Is Lagrangian weight crucial in the direct forcing immersed boundary method?

Kun Zhou¹, Zhou Ding and Ke Sun
The State Key Laboratory of Refractories and Metallurgy, Wuhan University of Science and Technology, Wuhan, China

¹Email: zhou.kun@wust.edu.cn

Abstract. Particle resolved direct numerical simulation (PR-DNS) is one of the most powerful research tools for particle laden flows. Among a few most popular PR-DNS methods, the direct forcing immersed boundary method (DF-IBM) has obtained great success and has been adopted in various simulations of rigid particulate flows. Within DF-IBM, Eulerian and Lagrangian frameworks are used to depict the continuum and dispersed phases, respectively. Interpolation between the two frameworks is accomplished through a discrete delta function. It is generally believed that a Lagrangian weight attached to each Lagrangian marker, which is distributed on a particle’s surface, needs to be carefully chosen. To be more specific, the Lagrangian weight is supposed to match the local Eulerian cell. The matching requirement is not trivial for non-uniform Eulerian mesh or irregular shaped particles. There are various methods developed to calculate the Lagrangian weight. Here, the Lagrangian weights in a few testing cases have been calculated following two intuitively “straightforward” methods. It turns out there are substantial discrepancies in the Lagrangian weights obtained from different methods. However, further numerical examples demonstrate that such discrepancies have negligible effects on the flow dynamics. So a natural question is raised: Is Lagrangian weight crucial in the direct forcing immersed boundary method? A negative answer to this question is suggested. More detailed analysis is provided in a forthcoming paper.

1. Introduction
Particle laden flows are ubiquitous in both nature and industry processes. According to the relative size of particle to the smallest characteristic flow length scale, the simulation techniques can generally be grouped into three categories [1], Eulerian-Eulerian method, point-particle method, and particle-resolved method. When particles are small and usually of very high number density (such as aerosol particles), the dispersed phase can be readily treated as a continuum in the Eulerian framework. If particles become larger while still much smaller than the smallest flow scale (e.g., the Kolmogorov scale in turbulence), the momentum coupling between fluid and particles needs better treatment than that in the Eulerian-Eulerian method. Point-particle method is the most appropriate choice under such condition. If particles are even larger, the interface between fluid and particles is required to be resolved, hence the so-called particle-resolved method. From another perspective, with regard to the particle loading rate (volume fraction or mass fraction), fluid and particles may be one-way coupled (particles are assumed passive to the flow field), two-way coupled (fluid-particles coupling), and four-way coupled (coupling among particles along with fluid-particles coupling).

Among various particle-resolved methods [2], immersed boundary method (IBM), invented by Peskin [3] during simulating the flow around heart valves (coupling between fluid and elastic bodies),
has received much attention and has been developed to simulate various particulate flows with great success. The direct forcing immersed boundary method (DF-IBM) [4] is the most popular to simulate rigid particulate flows. Within DF-IBM, Eulerian and Lagrangian frameworks are used to depict the continuum and dispersed phases, respectively. Uniform Eulerian mesh is usually adopted for the continuum phase (which occupies the whole domain possessed by fluid and particles). Particles are tracked through the Lagrangian framework. Each particle is discretized by Lagrangian markers on its surface, and such markers work as agents to exchange momentum between the Eulerian and Lagrangian frameworks. From the very beginning, it is suggested [4] that a Lagrangian weight, attached to each Lagrangian marker, needs to be carefully chosen so as to interpolate the momentum correctly between the Eulerian and Lagrangian frameworks. To be more specific, the Lagrangian weight is supposed to match the local Eulerian cell. It is generally believed such a matching is necessary for accurate simulation. Just recently, Tschisgale et al [5] reiterated such “straightforward” requirement through a thought experiment. The matching requirement is not trivial for non-uniform Eulerian mesh or irregular shaped particles. Pinelli et al [6] developed a polynomial interpolation scheme to deal with non-uniform Eulerian mesh, and also suggested to use a recovery condition for the interaction force to determine the Lagrangian weight. Akiki and Balachandar [7] used a collocation method through matching the known Stokes solution for a sphere to determine Lagrangian weight.

Here, through calculating the Lagrangian weights in a few testing cases, using two different intuitively “straightforward” methods, it is shown that there are substantial discrepancies between the weights obtained from different methods. However, in simulations of uniform ambient flow around a fixed sphere with very different weights, it is found that the drags on the sphere are nearly identical. Through these concrete numerical examples, this paper raises the problem “Is Lagrangian weight crucial in direct forcing immersed boundary method?” A negative answer to this question is suggested [8]. More detailed analysis is provided in a forthcoming paper.

