbf{u}^{n,k-2}) \quad (3a)$$

$$\frac{\hat{\mathbf{u}}^{n,k} - \tilde{\mathbf{u}}^{n,k}}{\Delta t} = \nu\beta_k L(\hat{\mathbf{u}}^{n,k} - \mathbf{u}^{n,k-1}) + \mathbf{f}^{n,k} \quad (3b)$$

$$\frac{\mathbf{u}^{n,k} - \hat{\mathbf{u}}^{n,k}}{\Delta t} = -\nabla \phi^{n,k}, \quad (3c)$$

where $k = 1, 2, 3$ denotes the sub-step number, such that $\mathbf{u}^{n,0} = \mathbf{u}^n$, $\mathbf{u}^{n,3} = \mathbf{u}^{n+1}$ and $k - 2$ is ignored for $k = 1$; $\alpha_k, \beta_k, \gamma_k, \zeta_k$ ($k = 1, 2, 3$) are the Runge-Kutta coefficients [26],

$$\begin{aligned} \alpha_1 = \beta_1 = \frac{4}{15}, \quad \alpha_2 = \beta_2 = \frac{1}{15}, \quad \alpha_3 = \beta_3 = \frac{1}{6}, \quad \gamma_1 = \frac{8}{15}, \quad \gamma_2 = \frac{5}{12}, \quad \gamma_3 = \frac{3}{42} \\ \zeta_1 = 0, \quad \zeta_2 = -\frac{17}{60}, \quad \zeta_3 = -\frac{5}{120}; \end{aligned} \quad (4)$$

$\tilde{\mathbf{u}}^{n,k}$ and $\hat{\mathbf{u}}^{n,k}$ are the intermediate velocities, between $(k - 1)$ th and k th sub-steps, computed with (3a) and (3b), respectively. This implementation of the RK scheme is explicit, i.e., the advection term is treated explicitly, and the diffusion term implicitly. Accounting for (2) at k th step, $\nabla \cdot \mathbf{u}^{n,k} = 0$ and (3c) yields an equation for ϕ^k ,

$$\nabla^2 \phi^{n,k} = \frac{1}{\Delta t} \nabla \cdot \hat{\mathbf{u}}^{n,k}. \quad (5)$$

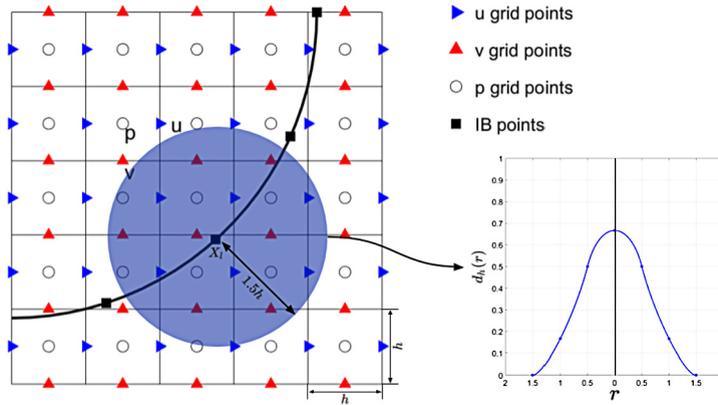


Fig. 1. Schematic representation of the immersed boundary method on a two-dimensional staggered Cartesian grid.

Once it is solved, flow velocity $\mathbf{u}^{n,k}$ is obtained from (3c) and pressure $p^{n,k}$ is calculated with

$$\nabla \phi^{n,k} = (\alpha_k + \beta_k) \nabla p^{n,k} - \nu \beta_k L(\mathbf{u}^{n,k} - \hat{\mathbf{u}}^{n,k}). \tag{6}$$

Equations (3)–(6) are solved on a rectangular Cartesian grid consisting of N_x and N_y nodes in the x and y directions, respectively.

The IBM replaces the surface of an immersed solid body, Γ , with appropriate momentum forces at Cartesian grid points adjacent to Γ . We use lower case letters to denote the variables on a Cartesian grid and corresponding upper case letters to denote their immersed boundary counterparts. Thus, let $\mathbf{X} = \{\mathbf{X}_1, \dots, \mathbf{X}_M\}$ denote a set of M discretization points used to represent Γ . Since the immersed boundary does not necessarily coincide with the Cartesian grid points (Fig. 1), an interpolation scheme is required to exchange information between the underlying Cartesian grid and the immersed boundary points \mathbf{X}_l . Specifically, velocity at the latter points is approximated with

$$\tilde{\mathbf{U}}^{n,k}(\mathbf{X}_l) = \int_{\Omega} \tilde{\mathbf{u}}^{n,k}(\mathbf{x}) \delta(\mathbf{x}_{i,j} - \mathbf{X}_l) d\mathbf{x} \approx h^2 \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \tilde{\mathbf{u}}^{n,k}(\mathbf{x}_{i,j}) \delta_h(\mathbf{x}_{i,j} - \mathbf{X}_l), \quad l = 1, \dots, M, \tag{7}$$

where $\delta_h(\cdot)$ is a discrete approximation (with support related to the mesh size h) of the Dirac delta function $\delta(\cdot)$. Among various discrete approximations [27], we chose [28]

$$\delta_h(\mathbf{x} - \mathbf{X}) = d_h(x - X) d_h(y - Y), \tag{8}$$

since it strikes a good balance between numerical efficiency and smoothing properties [29]. Here d_h is defined by

$$d_h(r) = \frac{1}{6h} \begin{cases} 5 - 3|r| - \sqrt{1 - 3(1 - |r|)^2}, & \text{for } 0.5 \leq |r| \leq 1.5 \\ 2 + 2\sqrt{1 - 3|r|^2}, & \text{for } |r| \leq 0.5 \\ 0, & \text{otherwise,} \end{cases} \tag{9}$$

with $r = (x - X)/h = (y - Y)/h$. While (8) and (9) are written in two spatial dimensions for a square grid, they can be readily generalized for three dimensions and rectangular grids.

Forces $\mathbf{F}_l = \mathbf{F}(\mathbf{X}_l)$ are constructed to satisfy the no-flow ($\mathbf{u} \cdot \mathbf{n} = 0$, with \mathbf{n} denoting the unit normal vector to Γ) and no-slip ($\mathbf{u} \cdot \mathbf{s} = 0$, with \mathbf{s} denoting the unit tangential vector to Γ) boundary conditions on Γ or, more specifically, at points \mathbf{X}_l ($l = 1, \dots, M$). Suppose that an immersed body with surface Γ is moving through the fluid with velocity $\mathbf{U}_{\text{body}}^{n,k-1}$, which is computed with an equation of motion at the previous Runge-Kutta step; for a stationary body, $\mathbf{U}_{\text{body}} \equiv \mathbf{0}$. The Mohd-Yusof direct forcing method [19] estimates forces $\mathbf{F}(\mathbf{X}_l)$ at the k th time step as

$$\mathbf{F}^{n,k}(\mathbf{X}_l) = \frac{\mathbf{U}_{\text{body}}^{n,k-1}(\mathbf{X}_l) - \tilde{\mathbf{U}}^{n,k}(\mathbf{X}_l)}{\Delta t}, \quad l = 1, \dots, M. \tag{10}$$

These forces are then extrapolated to the neighboring Cartesian grid points, $\mathbf{x}_{i,j}$, and incorporated into the volumetric force \mathbf{f} ,

$$\mathbf{f}^{n,k}(\mathbf{x}_{i,j}) = \int_{\Omega} \mathbf{F}^{n,k}(\mathbf{X}_l) \delta_h(\mathbf{x}_{i,j} - \mathbf{X}_l) d\mathbf{X}_l \approx \sum_{l=1}^M \Delta V_l \mathbf{F}^{n,k}(\mathbf{X}_l) \delta_h(\mathbf{x}_{i,j} - \mathbf{X}_l), \tag{11}$$

where ΔV_l is the volume associated with the l th IB point.

This numerical scheme allows relatively large time steps ($\text{CFL}=\sqrt{3}$) and requires low memory storage [30]. However, computation of (3) must be repeated at each sub-step, which significantly increases the computational cost associated with solving the Poisson equation (5). Our strategy is to modify (3a) in order to eliminate the need for computing the Poisson equation at the sub-steps.

