

## Research Article

# A Three-Dimensional, Immersed Boundary, Finite Volume Method for the Simulation of Incompressible Heat Transfer Flows around Complex Geometries

Hassan Badreddine,<sup>1,2</sup> Yohei Sato,<sup>3</sup> Matthias Berger,<sup>1,2</sup> and Bojan Ničeno<sup>3</sup>

<sup>1</sup>Future Cities Laboratory, Singapore-ETH Centre, 1 Create Way, No. 06-01 CREATE Tower, Singapore 138602

<sup>2</sup>Department of Architecture, ETH Zurich, Building HIT, Wolfgang-Pauli Str. 27, 8093 Zurich, Switzerland

<sup>3</sup>Nuclear Energy and Safety Department, Paul Scherrer Institute (PSI), 5232 Villigen, Switzerland

Correspondence should be addressed to Hassan Badreddine; [hassan@arch.ethz.ch](mailto:hassan@arch.ethz.ch)

Received 20 December 2016; Revised 3 April 2017; Accepted 30 April 2017; Published 19 June 2017

Academic Editor: Michael Harris

Copyright © 2017 Hassan Badreddine et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The current work focuses on the development and application of a new finite volume immersed boundary method (IBM) to simulate three-dimensional fluid flows and heat transfer around complex geometries. First, the discretization of the governing equations based on the second-order finite volume method on Cartesian, structured, staggered grid is outlined, followed by the description of modifications which have to be applied to the discretized system once a body is immersed into the grid. To validate the new approach, the heat conduction equation with a source term is solved inside a cavity with an immersed body. The approach is then tested for a natural convection flow in a square cavity with and without circular cylinder for different Rayleigh numbers. The results computed with the present approach compare very well with the benchmark solutions. As a next step in the validation procedure, the method is tested for Direct Numerical Simulation (DNS) of a turbulent flow around a surface-mounted matrix of cubes. The results computed with the present method compare very well with Laser Doppler Anemometry (LDA) measurements of the same case, showing that the method can be used for scale-resolving simulations of turbulence as well.

## 1. Introduction

Computational fluid dynamics (CFD) has reached a development level at which it is routinely applied in the industrial environment, at least for single-phase flows. Grid generation techniques and methods for discretization of governing equations all play important roles in successful application of CFD. Clearly, flow simulation is facilitated by the ever-increasing computational power on the one hand and the development of more efficient numerical methods on the other. Prominent examples of these algorithmic advances are the multigrid methods [1] and direct solvers based on Fast Fourier Transforms (FFTs). The most efficient numerical algorithms are usually defined for structured orthogonal grids, suitable only for problems occurring in simple, canonical geometries. Compared to unstructured grids, structured grids take less computational time and data storage.

Flow simulations around complex geometries can be achieved in two ways: either by using unstructured codes already applicable for complex geometries or by tailoring the existing highly efficient and/or highly accurate structured methods to nontrivial geometries. The former approach has already been adopted by commercial CFD vendors and, as such, represents the state of the art. Indeed, leading commercial CFD solvers are based on unstructured grid methods and can be applied to complex geometries. The latter approach can go in two directions. One possibility is to define the underlying highly efficient/accurate method to boundary fitted grids [2, 3]. The other possibility is to leave the basic numerical method in its original form, that is, to discretize the system of governing equations using the original method on orthogonal grids and then to make the necessary changes to include the effects of a complex surface immersed in the computational domain. Such methods, usually referred to as

immersed boundary methods (IBMs), can conveniently be divided into two broad classes (*forcing* and *cut-cells methods*), depending on the way the boundary conditions are imposed on the immersed body.

In the *forcing* methods, one can find the continuous forcing approaches [4–7] where a continuous force is added to the governing equations before discretization, making the formulation of such approaches independent of the spatial discretization. On the other hand, in the discrete forcing approaches [8–12], the governing equations are first discretized on the grid neglecting the presence of the immersed body. Then interpolation schemes are used for the grid points at the vicinity of the immersed body in such a way that the required boundary condition at the interface is satisfied. Compared to continuous forcing approaches, discrete forcing approaches have the advantage of allowing direct control over stability and numerical accuracy.

In the second class of IBM, the *cut-cells* methods, once the body is immersed in the domain, and the intersections between computational cells and immersed body are evaluated and the fraction of the cell volumes and cell face surfaces are computed. This procedure is required in order to modify the discretized system of equations for the truncated cells for the purpose of imposing the boundary conditions on the immersed body. Developments in this area include [13–16]. A comparative overview on IBM approaches is given in [17, 18].

Several works exist in the literature regarding extending IBM approaches to solve the energy equation around complex geometries by imposing Dirichlet and Neumann boundary conditions [19–27]. Conjugate heat transfer simulations, where the energy equation is also solved inside the immersed body, using IBM were also reported in [28–31]. However, to the best of the authors' knowledge, most of these works are based on the extension of the IBM *forcing* method to heat transfer. In this work, the implementation and development of a new IBM based on a *cut-cell* approach to simulate heat transfer flow will be presented.

## 2. Numerical Method

A new cut-cell approach is described in this section and implemented in the in-house code PSI-BOIL [32]. In this work, we assume incompressibility of fluid and constant physical properties. The governing equations are the Navier-Stokes equations and the energy balance equation written here in integral formulation:

$$\oint_S \mathbf{u} d\mathbf{S} = 0, \quad (1)$$

$$\int_V \frac{\partial \mathbf{u}}{\partial t} dV + \oint_S \mathbf{u}\mathbf{u} d\mathbf{S} = -\frac{1}{\rho} \oint_S p d\mathbf{S} + \nu \oint_S \nabla \mathbf{u} d\mathbf{S} + \mathbf{F}_{\text{Boussinesq}}, \quad (2)$$

$$\int_V \frac{\partial T}{\partial t} dV + \oint_S \mathbf{u}T d\mathbf{S} = \frac{\lambda}{\rho c_p} \oint_S \nabla T d\mathbf{S}. \quad (3)$$

In the above equations,  $\rho$  is the density,  $\mathbf{u}$  is the velocity vector,  $\nu$  is the kinematic viscosity,  $T$  is the temperature,  $\lambda$  is the thermal conductivity, and  $c_p$  is the specific heat capacity.  $\mathbf{F}_{\text{Boussinesq}}$  represents the Boussinesq approximation.  $V$  is the cell volume and  $S$  is the area vector of cell surface.

The governing equations are integrated in time using a semi-implicit projection method [33]. For the spatial discretization, the orthogonal staggered finite volume method is used, with the second-order central difference being used for the diffusive terms and advection terms. Momentum equations are first solved for a tentative velocity field,  $\mathbf{u}^*$ , which is not divergence-free (i.e.,  $\nabla \cdot \mathbf{u}^* \neq 0$ ):

$$\int_V \frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} dV + \mathbf{H}^{n-1/2} = \mathbf{P}^n + \mathbf{D}^{n+1/2} + \mathbf{F}^n. \quad (4)$$

Here,  $\mathbf{H}$ ,  $\mathbf{D}$ , and  $\mathbf{P}$  represent the advective, viscous, and pressure terms in the momentum equations (see (4)), respectively. The superscript  $n$  represents current time step and  $n - 1/2$  indicates Adams-Bashforth time discretization scheme, whereas  $n + 1/2$  indicates an implicit time level. Once the tentative velocity field  $\mathbf{u}^*$  is available, the algorithm proceeds by the solution of the pseudo-pressure  $\tilde{p}$  elliptic equation:

$$\oint_S \nabla \tilde{p} d\mathbf{S} = \frac{\rho}{\Delta t} \oint_S \mathbf{u}^* d\mathbf{S}. \quad (5)$$

The final step of the algorithm is projection of the velocity onto a divergence-free field  $\mathbf{u}^{n+1}$ :

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \frac{1}{\rho} \nabla \tilde{p} \Delta t, \quad (6)$$

$$p^{n+1} = p^n + \tilde{p}. \quad (7)$$

As argued in [34], the pressure in this method converged only at a first-order rate. To converge at a second-order term, the pressure should be updated as follows:

$$p^{n+1} = p^n + \tilde{p} - \frac{\nu \Delta t}{2} \nabla^2 \tilde{p}. \quad (8)$$

Diagonally preconditioned conjugate gradient method is used to solve (4) and a variant of the additive correction multigrid method [35] is used to solve (5). However, it should be noted that multigrid methods, in certain circumstances, could be inefficient and require large computational time [22]. Other solvers such as FFT [36] could be also used.

### 2.1. Modification for an Immersed Body

**2.1.1. Cutting the Cells.** In the first step of our immersed body approach, the immersed body is imported from a file in stereolithographic (STL) format. STL files define triangulated surfaces by the coordinates of the triangle's vertices and normal surface vectors. The next step is to construct the intersection between the edges of the Cartesian grid cell with the triangles of the immersed body. The edge-triangle intersections are then used to define the cutting plane for cells  $i, j, k$  (Figure 1).

