Eleventh International Conference on CFD in the Minerals and Process Industries
CSIRO, Melbourne, Australia
7-9 December 2015

DIRECT NUMERICAL SIMULATION OF FLOW THROUGH A SOLID FOAM: 3D μ-CT IMAGE TO AN IMMERSED BOUNDARY METHOD (IBM) BASED CFD MODEL

Saurish Das, J.A.M. (Hans) Kuipers & Niels G. Deen*

Department of Chemical Engineering and Chemistry, Technical University of Eindhoven,
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
*Corresponding author, E-mail address: N.G.Deen@tue.nl

ABSTRACT

Due to the random and complex geometrical shapes, most of the work on open-cell solid foams are experimental, and only a limited number of numerical studies are available in the literature. In the present contribution a sharp Immersed Boundary Method (IBM) is developed to fully resolve the random structure of the solid foam on a non-body fitted Cartesian computational grid. 3D image data from a Micro-CT (computed tomography) scan of an actual foam geometry is converted into unstructured triangular elements (surface mesh) and incorporated as immersed surface in the present sharp interface method. The novel feature of the fluid–solid coupling technique is the direct (i.e. implicit) incorporation of the no-slip boundary condition (with a second-order method) on the surface of the particles at the level of the discrete momentum equations. The use of a Cartesian grid for flow makes this method robust and computationally friendly. At the same time it avoids the tedious volumetric mesh generation process for very complex shapes. After presenting some verification cases, we perform simulation results for flow through a section of a typical random foam structure generated from a CT-scan technique. The Reynolds number is varied from creeping flow to Re ~ 40. Finally, a pressure drop correlation is proposed for this particular foam sample.

NOMENCLATURE

d_p	equivalent particle diameter
f	friction factor
F	dimensionless drag force
p	pressure
S_a	specific surface area
t	time
u_s	superficial velocity
$\bar{C}$	Convective Flux
$\bar{n}$	unit normal vector
$\bar{u}$	velocity
$\bar{x}$	position vector of cell-centre
ρ	fluid density
μ	fluid viscosity
ϕ	any flow variable
ξ	dimensionless distance
ε	porosity
δp	pressure correction

INTRODUCTION

Open-cell solid foam packing represents a type of structured porous media that offer high specific surface area with a low pressure drop per unit length. Solid foams may be produced from a variety of materials like metal, ceramics, carbon, polymer, etc. Historically, solid foams made by metal are used mainly in heat transfer devices like heat exchangers, thermal energy absorbers, vaporizers, heat shielding devices etc. Due to their good mechanical and thermal behaviour solid foams have also gained attention in several other applications like high temperature filters, pneumatic silencers, burn rate enhancer, catalytic reactors etc. Over the last couple of decades open cell foams (metal or ceramics) has been used as catalyst support in fixed bed catalytic reactors. The use of foams improves gas-liquid contacting, by enhancing the mass-transfer rate with minimal pressure drop as compared to other packing structures. Also, for the case of exothermic reactors where heat needs to be removed from the bed, use of metallic foams enhances heat removal rate.

A large number of experimental studies were reviewed by Edouard et al. (2008) and Dietrich et al. (2012). They reported very large deviations between the existing pressure drop correlations due to (1) morphological differences between different commercially available solid foams and (2) experimental inaccuracies. In contrast, numerical studies at pore scale level are capable to overcome all experimental uncertainties, but most of those studies have only focused on idealized unit cells to represent a solid foam. In recent years researchers used high resolution X-ray computed micro-tomography (μ-CT) techniques to reconstruct random complex foam structures to accurately calculate morphological properties (surface area, window opening, strut diameter, porosity etc.) and subsequently incorporate these into a CFD domain to study fluid flow [Petrasch et al. (2008), Magnico et al. (2009)].

In the present contribution we propose an efficient and accurate sharp interface Immersed Boundary Method (IBM) to fully resolve a random foam structure. The Immersed Boundary Method is a novel computational

technique which is an alternative to body conformal grid methods for flow over complex shapes. It avoids the challenging high quality grid generation process and the development of a CFD code for unstructured grids. Moreover, simple structured grid based codes are better than unstructured grid codes in terms of memory requirement to store the grid information, computational cost relating with convergence characteristics and the efforts involved to generate a good quality mesh. At the same time IBM can easily be coupled with other transport equations [Deen et al. (2014)]. On the other hand, body fitted grids has a great ability for local mesh refinement in zones of interest where sharp gradients occur. In case of structured meshes uniform grid refinement would be required, which would take away some of the advantages.