3. Accelerated IBM

Our time advancement follows the procedure proposed in [30]. The first sub-step of time stepping from \mathbf{u}^n to \mathbf{u}^{n+1} by (3a)–(3c) reads as

$$\frac{\tilde{\mathbf{u}}^{n,1} - \mathbf{u}^n}{\Delta t} = \nu(\alpha_1 + \beta_1)L(\mathbf{u}^n) - \gamma_1 N(\mathbf{u}^n) \quad (12)$$

$$\frac{\hat{\mathbf{u}}^{n,1} - \tilde{\mathbf{u}}^{n,1}}{\Delta t} = \nu\beta_1 L(\hat{\mathbf{u}}^{n,1} - \mathbf{u}^n) + \mathbf{f}^{n,1} \quad (13)$$

$$\frac{\mathbf{u}^{n,1} - \hat{\mathbf{u}}^{n,1}}{\Delta t} = -\nabla\phi^{n,1}, \quad (14)$$

where the boundary force $\mathbf{f}^{n,1}$ is calculated using (10)–(11). The second sub-step without time splitting has the form

$$\frac{\mathbf{u}^{n,2} - \mathbf{u}^{n,1}}{\Delta t} = \nu\alpha_2 L(\mathbf{u}^{n,1}) + \nu\beta_2 L(\mathbf{u}^{n,2}) - \gamma_2 N(\mathbf{u}^{n,1}) - \zeta_2 N(\mathbf{u}^n) - (\alpha_2 + \beta_2)\nabla p^{n,2}. \quad (15)$$

Substituting $\mathbf{u}^{n,1} = \hat{\mathbf{u}}^{n,1} - \Delta t\nabla\phi^{n,1}$ from (14) into (15), we obtain the second sub-step with time-splitting,

$$\frac{\tilde{\mathbf{u}}^{n,2} - \hat{\mathbf{u}}^{n,1}}{\Delta t} = \nu(\alpha_2 + \beta_2)L(\hat{\mathbf{u}}^{n,1}) - \nu\alpha_2\nabla(\nabla \cdot \hat{\mathbf{u}}^{n,1}) - \gamma_2 N(\hat{\mathbf{u}}^{n,1} - \Delta t\nabla\phi^{n,1}) - \zeta_2 N(\mathbf{u}^n) \quad (16)$$

$$\frac{\hat{\mathbf{u}}^{n,2} - \tilde{\mathbf{u}}^{n,2}}{\Delta t} = \beta_2 L(\hat{\mathbf{u}}^{n,2} - \hat{\mathbf{u}}^{n,1}) + \mathbf{f}^{n,2} \quad (17)$$

$$\frac{\mathbf{u}^{n,2} - \hat{\mathbf{u}}^{n,2}}{\Delta t} = -\nabla\phi^{n,1} - (\alpha_2 + \beta_2)\nabla p^{n,2} + \nu\beta_2 L(\mathbf{u}^{n,2} - \hat{\mathbf{u}}^{n,2}) = -\nabla\phi^{n,2}. \quad (18)$$

Instead of solving the Poisson equation (5) at the first sub-step, we approximate $\nabla\phi^{n,1}$ with its counterpart at the end of the previous time step, $\nabla\phi^n$. This approximation has second-order accuracy. Indeed, it follows from (6) that

$$\begin{aligned} \Delta t\nabla\phi^{n,1} &= (\alpha_1 + \beta_1)\Delta t\nabla p^{n,1} - \nu\beta_1\Delta tL(\mathbf{u}^{n,1} - \hat{\mathbf{u}}^{n,1}) \\ &= (\alpha_1 + \beta_1)\Delta t\nabla p^{n,1} + \mathcal{O}(\Delta t^2). \end{aligned}$$

From the second-order Taylor expansion $\nabla p^{n,1} = \nabla p^n + \Delta t\frac{\partial}{\partial t}(\nabla p^n) + \mathcal{O}(\Delta t^2)$ and the relation $p = \phi + \mathcal{O}(\Delta t)$ [31],

$$\begin{aligned} \Delta t\nabla\phi^{n,1} &= (\alpha_1 + \beta_1)\Delta t\nabla p^n + \mathcal{O}(\Delta t^2) \\ &= (\alpha_1 + \beta_1)\Delta t\nabla\phi^n + \mathcal{O}(\Delta t^2). \end{aligned}$$

Thus, the advancement from the first sub-step to the second is carried out without solving (14), while maintaining the second-order accuracy in time. The same procedure is used in the last sub-step with $\mathbf{u}^{n,2} = \hat{\mathbf{u}}^{n,2} - \Delta t\nabla\phi^{n,2} = \hat{\mathbf{u}}^{n,2} - \Delta t\sum_{l=1}^2(\alpha_l + \beta_l)\nabla\phi^n$.

The modified method calculates $\hat{\mathbf{u}}$ at the first and second sub-steps without projection onto the divergence-free velocity field by (14) and (18). Only at the last sub-step, the projection into the divergence-free field is required. Fig. 2 exhibits the divergence (L^2) norm, $\|\nabla \cdot \mathbf{u}^{n,k}\|_{L^2(\Omega)}$, at each sub-step. As expected, the velocity field $\hat{\mathbf{u}}$ at the first and second sub-step does not satisfy the divergence-free constraint because the Poisson equation for pressure is not being solved. Only at the last (third) sub-step, the divergence-free constraint is enforced. This results in a significant reduction in computational cost [30].

Equation (16) contains an additional term, $\nabla(\nabla \cdot \hat{\mathbf{u}}^{n,k-1})$, which is absent in the original IBM formulation (3a) and has to be calculated at every sub-step. However, this term does not incur an additional cost because its computation is required anyway in (5). The accuracy in the time advancement remains second order.

The flexibility and robustness of IBM make it an ideal tool for simulation of fluid flow in domains with complex geometries. We use our accelerated IBM to model flow around inclusions with deterministic and random surfaces.

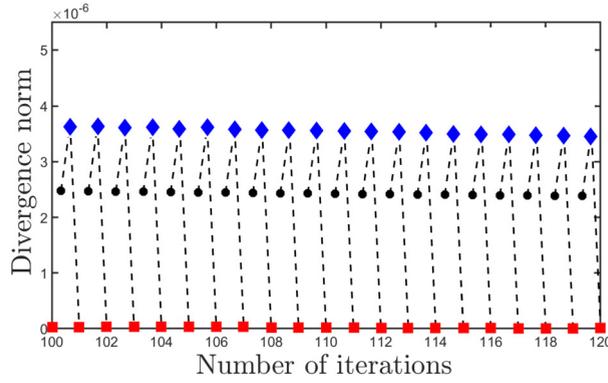


Fig. 2. L^2 norm of the velocity divergence, $\|\nabla \cdot \mathbf{u}^{n,k}\|_{L^2(\Omega)}$, at every sub-step. The velocity field $\hat{\mathbf{u}}$ at the first (black circles) and second (blue diamonds) sub-steps does not satisfy the divergence-free constraint because the Poisson equation for pressure is not being solved. The latter is done only at the final, third sub-step (red squares). (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

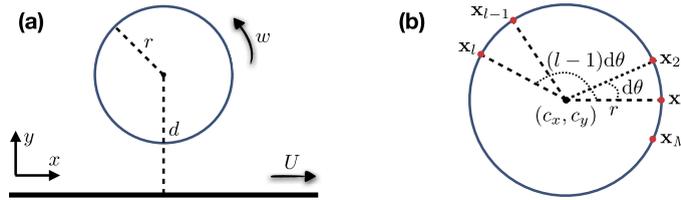


Fig. 3. (a) Wannier flow refers to creeping flow past a rotating circular cylinder near a moving wall. (b) Generation of the IB points.

4. Flow around inclusions with deterministic surfaces

We start by demonstrating the overall accuracy and efficiency of our modified IBM on two classical two-dimensional problems. The first deals with the Wannier problem of Stokesian flow around a cylinder in the vicinity of a moving plate. It serves to validate the accuracy of our method because it admits an analytical solution. The second problem considers steady-state and transient flows over a stationary circular cylinder. It has been widely investigated, both experimentally and numerically, and often serves as a benchmark. Then, we investigate flow around cylinders with sinusoidal surfaces to demonstrate the flexibility of our method.

4.1. Wannier flow

Wannier [32] considered creeping (low Reynolds number) flow past a rotating rigid circular cylinder next to a moving wall (Fig. 3a). Steady flow of an incompressible Newtonian fluid flow with dynamic viscosity ν is described by the Stokes and continuity equations,

$$\nu \nabla^2 \mathbf{u} = \nabla p, \quad \nabla \cdot \mathbf{u} = 0. \tag{19}$$

Suppose that the cylinder radius is r , its rate of rotation is w , the distance from the center of the cylinder to the moving wall is d , and the velocity of the wall is U . Then the components of the flow velocity vector $\mathbf{u} = (u, v)^\top$ are given by [32]

$$u = U - 2(A + F\tilde{y}) \left[\frac{s + \tilde{y}}{\alpha} + \frac{s - \tilde{y}}{\beta} \right] - F \ln \frac{\alpha}{\beta} - B \left[\frac{s + 2\tilde{y}}{\alpha} - \frac{2\tilde{y}(s + \tilde{y})^2}{\alpha^2} \right] - C \left[\frac{s - 2\tilde{y}}{\beta} + \frac{2\tilde{y}(s - \tilde{y})^2}{\beta^2} \right] \tag{20a}$$

$$v = 2x \left[(A + F\tilde{y}) \left(\frac{1}{\alpha} - \frac{1}{\beta} \right) - B \frac{\tilde{y}(s + \tilde{y})}{\alpha^2} - C \frac{\tilde{y}(s - \tilde{y})}{\beta^2} \right] \tag{20b}$$

where $\tilde{y} = y + d$, $\alpha = x^2 + (s + \tilde{y})^2$, $\beta = x^2 + (s - \tilde{y})^2$, $s = \sqrt{d^2 - r^2}$, $\gamma = (d + s)/(d - s)$, $\tau = \omega r^2/2s$, $A = -d(F + \tau)$, $B = 2(d + s)(F + \tau)$, $C = 2(d - s)(F + \tau)$, $F = U/\ln \gamma$.