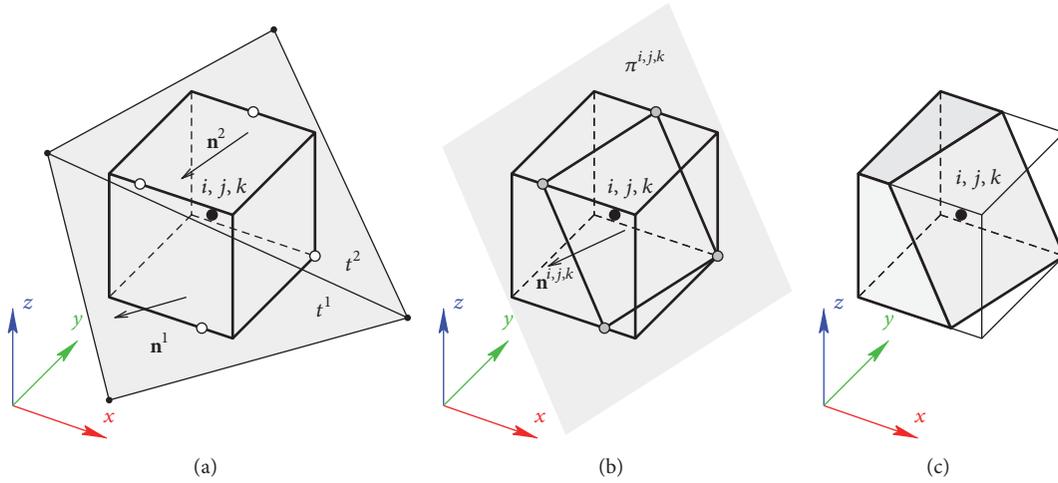


FIGURE 1: Procedure for immersing a body into a Cartesian grid: (a) intersecting the edges of cell  $i, j, k$  with triangulated surface  $t^1, t^2$ ; (b) creation of the cutting plane  $\pi^{i,j,k}$ ; and (c) definition of the final computational cell.

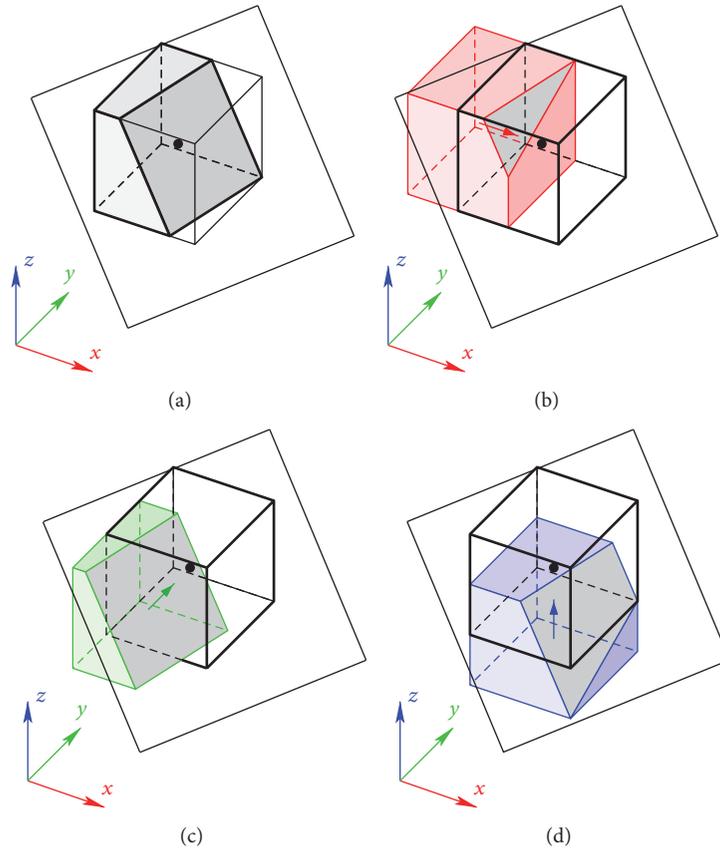


FIGURE 2: Each grid is cut separately: (a) scalar cell and (b), (c), and (d) momentum cells in  $x$ -,  $y$ -, and  $z$ -directions, respectively.

Since we are using a staggered grid approach, we have four finite volume grids: one for the scalar variables and one each for the three velocity components. To cope with immersed boundaries, we cut all three grids separately, using the procedure outlined in Figure 1, as illustrated in Figure 2.

**2.1.2. Cell Classification.** Once the scalar cells (pressure and temperature cells) are cut, they are classified into three sets:

- (i) *Cut* cells, belonging to set C, are the cells which are intersected by the immersed boundary. The fluid volume fraction  $f_V$  for these cells is  $0 < f_V < 1$ .

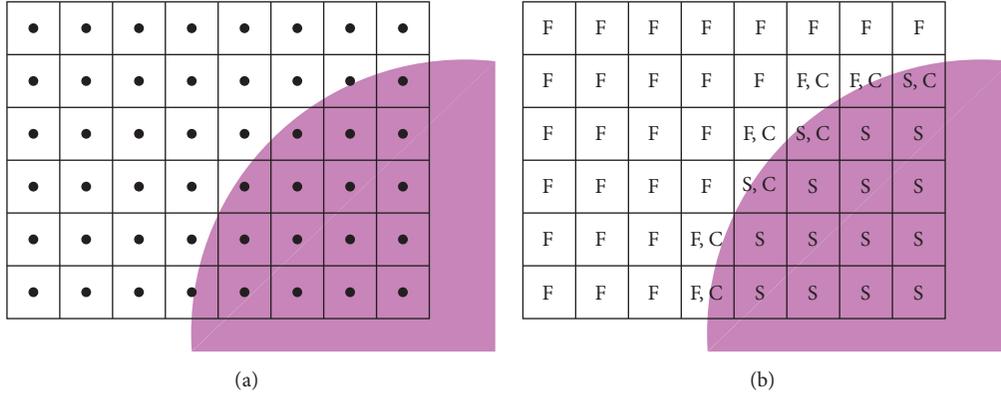


FIGURE 3: A 2D computational domain with classified cells: (a) scalar cell centers and (b) sets. The purple shaded area represents immersed body.

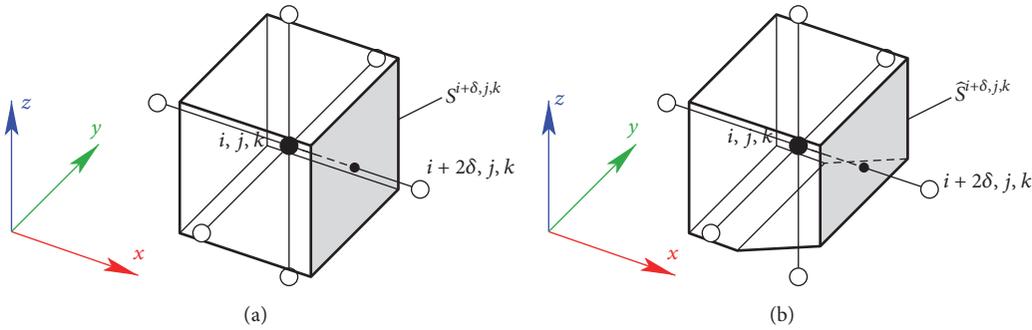


FIGURE 4: A cell face,  $i + \delta, j, k$  before (a) and after (b) cutting. Cell face area before the cutting was  $S^{i+\delta, j, k}$  and after  $\hat{S}^{i+\delta, j, k}$ .

- (ii) *Fluid* cells, belonging to set F, are the cells whose center remains in the fluid part of the domain, no matter if they are cut or not. This means that  $f_V > 0.5$  for these cells.
- (iii) *Solid* cells, belonging to set S, are all those which are not in set F, that is the cells whose center lies in the solid part of the domain.

The classification of momentum (velocity) cells, whether they are ON (in the fluid) or OFF (in the immersed body), is related to the scalar cells. For instance, the  $u$ -component velocity cell is considered to be in the fluid if the two surrounding scalar cells in  $I$ -direction (i.e., cell (i) and cell (i-1)) are both in fluid). A similar procedure is applied to the other two velocity components.

Figure 3(a) shows a 2D computational grid, with each scalar cell center being denoted with a black dot. Immersed body is represented by a shaded area. The cells whose cell center lies in fluid belong to set F, while the rest of the cells belong to set S as illustrated in Figure 3(b). The cells which are cut with the immersed body also belong to set C.

