A sharp-interface Immersed Boundary Method to simulate convective and conjugate heat transfer through highly complex periodic porous structures

Saurish Das, A. Panda, N.G. Deen, J.A.M. Kuipers

Multiphase Reactors Group, Department of Chemical Engineering and Chemistry, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands

Multiphase and Reactive Flows Group, Department of Mechanical Engineering, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands

Highlights
- A versatile immersed boundary method for highly complex porous structures.
- Conjugate heat transfer and Neumann boundary condition for temperatures.
- Implementation of temperature and velocity periodic boundary conditions.
- Simulation of highly complex open-cell foam geometry.

Graphical Abstract

Abstract
Immersed boundary method (IBM) based CFD code has helped considerably in avoiding the tedious grid generation process in fluid flows involving complex geometries. In this work, we have developed an IBM framework to simulate flow, convective heat transfer as well as conjugate heat transfer through a highly complex random porous structure. In this framework, we can incorporate any complex solid body as a triangulated surface mesh and an accurate algorithm is proposed to identify solid cells and fluid cells. Moreover, a detailed implementation of periodic boundary condition for velocity and temperature is presented. Detailed code verification process is performed to demonstrate that the method is second-order accurate for both the velocity and temperature fields for all the boundary conditions considered. The developed scheme is shown to be applicable for convective and conjugate heat transfer through highly complex computer-generated realistic open-cell solid foams in a periodic Cartesian domain.

1. Introduction
Fluid flow through a porous medium is a subject of common interest for many separate emerging fields of studies, e.g. reaction engineering, groundwater engineering, biomedical application, reservoir engineering, only to name a few. The understanding of transport through a porous medium is a challenging task because, in most of the cases the geometry of the solid structure is random and not very clearly defined. The deformation of the solid structure due to fluid pressure, disparity in length scales, heat/species transport etc. bring the an additional level of complexity for direct numerical simulation (DNS) of the porous medium. Fully resolved simulations provide a fundamental understanding of fluid flow and transport phenomena through porous media and help in developing closures which are then used for macro-scale modeling (Whitaker, 1998) for industrial scale applications.
In this contribution, we are focusing on developing a versatile and accurate Immersed Boundary Method (IBM) based framework to perform DNS studies of a complex porous medium. The Immersed Boundary Method is a substitute to the body fitted mesh based methods for flow over complex geometries. This facilitates easy mesh generation process and avoids the use of unstructured grids. On top of this simple Cartesian grid-based codes have certain advantages over unstructured grid-based codes viz. low memory overhead, efficient data structures for book-keeping, vectorization and better convergence characteristics. Additionally, other transport equations can easily be coupled using an IBM framework. On the contrary, body-fitted grids provide ease of local mesh refinement in zones where sharp gradients occur which will result in computational overhead in case of structured meshes.

There exists several different kinds of IBM framework to apply no-slip boundary condition (for stationary bodies) or prescribe velocity (for moving bodies) at the solid surface. The momentum forcing method (first proposed by Peskin (1972)) is a widely used approach where the velocity condition at the solid surface is applied by introducing a source term in the momentum equation. This forcing function or the source term is applied to the vicinity of the solid surface and varies smoothly where it is maximum at the fluid-solid interface. Generally, in a time-step, a feedback loop or an iterative cycle is used to update the magnitude of the forcing function using the difference between the desired fluid velocity (at the solid surface) and the fluid velocity in the last iteration (Goldstein et al., 1993; Saiki and Biringen, 1996; Kriebitzsch, 2011; Tang et al., 2014). To avoid this feedback mechanism to calculate forcing function, in the ‘direct forcing’ approach the forcing term is calculated by the momentum balance at the ‘forcing points’ placed at the solid surface (Uhlmann, 2005; Kempe and Fröhlich, 2012; Kim and Choi, 2006; Taira and Colonius, 2007; Tenneti et al., 2011). Although most of the recently developed momentum forcing methods exhibit a second order spatial convergence, however, it demands a higher grid resolutions due to smear out the boundary effects. It is more appropriate for moving body problems. The ghost-cell immersed boundary method (Tseng and Ferziger, 2003; Deen et al., 2012; Ghias et al., 2004; Mittal et al., 2008; Pan, 2012; Majumdar et al., 2001) is more effective for complex geometries where the boundary conditions at the fluid-solid interface are applied during the discretization of the transport equations. Due to the sharp representation of fluid-solid interface, this method sometimes struggles with a spurious oscillation of flow variables (Tseng and Ferziger, 2003). Another approach, the cut-cell immersed boundary method (Chen and Botella, 2010; Udaykumar et al., 2001; Udaykumar et al., 2002; Ye et al., 1999) generates body conformal cells around the fluid-solid interface by the dividing the near surface Cartesian cells ‘cut’ by the fluid-solid interface. The momentum equations are solved for all the cells (Cartesian) inside the fluid, as well-as for all those ‘cut-cells’. However, in 3D, the calculation of convective and diffusive fluxes through the complex faces of those cut-cell control volumes are not trivial. As compared to other methods, this method offers very good local and global conservation of mass and momentum. Extension of the immersed boundary methodology to the solution of the energy equation is conceptually very similar when solid surface is at a constant temperature (Dirichlet condition). However, the treatment for constant heat flux (Neumann) condition at the solid surface, particularly in the ghost-cell method based framework is rare and not very trivial. To numerically predict transport characteristics through a porous medium, both the boundary conditions are required. For example, consider a porous structure which is used as catalyst support for non-isothermal reactions. A fast reaction coupled with high thermal conductivity of the solid will result in a constant surface temperature due to zero temperature gradient inside the solid. A slow reaction with low thermal conductivity of the solid would make constant heat flux boundary condition more realistic to use because of the substantial temperature gradients inside the solid. Authors like Kim and Choi (2006), Zhang et al. (2008), Pacheco et al. (2005) and Pan (2012) had proposed the treatment for both the temperature Dirichlet and Neumann boundary conditions, however with an IBM framework based on a forcing function. In heat transfer phenomena another frequently encountered situation is the need to study the transport of heat inside the solid as-well-as the fluid region (conjugate heat transfer (CHT)). This phenomenon is required more often to develop closures for effective thermal conductivity or thermal dispersion. Very few such boundary non-conforming techniques for conjugate heat transfer problems are available in the literature. A simple CHT model for solidification application was demonstrated by Marella et al. (2005) using a level-set field. Of late, Nagendra et al. (2014) developed a CHT model using ghost-cell IB technique in a curvilinear grid based solver.

In this paper, we present a numerical framework capable of solving highly complex porous structure in a domain comprising of structured grid. The geometrical details of the complex porous structure are described by a surface mesh with triangular elements. A Stereo-Lithography (STL) file format is used where each triangle is characterised by three vertices and an unit normal (pointing from solid to fluid). In the literature, there also exist few implementations (mainly forcing function based) that are capable of considering irregular geometries. However, those methods might fail for geometries with high complexities and very sharp edges. In the present ghost-cell method special attention has been given to such cases. The ongoing research of X-ray micro-computed tomography (μ-CT) imaging has led to an increased demand for modeling of complex irregular structures. Our in-house code is capable of incorporating such complex geometrical details in the form of a triangulated surface mesh. An accurate algorithm is proposed to identify solid-cells and fluid-cells. Often, pore-scale simulations are performed in a periodic domain, considering a single representative elementary volume (REV). A detailed implementation of periodic boundary condition for velocity as-well-as temperature field has been presented in this work. Detailed code verification process is performed to demonstrate that the method is second-order accurate for both velocity as-well-as temperature field for all the boundary conditions (Dirichlet, Neumann and CHT). The developed scheme is shown to be applicable for convective and conjugate heat transfer through a highly complex computer-generated realistic open-cell solid foams (Das et al., 2018) in a periodic Cartesian domain.

2. Numerical methodology

The mass and momentum conservation equations are solved in a 3D staggered Cartesian computational domain. The following governing equations are solved for velocity ($\mathbf{u}$) and pressure ($p$):

$$\nabla \cdot \mathbf{u} = 0$$

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho \nabla \cdot (\mathbf{u} \mathbf{u}) = -\nabla p + \nabla \cdot \left[ \mu \left( \nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) \right]$$

where $\rho$ is the density and $\mu$ is the viscosity of the fluid. In the current finite volume implementation the transport equations are integrated for each control volume (CV) where the convective flux is calculated using a total variation diminishing (TVD) Min-mod scheme (treated explicitly) and the Central Difference scheme (treated implicitly) is used for the discretization of the diffusion terms which is second order accurate. A fractional-step method is employed for incorporating the pressure-velocity coupling, in which, first a tentative velocity field $\mathbf{u}^*$ is computed from the dirre-
tised momentum equation (Eq. (2)) using the old time level pressure \((p^n)\). Then the pressure Poisson equation 
\[
\left(\nabla \cdot \left( \frac{\rho_f^2}{\mu} \nabla \Delta p \right) \right) = \nabla \cdot \mathbf{v}
\]
is solved to calculate the new pressure field \(p^{n+1} = p^n + \Delta \phi\). The divergence free velocity for the new time-step is computed as, \(\mathbf{u}^{n+1} = \mathbf{u}^n - \frac{\partial p}{\partial x} \nabla \Delta p\). More details of the present finite volume implementation can be found in our previous work (Das et al., 2016, 2017a,b).

