A Gaussian-Like Immersed-Boundary Kernel with Three Continuous Derivatives and Improved Translational Invariance

Yuan-Xun Bao[a,*], Jason Kaye[a], Charles S. Peskin[a]

[a]*Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street, New York, NY, 10012, USA*

Abstract

The immersed boundary (IB) method is a general mathematical framework for studying problems involving fluid-structure interactions in which an elastic structure is immersed in a viscous incompressible fluid. In the IB formulation, the fluid described by Eulerian variables is coupled with the immersed structure described by Lagrangian variables via the use of the Dirac delta function. From a numerical standpoint, the Lagrangian force spreading and the Eulerian velocity interpolation are carried out by a regularized, compactly supported discrete delta function, which is assumed to be a tensor product of a single-variable immersed-boundary kernel. IB kernels are derived from a set of postulates designed to achieve approximate grid translational invariance, interpolation accuracy and computational efficiency. In this note, we present a new 6-point immersed-boundary kernel that is C^3 and yields a substantially improved translational invariance compared to other common IB kernels.

Keywords: Immersed boundary method, fluid-structure interaction, discrete delta function, immersed-boundary kernel, force spreading, velocity interpolation, translational invariance

1. Introduction

The immersed boundary (IB) method was originally proposed by Peskin to study flow patterns around heart valves [1]. In the IB formulation, a viscous incompressible fluid described by Eulerian variables is assumed to occupy the entire domain, which contains an immersed structure, described by Lagrangian variables, that moves with the fluid and exerts a force on the fluid. In the spatially discretized setting, the fluid domain is represented by a uniform Eulerian grid and the immersed structure is configured as a collection of Lagrangian points or markers. The IB kernel plays a key role in communicating between the Eulerian and Lagrangian grids by spreading applied forces to the fluid and interpolating Lagrangian marker velocity. There are three main criteria for constructing an ideal IB kernel: grid translational invariance, interpolation accuracy and computational efficiency. It is a desirable property of an IB kernel to perform force spreading and velocity interpolation that are independent of the position of Lagrangian markers relative to the Eulerian computational grid. In this case, if the IB method were applied to a translation-invariant linear system like the Stokes equations on a periodic domain, the results would remain the same despite shifts in position of Lagrangian markers relative to the Eulerian grid [2]. There are functions that might serve as candidates for an IB kernel in terms of exact grid translational invariance; for example, the sinc function $\sin(r)/r$. However, the sinc function is not computationally efficient because its support is unbounded. In fact, as we discuss later, exact grid translational invariance is inconsistent with the assumption of compact support [2]. The process of constructing a computationally efficient IB kernel that simultaneously has good interpolation accuracy and translational invariance is non-trivial.

In the IB method, the 3D discrete delta function is assumed to be represented by a tensor product of a single-variable kernel $\phi(r)$,

$$\delta_h(\mathbf{x}) = \frac{1}{h^3}\phi\left(\frac{x_1}{h}\right)\phi\left(\frac{x_2}{h}\right)\phi\left(\frac{x_3}{h}\right), \tag{1.1}$$

*Corresponding author

Email addresses: billbao@cims.nyu.edu (Yuan-Xun Bao), jkaye@cims.nyu.edu (Jason Kaye), peskin@cims.nyu.edu (Charles S. Peskin)

where x_1, x_2, x_3 are the Cartesian components of $\mathbf{x}$ and h is the meshwidth. This representation is not essential, but it significantly simplifies the discussion, since the single-variable kernel $\phi(r)$ is the object of interest. We first require that $\phi(r)$ be continuous for all real r and have compact support, i.e., $\phi(r) = 0$ for $|r| \geq r_s$, where r_s is the radius of support. Continuity of ϕ is assumed in order to avoid sudden jumps in the interpolated velocity or applied force as Lagrangian markers move through the Eulerian grid. It turns out that most IB kernels are C^1 even though the higher regularity is not explicitly assumed. The reason for that is still a mystery, but higher regularity of the IB kernel is a nice feature to have in certain applications, such as the interpolation of derivatives or the spreading of a force dipole. Compact support of ϕ is required for computational efficiency.