2. Methodology

2.1. Immersed boundary method

The DF-IBM developed by Uhlmann [4] has been used in many simulations of particle laden flows. The method adopts an Eulerian-Lagrangian framework to depict the fluid and particles. The whole domain possessed by fluid and particles is discretized by the Eulerian mesh (generally uniform). The momentum equation in the Eulerian framework is the Navier-Stokes equation with additional source term coming from immersed particles

\[
\frac{du}{dt} = -u \cdot \nabla u - \frac{1}{\rho_f} \rho \nabla p + v(\nabla^2 u) + f_{IBM} \equiv rhs + f_{IBM}
\]

(1)

where \( u, p, \rho_f \), and \( v \) are the fluid velocity, pressure, density, and kinematic viscosity, respectively, and \( f_{IBM} \) is the force term to take into account the fluid-particle interaction.

For a particle, the translational dynamic equation is [5]

\[
m_p \frac{du_p}{dt} = -\int_L \rho_f f_{IBM} dV + \oint_{\Gamma^-} \rho_f u d\Gamma + V_p (\rho_p - \rho_f) g
\]

(2)

where \( m_p, u_p, \rho_p \), and \( V_p \) are the mass, velocity, density, and volume of the particle, respectively, and \( g \) is the gravitational acceleration. The integral domain \( L \) is the support of the force field \( f_{IBM} \), which is assumed to be a thin layer, and \( \Gamma^- \) is the inner region of the particle, which is also occupied by the fluid simultaneously.

Define the preliminary velocity \( \tilde{u} \) as

\[
\tilde{u} = u^{n-1} + \int_{t_{n-1}}^{t_n} rhs dt
\]

(3)

and its corresponding velocity in Lagrangian frame \( \tilde{U} \) is
\[ \bar{U} = \sum \tilde{u} \delta_h h^3 \]  

where \( \delta_h \) is the vector discrete delta function, and \( h \) is the Eulerian grid size (assumed to be uniform). A common practice for the scalar delta function is

\[ \delta_h(r) = \frac{1}{h} \psi(r) \]  

where \( \psi(r) \) is a piecewise function, and \( r = (x - X)/h \) (where \( x \) and \( X \) are the coordinates of Eulerian and Lagrangian points, respectively). The vector delta function is obtained through the direct product of the scalar function in each dimension (\( x, y, \) and \( z \)). Let \( u_f \) denote the fluid-particle interface velocity, which is determined through the rigid body motion of particle

\[ u_f = u_p + \omega_p \times r \]  

where \( \omega_p \) is the particle angular velocity, and \( r \) is the position vector of particle surface point with respect to the center of mass of the particle. Define the Lagrangian force \( F \) as

\[ F = \frac{u_f - \bar{U}}{\Delta t} \]  

where \( \Delta t \) is the discrete time step, and let

\[ f_{IBM} = \sum_i F \delta_h W_i \]  

where \( N \) is the total number of Lagrangian markers, and \( W_i \) is the weight assigned to marker \( I \).

The computational code used here adopts a spectral method for a canonical channel domain. A three order Runge-Kutta method is used to integrate with time. The additional force term in the Navier-Stokes equation is evaluated in the physical space as depicted above. The details of the method can be found in Reference [7].

2.2. Determination of the Lagrangian weight

In immersed boundary method, the force density \( F \) in Lagrangian frame can be obtained through interpolation the corresponding field \( f \) in Eulerian frame

\[ F(X_i) = \sum_i f(x_i) \delta_h (x_i - X_i) w_i \]  

where \( n \) includes all Eulerian grid points \( x_i \), \( w_i \) is the weight associated to \( x_i \), which is always taken as the Eulerian grid cell volume \( h^3 \) for uniform grid (or area \( h^2 \) for 2D domain). The kernel \( \delta_h \) satisfies

\[ \sum_i \delta_h (x_i - X) h^3 = 1 \]  

\[ \sum_i (x_i - X) \delta_h (x_i - X) h^3 = 0 \]  

On the other hand, the distribution operation from Lagrangian to Eulerian is

\[ f(x_i) = \sum_i F(X_i) \delta_h (x_i - X_i) W_i \]  

where \( N \) refers to the total number of Lagrangian markers, and \( W_i \) is the weight assigned to the marker \( X_i \). Due to the kernel properties (Equations (10) and 11)), we have

\[ \sum_i f(x_i) w_{-i} = \sum_i F(X_i) W_i \]  

\[ \sum_i x_i \times f(x_i) = \sum_i X_i \times F(X_i) W_i \]
i.e., the total force and torque are conserved during the transfer between Eulerian and Lagrangian frameworks. It is worth pointing out that the conservation is independent of the choice of the weight \( W \), only if the Eulerian grid is uniform and the corresponding weight \( w \) is constant.