Once the velocity field for the current time level \((n+1)\) is obtained, we solve for the temperature field for the fluid phase as-wells-as the solid phase (only for case of CHT). The following governing equations are solved for fluid temperature \((T_f)\) and solid temperature \((T_s)\) respectively:

**Fluid** :
\[
\rho_f C_{pf} \frac{\partial T_f}{\partial t} + \rho_f C_{pf} \nabla \cdot \mathbf{u}_f T_f = k_f \nabla^2 T_f
\]

**Solid** :
\[
\rho_s C_{ps} \frac{\partial T_s}{\partial t} = k_s \nabla^2 T_s
\]

where \(\rho_f, C_{pf}\) and \(k_f\) are the density, specific heat capacity and thermal conductivity of the fluid, and \(\rho_s, C_{ps}\) and \(k_s\) are the density, specific heat capacity and thermal conductivity of the solid, respectively. Similar to the momentum equation, the convective thermal flux is calculated using the Central Differencing scheme whereas the conductive flux is calculated using the Central Differencing scheme.

The discretization of the above equations leads to a set of algebraic equations of the following generic form,

\[
A_C \phi_C + \sum_{nb} A_{nb} \phi_{nb} = B_C
\]

where \(\phi\) corresponds to one of the fluid velocity components \((u, v, \text{or } w)\) or temperature \((T)\) or pressure correction \((\Delta p)\). The subscript ‘C’ indicates the cell-center and ‘nb’ represents the six neighbouring cells. We use the notation \(Xp, Xn, Yp, Yn, Zp\) and \(Zn\) for the six neighbouring cells, e.g. \(Xp\) is cell in the positive \((x^+)\) x-direction from the cell-center, \(Xn\) is for the negative \((x^-)\) x-direction etc. The coefficient \(A_C\) depends on the fluid property, grid size \((\Delta x)\) and time-step size \((\Delta t)\), whereas all the explicit terms are contained in \(B_C\).

The discretized Eq. (5) is derived for each computational cells in the domain which forms a system of linear equations \((A \cdot \phi = B, \text{where } A, B \text{ and } \phi \text{ represent coefficient, solution and resultant matrix, respectively})\) which is then solved for \(\phi\) using a parallel solver based on Block-Incomplete Cholesky Conjugate Gradient (B-ICCG) method.

### 3. Immersed boundary method

#### 3.1. General description and Dirichlet boundary condition

A 2D implementation of the current IBM framework is depicted in Fig. 1. At first cells are flagged depending upon the location of the cell centre viz. “solid cell” (cell-center inside the solid body) or “fluid cell” (cell-center in the fluid zone). “IB-fluid cells” are fluid cells which have at least one solid cell as a neighbour and that solid cell is then termed as “IB-solid cell”. Sometimes, in the literature the IB-solid cells are called ‘ghost-cells’. The numerical procedure for flagging solid cell and fluid cells for a complex immersed body is described in Section 3.4. In Fig. 1 IB-fluid cell C neighbours two IB-solid cells, \(Xn\) and \(Yn\). The normalized distance between the IB-solid cell \(Xn\) and the immersed boundary along the \(x\)-direction is calculated as \(\xi_n = XnST/\Delta x\). Similarly, the distance between the IB-solid cell \(Yn\) and the immersed boundary along the \(y\)-direction is calculated as \(\xi_y = YnST/\Delta y\). The direction of \(\xi_x\) or \(\xi_y\)-axis will always be from the solid cells pointing towards the fluid cells parallel to the global \(x\)-or \(y\)-axis. To enforce a Dirichlet condition \((\phi = \phi_D)\) at the fluid-solid interface, the value of the IB-solid cells are ‘temporarily’ represented in terms of relevant neighbouring cells in a way that it satisfies \(\phi = \phi_D\) at the solid surface. For velocity components \(\phi_v = 0\), represents the no-slip condition whereas for temperature \(\phi_T = T_s\) represents an iso-thermal solid temperature boundary condition. We express the values of the IB-solid cells as,

\[
\phi_{Xn} = \xi_x \phi_C + \beta_x \phi_{Xp} + \gamma_x
\]

\[
\phi_{Yn} = \xi_y \phi_C + \beta_y \phi_{Yp} + \gamma_y
\]

The variables \(\xi_x, \beta_x \text{ and } \gamma_x\) \((\xi_y, \beta_y \text{ and } \gamma_y)\) can be found using an quadratic fit between \(S1 = C - Xp, S2 = C - Yp\). To avoid singularity (Das et al., 2017a) we use a linear fit between \(C - Xp\) or \(C - Yp\) respectively.

---

**Fig. 1.** Incorporation of the boundary condition at the solid surface. The solid surface is represented by a surface mesh of triangular elements (line segments in 2D). For the IB-fluid cell C, the distance (normalized by the grid spacing \(\Delta x\)) from the neighbouring cells (i.e. \(Xn\) and \(Yn\)) to the solid surface in the direction of the grid lines (i.e. \(\xi_x - \xi_n\) and \(\xi_y - \xi_m\)) are required to implement the boundary conditions exactly at the solid surface. The \(\xi_x\) values are stored in a hash table format. Hash codes are stored in the flag matrix for IB-fluid cells and each of these point to a table where six \(\xi_x\) values are stored. This procedure needs to be carried out for the three velocity components \((u, v \text{ and } w)\) and temperature \((T)\) in the staggered computational grid.
Yp when S1 or S2 are very close to C (ξ_{s0} → 0 or η_{s0} → 0). Though for such cells the local accuracy of the method decreases, the number of such cells is very small, and as a result the global accuracy still remains second order in space. Following Deen et al. (2012), we express the values of α, β and γ in the following generalized compact form,

\[
\begin{align*}
\alpha_s &= -\frac{2\xi_s}{1-\xi_s^2}, \\ \beta_s &= \frac{\xi_s}{1-\xi_s^2}, \\ \gamma_s &= \frac{2}{1-\xi_s^2} \phi_s & \text{when } 0 \leq \xi_s < 0.999 \\
\alpha_s &= 0, \quad \beta_s = -\frac{1}{1-\xi_s^2}, \quad \gamma_s = \frac{2}{1-\xi_s^2} \phi_s & \text{when } 0.999 \leq \xi_s \leq 1
\end{align*}
\]  

(8)

where ξ_s = ξ_{s0} for Eq. (6) and η_s = η_{s0} for Eq. (7). For the presence of IB-solid cell in the negative x-direction, using Eq. (6), we replace η_{s0} in Eq. (5), and the updated sets of coefficients for IB-fluid cell C are obtained as,

\[
\begin{align*}
\tilde{A}_c &= A_c + \alpha_{s0} A_{x0} \\
\tilde{A}_{xp} &= A_{xp} + \beta_{s0} A_{x0} \\
\tilde{B}_c &= B_c - \gamma_{s0} A_{x0} \\
\tilde{A}_{x0} &= 0
\end{align*}
\]  

(9)

In a very similar way, φ_{s0} is replaced in Eq. (5) using Eq. (7) and newer sets of coefficients are again modified as,

\[
\begin{align*}
\tilde{A}_c &= \tilde{A}_c + \alpha_{s0} \tilde{A}_{y0} \\
\tilde{A}_{yp} &= \tilde{A}_{yp} + \beta_{s0} \tilde{A}_{y0} \\
\tilde{B}_c &= \tilde{B}_c - \gamma_{s0} \tilde{A}_{y0} \\
\tilde{A}_{y0} &= 0
\end{align*}
\]  

(10)

The final coefficients (\tilde{A}_c) do not depend on the sequence in which the coefficients are modified by using Eqs. (9) and (10). Note that Eq. (5) is also formed for solid-cells and to enforce φ = φ_s, we set the central coefficient, A_c = 1, the neighbouring coefficients A_{x0} = 0 and B_c = φ_s. The procedure mentioned above is carried out for all the three velocity components and temperature. It is to be noted that since the velocity field inside the solid body is already divergence free, no strict treatment to the pressure field is required (like \tilde{p} = 0 at the solid wall). While computing the pressure field inside the solid body, the pressure correction step causes a very small value of the velocity (O(u · 10^{-3})) inside the solid, which has insignificant effect on the overall simulation results.

It is evident from the above discussion that any complex surface can be accommodated in a Cartesian grid by cell flags and the ξ_s values at the IB-fluid cells. The values of ξ_s can be obtained from the intersection between the triangular element and the grid line (5). We present a hash table data structure (Fig. 1), which stores a unique mapping of the cell key (obtained by numbering all the IB-fluid cells) and the associated ξ_s values for all the six directions (for 3D). If there is only one neighbouring solid-cell then only one value of ξ_s is needed. The STL mesh is no longer needed once the hash table has been generated.

It is worth to point out that unlike other ghost-cell methods (Tseng and Ferziger, 2003), we do not implicitly define a unique value to the IB-solid cell. For instance, in Fig. 1, while solving for IB-fluid cell ‘6’ the φ value assigned at IB-solid cell ‘a’ will be different for the assigned φ value at IB-solid cell ‘a’ while solving for IB-fluid cell ‘7’. It allows us to achieve very strong fluid-solid coupling even for a coarser grid.