**2.1.3. Inertial Terms.** Cuts performed on the computational cells change their geometrical properties, and this has to be incorporated in the discretized system of equations. Modification of the inertial terms for the presence of an

immersed body is straightforward. For a cell containing an immersed body, the inertial term, in terms of a generalized variable  $\phi$ , is multiplied by the volume fluid fraction of cell  $f_{V_{i,j,k}}$ :

$$\int_V \frac{\phi^* - \phi^n}{\Delta t} dV \approx \hat{I}_{i,j,k}^* - \hat{I}_{i,j,k}^n = (\phi_{i,j,k}^* - \phi_{i,j,k}^n) \frac{V_{i,j,k} f_{V_{i,j,k}}}{\Delta t}. \quad (9)$$

**2.1.4. Advective Terms.** Advective terms for the immersed body are affected by the changes in cell face areas. For each cell face which is cut, the fraction remaining in the fluid part of the cell is first estimated. As an example, the cell face at  $i + \delta, j, k$  before and after immersing a body is illustrated in Figure 4.

The fraction of the cell face area remaining in the fluid is

$$f_{i+\delta, j, k}^S = \left( \frac{\hat{S}}{S} \right)_{i+\delta, j, k}. \quad (10)$$

Using equations like (10) for all faces in the computational domain which are cut, the advective term for the general

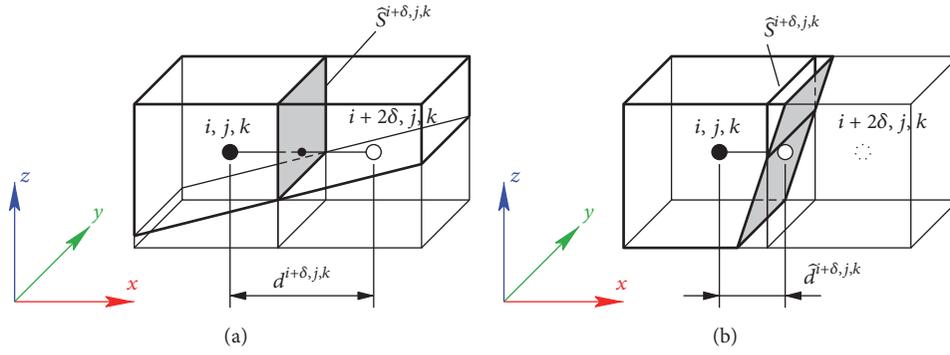


FIGURE 5: (a) Both cells are in the fluid. (b) Cell  $i + 2\delta, j, k$  in the solid. The effective diffusive flux through the connection diminishes in proportion to the reduced face area and reduced distance between cells.

scalar variable  $\phi$  for the immersed body can be evaluated as follows:

$$\oint_S \mathbf{u}\phi \, dS \approx \sum_{\delta=-1/2}^{1/2} (\widehat{H}_{i+\delta, j, k} + \widehat{H}_{i, j+\delta, k} + \widehat{H}_{i, j, k+\delta}), \quad (11)$$

where  $\widehat{H}$  represents those advective terms modified for the presence of the immersed boundary, computed from

$$\forall (i, j, k) \in F: \widehat{H}_{i+\delta, j, k} = H_{i+\delta, j, k} f_{i+\delta, j, k}^S \quad (12)$$

$$\text{if } (i + 2\delta, j, k) \in C,$$

where  $H_{i+\delta, j, k} = (\mathbf{u}\phi S_x)_{i+\delta, j, k}$  represents the unmodified advective term. Analogous expressions could be easily formulated for the  $y$ - and  $z$ -directions.

**2.1.5. Diffusion Terms.** The diffusion terms for the case of an immersed body can be written as follows:

$$\oint_S \nabla\phi \, dS \approx \sum_{\delta=-1/2}^{1/2} (\widehat{D}_{i+\delta, j, k} + \widehat{D}_{i, j+\delta, k} + \widehat{D}_{i, j, k+\delta}). \quad (13)$$

For the evaluation of the diffusion terms, modified for the presence of the immersed body ( $\widehat{D}$ ), two different cases have to be considered. The first occurs if both cells surrounding the face for which the diffusive flux is estimated belong to the fluid set (Figure 5(a)). In such a case, the distance between the cells remains unchanged, but the cell face area between them changes, affecting the diffusive flux through the cell face, and is accounted for with the factor introduced in (10). The second case occurs if one cell, say  $i + 2\delta, j, k$ , is in the solid (Figure 5(b)). In such a case, the cell in the solid becomes effectively a boundary cell, and distance between the cells decreases. That is accounted for by defining the ratio between the new  $\widehat{d}$  and the old distances between the cells:

$$f_{i+\delta, j, k}^d = \left( \frac{\widehat{d}}{d} \right)_{i+\delta, j, k}. \quad (14)$$

To summarize, corrections which need to be made for the diffusive fluxes are as follows:

$$\forall (i, j, k) \in F: \widehat{D}_{i+\delta, j, k} = \frac{D_{i+\delta, j, k} f_{i+\delta, j, k}^S}{f_{i+\delta, j, k}^d} \quad (15)$$

$$\text{if } (i + 2\delta, j, k) \in C,$$

where  $D_{i+\delta, j, k} = (\phi_{i+2\delta, j, k} - \phi_{i, j, k})(S_x/d)_{i+\delta, j, k}$  represents the unmodified diffusive flux. It should be noted that same procedure is also used to discretize the elliptic equation for the pseudo-pressure (see (5)).

The above treatment for the diffusion terms is valid for the momentum equations and for the heat equation when fixing the velocity or the temperature to a predefined value is required. Another approach for the diffusion term in the heat equation should be taken in case conduction is allowed inside the solid. For instance, when both cells  $i, j, k$  and  $i + 2\delta, j, k$  are cut but their centers are in the fluid (similar to Figure 5(a)), the diffusive flux at the cell face  $i + \delta, j, k$  is evaluated as follows:

$$\widehat{D}_{i+\delta, j, k} = \frac{\lambda_f \widehat{S}_{i+\delta, j, k} + \lambda_s (S_x - \widehat{S}_{i+\delta, j, k})}{\widehat{d}^{i+\delta, j, k}} (\phi_{i+2\delta, j, k} - \phi_{i, j, k}), \quad (16)$$

where  $\lambda_f$  and  $\lambda_s$  are the thermal conductivity of the fluid and solid, respectively. The second case is when the center of cell  $i, j, k$  is in the fluid and the center of  $i + 2\delta, j, k$  is in the solid (similar to Figure 5(b)). In this case, the diffusive flux at cell face  $i + \delta, j, k$  is multiplied by

$$\widehat{D}_{i+\delta, j, k} = S_x \cdot \frac{\lambda_f \lambda_s}{\lambda_f (d^{i+\delta, j, k} - \widehat{d}_{i+\delta, j, k}) + \lambda_s \widehat{d}_{i+\delta, j, k}} (\phi_{i+2\delta, j, k} - \phi_{i, j, k}). \quad (17)$$

The above equation can be proved by considering the continuity of the diffusion flux at the interface between the solid and the fluid:

$$\lambda_f \frac{T_I - T_{ijk}}{\hat{d}^{i+\delta,j,k}} = \lambda_s \frac{T_{i+2\delta,j,k} - T_I}{\hat{d}^{i+\delta,j,k} - \hat{d}^{i+\delta,j,k}}, \quad (18)$$

where  $T_I$  is the temperature at the interface. The diffusion flux at cell face  $i + \delta, j, k$  is

$$\lambda \frac{\partial T}{\partial x_{i+\delta,j,k}} \approx \lambda_f \frac{T_I - T_{ijk}}{\hat{d}^{i+\delta,j,k}}. \quad (19)$$

From (18),  $T_I$  could be formulated in terms of  $T_{ijk}$  and  $T_{i+2\delta,j,k}$  and then inserted in (19) to get (17).

The corrections for the presence of the immersed body, defined for the inertial term by (9), the advection term by (12), and the diffusive term by (15), are performed before compiling the final discretized system of equations. Once this is done, additional care has to be taken for the cells that belong to the solid set. Their coefficients in the discretized system have to be modified in a way which ensure that the values taken by the variables (temperature and velocity) in these cells remain equal to the predefined value (in case no conduction is solved in the solid).

**2.1.6. Velocity Projection.** The discrete form of the velocity projection equation (6) is

$$u_{i+\delta,j,k}^{n+1} = u_{i+\delta,j,k}^* + \frac{\Delta t}{\rho} \frac{\tilde{P}_{i,j,k} - \tilde{P}_{i+2\delta,j,k}}{d_{i+\delta,j,k}}. \quad (20)$$

It should be noted that the velocity is only updated for the momentum cells which are inside the fluid (i.e., when the pressure scalar cells  $i, j, k$  and  $i + 2\delta, j, k$  are both in the fluid).