In the simulations reported below we set $r = 0.5$ m, $d = 1.5$ m, $U = 1$ m/s, and choose the simulation domain to be a square of length 4 m centered around the cylinder. The velocity $\mathbf{u}(\mathbf{x})$ from the analytical solution (20) is enforced on the square’s boundaries. The circular cylinder is represented by a set of M IB points whose coordinates $\mathbf{x}_l = (X_l, Y_l)$, with $l = 1, \dots, M$, are

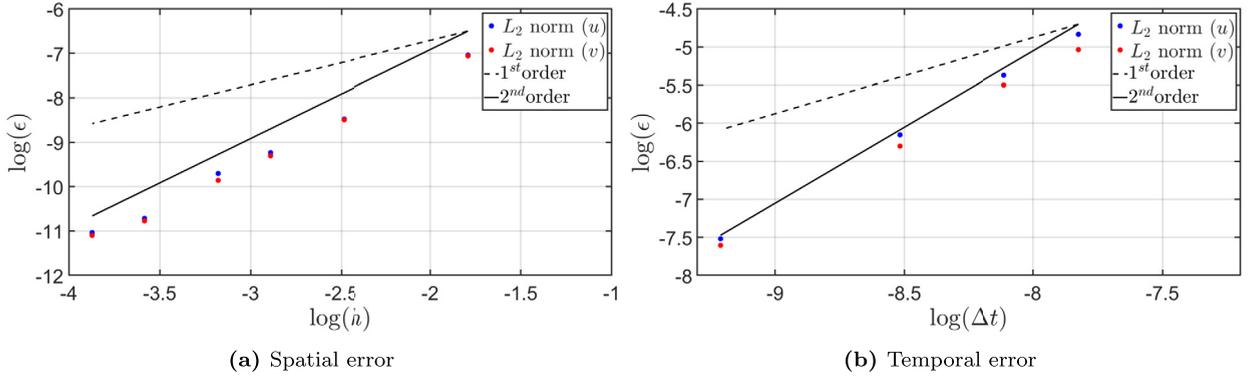


Fig. 4. The L_2 norms of the logarithm of the error ϵ in computing components u and v of the velocity field \mathbf{u} for Wannier flow. The errors are plotted as functions of the logarithm of (a) grid size h and (b) time step Δt .

Table 1

The CPU time required to complete one time step.

Nodes	CPU Time		Difference	Improvement (%)
	Original method	Modified method		
24×24	0.0053	0.0044	0.0009	16.98
48×48	0.0093	0.0080	0.0012	14.13
96×96	0.0310	0.0251	0.0059	19.03
192×192	0.1193	0.0968	0.0225	18.86

$$X_l = c_x + r \cos[(l-1)d\theta], \quad Y_l = c_y + r \sin[(l-1)d\theta]. \quad (21)$$

Here r is the radius of the circular cylinder, c_x and c_y are the coordinates of its center on the Cartesian grid, and $d\theta$ is the angular difference between two adjacent IB points. We set the first IB point at the far right side on the horizontal center line and add points through the direction of counter clock wise. Fig. 3b describes the generation of IB points of the circular cylinder. Three cases are solved on four different uniforms of grids consisting of 24×24 , 48×48 , 96×96 , and 192×192 nodes. To ascertain the accuracy of our accelerated IBM (12)–(18), we compare its prediction of the steady-state velocity $\mathbf{u}_i \equiv \mathbf{u}(\mathbf{x}_i)$ with its analytical counterpart in (20) evaluated at the corresponding grid points \mathbf{x}_i . The numerical solution $\mathbf{u}(\mathbf{x})$ is obtained by time-stepping (12)–(18) until the steady state is reached; the latter is defined as the time at which the velocity relative residual between a current and previous time steps falls below 10^{-5} .

Fig. 4 exhibits the L_2 error norms, for both of u and v , as function of the logarithm of grid size h and time step Δt . The L_2 error norms of the difference between the numerical and analytical solutions are defined as

$$L_2 = \left\{ \frac{1}{N_x N_y} \sum_{n=1}^{N_x N_y} (f_n^{\text{ana}} - f_n^{\text{num}})^2 \right\}^{1/2}, \quad (22)$$

where f_n^{ana} and f_n^{num} are the analytical and numerical solutions at the n th point on the numerical grid, and N_x and N_y are the number of points in the x and y direction respectively. The dashed and solid lines represent the -1 and -2 slopes, respectively. The slope of -2 in the dependence of the error norms on h indicates the second-order spatial accuracy of our algorithm. The slope of the dependence of the error norms on Δt is slightly less than -2 , indicating approximately (but not quite) second-order accuracy in time.

Table 1 contains the CPU times required to complete one time step in the previous and current methods on different grids. The reported CPU times are averaged over 1000 time steps. Our approach reduces the computational time on average by 17%, when the quad-cores (Intel Xeon E3-1220 V2) CPU with 20Gb RAM workstation is used.

4.2. Flow over a stationary circular cylinder

Flow over a stationary circular cylinder provides another test problem, which has been investigated both experimentally [33] and numerically [34,35]. The flow remains steady and symmetric up to the Reynolds number of about $\text{Re} = 47$. As the Reynolds number increases, the flow becomes unstable and exhibits a periodic shedding of vortices.

For a cylinder with diameter D , we chose the computational domain to be a rectangle of length $30D$ and width of $15D$, and place the cylinder's center at distance $14D$ from the inlet and $7.5D$ from the bottom, in order to minimize the effects of the (artificial) rectangle's boundaries on the flow patters. A uniform flow velocity $u = U_\infty$ is applied at the inlet ($x = 0$) and homogeneous Neumann boundary conditions are enforced on the lateral boundaries ($y = 0$ and $y = 30D$). These and

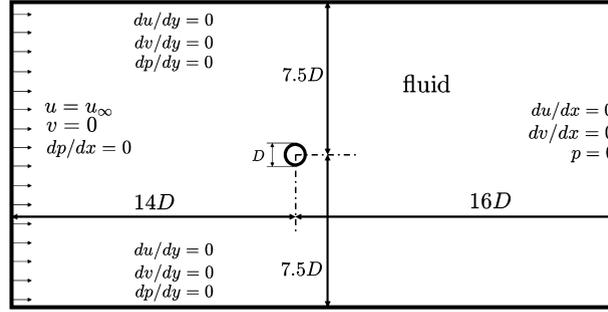


Fig. 5. Simulation domain and boundary conditions for flow over a stationary circular cylinder in a channel.

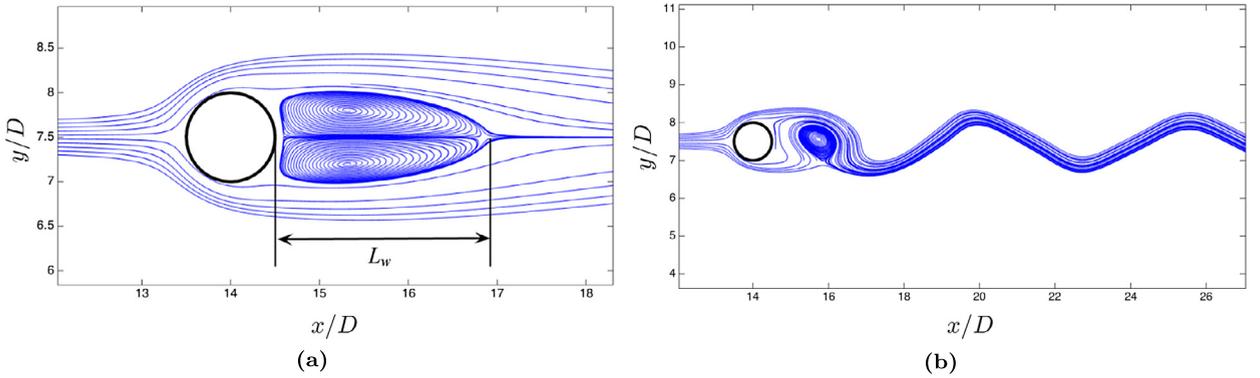


Fig. 6. Streamlines and recirculation for (a) $Re = 40$ and (b) $Re = 100$.

remaining boundary conditions are shown in Fig. 5, together with the simulation domain. The latter is discretized with a uniform square mesh of size $\Delta x = \Delta y$, and the immersed boundary is discretized with $\Delta s \approx \Delta x = \Delta y$.

The performance of the modified IBM is analyzed, for the Reynolds numbers of $Re \equiv u_\infty D / \nu = 40$ and 100, in terms of three quantities of interest: drag (C_D) and lift (C_L) coefficients and the Strouhal number St , which are often used to compare numerical results. The former two are defined as

$$C_D \equiv \frac{F_D}{U_\infty^2 D / 2} \quad \text{and} \quad C_L \equiv \frac{F_L}{U_\infty^2 D / 2}, \quad (23)$$

where the drag (F_D) and lift (F_L) forces are computed by integrating the forces acting on the fluid $\mathbf{f}(\mathbf{x}) = (f_x, f_y)^\top$ over a square domain Ω enclosing the cylinder [11],

$$F_D = - \int_{\Omega} f_x(\mathbf{x}) d\mathbf{x} \approx -\Delta x \Delta y \sum_{\mathbf{x} \in \Omega} f_x(\mathbf{x}), \quad (24)$$

$$F_L = - \int_{\Omega} f_y(\mathbf{x}) d\mathbf{x} \approx -\Delta x \Delta y \sum_{\mathbf{x} \in \Omega} f_y(\mathbf{x}). \quad (25)$$

The Strouhal number $St \equiv f_s D / U_\infty$ is used as a measure of the frequency of vortex shedding. When the Reynolds number Re increases beyond its critical value, which for a cylinder in the infinite stream is $Re \approx 47$, the flow becomes unsteady and vortices are shedding from the cylinder with frequency f_s .