3.2. Neumann boundary condition

In the case of Neumann boundary condition, the gradient of φ normal to the solid surface is prescribed. For a constant heat flux at the solid surface, the expression becomes

\[
-k_f \frac{\partial T_s}{\partial n} = \bar{q}_s
\]  

(11)

where \bar{q}_s is the prescribed uniform surface heat flux, k_f is the fluid conductivity and \bar{n} is the surface normal pointing outward from the solid surface. At first, the temperature at the fluid-solid interface is calculated using the condition described in Eq. (11). It is then used as a Dirichlet boundary condition as elaborated above. It is to be noted that the temperature field calculated at the previous time-step is used to compute the “fixed” wall temperature for the new-time level and thus the wall temperature lags by one time step although the overall stability of the framework is ensured by the implicit implementation of the Dirichlet boundary condition. For the Neumann boundary condition, the fluid-solid interface temperature is also stored in the hash table in addition to the ξ_s values. In Fig. 2 the detailed implementation is presented. For a second order accurate interpolation to calculate the surface temperature, two probes (d_1 and d_2) are created towards the fluid-cells normal to IB. The probes are placed at a distance Δn and 2Δn from the fluid-solid interface:

\[
d_1 = \bar{d}_1 + \Delta n \cdot \bar{n}; \quad d_2 = \bar{d}_1 + 2\Delta n \cdot \bar{n}
\]  

(12)

where \bar{d}_1 represents the location where the grid line intersects the fluid-solid interface. It is a natural choice to use Δn = Δx, where Δx is the Cartesian grid spacing. A Taylor series expansion of Eq. (11) is used to compute the surface temperature from the interpolated temperature field at the probe locations given by:

![Fig. 2. Implementation of Neumann boundary condition.](image-url)
\[ T_S = \frac{1}{3}(4T_1 - T_2 - 2G\Delta n) \]  
where \( T_1 \) and \( T_2 \) are the temperature at the probes \( d_1 \) and \( d_2 \). For constant heat flux case, \( G = -\bar{q_1}/k_f \).

For special cases a first order formulation is used to obtain the surface temperature which occurs when the second probe lies outside the bounds of the domain or inside the solid. For this, once can write the surface temperature from Eq. (11) as

\[ T_S = T_1 - G\Delta n \]

The temperature values at the probe locations are computed using trilinear interpolation by way of volumetric averaging of the temperature from the neighbouring cells (eight in 3D or four in 2D). In Fig. 2, to calculate the temperature at \( d_1 \), the temperature of the cells 1, 2, 3 and 4 are required. If one of the surrounding cell-centers lie inside the solid (for the Neumann condition, the temperature inside the solid has no meaning), we increase the distance \( \Delta n \) in such a way that all the cell-center involved in trilinear interpolation remain inside the fluid.

3.3. Conjugate heat transfer between solid and fluid

In the specific case of conjugate heat transfer, the energy equations inside the solid is also solved along with that of the fluid region. Grid cells whose cell-center lie inside the solid region are assigned the material properties of the solid. Additionally, the continuity of the temperature and the normal heat flux at the fluid-solid interface needs to be enforced i.e.

\[ T_f|_s = T_s|_s \quad \text{and} \quad -k_f \frac{\partial T_f}{\partial n}|_s = -k_s \frac{\partial T_s}{\partial n}|_s \]  

Sub-scripts ‘s’ and ‘f’ represent solid and fluid respectively. The implementation of CHT closely follows that of Neumann boundary condition where first, the fluid-solid interface temperature satisfying Eq. (15) is obtained which, then is used as the Dirichlet boundary condition. This fluid-solid interface temperature is calculated with the help of four probes (two each in the fluid and the solid region) satisfying Eq. (15) which is then stored in the hash table. It is schematically shown in the Fig. 3a for a 2-D case where

\[
\begin{align*}
\bar{d}_1 &= d_1 + \Delta n \cdot \bar{n}; & \bar{d}_2 &= d_2 + 2\Delta n \cdot \bar{n}; \\
\bar{d}_3 &= d_3 - \Delta n \cdot \bar{n}; & \bar{d}_4 &= d_4 - 2\Delta n \cdot \bar{n}
\end{align*}
\]

where \( \bar{d} \) represents the location of the fluid-solid interface. A Taylor series expansion of Eq. (15) is used to compute the surface temperature given by:

\[ T_S = \frac{k_f(4T_1 - T_2) + k_s(4T_3 - T_4)}{3(k_f + k_s)} \]

where \( T_1 \) and \( T_2 \) are the interpolated temperature values at the probe locations \( d_1 \) and \( d_2 \) (both on the fluid side), whereas \( T_3 \) and \( T_4 \) are those at the probe locations \( d_3 \) and \( d_4 \) (both on the solid side). In Fig. 3, \( F_1 \) and \( F_2 \) are the two fluid cells and \( S_1 \) and \( S_2 \) are the two solid cells.

In the case of \( F_1 \), the direct forcing method is applied by modifying the coefficients obtained from the discretizing energy equation viz. Eqs. (9) and (10). To modify the coefficients, surface temperature \( T_S \) and the non-dimensional distance \( \xi \) are required. These values are obtained from the hash table corresponding to the unique key attributed to the cell \( F_1 \). Similarly, in the case of \( S_1 \) the coefficients need to be modified using the surface temperature \( T_S \) and the non-dimensional distance \( 1 - \xi \). This leads to decoupling of temperature fields in the fluid and the solid cells which then can be solved with any wall boundary conditions. For complex solid geometries, certain specific cases may arise such as the second fluid/solid probe falls in the solid/fluid region or outside the domain boundaries. In these special cases, we resort to the use of linear interpolation to compute the solid temperature given by:

\[ T_S = \frac{k_f T_1 + k_s T_3}{k_f + k_s} \]

Sensitivity of the solution to the choice of the probe distance from fluid-solid interface (i.e. \( \Delta n \)) has also been checked by choosing \( \Delta n = \Delta x \) and \( \Delta n = 2\Delta x \). We observe negligible difference in the temperature field and overall heat transfer coefficient (\( \sim \)2%). However, for higher \( Re \) cases, this may not be valid because the second probe may fall outside the very thin boundary layer thus requiring proper calibration or grid refinement.

Fig. 3. Methodology for conjugate heat transfer (CHT) between solid and fluid.
3.4. An efficient method for identifying fluid cells and solid cells

For geometries that can be expressed by an analytical form flagging a cell center as fluid cell or solid cell is trivial. However, for complex geometries represented by a number of surface elements it is a classical 'point-in-polyhedron' problem in computational geometry. Several methods which are very efficient and flawless for small geometries have already been developed in the past. Moreover, very sharp edges of the foam sample require special attention during the identification of the cells. For this purpose we developed an accurate and efficient method where we loop over all the triangular elements and mark only the Eulerian grids in the vicinity of an individual triangular element (Fig. 4). The steps enlisted below allow for flagging of the cells in the Cartesian grid:

**Step 1:** First, a bounding box around a triangular element is created after locating the centroid of the triangular element in the Eulerian grid. Each cell in the bounding box is then checked and flagged as either a solid or a fluid cell. In Fig. 4a, this procedure is schematically shown in 2D for the triangular element \( t_s \), where inside the bounding box all the cells \((C_1 \text{ to } C_6)\) are checked. For each cell in the bounding box, grid lines connecting the cell center with each of the six neighbouring cell center is checked for intersection with the triangular element by computing the following dot product:

\[
v = (x_c - x_{nb}) \cdot \hat{n}
\]

where \( x_c \) and \( x_{nb} \) are the position vector of the cell in the bounding box and its neighbouring cell, respectively, whereas \( \hat{n} \) is the surface normal of the triangular element. If \( v > 0 \), the cell center \( x_c \) is marked as a fluid-cell. In Fig. 4a, if we examine cell center \( C_3 \), grid lines joining its two neighbouring cell centers \((C_1 \text{ and } C_2)\) intersect the triangular element \( t_s \). The vector quantity \((x_c - x_{nb})\) is represented by \( p \) and for both the cells \((C_2 \text{ and } C_3)\): \( p \cdot \hat{n} < 0 \), hence the cell \( C_3 \) is assigned as fluid cell. Similarly, cell \( C_7 \) and \( C_8 \) are assigned fluid cells. In a three dimensional array (e.g. \( f\{1,3,k\} \)) where \( i, j \) and \( k \) are the running indexes in \( x, y \) and \( z \) direction) the computational cells are mapped and flagged. This results in a hollow shell of IB-fluidic flags around the immersed boundary (Fig. 4b1). The size of the bounding box is based on the relative size of triangular element and Eulerian grid size facilitating proper enclosing. In our case where the triangular element size and the grid size are similar, a stencil size of \( 3 \times 3 \times 3 \) is sufficient. We define \( f\{1,3,k\} = -1 \) for solid cells, \( f\{1,3,k\} = -1 \) for solid cells and \( f\{1,3,k\} = 0 \) when cell are not marked.

**Step 2:** Again we loop over all the triangular elements and flag the solid cells in the vicinity of the immersed boundary in a similar way (Fig. 4b2). We flag the fluid cells and solid cells in two different loops of the triangular elements to avoid issues related to sharp edges, which will be discussed in Appendix A.