### 3. Results

**3.1. Steady-State Heat Conduction with a Source Term.** As a first step to assess the performance of our immersed boundary method, we solve the heat conduction equation with a source term:

$$\int_V \frac{\partial T}{\partial t} dV = \oint_S \lambda \nabla T \cdot dS + \int_V \dot{q} dV \quad [W]. \quad (21)$$

The computational domain (Figure 6) is a rectangular channel of dimension  $1.22 \times 0.125 \times 1$  in the  $x$ -,  $y$ -, and  $z$ -directions, respectively. A rectangular immersed body of width 0.22 divides the domain into two parts: the  $x$ -coordinates of the “solid” domain range from  $-0.72$  to  $-0.5$ , while the  $x$ -coordinates of the “solid-source” domain range from  $-0.5$  to  $0.5$ . The full domain is meshed with  $64 \times 2 \times 64$  grid cells. Given the fact that the solution of this case is two-dimensional, only two grid cells are used for the  $y$ -direction. The heat equation (see (21)) is solved with the source term set to  $100 \text{ W/m}^3$  inside the “solid-source” domain and to zero inside the “solid” domain. The following boundary conditions are used for temperature  $T$ :

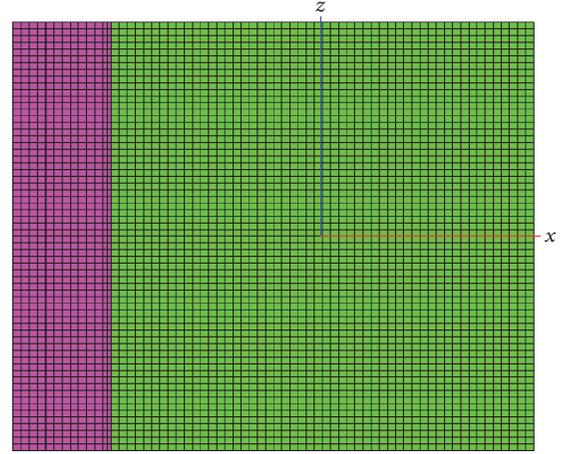


FIGURE 6: The computational domain used to solve the heat conduction equation. The green cells belong to the “solid-source” domain and the purple cells to the immersed body (“solid” domain).

- (i)  $T = 289 \text{ [K]}$  at  $x = -0.72$  and  $T = 300$  at  $x = 0.5$
- (ii) Periodic boundary condition in the  $y$ -direction
- (iii)  $\partial T / \partial n = 0$  in the  $z$ -direction.

The thermal conductivity  $\lambda$  for the “solid-source” domain was set to 0.5. Three different cases are considered: the first one without the immersed body (i.e., the domain size in the  $x$ -direction ranges from  $-0.5$  to  $0.5$  with  $T$  being set to 300 for the west and east boundaries). In this case, (21) has an analytical solution:

$$T(x) = \frac{\dot{q}}{2\lambda} \left( (x - 0.5) - (x - 0.5)^2 \right) + 300. \quad (22)$$

For the other cases, the immersed body is included with  $\lambda(\text{body})$  taking two values: one and two. For  $\lambda(\text{body}) = 1$ , the temperature at the interface between the fluid and body domains would be equal to 300. This will allow us to compare the results with the analytical data.

The diffusion term was discretized in time using Crank-Nicolson scheme and in space using the second-order central difference scheme. Equation (21) is solved until steady-state solution is reached. Temperature profiles for all the cases are shown in Figure 7. Excellent agreement with the analytical solution is obtained for the case simulated without IBM and for  $\lambda(\text{body}) = 1$ . Inside the body zone, a linear profile for the temperature is observed as expected since the heat equation is solved without a source term there. This profile is also steeper for  $\lambda = 1$  compared to  $\lambda = 2$  as expected.

**3.2. Natural Convection in a Square Cavity.** In this section, we validate our immersed boundary method for natural convection flow in square domain named “fluid.” The computational domain is similar to the one used in Section 3.1. The only difference is that the width of the immersed body is 0.1 instead of 0.22. The fluid governing equations (see (1) and (2)) and the energy balance equation (see (3)) are solved inside the fluid domain and the temperature is fixed to  $T_{\text{hot}} = 301 \text{ K}$

TABLE 1: Comparison between the IBM and the benchmark (data) solutions for the natural convection simulation in a square cavity.

Ra	$10^3$		$10^4$		$10^5$		$10^6$	
	IBM	data	IBM	data	IBM	data	IBM	data
$\overline{Nu}$	1.119	1.118	2.249	2.243	4.528	4.519	8.859	8.800
$u_{\max}$	3.645	3.649	16.220	16.178	34.755	34.730	64.854	64.630
$z_{\max}$	0.816	0.813	0.824	0.823	0.855	0.855	0.847	0.850
$w_{\max}$	3.732	3.697	19.766	19.617	68.709	68.590	219.696	219.360
$x_{\max}$	0.179	0.178	0.119	0.119	0.067	0.066	0.0418	0.0379

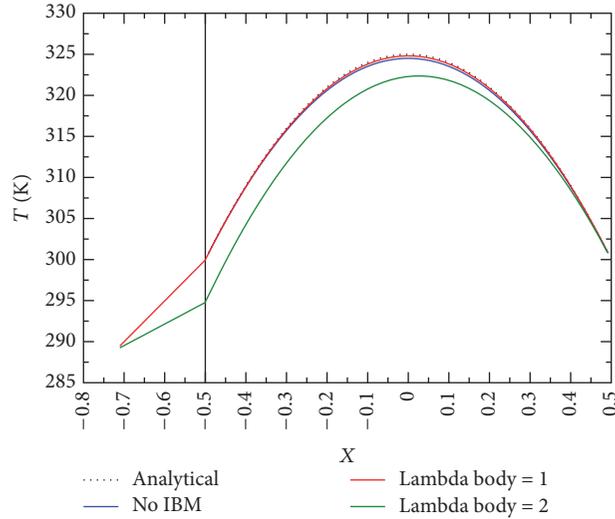


FIGURE 7: Temperature profile inside the domain. The vertical black line corresponds to the interface between the “solid-source” and body domains.

inside the body zone. The following boundary conditions are used:

- (i)  $T_{\text{cold}} = 300 \text{ K}$  at  $x = 0.5$
- (ii)  $\partial T / \partial n = 0$  for the top and bottom faces ( $z$ -direction)
- (iii) Periodic boundary condition for  $T$  and  $\mathbf{u}$  at the south and north faces ( $y$ -direction)
- (iv) No-slip boundary condition for  $\mathbf{u}$  at  $x$ - and  $z$ -directions,
- (v)  $\partial \bar{p} / \partial n = 0$ ,

where  $n$  is the normal vector at corresponding boundary. The whole domain is meshed with  $128 \times 2 \times 128$  grid cells. Only two cells are used for the  $y$ -direction because the solution is two-dimensional for this case. This problem can be characterized with two nondimensional numbers: the Rayleigh ( $Ra = g\beta\Delta TL^3/\nu\alpha$ ) and the Prandtl ( $Pr = \nu/\alpha$ ) number. Four cases are simulated with different Rayleigh numbers ( $10^3$  to  $10^6$ ). Prandtl number is set to 0.71 for all the cases. The governing equations are solved until steady state is reached.

Results of averaged Nusselt number and maximum vertical dimensionless velocity on the horizontal mid-plane and its location and maximum horizontal dimensionless velocity on the vertical mid-plane and its location are compared

against the benchmark data published in [37]. The dimensionless velocity is obtained by scaling the velocity with  $\alpha/L$ , where  $\alpha$  is the thermal diffusivity and  $L$  is the length of square cavity. The average Nusselt number  $\overline{Nu}$  is defined as follows:

$$\overline{Nu} = \int_{\text{hot wall}} \frac{\partial T}{\partial x} dz. \quad (23)$$

Results are presented in Table 1 and very good agreement is achieved compared to the benchmark data. The temperature profile for  $Ra = 10^5$  case is shown in Figure 8.

**3.3. Natural Convection in a Square Enclosure with a Circular Cylinder.** In this section, the simulation of natural convection in a square enclosure of length one with an immersed circular cylinder of radius 0.2 centered at the center of the enclosure is presented. The temperature of the walls of the square enclosure is fixed to  $T_{\text{cold}} = 300 \text{ K}$ , while  $T_{\text{hot}} = 301 \text{ K}$  is imposed on the cells that are inside the circular cylinder. Prandtl number was set to 0.71 and four different cases were simulated depending on Rayleigh number that ranges from  $10^3$  to  $10^6$ .

The domain (Figure 9) is meshed with 256 cells in the  $x$ - and  $z$ -directions, while only two cells were used in  $y$ -directions, where periodic boundary condition was used, since the solution is two-dimensional ( $x$ - $z$  plane).