There exists a large variety of IBM methods and an excellent literature review on this was undertaken by Mittal and Iaccarino (2005). The present methodology is an extension of the implicit second order accurate IBM method proposed by Deen et al. (2012) for complex shaped object. In the current framework any complex shaped hollow or solid object can be placed in a structured Cartesian grid in the form of a surface mesh. After discussing the numerical methodology briefly, here we present few validation cases using simple geometries. Subsequently, it is applied to simulate flow through a section of a typical complex foam structure generated from a Micro-CT (computed tomography) scan through 3D imaging techniques. Simulations are performed in both the Darcy and Forchheimer regime. Finally, a pressure drop correlation as a function of Re for this specific foam structure is proposed which is useful for more coarse grained simulation techniques.

MODEL DESCRIPTION

Numerical Model and Solution Methodology

The conservation equations of mass and momentum for an incompressible, Newtonian fluid read:

$$\nabla \cdot \bar{u} = 0 \tag{1}$$

$$\rho \frac{\partial \bar{u}}{\partial t} + \rho \nabla \cdot \left(\overline{uu} \right) = -\nabla p + \mu \nabla^2 \bar{u} \tag{2}$$

where ρ and μ are the fluid density and viscosity respectively. In the current Finite Volume (FV) implementation transport equations are time-integrated for each staggered computational grid cell and the discretized form of the momentum equation is obtained as:

$$\rho \bar{u}^{n+1} = \rho \bar{u}^{n} + \Delta t \left[-\nabla p^{n+1} + \mu \nabla^2 \bar{u}^{n+1} - \left\{ \bar{C}_{FOU}^{n+1} + \left(\bar{C}_{Min-Mod}^{n} - \bar{C}_{FOU}^{n} \right) \right\} \right] \tag{3}$$

where n indicates the time level and $\bar{C}$ indicates the net-convective flux:

$$\bar{C} = \rho \left(\nabla \cdot \overline{uu} \right) \tag{4}$$

A deferred correction method has been incorporated in the convection term, where both the First Order Upwind (FOU) scheme and the total variation diminishing (TVD) Min-mod scheme are used to calculate the convection flux. The flux based on the FOU scheme $\left(\bar{C}_{FOU} \right)$ is used as a predictor and treated implicitly, while the difference between the TVD and FOU schemes is used as a corrector that is treated in an explicit manner. The second order accurate central difference (CD) scheme is used for discretizing the diffusion terms. The discretized momentum equation is solved with a fractional step method, where at the first-step the tentative velocity field $\bar{u}^{**}$ is computed from:

$$\rho \bar{u}^{**} = \rho \bar{u}^{n} + \Delta t \left[-\nabla p^{n} + \mu \nabla^2 \bar{u}^{**} - \left\{ \bar{C}_{FOU}^{**} + \left(\bar{C}_{Min-Mod}^{n} - \bar{C}_{FOU}^{n} \right) \right\} \right] \tag{5}$$

To find $\bar{u}^{**}$ we need to solve a set of linear equations. To this end, we use a robust and efficient parallel Block Incomplete Cholesky Conjugate Gradient (B-ICCG) solver. The velocity at the new time level *n+1* can be obtained from:

$$\bar{u}^{n+1} = \bar{u}^{**} - \frac{\Delta t}{\rho} \nabla \left(\delta p \right) \tag{6}$$

where $\delta p = p^{n+1} - p^{n}$, represents the pressure correction. Since $\bar{u}^{n+1}$ need to satisfy the continuity equation, the pressure Poisson equation is obtained as:

$$\nabla \cdot \left\{ \frac{\Delta t}{\rho} \nabla \left(\delta p \right) \right\} = \nabla \cdot \bar{u}^{**} \tag{7}$$

which is again solved by the B-ICCG sparse matrix solver.

Immersed Boundary Method

In this paper, an implicit (direct) second order accurate IBM method proposed by Deen et al. (2012) has been extended for complex geometries. There exists a large variety of CAD mesh formats to represent complex geometries. In the current framework we have chosen a surface mesh with triangular elements to represent all immersed bodies. Triangular elements are stored in a STL (Stereo-Lithography) file format and are described as a raw unstructured triangulated surface by the unit normal and vertices (ordered by the right-hand rule) of the triangles using a three-dimensional Cartesian coordinate system [Fig. 1a]. The normal of each triangle always points outwards.