**Step 3:** Until now we have flagged all the solid cells \( f\{1,3,k\} = -1 \) and fluid cells \( f\{1,3,k\} = 1 \) enclosing the solid body. Next we need to assign solid flags inside the solid body. This is done by sequential tracing in all three directions in a 3D system. We move along the positive \( x \)- direction (i.e. with increasing index \( j \)) and we check the condition \( f\{1,3,k\} = 0 \) and \( f\{1,3,k\} = -1 \), and when this is satisfied we update \( f\{1,3,k\} = -1 \). The procedure will flag all cells inside the solid and it will automatically stops when it reaches a solid cell on the other end of the body (Fig. 4b3).

The above procedure works perfectly for periodic solid bodies. If the triangular surface mesh and/or the initial marking of the solid and fluid cells are not closed this method will fail.

3.5. Periodic boundary treatment

The boundary condition for both the momentum and energy at the outer boundaries of the domain is applied in an implicit manner, at the level of discretization, using fictitious cells outside the domain. First we will discuss, very briefly, the generalization of the boundary condition and then the periodic boundary implementation. Due to the staggering of the computational grids, the velocity normal to the boundary faces defined exactly on the boundary and hence no ghost cells are required to implement the boundary condition. However, for tangential velocity components, temperature and pressure the implicit boundary condition implementation is shown in Fig. 5a for a generalized quantity \( \phi \). The value of the ghost-cell \( \phi_{\text{ext}} \), \( \phi_C \) and \( \phi_{\text{wp}} \) as,

\[
\phi_{\text{ext}} = \alpha \phi_C + \beta \phi_{\text{wp}} + \gamma
\]

Very similar to the present immersed boundary method, the values of \( \alpha, \beta \) and \( \gamma \) can be obtained by making a quadratic fit between \( \phi_{\text{ext}}, \phi_C \) and \( \phi_{\text{wp}} \), and are listed below for different boundary types:

1. Inlet boundary: \( \alpha = 0, \beta = 0 \) and \( \gamma = \phi_{\text{inlet}} \), where \( \phi_{\text{inlet}} \) is the value at the inlet.
2. Boundary with a specified value \( (\phi = \phi_B) \), \( \alpha = -2, \beta = \frac{1}{2} \) and \( \gamma = \frac{3}{2} \phi_B \).
3. Boundary with a specified flux \( (\frac{d\phi}{dx} = q) \), \( \alpha = 1, \beta = 0 \) and \( \gamma = -q \Delta x \), where \( \Delta x \) is the grid spacing.

Fig. 4. Methodology for identifying fluid cells and solid cells.
In the discrete form of the momentum, pressure Poisson and energy equations (Eq. (5)), the values of the ghost-cells are replaced and the new set of coefficients are formed using Eq. (9).

The periodic boundary implementation for velocity and pressure is based on Patankar et al. (1977). The advantages of using periodic conditions (fully-developed flow) are that there are no entrance length effects and simulations can be performed in a reasonably small domain. Across the periodic domain the velocity field repeats itself and the pressure gradient is constant. At any point \((x, y, z)\) in the periodic domain with size \(L\), the velocity and pressure satisfies, \(u(x, y, z) = u(x + L, y, z)\) and \(p(x, y, z) - p(x + L, y, z) = \beta L\), where \(\beta\) is the pressure gradient in the streamwise flow direction (not to confuse with the coefficients). Using \(\beta\), we can define the pressure at any point \((x, y, z)\) as, \(p(x, y, z) = -\beta x + P(x, y, z)\) where \(P(x, y, z)\) is the periodic pressure field and relates to the details of local flow features. The great advantage of the above simple formulation is that now the pressure field is also self-similar at periodic faces, i.e. \(P(x, y, z) = P(x + L, y, z)\). The pressure gradient \(\beta\) appears as a source term in the momentum equation and the pressure-velocity coupling appears in terms of the periodic pressure field \(P\). The quantity \(\beta\) needs to be assigned a priori in order to control the mass flow rate and the Reynolds number \(Re\) of the simulation.

Similarly, the temperature field can be solved in a periodic domain. In the direction of the flow, the average temperature changes due to the heat transfer between the fluid and the solid body. For moderate to high Peclet number \((Pe)\) flows (negligible axial conduction), if the solid walls (domain boundaries or immersed boundary) are subjected to a constant heat flux, then the temperature field is piece wise linear: \(T(x, y, z) = T(x + L, y, z) + \left[T_{ul0} - T_{ul0}\right]\) where \(T_{ul0}\) and \(T_{ul0}\) represent the cup-average bulk temperature at the upstream and downstream of the periodic domain (refer Fig. 5b). However, at very low \(Pe\), due to axial conduction such condition is not valid and the boundary treatment is discussed later on. At moderate to high \(Pe\), if the walls are at constant temperature, \(T_{s}\), the non-dimensional temperature field \(\Theta\) repeats its exact morphology in the streamwise flow direction, i.e. \(\Theta(x, y, z) = \Theta(x + L, y, z)\), where \(\Theta = (T(x, y, z) - T_{s})/(T_{ul0} - T_{s})\).

Following Beale et al. (2005), both conditions can be combined as,
\[
T(x, y, z) = PT(x + L, y, z) + Q
\]  

(21)

Constant wall temperature :
\[
\begin{align*}
P &= \frac{\frac{T_{b| ul0} - T_{s}}{\frac{T_{ul0} - T_{s}}{x}}} \\
Q &= T \left(1 - \frac{Q}{T_{s}}\right)
\end{align*}
\]  

(22)

Constant wall heat-flux :
\[
\begin{align*}
P &= 1 \\
Q &= T_{b| ul0} - T_{b| ul1}
\end{align*}
\]  

(23)

These conditions are incorporated directly into the discretization of the temperature at the grid points that are adjacent to the periodic boundaries. Fig. 5b schematically shows a single layer of computational cells where flow and temperature field are periodic along the \(x\)-direction. To incorporate the temperature jump condition (Eq. (21)) across the periodic faces, similar to IBM, the coefficients of the discretised energy equation are modified for boundary adjacent cells. The coefficients of the \(P\) and \(Q\) require both the upstream and downstream temperature control volumes (i.e. \(T_{1}\) and \(T_{\infty}\)) are modified Using Eq. (22) to Eq. (23) as,
\[
\begin{align*}
\tilde{A}_{1\infty, xn} &= PA_1\tilde{n}_{1\infty, xn} \\
\tilde{B}_1 &= B_1 - QA_1\tilde{n}_{1\infty, xn} \\
\tilde{A}_{\infty, xn} &= A_{\infty, xn} + \frac{Q}{P}A_1\tilde{n}_{1\infty, xn} \\
\tilde{B}_\infty &= B_{\infty} + \frac{Q}{P}A_1\tilde{n}_{1\infty, xn}
\end{align*}
\]  

(24)

The constants \(P\) and \(Q\) require both the upstream and downstream bulk temperatures. For a particular simulation we fixed the upstream bulk temperature \((T_{ul0})\) to a reference value \((T_{ref})\), using the latest available temperature field \((T^n)\) to calculate \(P\) and \(Q\) to apply boundary condition (Eq. (24)) for the next time level \((T^{n+1})\). Once the simulation reaches a steady state (or pseudo steady state for high \(Re\) flows) the values of \(P\) and \(Q\) attains a nearly constant value.

Ideally, for periodic pressure and velocity fields we do not need any ghost-cells, where our in-house periodic solver (B-ICCG)
performs the necessary data exchange across the periodic face during the iterative solution of $\mathbf{A} \cdot \mathbf{\phi} = \mathbf{B}$. However, for convective flux calculations for the next time-step, we use the temperature field of the previous time-step, which requires the temperature at ghost cells (refer Fig. 5b). The TVD scheme for convective flux calculations requires two computational cells in the upstream direction. As a result for the current implementation, we create two layers of ghost cells across the periodic faces, schematically shown in Fig. 5b. After solving the temperature field inside the domain, the temperature at the ghost cells are updated as,

$$
\begin{align*}
T_{-1} &= PT_{mx-1} + Q \\
T_0 &= PT_{mx} + Q \\
T_{mx+1} &= \frac{1}{3} T_1 - \frac{2}{3} Q \\
T_{mx+2} &= \frac{1}{3} T_2 - \frac{2}{3} Q
\end{align*}
$$

(25)

We have performed all the heat transfer simulations at Prandtl number $(Pr = \mu C_p/k)$ equal to unity, which assures equal boundary layer thickness (and as a result same grid requirements) for velocity and temperature. Since solid thermal conductivity is high (i.e. very low Biot number) for metallic foams, the temperature gradient inside the solid is neglected. At moderate to high Re flows, more precisely at high Peclet number $(Pe = Re \times Pr)$, the axial conduction (fluid side) is negligible and one can observe a self-similar, repeating non-dimensional temperature $(\theta)$ field in the stream-wise flow direction. However, at low Peclet number, due to finite axial conduction we cannot use the scaled-periodic temperature boundary condition discussed earlier. For such cases a Danckwerts’ boundary condition (Danckwerts, 1953) is applied across the stream-wise flow direction where the velocity field is periodic. It assumes that at low Pe, for thermally fully developed flows,

$$
\begin{align*}
\rho C_p u T_{\text{ref}} &= \rho C_p u_n T - k \frac{dT}{dn} \\
\frac{dT}{dn} &= 0
\end{align*}
$$

(26)

at the inlet boundary

at the outlet boundary

where the reference temperature, $T_{\text{ref}}$ can be considered as a developing far stream temperature field. It has no influence on the non-dimensional quantities and in the present case we have used $T_{\text{ref}} = 0$.