The function $\phi(r)$ is also required a subset of the following moment conditions:

$$\begin{aligned}
&\text{(i)} && \text{Zeroth moment:} && \sum_j \phi(r-j) = 1, \\
&\text{(ii)} && \text{Even-odd:} && \sum_{j\,\text{even}} \phi(r-j) = \sum_{j\,\text{odd}} \phi(r-j) = \frac{1}{2}, \\
&\text{(iii)} && \text{First moment:} && \sum_j (r-j)\,\phi(r-j) = 0, \\
&\text{(iv)} && \text{Second moment:} && \sum_j (r-j)^2 \phi(r-j) = K, \text{ for some constant } K, \\
&\text{(v)} && \text{Third moment:} && \sum_j (r-j)^3 \phi(r-j) = 0.
\end{aligned}$$

The motivation of imposing moment conditions is well discussed in [2, 3]. Briefly, the zeroth moment condition implies that the total force is the same in the Eulerian and Lagrangian grids when δ_h is used for force spreading. The even-odd condition implies (i), and was originally proposed to avoid the "checkerboard" instability that may arise from using a collocated-grid fluid solver. Liu and Mori [3] generalized this condition to the so called "smoothing order" condition and showed that it has the effect of suppressing high-frequency errors and preventing Gibbs-type phenomena. Conservation of total torque relies on the first moment condition. Moreover, (i) and (iii) guarantee that a smooth function is interpolated with second-order accuracy when δ_h is used for interpolation. The second moment condition with $K = 0$ and the third moment condition are needed to derive kernels with a higher order of interpolation accuracy.

In addition to moment conditions, $\phi(r)$ is required to satisfy the sum-of-squares condition,

$$\sum_j (\phi(r-j))^2 = C, \text{ for some constant } C. \tag{1.2}$$

The sum-of-squares condition (1.2) is a weaker version of exact grid translational invariance,

$$\tilde{\phi}(r_1, r_2) = \sum_j \phi(r_1 - j)\,\phi(r_2 - j) = \Phi(r_1 - r_2), \text{ for all } r_1, r_2. \tag{1.3}$$

In other words, the coupling of $\phi(r)$ between any arbitrary two points r_1, r_2 is a function of $r_1 - r_2$ only. However, it can be shown that (1.3) is inconsistent with the assumption of ϕ being compactly supported [2]. The sum-of-squares condition does give some information about the coupling of ϕ, since it can be deduced from the Cauchy-Schwarz inequality that

$$\left|\tilde{\phi}(r_1, r_2)\right| = \left|\sum_j \phi(r_1 - j)\,\phi(r_2 - j)\right| \leq C, \text{ for all } r_1, r_2. \tag{1.4}$$

The inequality (1.4) guarantees that the coupling between two Lagrangian markers is strongest when the markers coincide, and furthermore (1.2) implies that the self-coupling is independent of the marker position.

In Table 1, we list some common IB kernels and the conditions they satisfy. The most widely used IB kernel is the standard 4-point kernel [2]. The standard 3-point kernel satisfies the zeroth moment condition but not the even-odd

IB Kernel	Support r_s	Even-Odd	0th Moment	1st Moment	2nd Moment	3rd Moment	Sum of Squares
Standard 3-point	1.5	✗	✓	0	✗	✗	$\frac{1}{2}$
Standard 4-point	2	✓	✓	0	✗	✗	$\frac{3}{8}$
Standard 6-point	3	✓	✓	0	0	0	$\frac{67}{128}$
New 6-point	3	✓	✓	0	$\frac{59}{60} - \frac{\sqrt{29}}{20}$	0	$\approx .326$

Table 1: Common immersed-boundary kernels and their properties.

condition. It was first introduced in an adaptive IB method using the staggered-grid discretization [4]. The standard 6-point kernel (with $K = 0$) satisfies all the moment conditions listed above [5]. It can be shown that the standard 6-point kernel interpolates cubic functions exactly and smooth functions with fourth-order accuracy. However, in our numerical experiments, its translational invariance is worse than that of the standard 4-point kernel. In terms of its defining postulates, our new 6-point kernel differs from the standard 6-point kernel only in the nonzero second-moment constant K (the sum-of-squares constant C is determined once K is fixed). The special choice of K given in (2.1) leads to a new 6-point kernel that is C^3 and significantly improves translational invariance compared with other IB kernels. Liu and Mori developed a MATLAB routine that automatically generates all the standard IB kernels as well as many others [3], but their code does not currently allow for the generalization to nonzero K.