Fig. 1b schematically represents the current IBM implementation. The enforcement of the no-slip boundary condition at the IB wall is handled at the level of the discretized momentum equation [Eq. 5]. In this direct forcing method at first all the cells are marked/ flagged as either “solid” (cell-center inside the solid body) or “fluid” (cell-center in the fluid zone). “IB-cells” are the special fluid cell which neighbor at least one solid cell and that neighboring solid cell named as “ghost-cell”. The following steps allow to mark the appropriate cells in staggered computational grid:

(1) At first, the centroid of the each triangular surface elements are located in an Eulerian background grid. A 3D

bounding box is created around the background grid and all the cells in the bounding box are need to check if they are either solid or fluid-cells. For each cell in the bounding box, grid lines are drawn connecting the cell center of that cell with each of its six neighboring cell centers. If any of these grid lines intersect the triangular element, the following dot product is evaluated for that pair of cells:

$$v = (\bar{x}_c - \bar{x}_n) \cdot \bar{\mathrm{n}} \quad \textbf{(8)}$$

where $\bar{x}_c$ is the position vector of the cell in the bounding box and $\bar{x}_n$ is the position vector of its neighboring cell, $\bar{\mathrm{n}}$ is the surface normal of the triangular element. If $v \leq 0$, cell *'n'* is marked as a solid-cell, else *'c'* is marked as a solid-cell. In a flag matrix the marked cells are represented as integers: solid-cells get flag -1 and fluid-cells get flag 0. This procedure will create a close hollow shell of solid flags around the IB. For proper enclosing, the size of the bounding box should be based on the relative size between the triangular element and Eulerian grid. For instance if the size of the triangular element is in the range to the size of the mesh, a stencil size of 3×3×3 is sufficient.

(2) IB-cells are marked in a similar way. A cell *'c'* will be flagged as a IB-cell when one of its neighbor (*'n'*) is a solid cell and $v \leq 0$ (Eq. 8). In the flag matrix the IB-cells are initially identified by integer 1 and later changed to a hash code, which is discussed later.

(3) Until now we have flagged all the solid-cells enclosing the solid body and all the IB-cells. Now we need to assign solid flag inside the solid body, which were initially flagged as fluid cell. This has to be done by sequential tracing in any of the three directions: for instance if we move along the positive x-direction, then any fluid-cell with index [*i*][*j*][*k*] that has a neighboring solid-cell at index [*i-1*][*j*][*k*] will be marked as a solid-cell. The procedure automatically stops when it reaches the solid cell in the other end of the body as it is guarded by a IB-cell. If the triangular surface mesh and/or the initial marking of the solid and IB-cells are not close this method will fail.

The no-slip boundary condition on the IB is enforced directly through the IB-cells. The algebraic form of the discretized momentum equation [Eq. 5] at cell-center *'c'* in Fig. 1b can be expressed as:

$$a_c \phi_c + \sum_{nb} a_{nb} \phi_{nb} = b_c \quad \textbf{(9)}$$

where ϕ corresponds to one of the fluid velocity components and *'nb'* indicates the cell-center of surrounding neighboring cells. The coefficients *'a'* depend on the fluid properties and the chosen grid resolution. All the explicit terms are collected in b_c. For a perspective of a IB-cell *'c'* , the no-slip condition is imposed by changing the central coefficient a_c and neighboring coefficients a_{nb} of Eq. 9. In the Fig. 1b IB-cell *'c'* neighbors a ghost-cell *'w'* and a fluid-cell *'e'*. The value of ϕ in the ghost-cell *'w'* is expressed as a linear combination of relevant fluid-cells in such a way that it satisfy no-slip boundary condition at the wall *('s')*. For this purpose a quadratic fit is used: $\phi = p\xi^2 + q\xi + r$, where ξ is the dimensionless distance from ghost-cell *'w'* as shown in Fig. 1b. The values of coefficients *p*, *q* and *r* are obtained from the known values of ϕ at the IB, which leads to:

$$\phi_w = -\frac{2\xi_s}{1-\xi_s}\phi_c + \frac{\xi_s}{2-\xi_s}\phi_e \quad \textbf{(10)}$$

The modified coefficients ($\hat{a}$) for Eq. 9 came out as:

$$\hat{a}_c = a_c + a_w\left(-\frac{2\xi_s}{1-\xi_s}\right) \quad \textbf{(11a)}$$

$$\hat{a}_e = a_e + a_w\left(-\frac{\xi_s}{2-\xi_s}\right) \quad \textbf{(11b)}$$

$$\hat{a}_w = 0 \quad \textbf{(11c)}$$

Note that Eq. 9 is also formed for solid-cells (and ghost cell) and to enforce zero velocity the central coefficient (a_c) is set to 1, while all other coefficients (i.e. a_{nb}, b_c) are set to 0.