Ideally, after the distribution and interpolation, a distribution recovers itself, i.e.,

\[
F(X_j) = \sum_{i=1}^{N} F(X_i) A_{ij} W_i
\]

where \( A_{ij} = \sum_{t} \delta_R(x_t - x_i) \delta_R(x_t - x_j) w_t \) is defined by the last step. To group the equations for all \( F(X_i), (I = 1, \ldots, N) \), then

\[
\begin{bmatrix}
A_{11} & A_{12} & \cdots & A_{1N} \\
A_{21} & A_{22} & \cdots & A_{2N} \\
\vdots & \vdots & \ddots & \vdots \\
A_{N1} & A_{N2} & \cdots & A_{NN}
\end{bmatrix}
\begin{bmatrix}
W_1 \\
W_2 \\
\vdots \\
W_N
\end{bmatrix} =
\begin{bmatrix}
F(X_1) \\
F(X_2) \\
\vdots \\
F(X_N)
\end{bmatrix}
\]

(15)

By requiring that the weight \( W \) matrix is independent of the actual force distribution \( F \), Pinelli et al [6] suggested to use the condition

\[
|AW - I| = 0
\]

(16)

where \( I \) is the identity matrix, and \( |\cdot| \) denotes the determinant. In fact, in a mathematically more precise way, Equation (17) is the necessary and sufficient condition that the eigenvalue equation \( BF = \lambda F \) (where \( B = AW \)) has nonzero solution. Specifically, the eigenvalue \( \lambda \) is 1 here. The condition Equation (17) does not guarantee Equation (16) is satisfied for any force distribution \( F \), but only for the eigenvector corresponding to the eigenvalue \( \lambda = 1 \). Equation (17) is not enough to determine the weight \( W \). So it was further suggested [6] to set

\[
\begin{bmatrix}
A_{11} & A_{12} & \cdots & A_{1N} \\
A_{21} & A_{22} & \cdots & A_{2N} \\
\vdots & \vdots & \ddots & \vdots \\
A_{N1} & A_{N2} & \cdots & A_{NN}
\end{bmatrix}
\begin{bmatrix}
W_1 \\
W_2 \\
\vdots \\
W_N
\end{bmatrix} =
\begin{bmatrix}
1 \\
1 \\ \vdots \\
1
\end{bmatrix}
\]

(18)

The above equation can be seen from another perspective. The distribution \( F \) makes up a \( N \) dimensional space, and it has \( N \) independent coordinate vectors, i.e., \( [F(X_1), 0, \cdots, 0], [0, F(X_2), 0, \cdots, 1], \cdots, [0, 0, \cdots, 0, F(X_N)] \). In Equation (16), after replacing \( N \) with its \( L \) coordinate vectors respectively, it gives \( N \) equations, which is exactly the above Equation (18). Very similar idea has been used to determine the weights in non-uniform Eulerian grid [7].

In order to have meaningful solution for \( W, A \) is required to be well behaved. When the Lagrangian grid is much denser than the Eulerian grid, \( A \) becomes bad conditioned, and the solution for \( W \) becomes oscillatory, even negative weights may happen (see below numerical examples). This Gibbs-like phenomenon was observed in their original derivation of the method [6]. Pinelli et al [6] also anticipated that singularity will happen when Lagrangian grid is much coarser (They did not investigate it in detail.). In fact, the above linear system is rather stable when the Lagrangian grid is not too much denser than the Eulerian grid, whatever how coarser it is. Details will be discussed later through numerical examples.