The steady-state flow regime with $Re = 40$ gives rise to a stationary symmetric wake behind the cylinder (Fig. 6a), which is consistent with the well established results [33–35]. Our predictions of both the length of the recirculation zone (L_w), defined as the distance between the two downstream stagnation points, and the drag coefficient C_D are compared with their counterparts obtained in the previous numerical results in Table 2. The close agreement between these predictions provides an additional verification of our numerical algorithm.

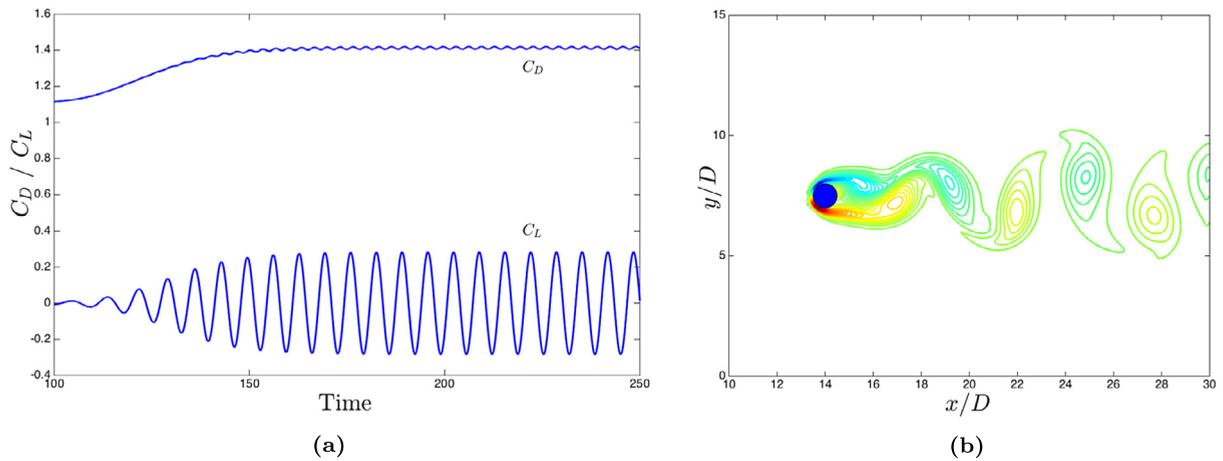
The transient flow regime with $Re = 100$ breaks the wake's symmetry (Fig. 6b) and leads to a periodic vortex shedding. Table 2 summarizes our predictions of the drag (C_D) and lift (C_L) coefficients and the Strouhal number St . These predictions are in agreement with those reported in the literature, which serves to further verify our numerical code.

The vorticity contours in Fig. 7 demonstrate the periodic shedding of vortices. The lift coefficient C_L oscillates with the shedding frequency f_s , while the drag coefficient C_D fluctuates with frequency $2f_s$. That is because the drag force F_D has

Table 2

Estimates of the drag (C_D) and lift (C_L) coefficients, the Strouhal number (St), and the length of the recirculation zone (L_w) computed with several numerical algorithms for $Re = 40$ and $Re = 100$. In theory, $C_L = 0$ at $Re = 40$. The slight deviation from $C_L = 0$ provides a measure of our method's accuracy; the other studies do not report their values of C_L . The Strouhal number St and the wake length L_w are not defined for $Re = 40$ and $Re = 100$, respectively.

	Source	C_D	C_L	L_w/D	St
Re = 40	[36]	1.51	-	2.35	-
	[37]	1.52	-	2.27	-
	[38]	1.54	-	2.54	-
	[39]	1.54	-	2.28	-
	[40]	1.52	-	2.35	-
	Present	1.57	3.01×10^{-6}	2.41	-
Re = 100	[23]	1.453 ± 0.011	± 0.339	-	0.169
	[11]	1.447 (mean)	± 0.330	-	0.165
	[41]	1.37 ± 0.009	± 0.323	-	0.160
	[36]	1.33 (mean)	± 0.332	-	0.165
	Present	1.413 ± 0.007	± 0.283	-	0.160

**Fig. 7.** (a) Variation of the drag and lift coefficients on a cylinder and (b) vorticity contour plot at $Re=100$.**Table 3**

Required average CPU time for running one time step for flow over a stationary circular cylinder.

Nodes	CPU Time		Difference	Improvement (%)
	Original method	Modified method		
150×75	0.0401	0.0331	0.0070	17.46
300×150	0.1572	0.1276	0.0296	18.83
450×225	0.3793	0.3025	0.0768	20.25
600×300	0.9171	0.7501	0.1670	18.21

one maximum and one minimum during the growth and shedding of each vortex, while the sign of the lift force F_L depends on the location of the vortex.

To demonstrate the computational savings of our method, we performed a set of numerical experiments on four different uniform grids: 150×75 , 300×150 , 450×225 , and 600×300 . Calculation of the average CPU time follows the same procedure as that used in the previous section. Table 3 provides a comparison of the CPU time required by our method to complete one time step with the corresponding CPU times of the original IBM. On average, our approach improves the computational efficiency by 18%.

4.3. Flow over a stationary circular cylinder with sinusoidal corrugation

To test the method's flexibility and accuracy, we simulate flow over a corrugated cylinder with a sinusoidal surface. The base smooth-cylinder representation of IB points is identical to (21). The constant radius r is replaced by an angle-dependent sinusoidal radius, r_s ,

$$r_s(l) = r + \epsilon r \cos[n(l - 1)d\theta] \quad (26)$$

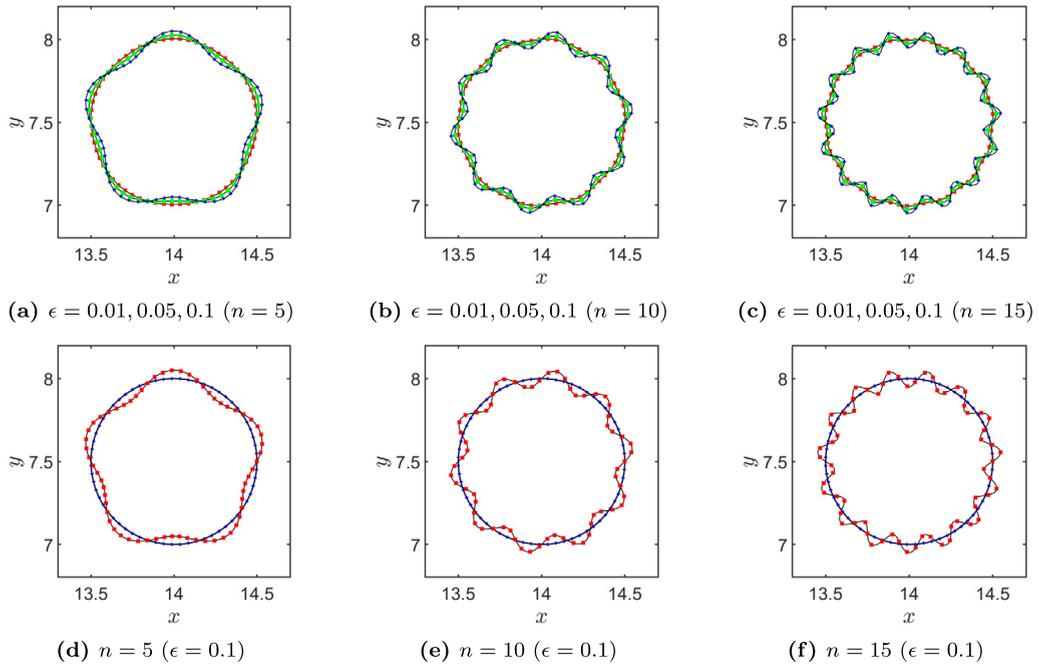


Fig. 8. Corrugated cylinders and their representations with IB points for sinusoidal surfaces with different values of the geometric parameters ϵ and n . In (a)–(c), the red dotted lines with squares correspond to $\epsilon = 0.01$, the green dotted line to $\epsilon = 0.05$, and the blue dotted line with diamonds to $\epsilon = 0.1$.