From the above methodology proposed by Deen et al. (2012) it is clear that to implicate IB in the Cartesian grid we only need the ξ_s values at the IB-cells. It can easily be calculated by the intersection between the triangular element and grid line(s) as shown in Fig. 1b. For a stationary IB, to reduce the computational time drastically, we stored ξ_s for each IB-cells in a hash table format (Fig. 1b), which is a data structure where a key maps to associated values. As shown in Fig. 1b all the IB-cells are numbered and these numbers are stored in the flag matrix. These numbers represent a hash code or key for the hash table where all six (for 3D) ξ_s values are stored. Please note that if a IB-cell shares only one neighboring solid-cell (which happens for most of the IB-cells) only one value of ξ_s is needed and the rest are irrelevant (stored 0 in the code). Once the hash table is generated, triangular surface mesh is no longer needed for stationary body case.

VERIFICATION AND VALIDATION

The above frame work has been rigorously verified and validated for a large range of problems. However in this current paper only two cases are shown, which will now be discussed.

Flow past periodic arrays of cubes with different orientation

To verify the current implementation, a STL surface mesh for a cube is placed at the center of 3-D periodic computational domain. The fluid is accelerated from rest by applying a constant body force or average pressure gradient in the x-direction. Two separate simulations are performed using the surface mesh for cubes of the same size but with different orientation in the plane normal to the flow [Fig. 2]. For an ideal case the flow field must be the same for the two cases. The simulation settings are tabulated in the Table 1.