When the Lagrangian grid is uniform, a rather robust scheme can be used to tackle the oscillatory solution problem. The matrix element \( A_{ij} \) depicts the correlation between two Lagrangian markers through the kernel function. If the Lagrangian markers are uniformly distributed, it is straightforward to check that \( A \) is a circulant matrix, i.e., the matrix is fully specified by the first row, all other rows are obtained through cyclic permutations of the first row. For example, if the Lagrangian grid size matches the Eulerian grid, then only three Lagrangian markers are correlated for the kernel Equation (5), and \( A \) looks like...
\[ A = \begin{bmatrix} A_1 & A_2 & 0 & \cdots & A_3 \\ A_2 & A_1 & A_3 & \cdots & 0 \\ \vdots & \ddots & A_1 & \ddots & \vdots \\ 0 & \cdots & \cdots & A_2 & \cdots \\ A_3 & 0 & 0 & \cdots & A_1 \end{bmatrix} \]  

where \( A_1, A_2, \) and \( A_3 \) denote the three non-zero elements. Since the Lagrangian grid is uniform, the corresponding weights should all be constant \( W_c \). The constant can be directly determined through \((A_1 + A_2 + A_3)W_c = 1\). Generally, \( W_c \) is just the reciprocal of the summation of elements in a row. In practice, it is often very difficult to distribute the markers uniformly. However, there are ways to generate quasi uniform distribution. When the distribution is not exactly uniform, \( A \) is no longer a circulant matrix. Since the kernel function is continuous, it guarantees that the actual \( A \) only differs from the ideal circulant matrix slightly. Under such condition, \( W_c \) can be determined as \( W_c = N / \text{sum}(A) \), which is just the averaged result over each row.

3. Numerical examples

In section 2.2, a “straightforward” distribution-interpolation process is introduced to determine the weight. In this section, a few numerical examples are provided to see how such distribution-interpolation process actually perform. Finally, simulations of a still sphere sitting in a channel flow with very different weights are presented.

3.1. Case: 1D

First, a single Lagrangian marker in a uniform 1D Eulerian grid is considered. For simplicity, the Lagrangian weight \( W \) is normalized by the Eulerian weight. For a Lagrangian force distribution \( F \) after the distribution-interpolation process, the recovered value is \( F^* = 0.5W F \). Only if \( W = 2 \), the force may recover itself. This result is unexpected. During the distribution process, no force is “lost”. However, the interpolation only gives half back! The result can be easily extended to higher dimensions, i.e., \( W = 4 \) for 2D and \( W = 8 \) for 3D are the required conditions for fully recovering.

Next, the case for two markers which are separated by \( \Delta X \) (normalized by the Eulerian grid size) is considered. The two weights calculated from Equation (18) are equal, as shown in Figure 1. When increasing the separation, the weight increases from 1 to 2, until the two markers are isolated (degenerated to the single marker case). As discussed in the last section, even when the weight \( W \) is determined through the recovery condition, an arbitrary distribution is not guaranteed to recover. Suppose \( F(X_1) = 1 \), and \( F(X_2) = 1 + dF \). Figure 2 shows that the relative errors \([F^*(X_1) - F(X_1)]/F(X_1)\) (dash line), and \([F^*(X_2) - F(X_2)]/F(X_2)\) (solid line), under varying \( \Delta X \). If \( F(X_1) \) and \( F(X_2) \) are equal, or the separation \( \Delta X \) is large enough, there is no error during the recovering. Otherwise, the error increases as the discrepancy on \( F \) between two markers increases. Denser Lagrangian markers distribution also incurs larger errors. For more than two Lagrangian markers, the weight may become negative as shown in Figure 3. But negative weight is believed make the fluid-particle coupling unstable, which has never been adopted in the literature.

To conclude, the distribution-interpolation process is generally not ideal, in other words, it is not able to recover the original distribution completely. The local error may be very large. It is impossible to choose a weight function that guarantees the recovery for all distribution. The good news is that the total force is fully recovered.

3.2. Case: 2D

The weight for a circle in a uniform 2D Eulerian grid is considered. The circle is discretized evenly with markers separated by \( \Delta X \) (arc length). The weight determined by the recovery condition is shown in Figure 4. When the separation is big enough (out of the support domain of the kernel function), the weight approaches 4. Specifically, when the separation is 1, the weight is around 2. This is in big
contrast to what suggested by Uhlmann [4], where the weight is set to 1 when the size of Lagrangian grid matches the Eulerian grid.

3.3. Case: 3D

The weight for a sphere in a uniform 3D Eulerian grid is considered. The sphere is discretized nearly uniformly with an iteration method [10]. The markers separation $\Delta X$ is calculated through $N\Delta X^2 = \pi D^2$, where $D$ is the diameter of the sphere. Figure 5 shows the weight with respect to the separation. Similar to the 2D case, the weight determined through the recovery condition is around 2 for $\Delta X = 1$, which is nearly two times that suggested by Uhlmann [4]. For $W = 1$, the separation should be $\Delta X = 0.71$.