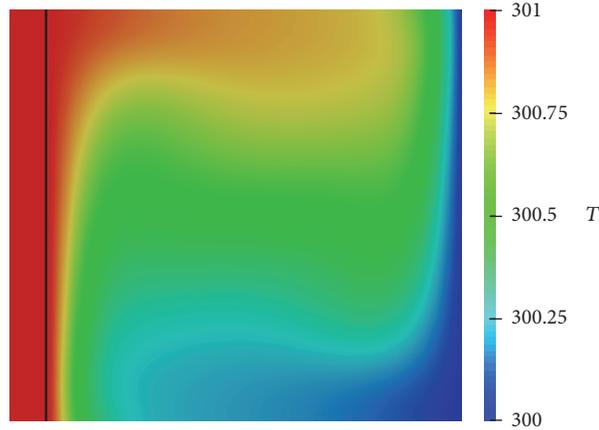


FIGURE 8: Temperature profile for natural convection flow in square cavity with  $Ra = 10^5$  simulated using the IBM approach. The vertical black line corresponds to the interface between the immersed body and the fluid domain.

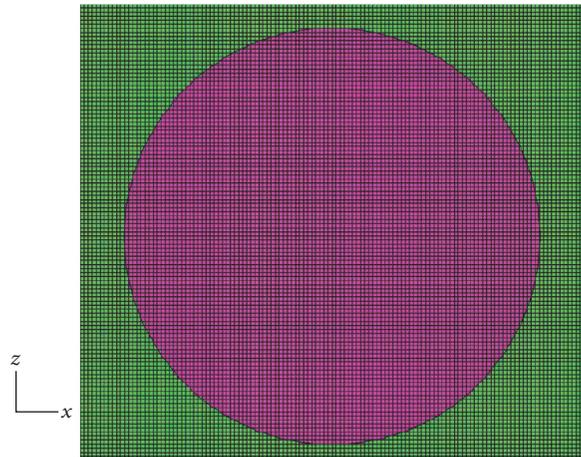


FIGURE 9: A section of the mesh used to simulate natural convection of heated cylinder. The green cells belong to the fluid domain and the purple cells to the immersed cylinder.

The governing equations are solved until steady state is reached. The local Nusselt number ( $Nu$ ) distribution along the surface of the square walls (Figure 10), isothermals (Figure 11), and streamlines (Figure 12) for all the cases were compared against results published in [25]. The local Nusselt number is defined as follows:

$$Nu = \frac{\partial T}{\partial n} \Big|_{\text{wall}}, \quad (24)$$

where  $n$  is the wall-normal vector. As can be seen, the results obtained using our IBM approach are in very good agreement with the results of Kim et al. [25].

**3.4. Turbulent Flow around a Matrix of Cubes.** In this section, we validate our immersed boundary method for isothermal turbulent flow around a surface-mounted matrix of cubes. As reference data, LDA measurements from [38] have been used. This case was used for 6th and 8th ERCOFTAC/IAHR/COST Workshops on refined flow and turbulence modeling [39, 40] and has been analyzed by a number of groups. The physics of

the flow is well understood and will not be explained here in great detail. Our focus in this work is to compare computed mean and fluctuating velocity profiles against the measured data.

The representation of the problem domain including all relevant dimensions is sketched in Figure 13. An equidistantly spaced matrix of cubes is mounted on the lower surface a rectangular channel. The channel height is  $H = 51$  mm, height of the cubes is  $h = 15$  mm, and the pitch between the cubes is  $S = 60$  mm. The experimental setup consists of  $25 \times 10$  cubes in the streamwise and spanwise directions, respectively. Measurements were taken around the 18th row of cubes, where the flow was expected to be spatially fully developed. This aspect makes this test case particularly attractive, as it allows analysis to be carried out in a periodic box, avoiding the specification of inlet conditions. The dashed lines in Figure 13 depict the computational domain. Dimensions are  $S \times S \times H$  in the streamwise ( $x$ ), spanwise ( $y$ ), and wall-normal directions ( $z$ ), respectively. The origin of the coordinate system is located on the lower surface, centrally

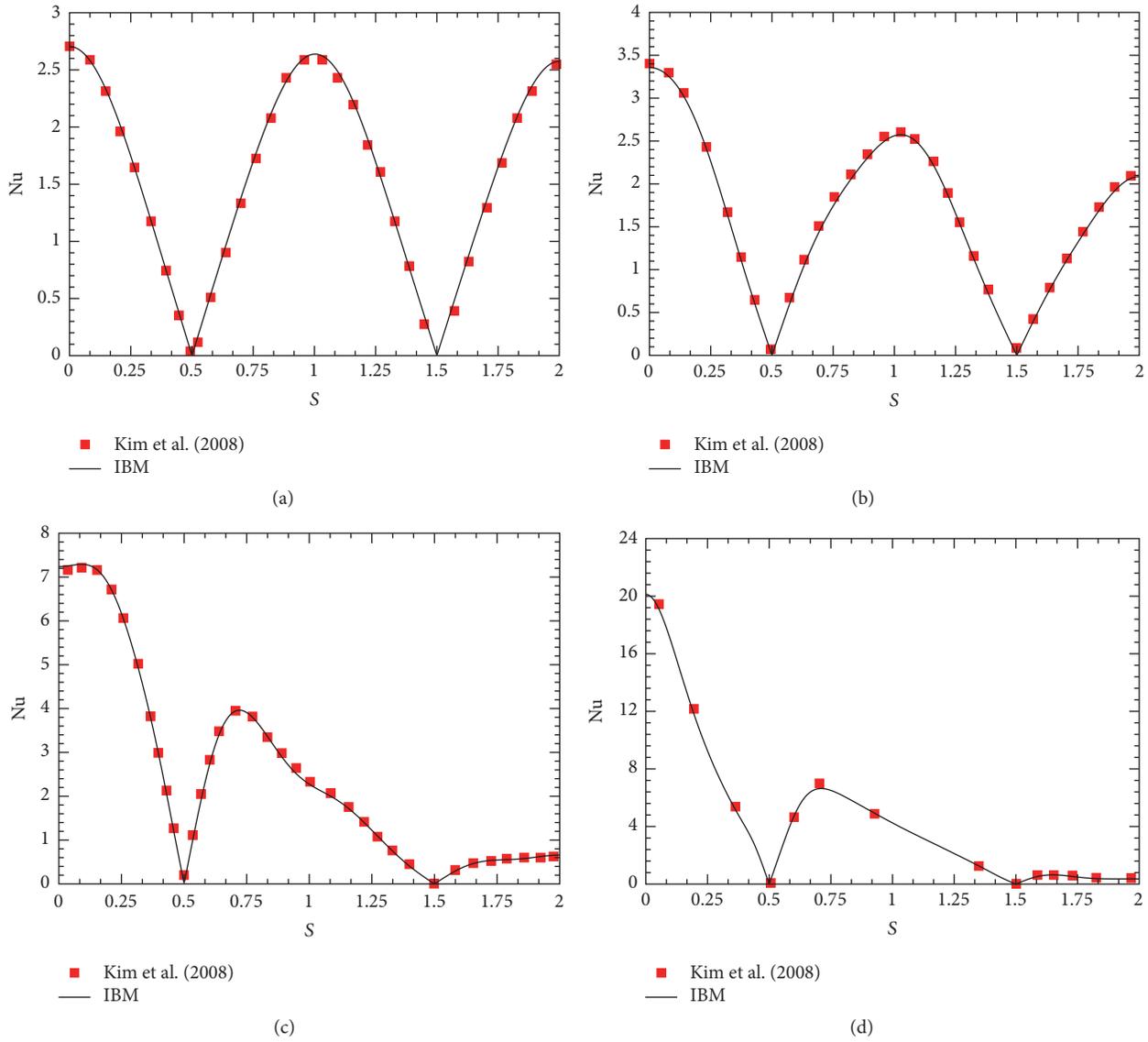


FIGURE 10: Local Nusselt number distribution along the surface of the square enclosure for  $Ra = 10^3$  (a),  $Ra = 10^4$  (b),  $Ra = 10^5$  (c), and  $Ra = 10^6$  (d).

with respect to the computational domain and the cube under consideration, as shown in Figure 13.

The working fluid is air, with a bulk velocity of  $U_b = 3.86$  m/s, resulting in a Re based on cube height of  $Re_h = 3854$ . This is a modest Re number, allowing simulations to be performed without the use of turbulence model, that is, as a DNS. Grid resolution is set to  $128 \times 128 \times 108$  cells in  $x$ -,  $y$ -, and  $z$ -directions, respectively. The grid is uniform in all directions; that is, no stretching towards the wall was used in any coordinate direction. Cell dimensions in  $x$ - and  $y$ -directions are  $\Delta x = \Delta y = 0.46875$  mm, and the number of cells in normal direction ( $z$ ) was chosen in such a way as to make  $\Delta z$  as close as possible to  $\Delta x$  and  $\Delta y$ ; in the present case,  $\Delta z = 0.472$ . The total number of cells was close to 1.8 million. For stability reason, the time step was set to  $2 \times 10^{-5}$ , resulting in a maximum Courant-Friedrich-Levy

(CFL) number around the value of 0.3. The total number of time steps performed in this simulation was  $10^5$ , but only the last  $8 \times 10^4$  were used for computation of turbulent statistics.