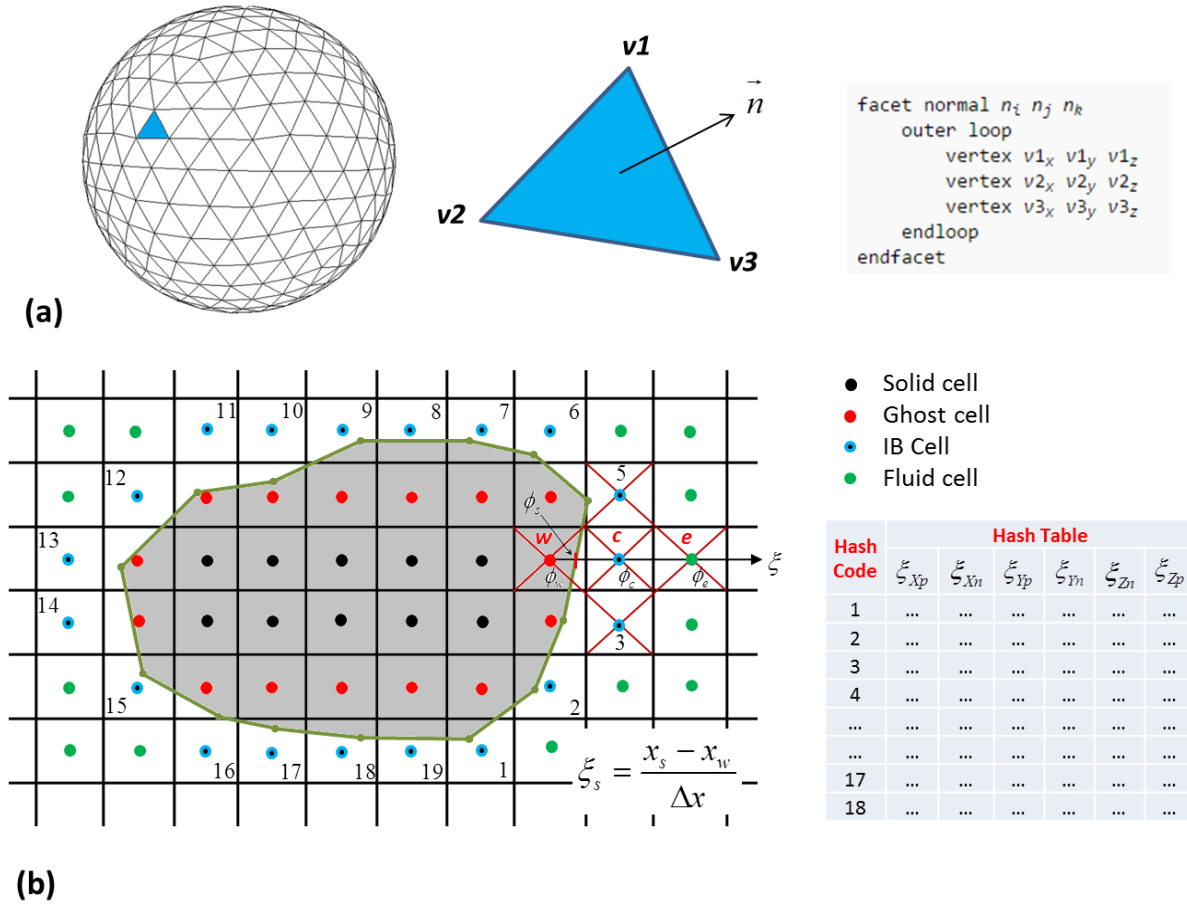

Figure1: Incorporation of the no-slip boundary condition at the surface of Immersed Body (IB). The IB is represented by a surface mesh of triangular elements (line segment in 2D). (a) The surface mesh for a sphere is made by triangular elements. In the STL file format the vertex of each triangle along with surface normal are stored to generate the solid surface. The surface normal always points outward. (b) The current IBM implementation procedure for irregular shaped bodies is schematically shown in 2D. At the beginning the computation cells are flagged either as 'solid cell', 'fluid cell' or 'IB-cell'. The intersection between the grid line and the IB surface i.e. ξ_s is required to implement the no-slip boundary condition and is stored in a Hash Table format. Hash codes are stored in the flag matrix for ghost cells and each of these point to a table where six ξ_s values are stored.

Table 1: Simulation settings for flow past periodic arrays of cubes.

Parameters	**Values**	**Unit**
Computational grid	60 × 60 × 60	(-)
Grid size	10^{-3}	m
Cube length	21×10^{-3}	m
Time step	10^{-4}	S
Fluid density	1.0	kg/m^3
Fluid viscosity	0.001	kg/(m s)
Applied pressure gradient	5.0	Pa/m

Fig. 2 shows the axial velocity profile normalized by the maximum velocity at the mid-plane (perpendicular to the flow) for the two different cases. The obtained superficial velocities respectively are 1.7802 m/s and 1.7788 m/s. A negligible difference is found between the two cases (deviates only 0.08%), which indicates a proper implementation. The slight deviation can be attributed to different spatial resolution per unit length of the cube and boundary effects.

Flow past periodic static arrays of spheres

For the creeping flow regime, Zick and Homsy (1982) proposed a semi-analytical method to obtain the drag acting on spherical particles for simple periodic arrays and tabulated the dimensionless drag force (F) as a function of porosity (ε).

$$F = \frac{F_{f \to s}}{3\pi\mu u_s d_p} \qquad \textbf{(12)}$$

where $F_{f\to s}$ is the total force exerted on a single particle, u_s represent superficial velocity, and d_p is the diameter of the particle. It provides a very good test case to validate the present model.

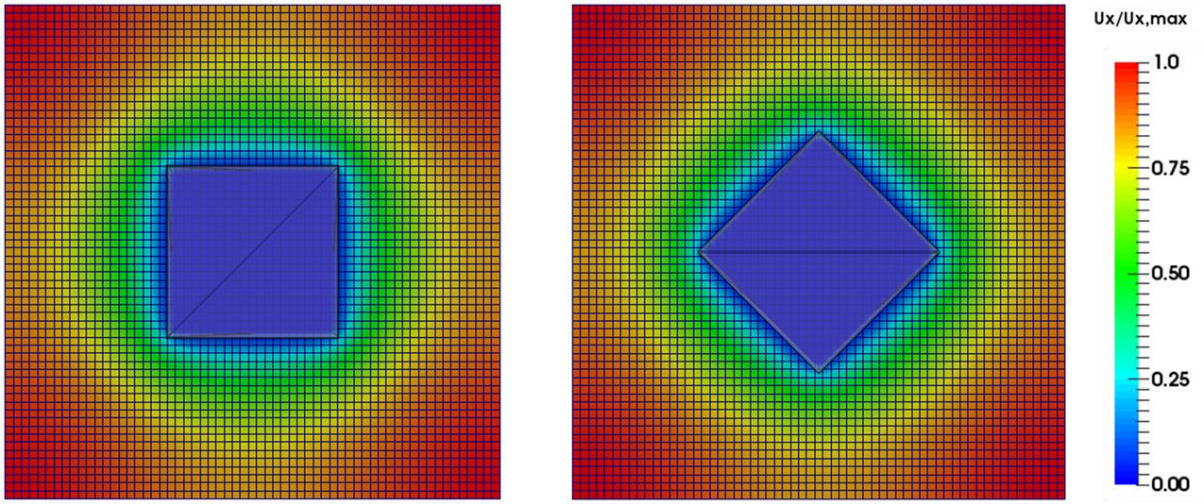

Figure 2: Axial velocity contours at the mid-plane (perpendicular to the flow direction) for flow past a periodic array of cubes. To check the current implementation the surface mesh for cubes of different orientation (rotated 45^0 parallel to the flow direction) are used and placed in same periodic computational domain. A constant pressure gradient is applied and for proper implementation it is expected to obtain same flow field. Superficial velocities are calculated and a very negligible difference about 0.08% is found which is due to the different special resolution per unit length of the cube and boundary effects.