3.6. Overview of the numerical method

We summarize here the step-by-step solution procedure. At the beginning of the solution, the triangulated surface mesh for the solid body is imported in the Cartesian computational domain. The cells are flagged as either fluid cell or solid cells using the algorithm described in Section 3.4. Based on the position of the triangulated mesh, the hash table (Fig. 1) is generated to store $\xi_5$ values. While solving for velocity, first a tentative velocity field $\mathbf{u}'$ is computed from the discritised momentum equation for three velocity components ($\mathbf{A} \cdot \mathbf{u}' = \mathbf{B}$, $\mathbf{A} \cdot \mathbf{u}' = \mathbf{B}$ and $\mathbf{A} \cdot \mathbf{w}' = \mathbf{B}$) using old time level pressure $(p^*)$. The coefficient matrix $\mathbf{A}$ is modified by using Eq. (20) to incorporate the domain boundary and by Eqs. (9) and (10) to incorporate the immersed boundaries where $\xi_5$ values from the hash table are used. After solving the discritised momentum equation using B-ICCG solver, the pressure Poisson equation is solved to compute the pressure correction $(\Delta p)$ at each cell using the B-ICCG solver ($\mathbf{A} \cdot \mathbf{\phi} = \mathbf{B}$). After computing a divergence free velocity field for the new time level using the pressure correction $(\Delta p)$, the ghost-cell values are updated.

Similarly the temperature field is computed by solving the associated discritised energy equation ($\mathbf{A} \cdot \mathbf{T} = \mathbf{B}$), where to incorporate boundary conditions the coefficient matrix $\mathbf{A}$ is modified (Eqs. (20) and (24)). After solving for the new time-level temperature field, temperatures at the ghost-cells are modified (Eq. (25) for periodic flow). For both the Neumann boundary condition and the CHT case, the fluid-solid interface temperature at the wall-grid line intersection point are also stored as well as updated at each time-step in the hash table.

4. Force and heat flux calculation

The use of a triangulated surface mesh allows us to calculate the local fluid-solid interaction force and heat flux. Often these entities are required to analyze the results and compute the motion of the solid moving body. The total surface force $(dF)$ at a single triangular element (Fig. 6a) has two parts, a pressure force $(dF_p)$ and a viscous force $(dF_v)$,

$$
dF = dF_p + dF_v = (-p \mathbf{n} dA) + (\bar{\mathbf{\tau}} \cdot \mathbf{n} dA)
$$

(27)

where $\bar{\mathbf{\tau}} = \mu \bigg( \nabla \mathbf{u} + \big( \nabla \mathbf{u} \big)^T \bigg)$ represents the stress tensor, whereas $\mathbf{n} = [n_x \, n_y \, n_z]$ and $dA$ are the surface unit normal and surface area, respectively. These forces are integrated over the surface to calculate the total forces,

$$
\mathbf{F} = -\int_{S} \rho \mathbf{n} dA + \int_{S} \bar{\mathbf{\tau}} \cdot \mathbf{n} dA
$$

(28)

The expression for the total rate of heat transfer from the solid to the fluid is given as:

$$
Q = -\int_{S} k(\nabla T \cdot \mathbf{n}) dA
$$

(29)

By looping over all the triangular elements of the surface mesh, we can numerically integrate the total force in Eq. (28). The term $\bar{\mathbf{\tau}} \cdot \mathbf{n}$ can be calculated at the center of gravity (CG) of each triangular element, which requires evaluation of the velocity derivatives in all the three directions, for which one can create probes in the respective direction ($x$-, $y$- and $z$-direction) and find the velocity derivatives at the surface using the velocity components at those probes. The velocity at a probe in a continuous velocity field can be obtained using trilinear interpolation.

For the local pressure force at the CG of a triangular element can be extrapolated using pressure at two probes perpendicular to the surface (i.e. $l_1$ and $l_2$ in Fig. 6a) while satisfying $\frac{d}{dn} = 0$. Using the pressure values, $p_1$ and $p_2$ at the location $l_1$ and $l_2$, pressure at the CG of triangular elements is calculated as,

$$
p_{l} = \frac{1}{3} (4p_1 - p_2)
$$

(30)

Similarly, to calculate the local skin-friction coefficient (in 2D), the local shear stress $(\tau_s)$ parallel to the solid surface can be calculated using two probes perpendicular to the triangular element as,

$$
\tau_s = \frac{\mu}{\Delta T} \frac{\partial u_1}{\partial n} = \frac{\mu}{\Delta T} \bigg( 2u_{T,1} - \frac{1}{2} u_{T,2} \bigg)
$$

(31)

where $u_1$ represents velocity component parallel to the solid surface and $\Delta n$ is the distance between the CG and probes shown in Fig. 6a.

For temperature Neumann condition, while calculating local $Nu$, the temperature at the IB is calculated using Eq. (12). Whereas, for constant surface temperature case, the local heat flux is calculated as,

$$
q_s = -k \frac{\partial T_s}{\partial n} = -\frac{k}{\Delta n} \bigg( \frac{3}{2} T_s + 2T_1 - \frac{1}{2} T_2 \bigg)
$$

(32)

The pressure, velocity and temperature at $l_1$ and $l_2$ are calculated using trilinear interpolation from the continuous pressure, velocity and temperature field. We have used $\Delta n = \Delta x$. However, for few cells we increase the probe distance so that all the cells involved in trilinear interpolation lie in the fluid zone.
For a closed IB surface, using Gauss theorem, the expression in Eq. (28) can be simplified and given as,

\[ \mathbf{T} = - \iint_S \rho \mathbf{v} \mathbf{n} dA + \mu \iint_S \nabla \mathbf{v} \cdot \mathbf{n} dA \]  

(33)

One of the main benefits of using Eq. (33) is that the calculation of the total force on any of the coordinate directions (\(x\) or \(y\) or \(z\)) requires velocity components in that direction (\(u\) or \(v\) or \(w\) respectively), which avoids error during interpolation in a staggered computational domain. For example, the force in the \(x\)-direction is:

\[ F_x = - \iint_S \rho U \mathbf{n} \mathbf{x} dA + \mu \iint_S \nabla U \mathbf{x} \cdot \mathbf{n} dA \]  

(34)

Similarly, the total rate of heat transfer can be calculated as:

\[ Q = -k \iint_S \frac{\partial T}{\partial n} \mathbf{n} dA \]  

(35)

The procedure to calculate the surface integration involved for computing total force and heat flux is shown schematically in Fig. 6b and in terms of generalized quantity \( I = \iint_S \frac{\partial \phi}{\partial \mathbf{n}} dA \).

The flow field can be assumed to be approximately 2D up to \( Re = 180 \) and beyond that the flow becomes 3D, and deviations obtained for a 2D solution become significant (Rajani et al., 2009). A cylinder of diameter \( d \) is placed in a 2-D rectangular computational domain of size \( 45d \times 30d \). The center of the cylinder is positioned at a distance \( 20d \) from the inlet. A uniform plug flow velocity, \( U_\infty \), at the inlet, a pressure outflow boundary condition at the outlet, free-slip boundary conditions for the outer domain boundaries, and no-slip boundary condition at the cylinder wall have been specified for the flow solver. For energy equation, a constant temperature, \( T_\infty \), is specified for the incoming fluid at the inlet, zero temperature gradient is specified at the outlet, and outer domain boundaries are set as adiabatic walls. At the cylinder wall two types of temperature boundary conditions, constant wall temperature (CWT) and constant wall flux (CWF) are specified.

The present framework allows to calculate the local force (pressure and viscous force) and heat transfer coefficient at the solid wall. At \( Re = 40 \), Fig. 7 shows the variation of the pressure coefficient \( C_p = \frac{P_\infty - P}{\frac{1}{2} \rho U_\infty^2} \) and skin-friction \( C_f = \frac{T_\infty - T_{wall}}{\frac{1}{2} \rho U_\infty^2} \) distribution at the cylinder surface. The numerical recipes to calculate local quantities at the cylinder wall are given in Eqs. (30) and (31). Two different grid resolutions are used: G20 (20 grid cells across the diameter of the cylinder) and G40. A very good agreement with the existing literature data (Tseng and Ferziger, 2003; Berthelsen and Faltinsen, 2008; Grove et al., 1964) is observed and one can clearly see that 20 grid cells across the cylinder diameter provides grid independent results, in terms of local accuracy.

5. Validation and verification test cases

5.1. Flow past a cylinder in an infinite domain

The flow field can be assumed to be approximately 2D up to \( Re = 180 \) and beyond that the flow becomes 3D, and deviations obtained for a 2D solution become significant (Rajani et al., 2009). A cylinder of diameter \( d \) is placed in a 2-D rectangular computational domain of size \( 45d \times 30d \). The center of the cylinder is positioned at a distance \( 20d \) from the inlet. A uniform plug flow velocity, \( U_\infty \), at the inlet, a pressure outflow boundary condition at the outlet, free-slip boundary conditions for the outer domain boundaries, and no-slip boundary condition at the cylinder wall have been specified for the flow solver. For energy equation, a constant temperature, \( T_\infty \), is specified for the incoming fluid at the inlet, zero temperature gradient is specified at the outlet, and outer domain boundaries are set as adiabatic walls. At the cylinder wall two types of temperature boundary conditions, constant wall temperature (CWT) and constant wall flux (CWF) are specified.