The computed mean and fluctuating streamwise and spanwise velocity profiles are compared with measurements from [38] in Figures 14–16. Figure 14 shows profiles in the vertical (top row) and horizontal (bottom row) mid-plane of the cube at  $x = -12$  mm. Figures 15 and 16 show the same, but for  $x = 3$  mm and  $x = 12$  mm, respectively. Computed mean velocity profiles match measured values well at all three measuring locations. The same can also be claimed for the streamwise velocity fluctuations. The most striking agreement is probably the one in the vertical plane at  $x = -3$  mm, shown in Figure 15(b) at the top. The sharp peak in the streamwise velocity fluctuations is captured exactly and is

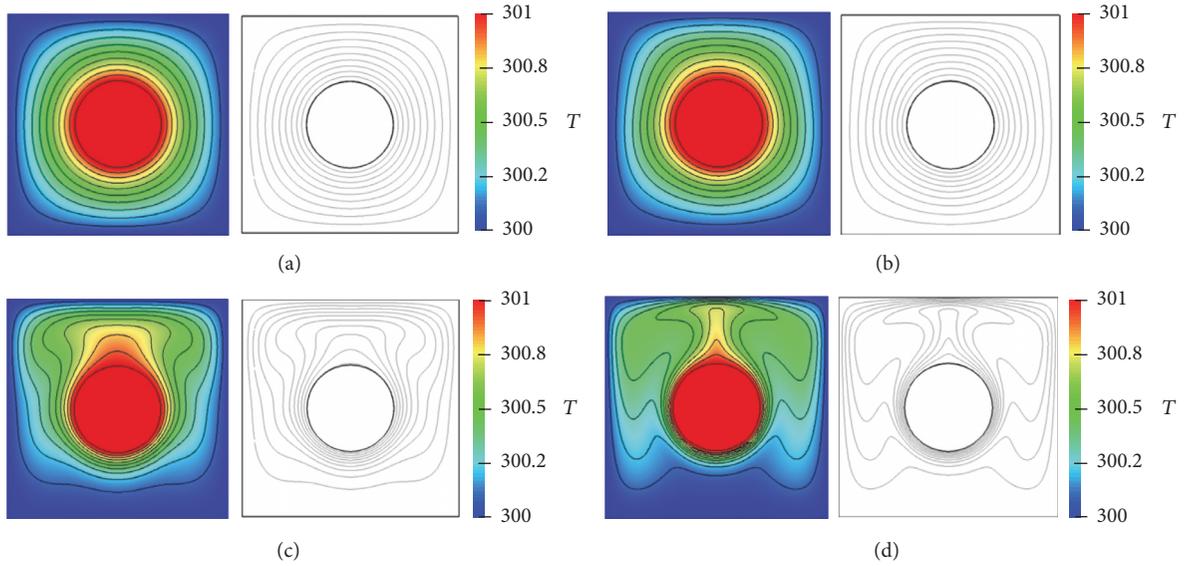


FIGURE 11: Comparison of isothermals contours simulated by our IBM approach (coloured figures) and by Kim et al. [25] for  $Ra = 10^3$  (a),  $Ra = 10^4$  (b),  $Ra = 10^5$  (c), and  $Ra = 10^6$  (d).

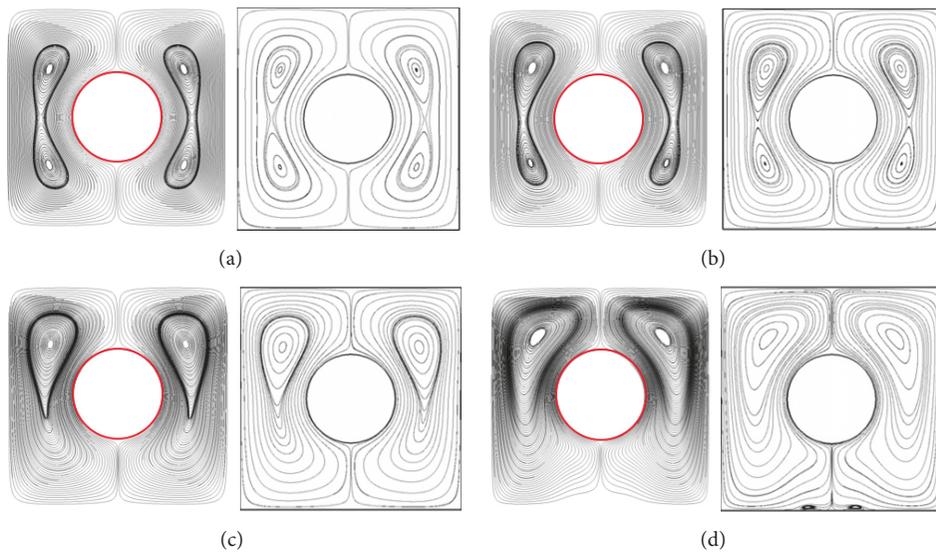


FIGURE 12: Comparison of velocity streamlines coloured by temperature simulated by our IBM approach (cylinder shown in red) and by Kim et al. [25] for  $Ra = 10^3$  (a),  $Ra = 10^4$  (b),  $Ra = 10^5$  (c), and  $Ra = 10^6$  (d).

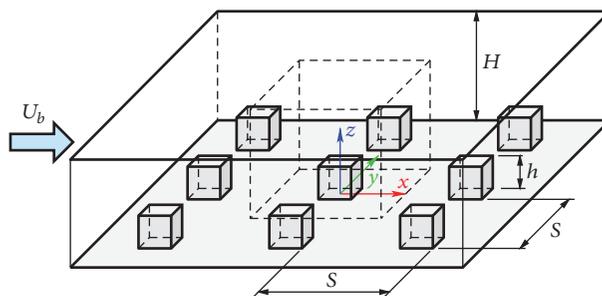


FIGURE 13: Schematic representation of the matrix of cubes test case with all relevant dimensions. The computational box is represented by the dashed lines, and coordinate system origin is indicated.

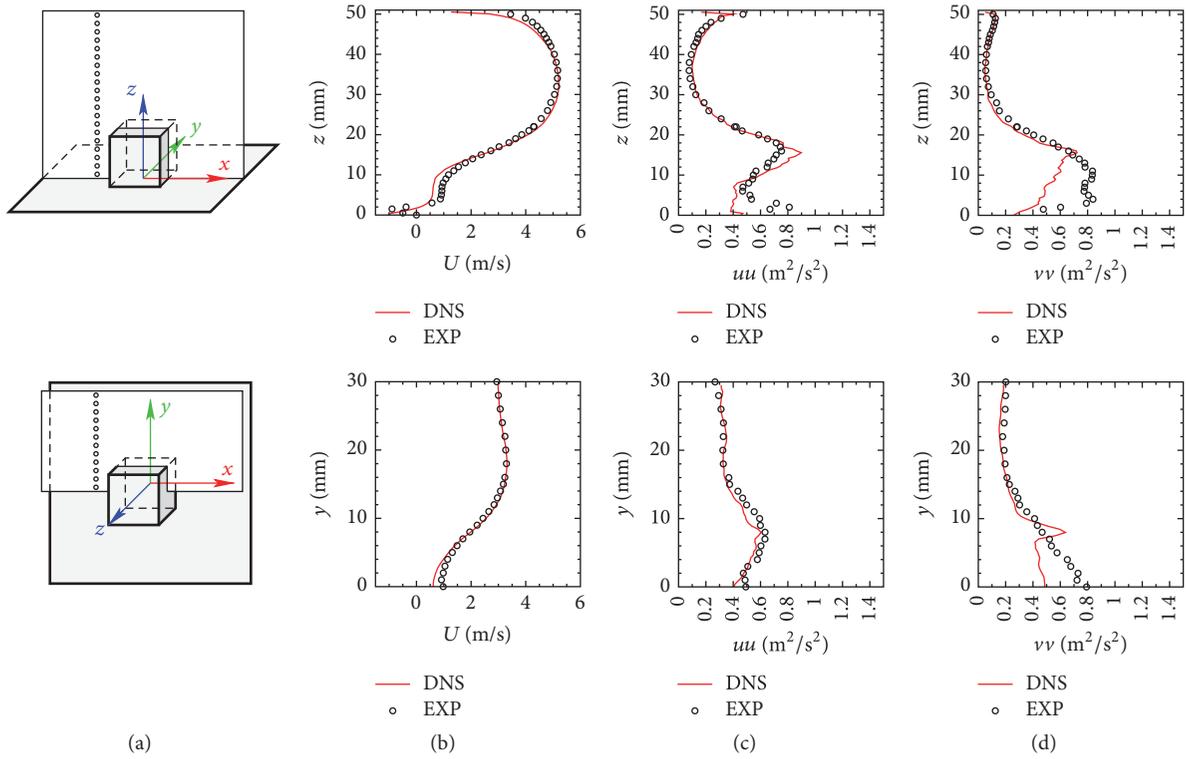


FIGURE 14: Comparison of computed and measured velocity at  $x = -12$  mm: (a) position of the measuring plane and line for LDA, (b) mean streamwise, (c) fluctuating streamwise, and (d) fluctuating spanwise. The top row shows comparison in vertical ( $x$ - $z$ ) plane and the bottom row in horizontal ( $x$ - $y$ ) plane. Code predictions are given by the solid line (DNS) and the measurements by open circles (EXP).