By changing the diameter of the sphere the solid volume fraction of system is varied and using periodic boundary conditions the drag force can be obtained under fully developed flow conditions. The fluid is accelerated from rest by applying a constant body force or an average pressure gradient in the x-direction. The magnitude of the pressure gradient is chosen in such a way that the flow field still remains in the creeping flow regime (Re << 1). Two different Eulerian grids namely (G1 and G2) have been used, whereas for both cases the same very fine surface mesh for the sphere was used [Fig. 3]. The simulation settings are tabulated in the Table 2.

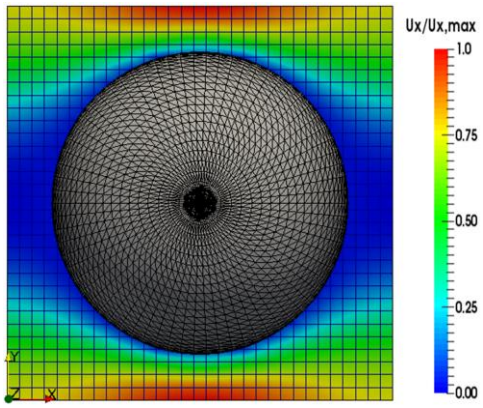

Figure 3: Creeping flow past a simple cubic packing of spheres ($1 - \varepsilon = 0.215$): single sphere in the periodic computational domain. Contours of axial velocity normalised by the maximum velocity are shown at the mid-plane (parallel to flow), along with the triangular surface mesh.

Table 2: Simulation settings for flow past static arrays of spheres at creeping flow regime.

Parameters		**Values**	**Unit**
Computational grid:	G1	$32 \times 32 \times 32$	(-)
	G2	$64 \times 64 \times 64$	(-)
Grid size:	G1	10^{-3}	m
	G2	5×10^{-4}	m
Time step		10^{-4}	S
Fluid density		1.0	kg/m^3
Fluid viscosity		0.05	kg/(m s)

In the Fig. 3, velocity contours for a typical simulation case is shown along with STL surface mesh for sphere and background Eulerian mesh.

Table 3 summarizes the different simulation cases. The average non-dimensional drag is tabulated as a function of the solid volume fraction ($1- \varepsilon$) and can be compared with analytical results of Zick and Homsy (1982). A very good match is found between the analytical and current numerical result.

FLOW THROUGH A TYPICAL RANDOM FOAM STRUCTURE

After developing and validating the numerical code the attention has been turned to its application for simulating flow past a typical random foam structure. In Fig. 4(a) the surface mesh for a typical section of the foam is shown. It is created using 3D image data generated from a Micro-CT (computed tomography) scan of an actual foam geometry [https://grabcad.com/library/auxetic-foam-sample-1]. The section of the foam is chosen such that the main morphological properties like porosity ($\varepsilon = 0.925$) and the specific surface area ($S_a = 116.6\ m^{-1}$) remain in a limit of ±0.5% compared to the large sample.

Table 3: Comparison of the current numerical results with the semi-analytical results for Stokes flow through simple cubic arrays of spheres.

$1-\varepsilon$	d_p (m)	F(-) Present	F(-) Zick and Homsy (1982)	$\Delta F/F$ (%)
0.524	0.0320	G1: 41.873	42.14	0.63
		G2: 41.866		0.65
0.449	0.0304	G1: 27.794	28.10	1.09
		G2: 27.802		1.06
0.343	0.0278	G1: 15.392	15.40	0.05
		G2: 15.381		0.12
0.215	0.0238	G1: 7.415	7.44	0.36
		G2: 7.403		0.52
0.126	0.0199	G1: 4.324	4.29	-0.74
		G2: 4.317		-0.58
0.064	0.0159	G1: 2.821	2.81	-0.41
		G2: 2.731		-0.73
0.027	0.0119	G1: 2.011	2.01	-0.13
		G2: 2.009		-0.03