3.4. A sphere in channel flow

A still sphere with diameter $D$ is in the middle of a channel. The channel has dimensions $x \times y \times z = 14D \times 50D \times 8D$, which is discretized with mesh of $210 \times 750 \times 141$. The flow is along the $y$ direction. Both the top and bottom wall are set to move along the $y$ direction. No pressure gradient is applied along the flow direction. The Reynolds number is set to $Re = 100$.

![Figure 1. Weight (two equal) with respect Lagrangian markers.](image1)

![Figure 2. Relative errors during the recovering of distribution and interpolation for two markers (1D). Dash and solid lines are for $F(X_1)$ and $(X_2)$, respectively. $dF$ is the discrepancy between $F(X_1)$ and $F(X_2)$.](image2)

![Figure 3. Weights for three markers with respect Lagrangian markers separation for (1D).](image3)

![Figure 4. Weight with respect to marker separation for a circle in uniform 2D Eulerian grid.](image4)
Both the recovery method and the one cell thickness method [4] to determine the weight have been tried. The number of markers has been varied from 258, 696 to 2328. For \( N = 696 \), the Lagrangian grid size matches the Eulerian grid. Figure 6 shows the drag coefficients corresponding to these 6 settings. All approach to the well adopted empirical value \( C_d = 1.1 \) for \( N = 696 \), which is based on the empirical formula [11]

\[
C_d = \frac{24}{Re} (1 + 0.1935Re^{0.6305})
\]  

(20)

The result is rather perplexing at first sight, given the fact that the weights determined by the two methods differ as large as 2 times. Further detailed analyses show that the Lagrangian weight is actually a relaxation factor. The ideal condition of zero velocity difference between fluid and particle at a Lagrangian marker’s position is not achievable. The Lagrangian weight controls how quick such ideal condition is approached. The larger the weight is, the faster the approaching rate is. However, numerical stability condition limits the maximum Lagrangian weight possible.

4. Conclusions

Direct forcing immersed boundary method is very efficient and popular in the simulations of rigid particulate flows. It is generally believed that the Lagrangian weight should match the local Eulerian cell so as to achieve accurate results, based on “straightforward” intuition about the force balance between fluid and particles. Since the matching requirement is not trivial for non-uniform mesh or irregular shaped particles, there are various endeavors to calculate the weights “correctly” through different methods.

Through a serial of numerical examples, it is shown that there are substantial discrepancies on the weights obtained from different methods. However, further numerical examples show that the discrepancies on the weights have negligible effects on the fluid dynamics. These simple simulation results explain the perplexity why such “obvious” discrepancies on the weights obtained from different methods are unnoticed for long time, since all methods are able to give good results irrespective of the actual weights used. So the answer to the title question is that Lagrangian weight is not crucial in the direct forcing immersed boundary method. More pertinent questions, such as What is the actual role of the Lagrangian weight? Is there optimal choice for the weight? etc., are answered in a forthcoming companion paper.
Acknowledgments
This work was supported by the NSF of China (Nos. 115722741, 1602179).

References
[1] Balachandar S and Eaton J K 2010 Annu. Rev. Fluid Mech. 42 111-133
[2] Maxey M 2017 Annu. Rev. Fluid Mech. 49 171-193
[3] Peskin C S 1972 J. Comput. Phys. 10 252-271
[4] Uhlmann M 2005 J. Comput. Phys. 209 448-476
[5] Tschisgale S, Kempe T and Fröhlich J 2017 J. Comput. Phys. 339 432-452
[6] Pinelli A, Naqavi I Z, Piomelli U and Favier J 2010 J. Comput. Phys. 229 9073-9091
[7] Akiki G and Balachandar S 2016 J. Comput. Phys. 307 34-59
[8] Zhou K and Balachandar S 2018 Investigation of direct forcing immersed boundary method, in APS-DFD: Atlanta, Georgia
[9] Roma A M, Peskin C S and Berger M J 1999 J. Comput. Phys. 153 509-534
[10] Persson P-O and Strang G 2004 SIAM Rev. 46 329-345
[11] Clift R, Grace J R and Weber M E 1978 Bubbles, Drops, and Particles (New York: Academic Press)