The present framework allows to calculate the local force (pressure and viscous force) and heat transfer coefficient at the solid wall. At \( Re = 40 \), Fig. 7 shows the variation of the pressure coefficient \( C_p = \frac{P_\infty - P}{\frac{1}{2} \rho U_\infty^2} \) and skin-friction \( C_f = \frac{T_\infty - T_{wall}}{\frac{1}{2} \rho U_\infty^2} \) distribution at the cylinder surface. The numerical recipes to calculate local quantities at the cylinder wall are given in Eqs. (30) and (31). Two different grid resolutions are used: G20 (20 grid cells across the diameter of the cylinder) and G40. A very good agreement with the existing literature data (Tseng and Ferziger, 2003; Berthelsen and Faltinsen, 2008; Grove et al., 1964) is observed and one can clearly see that 20 grid cells across the cylinder diameter provides grid independent results, in terms of local accuracy.

Similarly, at \( Re = 40 \) and \( Pr = 0.7 \), Fig. 8a shows the variation of the local Nusselt number \( Nu = \frac{h \cdot \delta}{k} \) along the cylinder surface when a constant wall temperature \( T_{wall} \) is specified. As expected, in this case too we found that G20 grid provide grid independent results. In Fig. 8b we compare the local Nusselt number \( Nu \) variation with literature data when a constant heat flux \( q \) is specified at...
the surface. In this case we use the probe distance equal to the grid spacing, i.e. \( \Delta n = \Delta x \). However, for certain cells we adjust (increase) the distance \( \Delta n \) such that during the trilinear interpolation it uses only the temperature of the fluid cells. For this case too we found very good agreement with literature data even with \( G_{20} \) grid. In Fig. 8-c we examine the effect of probe distance in the implementation of Neumann boundary condition. We report very negligible (less-than 2\%) maximum deviation at the stagnation point. The average Nusselt number with varying \( Re \) is presented in Table 1 and compared with existing literature results.

5.2. Flow through periodic arrays of cylinders

In this section, the present model is used to study fluid flow and heat transfer through periodic arrays of infinitely long cylinders (another simplified model porous media). Simulations are performed using three different periodic configurations shown in Fig. 9a. The center to center distance between the cylinders is 3\( d \) where \( d \) represents the diameter of the cylinder and the periodic boundary condition is applied in the \( x \)-direction. It constitutes a system with porosity, \( \varepsilon = 0.913 \). For all the three configurations (C1 to C3), the same pressure gradient is applied which results in \( Re = 40 \). The cylinder is subjected to a constant wall temperature \( (CWT) \), \( T_s \) and the fluid Prandtl number, \( Pr = 1 \). For all the three cases 20 grid cells across the diameter are considered. Ideally, one can expect exactly the same profile of the non-dimensional temperature and the non-dimensional velocity for all the three configurations for proper treatment of periodic boundary condition. Along the line joining the center of the cylinders (line AB in Fig. 9a) the non-dimensional temperature (\( \Theta \)) and the non-dimensional axial velocity (\( u_s/U \)) are plotted for all the three configurations which shows an excellent agreement.

The non-dimensional drag (\( f \)) and average Nusselt number \( (Nu_{CWT}) \) are defined as,

\[
f = \frac{\beta}{\mu U/d^2}
\]

\[
Nu_{CWT} = \frac{d}{k} \frac{q}{\Delta T_{LMD}}
\]

where \( \Delta T_{LMD} = \frac{(T_s - T_{b, in}) - (T_s - T_{b, out})}{ln[(T_s - T_{b, in})/(T_s - T_{b, out})]} \)

The local Nusselt number (\( Nu_l \)) distribution on the cylinder surface for (a) CWT (at \( Re = 40 \)) and (b, c) CWF (\( Re = 45 \)) conditions when \( Pr = 0.7 \). \( G_{20} \) and \( G_{40} \) correspond to 20 and 40 grid cells across the diameter of the cylinder. In sub-figure (c), we examine the effect of probe distance (\( \Delta n \)) in the present implementation of Neumann boundary condition.

Fig. 7. Local pressure coefficient (\( C_p \)) and skin-friction (\( C_f \)) distribution on the cylinder surface at \( Re = 40 \). G20 and G40 correspond to 20 and 40 grid cells across the diameter of the cylinder.

Fig. 8. Local Nusselt number (\( Nu_l \)) distribution on the cylinder surface for (a) CWT (at \( Re = 40 \)) and (b, c) CWF (\( Re = 45 \)) conditions when \( Pr = 0.7 \). G20 and G40 correspond to 20 and 40 grid cells across the diameter of the cylinder.
The difference (LMTD) of inlet and outlet bulk temperatures ($T_b$; inlet and $T_b$; outlet) from the cylinder wall temperature ($T_s$). For the present case, we report $f = 3.238$ and $\text{Nu}_{CWT} = 2.542$.

The spatial accuracy (global) of the present method is demonstrated by a grid refinement study of a configuration where the cylinder is placed exactly in the middle of the periodic domain. We use in total six grid resolutions, $d = D_x = 10, 20, 40, 80$ and $160$, respectively, for error analysis. The relative errors, $|e_f|$ and $|e_{\text{Nu}}|$, on the total drag force ($f$) and the average Nusselt number ($\text{Nu}_{CWT}$) are computed by comparing coarser grids with the finest grid solution. Moreover, we also report the global error in momentum and heat balance in the periodic domain for different grid resolutions.

As mentioned earlier, for the solution of the momentum equations in a periodic domain, the periodic pressure $P$ is solved in the pressure-velocity coupling (refer Section 3.5). The total force $F$ (in the flow direction) on the solid body by the fluid is calculated by the surface integral, consisting of forces due to viscous shear ($F_s$) and pressure variation ($F_p$). Since here we are solving for periodic pressure $P$ instead of actual pressure $p$, the numerically calculated pressure force ($F_P$) only contains the force exerted due to the local pressure variation. In the actual system, due to the global pressure gradient it also experiences a buoyancy-type force $F_B$ and from the force balance we can write,

$$
\beta = \frac{F}{V} = \frac{1}{V} (F_s + F_p + F_B) = \frac{1}{V} (F_s + F_p + \beta V_s)
$$

$$
= \frac{1}{V} (F_s + F_p) + (1 - \varepsilon) \beta
$$

where $V$ is the total volume of the system and $V_s$ is the volume of the solid in the system, hence, the porosity, $\varepsilon = 1 - V_s/V$. From Eq. (42), we obtain:

$$
\beta = \frac{F_s + F_p}{\varepsilon V}
$$

In Eq. (43), the pressure gradient $\beta$ is an input for the simulation, while the quantity $F_s + F_p$ is calculated after the solution reaches a steady state. We define relative error in momentum balance as,

$$
e_M = \frac{\beta - \frac{F_s + F_p}{\varepsilon V}}{\beta}
$$

### Table 1
Comparison of the present computation of average Nusselt number ($\text{Nu}$) with literature results for flow past a static cylinder with constant wall heat flux condition when $Pr = 0.7$.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>2.03</td>
<td>2.04</td>
<td>2.00</td>
<td>2.03</td>
</tr>
<tr>
<td>20</td>
<td>2.76</td>
<td>2.78</td>
<td>2.71</td>
<td>2.74</td>
</tr>
<tr>
<td>40</td>
<td>3.75</td>
<td>3.77</td>
<td>3.67</td>
<td>3.74</td>
</tr>
<tr>
<td>100</td>
<td>G10: 6.468</td>
<td>–</td>
<td>–</td>
<td>5.886</td>
</tr>
<tr>
<td></td>
<td>G20: 5.896</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>G40: 5.873</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>150</td>
<td>7.15</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>200</td>
<td>8.09</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>
Similarly, we define a relative error in energy balance ($\epsilon_e$) by computing the total heat flux from the solid to fluid and the convective flux at the inlet and outlet of the domain. In Fig. 10a the relative errors ($\epsilon_e$ and $\epsilon_{Nu}$) has been plotted with dimensionless grid resolution ($d/\Delta x$). A second order spatial convergence (global accuracy) is clearly observed for non-dimensional drag and Nusselt number, whereas the total momentum and energy balance converge linearly with grid resolution. The relative accuracy and the order of convergence for the solution of the energy equation is higher compared to the solution of the momentum equation. It is worth to notice that a grid resolution of 10 cells across the diameter produces results within 3% accuracy for drag and within 1% accuracy for the Nusselt number.

Next, we consider the same system, however instead of an isothermal condition, a constant wall flux (CWF) is specified at the cylinder wall. The average Nusselt number ($Nu_{CWF}$) is defined similarly in terms of log mean temperature difference. However, in this case the $T_{avt}$ is defined in terms of average cylinder wall temperature which is computed numerically. We report $Nu_{CWF} = 3.763$ for the case of constant heat flux boundary condition. Similarly, in Fig. 10b the relative errors ($\epsilon_{Nu}$) has been plotted with dimensionless grid resolution ($d/\Delta x$). A second order spatial convergence (global accuracy) is clearly observed. In this case the total energy balance with grid resolution also follows second order trend. The relative accuracy and the order of convergence for the solution of the energy equation is higher compared to the solution of the momentum equation. A grid resolution of 10 cells across the diameter produces results within 2% accuracy.