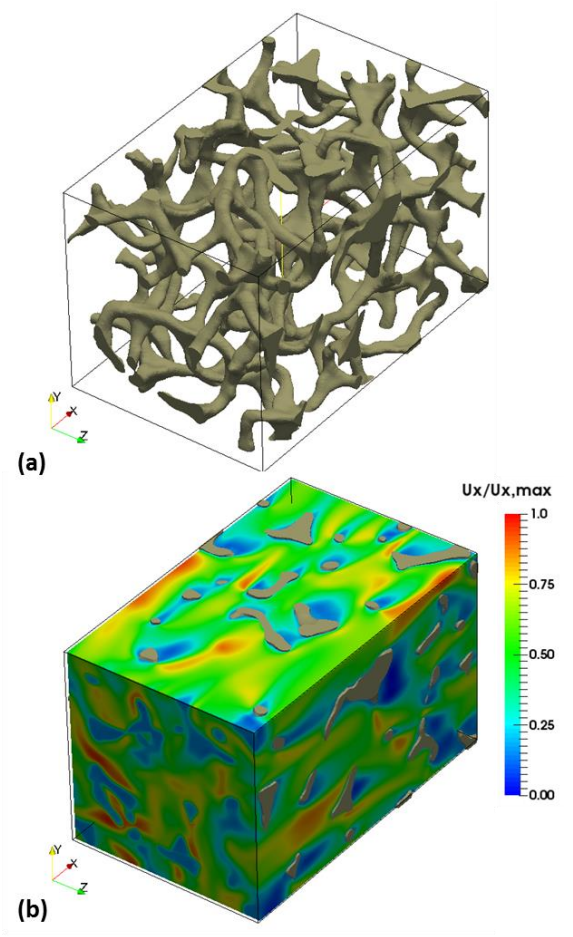

Figure 4: (a) Real foam sample in CFD domain. Periodic boundary condition is applied only in the x-direction, whereas free-slip boundaries are used on all other boundaries (b) Normalized x-velocity contours for the case of Re ~ 40.

The intention here is to find the pressure drop as a function of superficial velocity (Re) for a fully developed flow condition. A periodic boundary condition is applied in the x-direction , whereas free-slip boundaries are used on all other boundaries. As the foam sample is not periodic, we apply two empty layers of Eulerian grid cells both in the +x and –x direction beyond the physical extension of the foam. As a result numerically it represents an infinite domain in the x-direction where a number of same foam blocks are placed (in the x-direction) adjacent to each other with a small gap of four computational cell between them. The current scenario is not very different in comparison with flow through a continuous foam sample, considering the randomness in the foam geometry. Similar to previous cases the flow field is generated by applying a constant pressure gradient in the x-direction. By changing its magnitude, Re is varied from creeping flow regime to a value of Re ~ 40, where Re is calculated based on superficial velocity (u_s) and the characteristic length scale, which is chosen based on the equivalent spherical diameter as,

$$d_p = \frac{6V_s}{A_s} \tag{13}$$

V_s and A_s are the solid volume and surface area in the domain respectively. To check grid independence three Eulerian grids (G1, G2 & G3) of different resolution are used and drag forces are compared for Re ~ 40. The simulation settings are tabulated in Table 4. It is found that the G2 grid is sufficiently fine to have grid independent results [Table 5].

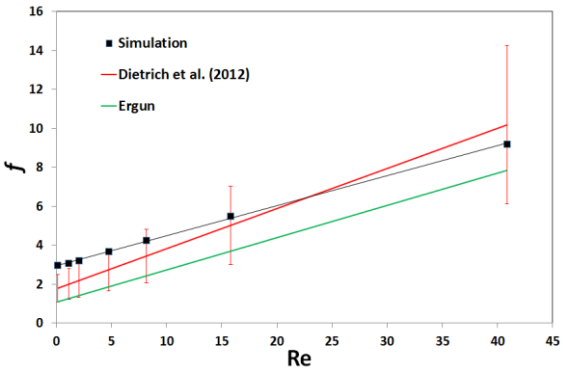

Figure 5: Non-dimensional pressure drop (f) as a function of Re for this particular foam sample ($\varepsilon = 0.925$).

To obtain the pressure drop correlation we start from the ansatz form of pressure drop:

$$\frac{\Delta p}{l} = a\frac{\mu}{d_p^2}u_s + b\frac{\rho}{d_p}u_s^2 \tag{14}$$

where actually $a = g(\varepsilon)$ and $b = h(\varepsilon)$**.** Note that since we are using a single foam sample with a single value of ε, for our case a and b are constant. The pressure drop will be normalized using viscous scale, and as a result the friction factor f , in our case is defined as:

$$f = \frac{\Delta p / l}{\mu u_s / d_p^2} = a + b\,\mathrm{Re} \tag{15}$$

Similar non-dimensional form for Ergun equation reads,

$$f_{Ergun} = \frac{150(1-\varepsilon)^2}{\varepsilon^3} + \frac{1.75(1-\varepsilon)}{\varepsilon^3}\,\mathrm{Re} \qquad \textbf{(16)}$$