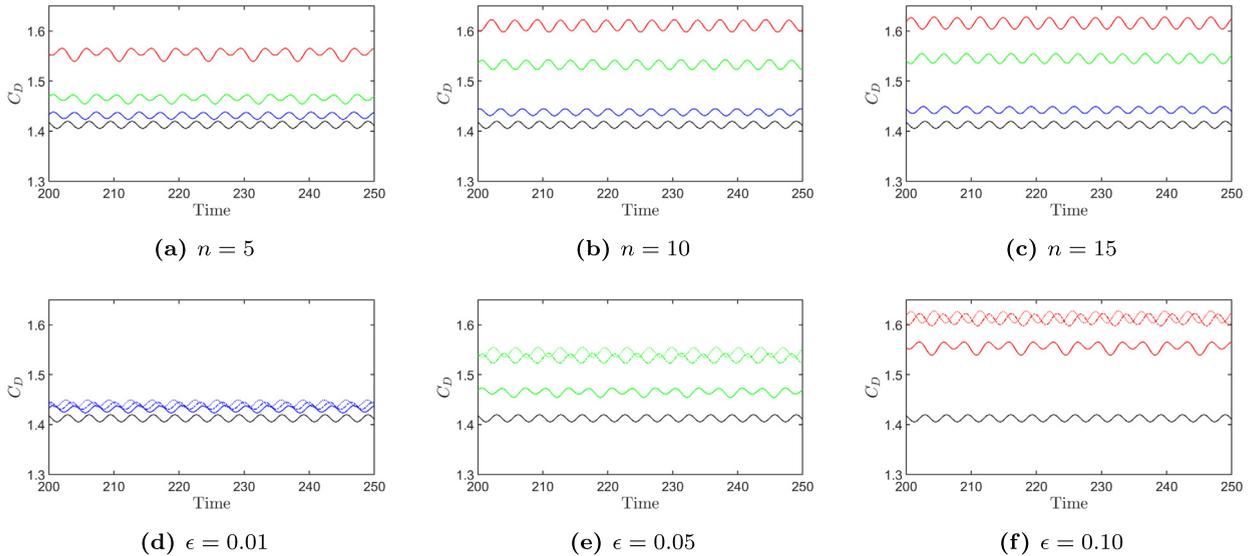


Fig. 9. Drag coefficient C_D of cylinders whose corrugated surface is characterized by amplitude ϵ and frequency n .

where ϵ indicates the relative roughness at the peak, and n represents the frequency. We consider three degrees of irregularity ($\epsilon = 0.01, 0.05$ and 0.1) with seven frequencies ($n = 1, 3, 5, 7, 10, 15$ and 20), resulting in the surfaces shown in Fig. 8.

Simulation results are exhibited in Fig. 9, with values of the computed drag coefficient for different surfaces summarized in Table 4. The black line in Fig. 9 represents the circular cylinder, while the blue, green and red lines correspond to the corrugated cylinders with $\epsilon = 0.01, 0.05$ and 0.1 , respectively. The drag coefficient C_D increases with both amplitude (ϵ) and frequency (n) of corrugation. The magnitude of C_D oscillations increases slightly with ϵ and is largely insensitive to values of n . For $\epsilon = 0.01$, the drag coefficient is sensitive to the corrugation frequency in the interval $5 \leq n \leq 15$ and insensitive otherwise. The trend is similar for larger values of ϵ , but the range of sensitive frequency increases.

Table 4
 Estimated drag coefficient C_D of a circular cylinder and corrugated cylinders characterized by amplitude ϵ and frequency n . The drag coefficient increases with the degree of roughness, as quantified by ϵ and n .

	Cylinder	ϵ	n						
			1	3	5	7	10	15	20
C_D	1.413	0.01	1.430	1.430	1.431	1.433	1.438	1.442	1.443
		0.05	1.425	1.434	1.464	1.494	1.533	1.545	1.546
		0.10	1.491	1.451	1.552	1.592	1.610	1.616	1.618

5. Flow around inclusions with randomly rough surfaces

Realistic surface roughness defies simple parameterizations that are used in its deterministic representations, such as that used above. Instead, rough surfaces are represented as random fields whose statistics are inferred from measurements [e.g., 1, 4–6,8, and the references therein]. In section 5.1, we describe three alternative strategies for generating such surfaces. Once generated, these surfaces are replaced by IB points and the boundary conditions are applied to calculate the local body force. Sections 5.2 and 5.3 provide two computational examples.

5.1. Probabilistic representations of surface roughness

We use three alternative strategies for generation of closed Lipschitz continuous random surfaces with given statistics: the moving average method (MAM) [7,8] and two implementations of the Karhunen-Loève (KL) expansion.

5.1.1. Moving average method

Consider a closed randomly rough circular surface with mean radius r . The rough surface, $R = R(s)$, where $s = r\theta$ with $\theta \in [0, 2\pi]$, is represented by a set of M random variables (IB points) $\{R_i = R(s_i)\}_{i=1}^M$ with $s_i = 2\pi r(i - 1)/M$. We use a Reynolds decomposition $R_i = r + R'_i$ to represent these random variables as the sums of the deterministic radius, r , of the corresponding smooth cylinder and the zero-mean random variables R'_i representing the local surface deviations from the smooth surface. To be specific, we take $\{R'_i\}_{i=1}^M$ to be correlated Gaussian variables with zero mean, standard deviation σ , and correlation length ℓ . MAM generates such correlated random variables in two steps. First, a set of N numbers $\{v_k\}_{k=1}^N$ is drawn from the normal distribution $\mathcal{N}(0, \sigma_v)$, where N ($N \gg M$) is selected to be large enough and the standard deviation σ_v is defined below. Second, a set of realizations, $\{r_i\}_{i=1}^M$, of the correlated random variables $\{R'_i\}_{i=1}^M$ is generated by a moving average procedure

$$r_i = \sum_{j=-L}^L w_j v_{j+i+L}, \quad i = 1, \dots, M, \quad L = \frac{N - M}{2}. \tag{27}$$

The $2L + 1$ weights w_j satisfy the normalization condition $\sum_{j=-L}^L w_j = 1$ and are computed from the correlation function of the random surface, $\rho(s; \ell)$. For example, if $\rho(s; \ell)$ is Gaussian, $\rho = \exp(-s^2/\ell^2)$, then [7]

$$w_j = \exp[-2(j\Delta s)^2/\ell^2], \quad j = -L, \dots, L, \tag{28}$$

where Δs is the length of IB segments, which we set to $\Delta s \approx \Delta x = \Delta y$. The standard deviation of the $\mathcal{N}(0, \sigma_v)$ is related to the standard deviation of the rough surface σ by (27),

$$\sigma^2 = \langle r_i^2 \rangle = \sigma_v^2 \sum_{j=-L}^L w_j^2. \tag{29}$$

5.1.2. Karhunen-Loève expansion

The KL expansion provides an alternative means to represent the random fluctuations $R'(s)$ with ensemble mean $\mathbb{E}[R'(s)] = 0$, finite variance $\mathbb{E}[R'(s)^2] = \sigma^2$, the covariance function $C(|s_1 - s_2|) = \mathbb{E}[R'(s_1)R'(s_2)] = \sigma^2 \rho(|s_1 - s_2|)$, and correlation function ρ . Specifically, it represents $R'(s)$ as a series

$$R'(s) = \sigma \sum_{k=1}^{\infty} \sqrt{\lambda_k} \psi_k(s) \xi_k, \tag{30}$$

where ξ_k are mutually uncorrelated random variables with zero mean and unit variance; and λ_k and $\psi_k(s)$ are the eigenvalues and eigenfunctions of the correlation function ρ . For an exponential correlation function, $\rho(|s_1 - s_2|) = \exp(-|s_1 - s_2|/\ell)$, the latter are given by [1],

$$\lambda_k = \frac{2\ell}{\ell^2\gamma_k^2 + 1} \quad \text{and} \quad \psi_k(s) = \frac{\ell\gamma_k \cos(\gamma_k s) + \sin(\gamma_k s)}{\sqrt{(\ell^2\gamma_k^2 + 1)T/2 + \ell}}, \quad k \geq 1. \tag{31}$$

Here T is the process length along s , and the constants γ_k ($k \geq 1$) are solutions of the transcendental equation $(\ell^2\gamma_k^2 - 1)\sin(\gamma_k T) = 2\ell\gamma_k \cos(\gamma_k T)$. We take random variables ξ_k to be uniformly distributed on the interval $[-\sqrt{3}, \sqrt{3}]$, which ensures that they have zero mean and unit variance.

Numerical implementation of KL expansions such as (30) requires one to truncate the infinite summation. The resulting truncation error depends on the correlation length ℓ . The larger the correlation length, the fewer terms in (30) are necessary to represent the random surface $R'(s)$ with a prescribed degree of accuracy. We use the number of terms, N_{KL} , that retains 95% of the full eigenspectrum,

$$\sum_{k=1}^{N_{\text{KL}}} \lambda_k \geq 0.95 \sum_{k=1}^{\infty} \lambda_k. \tag{32}$$

5.1.3. Nonstationary Karhunen-Loève expansion

Computation of the eigenvalues and eigenfunctions of the KL expansion can be simplified by modifying the statistical properties of the zero-mean random fluctuations $R'(s)$. Rather than prescribing the covariance function C of $R'(s)$, it is computed as solution of a stochastic Helmholtz equation [42]. If one imposes a condition $R'(0) = R'(\pi r) = 0$ on the $2\pi r$ -periodic random process $R'(s)$, then the KL expansion (30) takes the form of a Fourier series [6,42]

$$R'(s) = \sqrt{\frac{2}{\pi r}} \sum_{k=1}^{\infty} \frac{\ell^2}{1 + (k\ell/r)^2} \sin\left(\frac{ks}{r}\right) \xi_k. \tag{33}$$

Thus constructed random field $R'(s)$ is not stationary, e.g., its variance is not constant and differs from σ^2 . The randomly rough surface at the i th IBM point is now represented by

$$R_i = r + \epsilon R'(s_i), \tag{34}$$

where the parameter $\epsilon = \sigma / \max[\sigma_{R'_i}]$ controls the maximum deviation of the roughness height from the smooth surface of radius r .

5.2. Flow over a stationary circular cylinder with randomly rough surface

In the simulations reported here, we set the mean radius of the cylinder to $r = 0.5$ and the corresponding Reynolds number to $\text{Re} = 100$. We use moving average method (27) and the KL expansion (30) to generate random roughness. Within the statistical framework adopted here, the circular cylinder roughness is characterized by two parameters: the standard deviation σ and the correlation length ℓ . Fig. 10 exhibits typical realizations of the randomly rough surface of the cylinder for several values of the correlation length ℓ , normalized with the length of the process $T = 2\pi r$. The two representations of random surfaces generate visually similar representations of roughness.