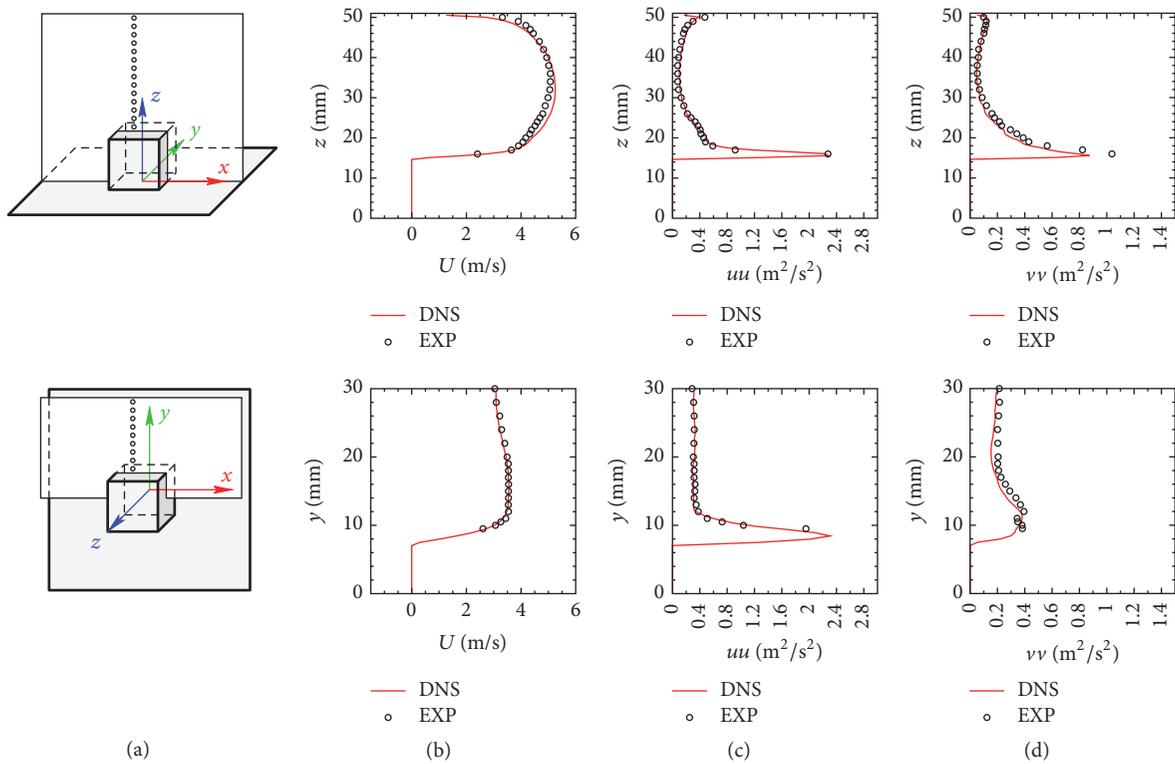


FIGURE 15: Same as in Figure 14 but for  $x = -3$  mm.

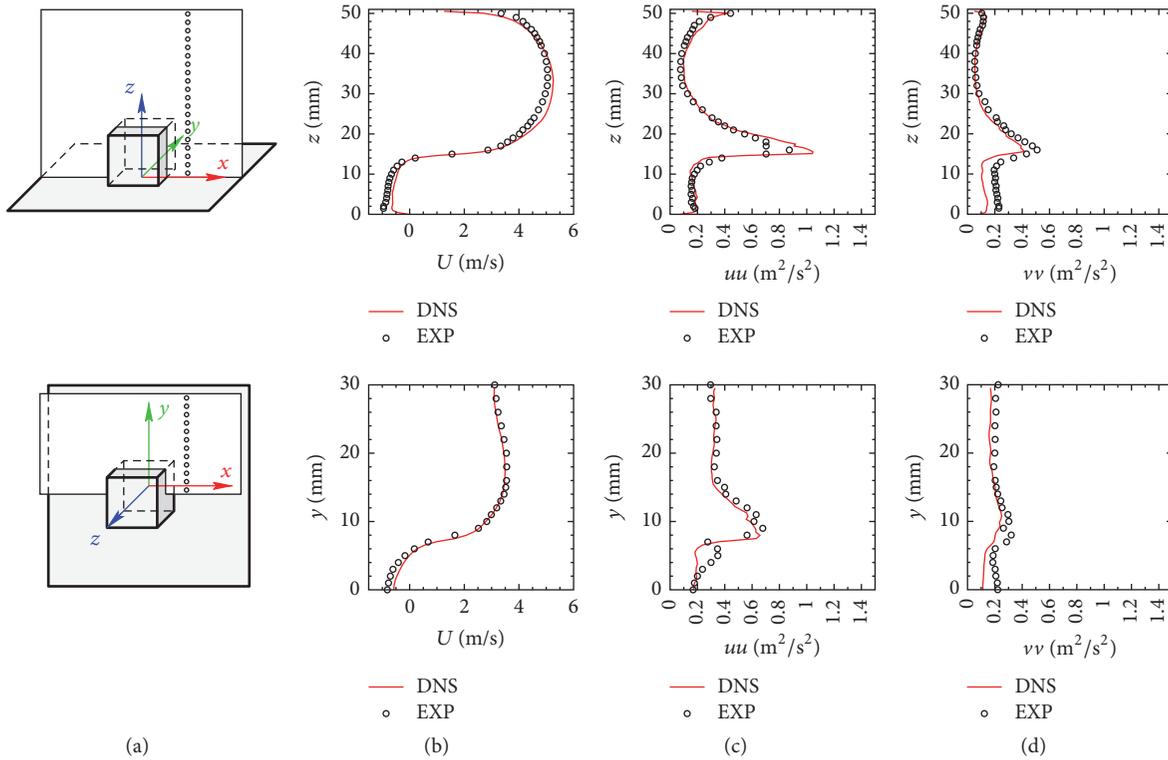


FIGURE 16: Same as in Figure 14 but for  $x = 12$  mm.

very close for the spanwise component as well. Overall, the agreement is good, and we can conclude that the immersed boundary approach we describe can be used for large-scale simulations of turbulence, at least for the modest Re number considered in this work. With increasing Re, a subgrid scale (SGS) model would need to be used, as well as wall functions. However, implementation of state-of-the-art SGS models on a Cartesian grid is a straightforward task, and this is an added benefit of the current approach.

#### 4. Conclusions

In this paper, a new immersed boundary method (IBM) based on the *cut-cell* approach to simulate three-dimensional, incompressible flows with heat transfer has been described. The governing equations are first discretized using the Cartesian, staggered, finite volume method. The resulting systems of linear equations are then corrected for the presence of the immersed body by recomputing individual entries where needed. The simple structure of the Cartesian grid facilitated usage of an efficient algebraic multigrid solver, but special care had to be taken with the modifications performed on the discretized system of equations not to hinder its convergence properties.

The verification tests comprise steady-state heat conduction equation with a source term solved inside a cavity coupled with the solution of the heat conduction equation without source term inside the immersed body. The parabolic temperature profile obtained inside the cavity compares very

well to the analytical solution, and temperature profile inside the immersed body was as expected linear. Next simulations of natural convection in cavity with and without immersed cylinder for different Rayleigh numbers were presented. Results in terms of Nusselt number distribution, isothermal contours, and velocity streamlines were in excellent agreement with results published in [25, 37]. Finally, the new IBM approach has been also validated in the turbulent flow regime, using Direct Numerical Simulation (DNS) of the flow around a surface-mounted matrix of cubes at  $Re = 3854$ . The computed results for velocity and velocity fluctuations are in good agreement with experimental Laser Doppler Anemometry (LDA) measurements, showing that the proposed method can be used for large-scale simulations of turbulence.

The general benefits of the IBM approach over body-fitted grid methods are also shared by the method proposed in this work. The structure of the Cartesian grid allows the usage of efficient linear solvers (multigrid) and easier optimization of the computer code for various computational platforms. Furthermore, the implementation of state-of-the-art physical models is a straightforward task on a Cartesian grid. Finally, using an IBM rids the analyst from the burden of a body-fitted mesh generation procedure.

#### Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## References

- [1] P. Wesseling and C. W. Oosterlee, "Geometric multigrid with applications to computational fluid dynamics," *Journal of Computational and Applied Mathematics*, vol. 128, no. 1-2, pp. 311–334, 2001.
- [2] G. E. Karniadakis, "Simulating turbulence in complex geometries," vol. 24, no. 6, pp. 343–362.
- [3] M. O. Deville, P. F. Fischer, and E. H. Mund, *High-Order Methods for Incompressible Fluid Flow*, vol. 9 of *Cambridge Monographs on Applied and Computational Mathematics*, Cambridge University Press, Cambridge, 2002.
- [4] C. S. Peskin, "Numerical analysis of blood flow in the heart," *Journal of Computational Physics*, vol. 25, no. 3, pp. 220–252, 1977.
- [5] D. Goldstein, R. Handler, and L. Sirovich, "Modeling a No-Slip Flow Boundary with an External Force Field," *Journal of Computational Physics*, vol. 105, no. 2, pp. 354–366, 1993.
- [6] M.-C. Lai and C. S. Peskin, "An immersed boundary method with formal second-order accuracy and reduced numerical viscosity," *Journal of Computational Physics*, vol. 160, no. 2, pp. 705–719, 2000.
- [7] K. Khadra, P. Angot, S. Parneix, and J.-P. Caltagirone, "Fictitious domain approach for numerical modelling of Navier-Stokes equations," *International Journal for Numerical Methods in Fluids*, vol. 34, no. 8, pp. 651–684, 2000.
- [8] J. Mohd-Yusof, "Combined immersed boundary/B-spline methods for simulation of flow in complex geometries," in *Annual Research Briefs*, pp. 317–328, Center for Turbulence Research, 1997.
- [9] E. A. Fadlun, R. Verzicco, P. Orlandi, and J. Mohd-Yusof, "Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations," *Journal of Computational Physics*, vol. 161, no. 1, pp. 35–60, 2000.
- [10] R. Verzicco, J. Mohd-Yusof, P. Orlandi, and D. Haworth, "Large eddy simulation in complex geometric configurations using boundary body forces," *AIAA Journal*, vol. 38, no. 3, pp. 427–433, 2000.
- [11] A. Gilmanov, F. Sotiropoulos, and E. Balaras, "A general reconstruction algorithm for simulating flows with complex 3D immersed boundaries on Cartesian grids," *Journal of Computational Physics*, vol. 191, no. 2, pp. 660–669, 2003.
- [12] E. Balaras, "Modeling complex boundaries using an external force field on fixed cartesian grids in large-eddy simulations," *Computers & Fluids*, vol. 33, no. 3, pp. 375–404, 2004.
- [13] T. Ye, R. Mittal, H. S. Udaykumar, and W. Shyy, "An Accurate Cartesian Grid Method for Viscous Incompressible Flows with Complex Immersed Boundaries," *Journal of Computational Physics*, vol. 156, no. 2, pp. 209–240, 1999.
- [14] G. Yang, D. M. Causon, and D. M. Ingram, "Calculation of compressible flows about complex moving geometries using a three-dimensional Cartesian cut cell method," *International Journal for Numerical Methods in Fluids*, vol. 33, no. 8, pp. 1121–1151, 2000.
- [15] H. S. Udaykumar, R. Mittal, and P. Rampunggoon, "Interface tracking finite volume method for complex solid-fluid interactions on fixed meshes," *Communications in Numerical Methods in Engineering*, vol. 18, no. 2, pp. 89–97, 2002.
- [16] M. P. Kirkpatrick, S. W. Armfield, and J. H. Kent, "A representation of curved boundaries for the solution of the Navier-Stokes equations on a staggered three-dimensional Cartesian grid," *Journal of Computational Physics*, vol. 184, no. 1, pp. 1–36, 2003.
- [17] R. Mittal and G. Iaccarino, "Immersed boundary methods," *Annual Review of Fluid Mechanics*, vol. 37, pp. 239–261, 2005.
- [18] H. Bandringa, *Immersed boundary methods [M.S. thesis]*, University of Groningen, Groningen, The Netherlands, 2010.
- [19] J. Kim and H. Choi, "An immersed-boundary finite-volume method for simulation of heat transfer in complex geometries," *KSME International Journal*, vol. 18, no. 6, pp. 1026–1035, 2004.
- [20] J. R. Pacheco, A. Pacheco-Vega, T. Rodić, and R. E. Peck, "Numerical simulations of heat transfer and fluid flow problems using an immersed-boundary finite-volume method on non-staggered grids," *Numerical Heat Transfer, Part B: Fundamentals*, vol. 48, no. 1, pp. 1–24, 2005.
- [21] D. Pan, "An immersed boundary method on unstructured cartesian meshes for incompressible flows with heat transfer," *Numerical Heat Transfer, Part B: Fundamentals*, vol. 49, no. 3, pp. 277–297, 2006.
- [22] P. Orlandi and S. Leonardi, "DNS of turbulent channel flows with two- and three-dimensional roughness," *Journal of Turbulence*, vol. 7, pp. 1–22, 2006.
- [23] A. Gilmanov and S. Acharya, "A computational strategy for simulating heat transfer and flow past deformable objects," *International Journal of Heat and Mass Transfer*, vol. 51, no. 17-18, pp. 4415–4426, 2008.
- [24] N. Zhang, Z. C. Zheng, and S. Eckels, "Study of heat-transfer on the surface of a circular cylinder in flow using an immersed-boundary method," *International Journal of Heat and Fluid Flow*, vol. 29, no. 6, pp. 1558–1566, 2008.
- [25] B. S. Kim, D. S. Lee, M. Y. Ha, and H. S. Yoon, "A numerical study of natural convection in a square enclosure with a circular cylinder at different vertical locations," *International Journal of Heat and Mass Transfer*, vol. 51, no. 7-8, pp. 1888–1906, 2008.
- [26] Z. Wang, J. Fan, K. Luo, and K. Cen, "Immersed boundary method for the simulation of flows with heat transfer," *International Journal of Heat and Mass Transfer*, vol. 52, no. 19-20, pp. 4510–4518, 2009.
- [27] C. Shu, W. W. Ren, and W. M. Yang, "Novel immersed boundary methods for thermal flow problems," *International Journal of Numerical Methods for Heat & Fluid Flow*, vol. 23, no. 1, pp. 124–142, 2013.
- [28] G. Iaccarino and S. Moreau, "Natural and forced conjugate heat transfer in complex geometries on cartesian adapted grids," *Journal of Fluids Engineering, Transactions of the ASME*, vol. 128, no. 4, pp. 838–846, 2006.
- [29] S. Kang, G. Iaccarino, and F. Ham, "DNS of buoyancy-dominated turbulent flows on a bluff body using the immersed boundary method," *Journal of Computational Physics*, vol. 228, no. 9, pp. 3189–3208, 2009.
- [30] K. Nagendra, D. K. Tafti, and K. Viswanath, "A new approach for conjugate heat transfer problems using immersed boundary method for curvilinear grid based solvers," *Journal of Computational Physics*, vol. 267, pp. 225–246, 2014.
- [31] P. Orlandi, D. Sassun, and S. Leonardi, "DNS of conjugate heat transfer in presence of rough surfaces," *International Journal of Heat and Mass Transfer*, vol. 100, pp. 250–266, 2016.
- [32] Y. Sato and B. Ničeno, "A sharp-interface phase change model for a mass-conservative interface tracking method," *Journal of Computational Physics*, vol. 249, pp. 127–161, 2013.
- [33] A. J. Chorin, "Numerical solution of the Navier-Stokes equations," *Mathematics of Computation*, vol. 22, pp. 745–762, 1968.

- [34] D. L. Brown, R. Cortez, and M. L. Minion, "Accurate projection methods for the incompressible Navier-Stokes equations," *Journal of Computational Physics*, vol. 168, no. 2, pp. 464–499, 2001.
- [35] B. R. Hutchinson and G. D. Raithby, "A multigrid method based on the additive correction strategy," *Numerical Heat Transfer*, vol. 9, pp. 511–537, 1986.
- [36] P. Orlandi, Ed., *Fluid Flow Phenomena: A Numerical Toolkit*, vol. 55 of *Fluid Mechanics and its Applications*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2000.
- [37] G. De Vahl Davis and I. P. Jones, "Natural convection in a square cavity: A comparison exercise," *International Journal for Numerical Methods in Fluids*, vol. 3, no. 3, pp. 227–248, 1983.
- [38] E. R. Meinders and K. Hanjalić, "Vortex structure and heat transfer in turbulent flow over a wall-mounted matrix of cubes," *International Journal of Heat and Fluid Flow*, vol. 20, no. 3, pp. 255–267, 1999.
- [39] K. Hanjalić and S. Obi, Eds., *Proceedings 6th ERCOFTAC/IAHR/COST Workshop on Refined Flow Modelling*, Delft University of Technology, Delft, The Netherlands, June 1997.
- [40] A. Hellsten and P. Rautaeimo, Eds., *Proceedings 8th ERCOFTAC/IAHR/COST Workshop on Refined Flow Modelling*, Helsinki University of Technology, Helsinki, Finland, June 1999.



**Hindawi**

Submit your manuscripts at  
<https://www.hindawi.com>