In Fig. 5 the numerically obtained friction factor is plotted as a function of Re. By curve fitting for this specific foam sample (ε = 0.925) we found that $a = 2.9363$ and $b = 0.1545$. Similarly using Eq. 16 one can obtain $a_{Ergun} = 1.066$ and $b_{Ergun} = 0.1658$, which indicates that in this range of Re the Ergun correlation underpredicts the pressure drop. The current simulations results are also compared with existing experimental correlation. Dietrich et al. (2012) reviewed experimental pressure drop correlation/ values by about 20 authors including around 100 different foam samples and proposed a pressure drop correlation. It consists of more than 2500 experimental values and most of the data point lies within an error range of ±40% of the proposed pressure drop correlation. In the current non-dimensional form of representation it reads:

$$f_{Dietrich} = \frac{247.5(1-\varepsilon)^2}{\varepsilon^3} + \frac{2.175(1-\varepsilon)}{\varepsilon^3}\,\mathrm{Re} \qquad \textbf{(17)}$$

Hence, $a_{Dietrich} = 1.759$ and $b_{Dietrich} = 0.2061$. Fig. 5 shows that for moderate Re the present simulation results lie well within the error range of the experimental correlation.

Table 4: Simulation settings for flow past a section of a typical solid foam sample.

Parameters		Values	Unit
Computational grid:	G1	180 × 150 × 150	(-)
	G2	320 × 200 × 200	
	G3	480 × 300 × 300	
Grid size:	G1	3.125×10^{-4}	m
	G2	2.34375×10^{-4}	
	G3	1.5625×10^{-4}	
Time step		10^{-5}	s
Fluid Density		10.0	kg/m^3
Fluid Viscosity		0.003	kg/(m s)

Table 5: Grid independence study at Re ~ 40. Calculated friction factor (*f*) and its deviation from the finest grid (G3).

Parameters		Values	Deviation *(%)*
f :	G1	9.39639	2.67
	G2	9.19439	0.47
	G3	9.15177	0.00

CONCLUSIONS

A sharp implicit Immersed Boundary Method (IBM) for complex geometries is developed to study the hydrodynamics for flow through a random open-cell solid foam at pore-scale level. The immersed body can be incorporated as a triangulated surface mesh on a Eulerian Cartesian background mesh. The method provides a versatile and accurate framework for flow through complex geometries. Surface meshes for ordered bodies are created and used to check the accuracy of the current simulation framework. The simulation results match well with existing literature data. Finally the current IBM based CFD model is used to simulate flow through a typical solid foam structure. A pressure drop correlation as a function of Re is derived for this typical foam structure. It is found that the Ergun correlation underpredicts the pressure drop. However for moderate Re the simulation results lie well within the error range of experimental correlation [Dietrich et al. (2012)] composed using a large number of foam sample. The internal pore structure of solid foam influence significantly its hydrodynamic behavior and the present method is very helpful for accurate prediction of pressure drop for a specific foam sample.

REFERENCES

DEEN, N. G., KRIEBITZSCH, S. H., VAN DER HOEF, M. A., & KUIPERS, J. A. M., (2012), "Direct numerical simulation of flow and heat transfer in dense fluid–particle systems", Chem. Eng. Sci. 81, 329-344.

DEEN, N. G., PETERS, E. A. J. F., PADDING, J. T., & KUIPERS, J. A. M., (2014), "Review of direct numerical simulation of fluid–particle mass, momentum and heat transfer in dense gas–solid flows", Chem. Eng. Sc., 116, 710-724.

DIETRICH, B., (2012), "Pressure drop correlation for ceramic and metal sponges", Chem. Eng. Sci. 74, 192-199.

EDOUARD, D., LACROIX, M., Huu, C. P. and LUCK, F., (2008), "Pressure drop modeling on solid foam: State-of-the art correlation", Chem. Eng. J. 144 (2), 299-311.

MAGNICO, P., (2009), "Analysis of permeability and effective viscosity by CFD on isotropic and anisotropic metallic foams", Chem. Eng. Sci. 64, 3564-3575.

MITTAL, R. & IACCARINO, G., (2005), "Immersed boundary methods", Annu. Rev. Fluid Mech. 37, 239–261.

PETRASCH, J., MEIER, F., FRIESS, H., STEINFELD, A., (2008), "Tomography based determination of permeability, Dupuit-Forchheimer coefficient, and interfacial heat transfer coefficient in reticulate porous ceramics", Int. J. Heat Fluid Flow 29, 315-326.

ZICK, A. A., & G. M., HOMSY, (1982), "Stokes flow through periodic arrays of spheres", J. of fluid mech. 115, 13-26.