5.3. Effective thermal conductivity of a model porous medium

The effective thermal conductivity of porous media is of great significance for volume-averaged macroscopic modeling of porous media. By utilizing the present CHT method one can very accurately calculate the effective thermal conductivity for any complex porous structure. Here, we consider a very simplified 2D porous medium made up of a solid material (subscript ‘s’) and a fluid (subscript ‘f’) filling a cylindrical pore having porosity, $\varepsilon = 0.196$ (Fig. 11a).

One can consider that the fluid and solid layers are oriented either parallel or perpendicular to the direction of the heat flow to obtain the upper and lower bound of the effective thermal conductivity.

The effective thermal conductivity of the parallel layers, $k_i$ (perpendicular layers, $k_o$) is given by the weighted arithmetic mean (harmonic mean) of the fluid conductivity ($k_f$) and the solid conductivity ($k_s$): $k_\parallel = (1 - \varepsilon)k_s + \varepsilon k_f$ and $k_\perp = \frac{k_s}{1 - \varepsilon} + \frac{\varepsilon}{k_f}$.

Using the upper and lower limit ($k_i$ and $k_o$) of the stagnant effective thermal conductivity ($k_{eff}$), several authors represent it as, $k_{eff} = k_f^{1-F} k_s^{1-F}$, where $0 \leq F \leq 1$.

To calculate $k_{eff}$, a temperature gradient is applied in the perpendicular direction to one of the faces and the heat flow rate along the temperature gradient is computed. Three different conductivity ratio, $K_r = k_i/k_o = 10, 100$ and 1000 are considered where $k_i = 1$. The temperature contours for $K_r = 10$ and $K_r = 1000$ are shown in Fig. 11a. One can observe a higher distortion of temperature contours for the case of $K_r = 1000$. We report $k_{eff} = 7.23, 67.70$ and 672.12 for $K_r = k_i/k_o = 10, 100$ and 1000, respectively, whereas $k_\parallel = 3.6, 4.9$ and 5.1, and $k_\perp = 8.24, 80.6$ and 804.2, respectively.

The spatial accuracy (global) of the present CHT method is demonstrated by a grid refinement study. We use in total six grid resolutions, $d/\Delta x = 10, 20, 40, 80$ and 160, respectively, and the relative errors in predicting $k_{eff}$ are computed by comparing coarser grids with the finest grid resolution (i.e. $d/\Delta x = 160$). In Fig. 11b the relative errors ($\epsilon$) has been plotted with dimensionless grid resolution ($d/\Delta x$) for different conductivity ratios. A second order spatial convergence (global accuracy) is clearly observed for all the cases, however the absolute accuracy decreases with increasing $K_r$.

5.4. Transient heat conduction of a sphere in an infinite domain

Further investigation focuses on the temporal accuracy of the conjugate heat transfer implementation. We consider a stagnant two-phase system where a sphere (subscript ‘1’) of diameter $d$ is buried in an infinite domain (subscript ‘2’). The initial temperature of the sphere ($T_i$) is higher than the initial temperature of the sur-
boundary effects and the simulations are performed up to \( T = \frac{1}{2} \). We consider equal density and specific heat capacity of the sphere and the surroundings, i.e. \( \rho_1 c_p = \rho_2 c_p \). However, we vary the conductivity ratio, \( K_r = k_2 / k_1 \), to 10, 100 and 1000. The sphere will lose heat to the surroundings until it reaches the thermal equilibrium condition. To represent non-dimensional time, we use Fourier number, \( F_0 = \alpha t / d^2 \), where \( \alpha = k_1 / (\rho_1 c_p) \). We use a domain size of \( 20d \times 20d \times 20d \) to avoid boundary effects and the simulations are performed up to \( F_0 = 0.025 \).

The present problem can be simplified in a 1D spherical coordinate system where the sphere surface exactly passes through a face-center of a finite volume grid. As a result, it is indeed possible to treat time in a fully implicit manner of the boundary fitted 1D grid. At the same time, as the model is 1D, one can use a very high grid resolution (2000 grid points across the diameter of the sphere). In Appendix B we present the above mentioned 1-D model. It can be considered as resolution free time-step independent method which is used to validate the current IBM-CHT model.

We compare the transient evolution of temperature from the present IBM-CHT model with the 1-D, body conforming, fully implicit model for different \( K_r \). Along the radial direction, we non-dimensionalize the local variation of temperature (\( T \)) as, \( \Theta_r = (T - T_1) / (T_1 - T_2) \), and the radial distance (\( r \)) as \( r^* = r / (0.5d) \). Hence, at the initial condition, inside the sphere \( (0 < r^* < 1) \), \( \Theta = 0 \) and outside the sphere \( (1 < r^* < \infty) \), \( \Theta = 1 \). Fig. 12a–c show the variation of radial temperature at \( F_0 = 0.00625, 0.0125 \) and 0.025 for different conductivity ratios. In these figures, we have used 20 cells across the diameter of the sphere, however we have varied the time-step size (\( \Delta t \)). We have used three different time-step size, \( \Delta t = 0.1 \Delta t_r, \Delta t_r, 10 \Delta t_r \) and 100\( \Delta t_r \) where \( \Delta t_r = \Delta x^2 / (2 \alpha) \), i.e. stable time-step for an explicit diffusion problem. We observe that the current IBM-CHT implementation (symbols) matches very closely with the 1D code (continuous lines) when time-step size is lower than 10\( \Delta t_r \). It is stressed here that from the numerical stability point of view the current implementation is unconditionally stable, however, as the surface temperature lags by one time-step, the accuracy of the transient response demands certain \( \Delta t \) restriction. Fig. 12-d shows the accuracy of the method in terms of spatial resolution. We choose three different grid resolutions, \( d / \Delta x = 10, 20 \) and 40, when \( \Delta t = 1 \Delta t_r \) at \( K_r = 1000 \). At this high conductivity ratio too \( d / \Delta x = 10 \) grid provides reasonably accurate results.

6. Convective and conjugate heat transfer through random open-cell foams

To demonstrate the capability of the present numerical framework, we perform convective and conjugate heat transfer simulations using a highly complex random open-cell solid foam geometry (refer Fig. 13) of porosity \( \epsilon = 0.948 \). Open-cell solid foam is a novel material consisting of interconnected ligaments forming a complex network of randomly oriented polyhedrons. Because of its excellent thermal, mechanical and acoustic properties, such foams are extensively used for several industrial applications. In the accompanying paper (Das et al., 2018) we have presented a methodology to generate realistic random micro-structures of open-cell solid foam. A open-source software Surface Evolver (Brakke, 1992) is used to generate foam geometry which can inherently mimic the foam forming process (i.e. minimization of surface energy). The generated random foam sample is periodic in all the three directions that help us to perform simulation in a fully-developed condition by utilizing periodic boundary condition. The random foam geometry is stored as a triangulated surface mesh using the Stereo Lithography (STL) file format and imported in the periodic CFD domain.

At first, we compute solid to liquid heat transfer with varying Reynolds number (\( Re \)) when the temperature of the foam geometry is maintained constant. The flow is accelerated by applying a constant body force (Section 3.5). The heat transfer calculations are performed with Prandtl number, \( Pr = 1 \). A temperature-periodic boundary condition (Eq. (22)) is used for \( Re \approx 10 \) to 500 across the periodic faces in the macroscopic flow direction, whereas for \( Re \approx 0.1 \) and 5, a Danckwert type boundary condition (Eq. (26)) across the periodic faces in the flow direction is used. All boundaries parallel to the flow direction are subjected to a zero heat flux condition. In this way one can compute the heat transfer coefficient at a thermally fully developed condition. Note that we use the superficial velocity and equivalent spherical diameter (6 times the ratio of the solid volume to the solid surface area) to define the Reynolds number. Fig. 13 shows the non-dimensional temperature contours at the mid-plane of the simulation domain for different flow Reynolds numbers. In the direction of the flow the average temperature increases (or decreases) due to fluid heating (or cooling) by the porous structure. However, at the thermally fully developed condition, the non-dimensional temperature field...
(H) repeats its shape in the streamwise flow direction. At high Reynolds number, the overall heat transfer enhances because of vigorous flow structures. At Re ≈ 500 the flow is unsteady. However, the magnitude of fluctuation is very small (maximum variation in Θ is ±1%).

In order to find the heat transfer coefficient (h), we calculate the total...
the temperature differences for heat transfer and observe a comparatively higher distortion of temperature contours at a mid-plane for the case of solid volume fraction, contour-lines are parallel, however one can crudely say $F \approx 0.5$. However, further parametric variation is needed to establish a correlation of $F$ in terms of $\varepsilon$ and $K_r$.