Our focus is on the effects of random roughness on the expected value of the drag coefficient C_D . We consider the range of standard deviations σ between $0.001r$ and $0.05r$, and the normalized correlation length ℓ/T varying between 0 (white noise) and 0.2. Table 5 and Fig. 11 collate predicted values of C_D . As expected, the drag coefficient increases with the standard deviation σ . For a given σ , uncorrelated or near-zero correlation lengths induce large variations of the drag coefficient, while the sensitivity to larger correlation lengths is smaller. The results are qualitatively similar to those reported in Section 4.3 if one swaps the roughness amplitude ϵ for σ and the corrugation frequency n for the normalized correlation length ℓ/T . The drag coefficient C_D increases with both σ and ℓ/T . The magnitude of C_D oscillations is affected by σ , while its dependence on ℓ/T is confined to small ranges of the latter.

Table 5 reveals that the predicted values of C_D are virtually insensitive to the method used to generate random roughness. The agreement between the two representations improves with ℓ/T . Finite-term truncation of KL expansions requires the correlation length ℓ to be non-zero and the spatial dimension of $R(s)$ increases as ℓ decreases. Yet, even for relatively small ℓ , the IBM remains computationally feasible.

Next we explore the computational efficiency of our method for randomly rough surfaces. The comparison metric and hardware are the same as described in Section 4.1; the uniform mesh size is now related to the normalized correlation length $\ell/T = 0.05, 0.02$ and 0.01 : $200 \times 100, 500 \times 250$ and 1000×500 , respectively. The standard deviation is set to $\sigma = 0.01$. Table 6 provides a comparison of the CPU time required by our method to complete one time step with the corresponding CPU times of the original IBM. Our approach improves the computational efficiency by about 20% even for the finest grid resolution.

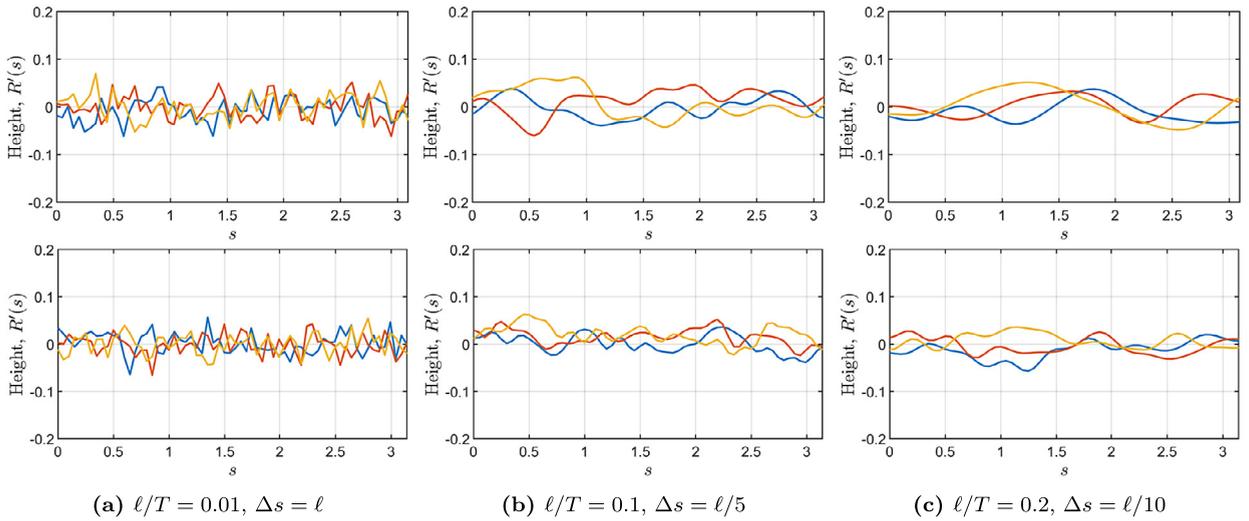


Fig. 10. Realizations of randomly rough surfaces with standard deviation $\sigma = 0.05r$ and three normalized correlation lengths ℓ/T (and spacing Δs between IBM points) generated with the moving average method (27) (top row) and the KL expansion (30) (bottom row).

Table 5

Estimated drag coefficients C_D for a circular cylinder and cylinders with random surface roughness characterized by standard deviation σ and normalized correlation length ℓ/T . The distance between IBM points is $\Delta s = \ell, 2\ell/3, \ell/3, \ell/6, \ell/10$ for the respective columns with non-zero ℓ/T . Random surface roughness is alternatively generated with the moving average method (MA) and the KL expansion (KL).

	Smooth cylinder	Roughness generation	σ	ℓ/T					
				0	0.01	0.02	0.05	0.1	0.2
C_D	1.413	MA	0.0005	1.413	1.413	1.413	1.413	1.413	1.413
			0.0025	1.416	1.413	1.416	1.412	1.413	1.412
			0.005	1.422	1.435	1.430	1.427	1.425	1.425
			0.01	1.470	1.425	1.453	1.427	1.426	1.427
			0.025	1.546	1.528	1.529	1.494	1.521	1.509
		KL	-	1.413	1.413	1.413	1.413	1.413	1.413
			-	1.415	1.413	1.414	1.415	1.416	1.415
			-	1.420	1.431	1.430	1.428	1.424	1.425
			-	1.466	1.461	1.450	1.428	1.429	1.428
			-	1.540	1.564	1.525	1.512	1.511	1.509

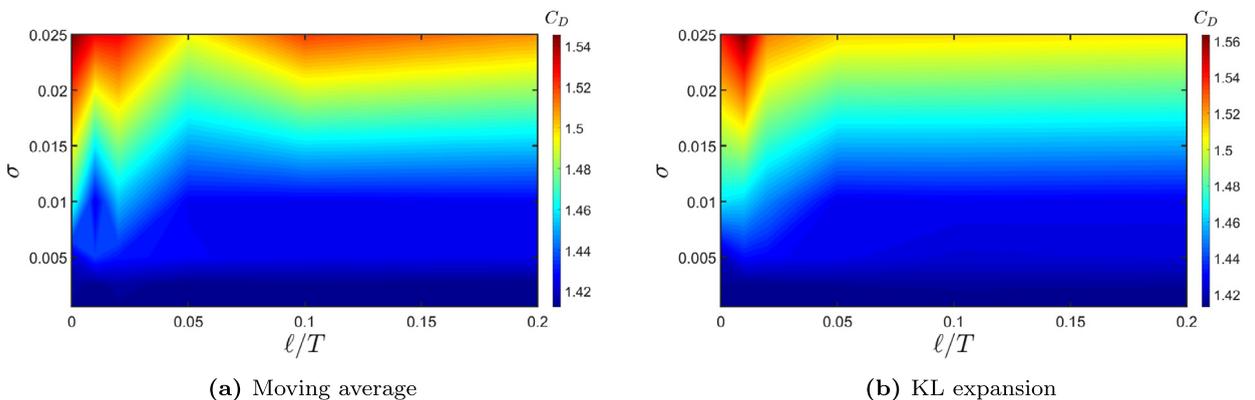


Fig. 11. Estimated drag coefficient C_D for cylinders with random surface roughness characterized by standard deviation σ and normalized correlation length ℓ/T .

Table 6
Average CPU time required for one time step to simulate flow over a stationary circular cylinder with randomly rough surface.

Nodes	CPU Time		Difference	Improvement (%)
	Original method	Modified method		
200 × 100	0.0614	0.0494	0.0120	19.54
500 × 250	0.4522	0.3621	0.0901	19.92
1000 × 500	2.2351	1.7874	0.4477	20.03

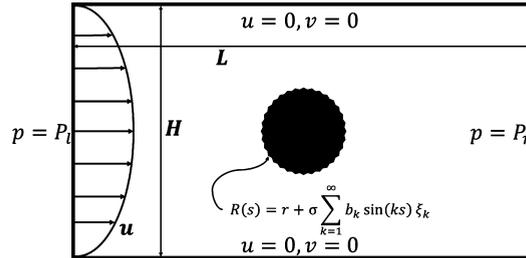


Fig. 12. Pressure driven creeping flow past a circular cylinder with random roughness represented by a KL expansion.

Table 7
The minimum number of terms, N , in the Fourier series (33) required to satisfy the criterion $\sum_{k=1}^N b_k^2 \geq \alpha \sum_{k=1}^{\infty} b_k^2$. The distance between IBM points is $\Delta s = \ell, 2\ell/3, \ell/3, \ell/6, \ell/10, \ell/20$ for the respective ℓ/T columns.

α (%)	N					
	$\ell/T = 0.005$	0.01	0.02	0.05	0.1	0.2
90	44	22	11	5	3	2
95	59	30	15	6	4	2

5.3. Creeping flow over a stationary circular cylinder with randomly rough surface

Consider incompressible viscous flow in a channel of width H and length L , which contains a cylinder of mean radius r (Fig. 12). Random roughness of the cylinder’s surface $R'(s)$ is now represented by the Fourier series (33). The creeping flow ($Re \ll 1$) is described by the Stokes equations (19) and is driven by an externally imposed pressure gradient, such that

$$p(0, y) = P_l, \quad p(L, y) = P_r, \quad 0 \leq y \leq H. \tag{35}$$

The normal components of the pressure gradient along the channel walls are

$$\frac{\partial p}{\partial y}(x, 0) = \frac{\partial p}{\partial y}(x, H) = 0, \tag{36}$$

and no-slip/flow conditions are imposed along both the channel walls and the cylinder surface,

$$u(x, 0) = u(x, H) = 0, \quad v(x, 0) = v(x, H) = 0, \quad \mathbf{u}(\mathbf{X}) \cdot \mathbf{n} = 0, \quad \mathbf{u}(\mathbf{X}) \cdot \mathbf{s} = 0. \tag{37}$$

Here \mathbf{X} , \mathbf{n} and \mathbf{s} denote the IBM points of the random surface, and unit normal and tangential vectors to the random surface, respectively. The channel width and length are set to $H = 6$ and $L = 10$, mean radius of the cylinder to $r = 1$, and the pressure drop to $\Delta P = -10$ ($P_l = 10, P_r = 0$).

The radius of the rough cylinder, $R = r + R'$, varies randomly with the angle $\theta \in [0, 2\pi]$. Its statistics are $\mathbb{E}[R] = r$ and $\ell/T = 0.01, 0.02, 0.05, 0.1$ and 0.2 . The parameter ϵ , which controls the maximum amplitude of surface roughness, is set to $0.01, 0.05$ and 0.1 . A value of ℓ determines the number of terms in the truncated Fourier series (33) that is required to capture a given portion of the entire spectrum. Table 7 shows the minimum number of terms that are required to cover 90% and 95% of the entire energy (in the L^2 -norm sense) for different correlation lengths. Fig. 13 shows realizations of the random surface roughness constructed with the truncated Fourier series (33) containing 95% of the spectrum.

Faxén [43] estimated the drag coefficient for Poiseuille flow of Newtonian fluids past a smooth circular cylinder placed symmetrically between two fixed parallel walls to be

$$C_D \approx \frac{6\pi}{f(\lambda) + g(\lambda)}, \tag{38}$$

where λ is the blockage factor defined as diameter of the cylinder to the width of channel ratio,

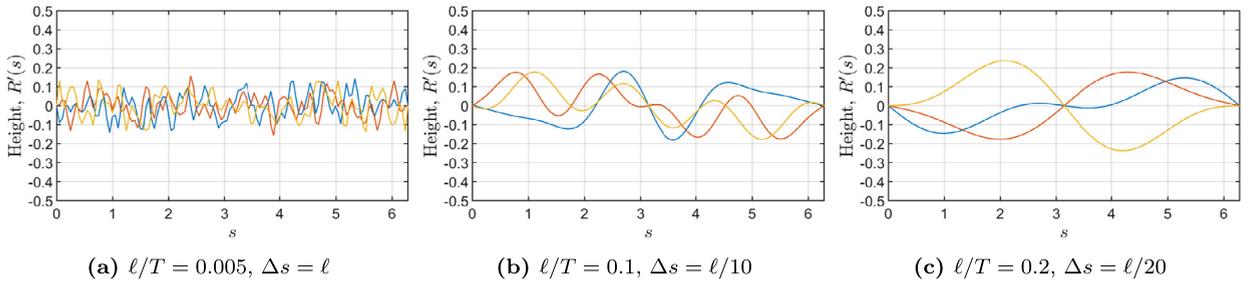


Fig. 13. Realizations of randomly rough surfaces with maximum amplitude roughness $\epsilon = 0.1r$ and three normalized correlation lengths ℓ/T (and spacing Δs between IBM points) generated with the truncated Fourier series (33).

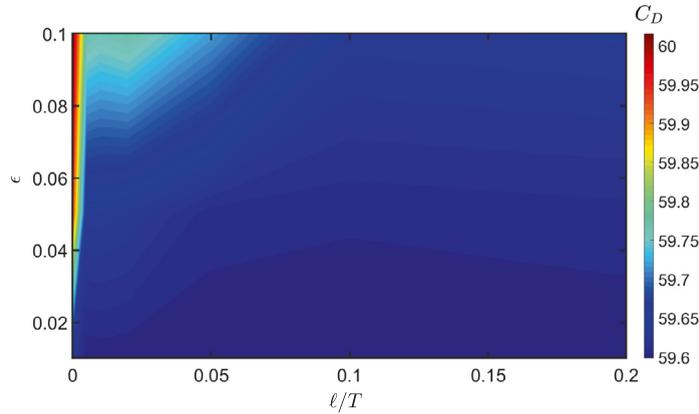


Fig. 14. Estimated drag coefficient C_D for cylinders with random surface roughness characterized by the maximum roughness amplitude ϵ and the correlation length ℓ .

$$f(\lambda) = A_0 - (1 + 0.5\lambda^2 + A_4\lambda^4 + A_6\lambda^6 + A_8\lambda^8) \ln(\lambda),$$

$$g(\lambda) = B_2\lambda^2 + B_4\lambda^4 + B_6\lambda^6 + B_8\lambda^8$$

with $A_0 = -0.9156892732$, $A_4 = 0.05464866$, $A_6 = -0.2646267$, $A_8 = 0.792986$, $B_2 = 1.26653975$, $B_4 = -0.91804$, $B_6 = 1.87710$, and $B_8 = -4.66549$. In our example, the blockage factor is $\lambda = 0.333$, which yields an estimate $C_D = 50.17$. An alternative estimate, provided by [44], is $C_D = 68.6$. Our simulation for the smooth cylinder yields $C_D = 59.5$.

Fig. 14 shows predicted values of the drag coefficient C_D for cylinders whose random roughness is characterized by the truncated Fourier series (33) for ϵ varying from 0.01 to 0.1 and ℓ/T varying from 0 to 0.2. As expected, C_D increases with the roughness amplitude σ and has large variations near zero correlation length, while larger correlation length results small variance.

Fig. 15 exhibits the corresponding velocity profiles. These are measured at the distance r ($x = 0.6L$) and $3r$ ($x = 0.8L$) behind the cylinder surface. Black solid lines indicate the profiles for the smooth cylinder. Larger velocity variations are observed near the cylinder surface with higher roughness amplitude, while the profile remains almost unchanged further away from the cylinder and near the wall due to the viscous effects.

6. Summary

We proposed an accelerated immersed boundary method (IBM) to simulate incompressible flows interacting with solid bodies. Our approach relies on the Uhlmann formulation of the fluid-solid interaction force [23], with the savings in computational cost stemming from the modification of the time advancement scheme which obviates the need to solve the Poisson equation for pressure at each sub-step. The velocity field is advanced without a divergence-free constraint in the first two sub-steps. The Poisson equation is solved only at the last sub-step. The proposed algorithm was tested on two classical benchmark problems: the Wannier problem of Stokesian flow around a cylinder in the vicinity of a moving plate, and free flow around a stationary circular cylinder.

A key innovation of our study is implementation of the IBM for flows over rough surfaces. Roughness is modeled by representing a close surface as a random field, which was generated with three alternative strategies: the moving average method (MAM) and two implementations of the Karhunen-Loève (KL) expansion. Regardless of the surface representation, the resulting randomness enters the IBM as a random force. We used this approach to model flows around a cylinder with rough surface; both free flow and flow in a channel were considered.

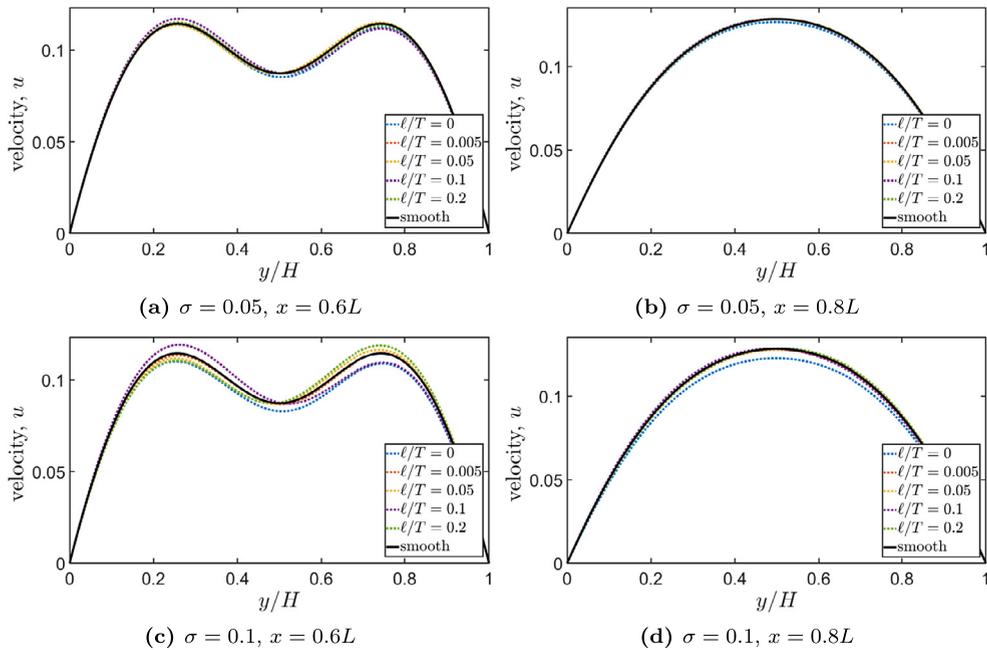


Fig. 15. Horizontal velocity profiles across the channel width at two spatial locations, $x = 0.6L$ and $x = 0.8L$, for different values of the standard deviation σ and correlation length ℓ of the randomly rough cylinder.