7. Conclusions

In the present work, we have proposed a versatile sharp interface second order accurate immersed boundary method to simulate flow and heat transfer through complex porous structures in a staggered Cartesian grid. Very complex solid geometrical shape can be imported as a triangulated surface mesh in a periodic computational domain. The present method is also capable of handling very sharp-edged solid geometry or zero thickness walls (e.g. baffles) that are very commonly encountered in complex porous structures. Dirichlet boundary condition has been implemented in a fully implicit manner where the boundary condition at the immersed surface has been enforced by ghost cells inside the solid. To implement conjugate heat transfer and Neumann boundary conditions, we have used the same framework, however, the surface temperature is computed from the previous time-step temperature field utilizing probes perpendicular to the immersed surface. For simulating heat and fluid flow through representative element volume (REV) of a porous medium, we have also developed a periodic boundary condition for both the velocity and temperature.

Extensive verification studies have been performed, and both velocity and temperature fields show second-order error convergence for all the boundary conditions. We have compared the local and global flow properties with existing literature results and a very good agreement has been reported. We have also simulated a transient heat conduction case where a hot sphere is placed in an infinite domain. This conjugate heat transfer problem has been simplified to a 1D spherical coordinate system which enables us to use a boundary fitted 1D grid with extremely high grid resolution and a fully implicit treatment of conjugate heat transfer boundary condition. We have compared the results obtained from the pre-
sent 3D IBM-CHT code with the 1D spherical coordinate code, and a very good spatial and temporal accuracy have been reported. Finally, the capability of the implementation has been demonstrated by performing convective and conjugate heat transfer through a periodic highly complex open-cell foam geometry. In the accompanying numerical method to develop momentum and thermal closures for random computer generated open-cell solid foams.

Acknowledgments

We thank SURF SARA (www.surfsara.nl) and NWO for the support in using the Cartesius supercomputer.

Appendix A. Special considerations for sharp solid edges and limiting cases

The present method is capable of tackling sharp solid edges (mainly for Dirichlet condition), as well as solid bodies with zero thickness (e.g. baffles), very commonly encountered in complex porous structures. However, certain cases need special consideration, schematically shown in Fig. 16. Fig. 16a shows a sharp solid porous structures. However, certain cases need special consideration, as-well-as solid bodies with zero thickness (e.g. baffles), very commonly encountered in complex limiting cases

\[
\begin{align*}
\tilde{A}_C &= A_C + x_{\phi_0}A_{\phi_0} + \chi_{\phi_0}A_{\phi_0} \\
\tilde{B}_C &= B_C - \gamma_{\phi_0}A_{\phi_0} - \chi_{\phi_0}A_{\phi_0} \\
\bar{A}_{\phi_0} &= 0 \\
\bar{A}_{\phi_0} &= 0
\end{align*}
\]

where the generalized forms of \( \alpha \) and \( \gamma \) are given by,

\[
\begin{align*}
\alpha_{\xi} &= -\frac{\xi}{\tau_{\xi}} & \text{when } 0 \leq \xi < 0.999 \\
\gamma_{\xi} &= 2\phi_{\xi} & \text{when } 0.999 \leq \xi \leq 1
\end{align*}
\]

For the presence of IB-solid cell \( C_9 \), the coefficients (\( \tilde{A} \) and \( \tilde{B} \)) are again modified using (10).

Fig. 16-c shows a limiting case where the cell center of \( C_4 \) exactly lies on the triangular element \( t_1 \). It creates a serious problem during flagging of fluid and solid cells. To avoid this, during the flagging of fluid cells (Section 3.4: Step 1), we inflate the surface mesh along its unit normal with a distance of approximately \( 10^{-3} \times \Delta x \). In the figure the red dotted line indicates the new position of triangular elements and as a result cell center \( C_4 \) is flagged as a fluid cell. However, the flagging of solid cells (Section 3.4: Step 2) is carried out using the actual surface mesh. This very small increase in surface mesh size is only utilized for flagging purposes and it also helps to avoid other limiting cases shown in Fig. 16-d and e.

Appendix B. Semi-analytical model for heat conduction of a sphere in an infinite domain

Consider a sphere (subscript ‘1’) of radius \( R \) buried in a sphere (subscript ‘2’) of infinite radius \( R_\infty \). The initial temperature of the inner sphere is \( T_1 \), different than the initial temperature of the surrounded sphere \( T_2 \). The heat equations in spherical coordinates with azimuthal and poloidal symmetry read,

\[
\begin{align*}
\frac{\partial T}{\partial t} &= \frac{1}{r^2} \frac{\partial}{\partial r} \left( r^2 \frac{\partial T}{\partial r} \right), & r \leq R \\
\frac{\partial T}{\partial t} &= \frac{1}{r^2} \frac{\partial}{\partial r} \left( r^2 \frac{\partial T}{\partial r} \right), & R \leq r \leq R_\infty
\end{align*}
\]
where \( r \) represents the radial coordinate and \( \alpha \) represents thermal diffusivity. In Fig. 17 the present 1D system is illustrated. The above equations need to be solved simultaneously with the following boundary and initial conditions,

\[
\begin{align*}
\frac{\partial T}{\partial r} & = 0 \\
T & = T_2 \\
\frac{\partial T}{\partial r} & = \frac{T_1 - T_2}{\Delta r} \\
T & = T_2 (r < R) \\
T & = T_1 (r > R)
\end{align*}
\]  

(50)

We discretize Eq. (49) on an uniform finite volume grid. The grid spacing \( \Delta r \) is chosen in a way that \( n_1 \Delta r = R \) and \( n_2 \Delta r = R_{\infty} - R \), where \( n_1 \) and \( n_2 \) are integers, represent the number of grid-cells in the inner and outer spheres, respectively. In this study we have used \( n_2 = 20n_1 \) to nullify the boundary effects of the infinite domain.

We use a second order Baker-Oliphant scheme for temporal discretization and the central difference scheme for spatial discretization. The generalized form of the discretization of Eq. (49) is given as,

\[
\frac{1}{2\Delta t} \left[ 3T_{i,1}^{n+1} - 4T_{i}^{n+1} + T_{i-1}^{n+1} \right] = \frac{\alpha}{\Delta r^2} \left[ r_2^2 T_{i,1}^{n+1} - (r_1^2 + r_2^2)T_{i}^{n+1} + r_1^2 T_{i-1}^{n+1} \right]
\]  

(51)

where sub-script \( i \) and super-script \( j \) represent running index for space and time, respectively. For both the inner ('1') and outer ('2') sphere [Fig. 17], Eq. (51) can be represented as

\[
A_{n,i} T_{i,1}^{n+1} + A_{c,i} T_{i}^{n+1} + A_{p,i} T_{i-1}^{n+1} = \beta_i
\]  

(52)

where \( A_{n,i} = -\frac{\alpha}{\Delta r^2} r_1^2 \), \( A_{c,i} = -\frac{\alpha}{\Delta r^2} r_2^2 \), \( A_{p,i} = \frac{\alpha}{\Delta r^2} (r_1^2 + r_2^2) \), \( A_{c} = \frac{\alpha}{\Delta r^2} r_2^2 \), and \( \beta_i = 2\Delta t \left[ 4T_{i}^{n+1} - T_{i-1}^{n+1} \right] \). By modifying the coefficients (\( A \) and \( \beta \)) the boundary conditions are applied. By satisfying the continuity of heat flux and the temperature at the surface of the inner sphere, we obtain temperature at the interface as,

\[
T_r = \frac{k_1(9T_{n-1} - T_{n-2}) + k_2(9T_{n+1} - T_{n+2})}{8(k_1 + k_2)}
\]  

(53)

This temperature can thus be enforced as Dirichlet boundary condition at the interface in the present body fitted 1D grid, however here we can do this procedure in a fully implicit manner. While solving for cell \( n_1 \), to enforce boundary condition at the interface, we represent the temperature of the cell \( n_1 + 1 \) in terms of \( T_{n+1} \) and \( T_{n-1} \) using a second order fit. Using Eq. (20), one can obtained,

\[
T_{n+1} = \frac{1}{3} (8T_r - 6T_{n+2} + T_{n+1})
\]  

(54)

For the cell \( n_1 \), after substituting \( T_{n+1} \) and \( T_r \) in Eq. (52), one can obtain the following:

\[
\tilde{A}_{n,n_1} T_{n_1}^{n+1} + \tilde{A}_{c,n_1} T_{n_1}^{n+1} + \tilde{A}_{p,n_1} T_{n_1}^{n+2} + \gamma T_{n_1}^{n+1} - \beta_{n_1} = 0
\]  

(55)

where

\[
\tilde{A}_{n,n_1} = A_{n,n_1} + A_{p,n_1} [k_2/3(k_1 + k_2)]
\]

\[
\tilde{A}_{p,n_1} = A_{p,n_1} [k_2/3(k_1 + k_2)]
\]

\[
\tilde{A}_{c,n_1} = A_{c,n_1} + A_{p,n_1} [k_2/3(k_1 + k_2)]
\]

\[
\gamma = -\frac{A_{p,n_1} [k_2/3(k_1 + k_2)]}{\beta_{n_1}}
\]

Similarly, while solving for cell \( n_1 + 1 \), we represent \( T_{n_1} \) in terms of \( T_{n_1+1} \) and \( T_{n_1+2} \), and we modify the coefficient of Eq. (52).

\[
\text{References}
\]


Fig. 17. 1-D Semi-analytical model for heat conduction of a sphere in an infinite domain.