Our analysis leads to the following major conclusions.

- Similar to the original IBM implementation, our algorithm achieves second-order accuracy in space and time.
- Our method is about 20% faster than the original IBM, while yielding consistent estimates of such quantities of interest as the drag and lift coefficients, the length of a recirculation zone in a cylinder's wake, and the Strouhal number.
- This computational speed up is observed for problems involving both smooth and randomly rough surfaces. Both versions of the IBM can handle rough surfaces with small correlation lengths.
- The drag coefficient of a cylinder with randomly rough surface increases with the roughness amplitude (standard deviation) and decreases with the roughness regularity (correlation length).
- Predicted values of the drag coefficient are virtually insensitive to the method used to generate random roughness. The agreement between the two representations, MAM and KL expansions, improves as the correlation length increases.

We considered problems with relatively simple geometries, but our computational framework can handle more complex or moving boundaries. Furthermore, it can be adapted to predict flow and transport of a conservative scalar in a vessel whose rough surface is under-specified by data and modeled as a random field [1]. It would do so without resorting to the stochastic mapping approach [5,4] that transform an original deterministic problem on a random domain into a stochastic problem on a deterministic domain.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors thank Gustaaf Jacobs for suggesting to us computational examples. This work was supported in part by the National Science Foundation under award number 1563614 and by Air Force Office of Scientific Research under award number FA9550-17-1-0417.

References

- [1] S.-W. Park, M. Intaglietta, D.M. Tartakovsky, Impact of stochastic fluctuations in the cell free layer on nitric oxide bioavailability, *Front. Comput. Neurosci.* 9 (2015) 131.
- [2] Y. Méheust, J. Schmittbuhl, Flow enhancement of a rough fracture, *Geophys. Res. Lett.* 27 (18) (2000) 2989–2992.

- [3] M. Duwensee, F.E. Talke, S. Suzuki, J. Lin, D. Wachenschwanz, Direct simulation Monte Carlo method for the simulation of rarefied gas flow in discrete track recording head/disk interfaces, *J. Tribol.* 131 (1) (2009) 35–60.
- [4] D. Xiu, D.M. Tartakovsky, Numerical methods for differential equations in random domains, *SIAM J. Sci. Comput.* 28 (3) (2006) 1167–1185.
- [5] D.M. Tartakovsky, D. Xiu, Stochastic analysis of transport in tubes with rough walls, *J. Comput. Phys.* 217 (1) (2006) 248–259.
- [6] M. Zayernouri, S.-W. Park, D.M. Tartakovsky, G.E. Karniadakis, Stochastic smoothed profile method for modeling random roughness in flow problems, *Comput. Methods Appl. Mech. Eng.* 263 (2013) 99–112.
- [7] J.A. Ogilvy, Computer simulation of acoustic wave scattering from rough surfaces, *J. Phys. D, Appl. Phys.* 21 (2) (1988) 260–277.
- [8] J.A. Ogilvy, J.R. Foster, Rough surfaces: Gaussian or exponential statistics? *J. Phys. D, Appl. Phys.* 22 (9) (1989) 1243–1251.
- [9] C.S. Peskin, Flow patterns around heart valves: a numerical method, *J. Comput. Phys.* 10 (2) (1972) 252–271.
- [10] C.S. Peskin, Numerical analysis of blood flow in the heart, *J. Comput. Phys.* 25 (3) (1977) 220–252.
- [11] M.-C. Lai, C.S. Peskin, An immersed boundary method with formal second-order accuracy and reduced numerical viscosity, *J. Comput. Phys.* 160 (2) (2000) 705–719.
- [12] E.M. Saiki, S. Biringen, Numerical simulation of a cylinder in uniform flow: application of a virtual boundary method, *J. Comput. Phys.* 123 (1996) 450–465.
- [13] C. Lee, Stability characteristics of the virtual boundary method in three-dimensional applications, *J. Comput. Phys.* 184 (2) (2003) 559–591.
- [14] R. Mittal, G. Iaccarino, Immersed boundary methods, *Annu. Rev. Fluid Mech.* 37 (1) (2005) 239–261.
- [15] D. Goldstein, R. Handler, L. Sirovich, Modeling a no-slip flow boundary with an external force field, *J. Comput. Phys.* 105 (2) (1993) 354–366.
- [16] C.S. Peskin, The fluid dynamics of computational methods, *Annu. Rev. Fluid Mech.* 14 (1982) 235–259.
- [17] R.P. Beyer, R.J. Leveque, Analysis of a one-dimensional model for the immersed boundary method, *SIAM J. Numer. Anal.* 29 (2) (1992) 332–364.
- [18] L.J. Fauci, A. McDonald, Sperm motility in the presence of boundaries, *Bull. Math. Biol.* 57 (5) (1995) 679–699.
- [19] J. Mohd-Yusof, Combined immersed boundary/B-spline methods for simulation of flow in complex geometries, Tech. rep., Center for Turbulence Research Annual Research Briefs, Stanford University, 1997.
- [20] E.A. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof, Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, *J. Comput. Phys.* 161 (1) (2000) 35–60.
- [21] J. Kim, D. Kim, H. Choi, An immersed-boundary finite-volume method for simulations of flow in complex geometries, *J. Comput. Phys.* 171 (1) (2001) 132–150.
- [22] 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.
- [23] M. Uhlmann, An immersed boundary method with direct forcing for the simulation of particulate flows, *J. Comput. Phys.* 209 (2) (2005) 448–476.
- [24] R. Ghias, R. Mittal, H. Dong, A sharp interface immersed boundary method for compressible viscous flows, *J. Comput. Phys.* 225 (1) (2007) 528–553.
- [25] F.H. Harlow, J.E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, *Phys. Fluids* 8 (1965) 2182–2189.
- [26] P.R. Spalart, R.D. Moser, M.M. Rogers, Spectral methods for the Navier-Stokes equations with one infinite and two periodic directions, *J. Comput. Phys.* 96 (2) (1991) 297–324.
- [27] C.S. Peskin, The immersed boundary method, *Acta Numer.* 11 (2002) 479–517.
- [28] A.M. Roma, C.S. Peskin, M.J. Berger, An adaptive version of the immersed boundary method, *J. Comput. Phys.* 153 (2) (1999) 509–534.
- [29] T. Kempe, J. Fröhlich, An improved immersed boundary method with direct forcing for the simulation of particle laden flows, *J. Comput. Phys.* 231 (9) (2012) 3663–3684.
- [30] H. Le, P. Moin, An improvement of fractional step methods for the incompressible Navier-Stokes equations, *J. Comput. Phys.* 92 (2) (1991) 369–379.
- [31] J. Kim, P. Moin, Application of a fractional-step method to incompressible Navier-Stokes equations, *J. Comput. Phys.* 59 (2) (1985) 308–323.
- [32] G.H. Wannier, A contribution to the hydrodynamics of lubrication, *Q. Appl. Math.* 8 (1950) 1–32.
- [33] C.H.K. Williamson, Vortex dynamics in the cylinder wake, *Annu. Rev. Fluid Mech.* 28 (1996) 477–539.
- [34] K. Taira, T. Colonius, The immersed boundary method: a projection approach, *J. Comput. Phys.* 225 (2) (2007) 2118–2137.
- [35] 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, *J. Comput. Phys.* 227 (10) (2008) 4825–4852.
- [36] J. Park, K. Kwon, H. Choi, Numerical solutions of flow past a circular cylinder at Reynolds numbers up to 160, *KSME Int. J.* 12 (6) (1998) 1200–1205.
- [37] T. Ye, R. Mittal, H.S. Udaykumar, W. Shyy, An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries, *J. Comput. Phys.* 156 (1999) 209–240.
- [38] A.L.F. Lima, E. Silva, A. Silveira-Neto, J.J.R. Damasceno, Numerical simulation of two-dimensional flows over a circular cylinder using the immersed boundary method, *J. Comput. Phys.* 189 (2003) 351–370.
- [39] M.N. Linnick, H.F. Fasel, A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains, *J. Comput. Phys.* 204 (1) (2005) 157–192.
- [40] S.C.R. Dennis, G.-Z. Chang, Numerical solutions for steady flow past a circular cylinder at Reynolds numbers up to 100, *J. Fluid Mech.* 42 (3) (1970) 471–489.
- [41] D.V. Le, B.C. Khoo, J. Peraire, An immersed interface method for viscous incompressible flows involving rigid and flexible boundaries, *J. Comput. Phys.* 220 (1) (2006) 109–138.
- [42] C.H. Su, D. Lucor, Covariance kernel representations of multidimensional second-order stochastic processes, *J. Comput. Phys.* 217 (1) (2006) 82–99.
- [43] H. Faxén, Forces exerted on a rigid cylinder in a viscous fluid between two parallel fixed planes, *Proc. Swed. Inst. Eng. Res.* 187 (1946) 1–13.
- [44] A.B. Richou, A. Ambari, J.K. Naciri, Drag force on a circular cylinder midway between two parallel plates at very low Reynolds numbers, part 1: Poiseuille flow (numerical), *Chem. Eng. Sci.* 59 (15) (2004) 3215–3222.