2. A new 6-point kernel

Our new 6-point kernel satisfies the sum-of-squares condition (1.2) and the moment conditions (ii)-(v) (and therefore (i)) with a special choice of

$$K = \frac{59}{60} - \frac{\sqrt{29}}{20}. \tag{2.1}$$

The derivation of this kernel follows the same procedure as that of the standard 4-point kernel [2]. First, the sum-of-squares constant C can be expressed in terms of K by considering the special case $r = 0$. Next, by restricting r to the interval $[0, 1]$, we have 6 unknowns: $\phi(r-3), \phi(r-2), \phi(r-1), \phi(r), \phi(r+1), \phi(r+2)$ and 6 equations (the even-odd condition accounts for two equations). By expressing all the other unknowns in terms of $\phi(r-3)$ using (ii)-(v), we can solve for $\phi(r-3)$ from the quadratic equation determined by (1.2). The continuity assumption of ϕ is now used to select the appropriate root to piece together a continuous function, i.e., by setting $r = 0$, we select the root that gives $\phi(-3) = 0$. As mentioned earlier, ϕ being C^1 follows implicitly from our defining postulates, i.e., $\phi'(-3) = 0$. We have the freedom to choose K so that $\phi''(-3) = 0$, which uniquely determines the special value (2.1). Magically, analogous to the scenario in which we obtain $\phi \in C^1$ by assuming only continuity of ϕ, the resulting 6-point kernel is C^3. We emphasize that *only* the choice of K given by (2.1) makes $\phi(r) \in C^3$, and that this is the smallest value of K for which $\phi(r) \geq 0$ for all r. For other choices of K, ϕ is only C^1, and there are some choices of K for which ϕ does not exist. The formula for the new 6-point kernel is given by

$$\alpha = 28\,, \tag{2.2}$$

$$\beta(r) = \frac{9}{4} - \frac{3}{2}(K + r^2)r + (\frac{22}{3} - 7K)r - \frac{7}{3}r^3\,, \tag{2.3}$$

$$\gamma(r) = -\frac{11}{32}r^2 + \frac{3}{32}(2K + r^2)r^2 + \frac{1}{72}\left((3K-1)r + r^3\right)^2 + \frac{1}{18}\left((4-3K)r - r^3\right)^2\,, \tag{2.4}$$

$$\phi(r-3) = \frac{-\beta(r) + \mathrm{sgn}\left(\frac{3}{2} - K\right)\sqrt{\beta^2(r) - 4\alpha\gamma(r)}}{2\alpha}, \tag{2.5}$$

$$\phi(r-2) = -3\phi(r-3) - \frac{1}{16} + \frac{K+r^2}{8} + \frac{(3K-1)r}{12} + \frac{r^3}{12}, \tag{2.6}$$

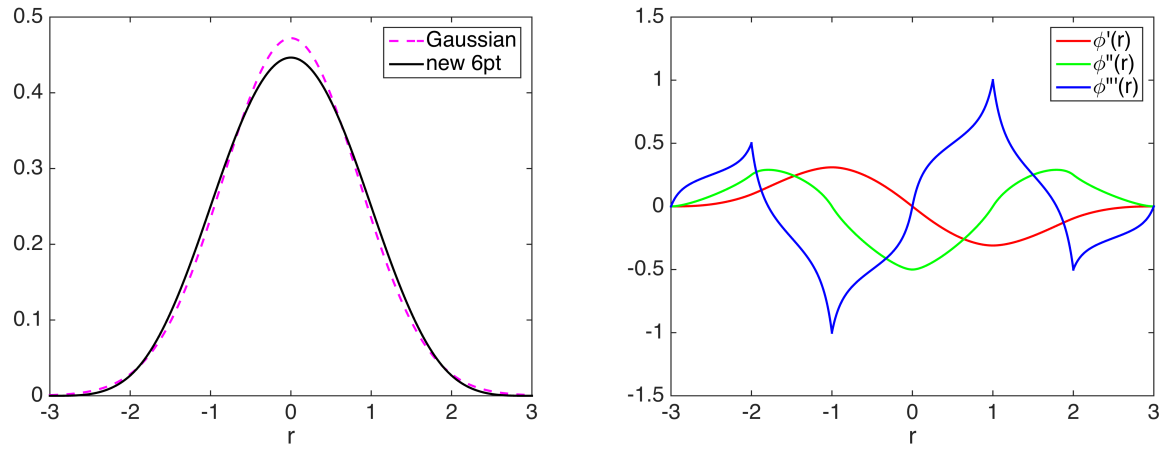
$$\phi(r-1) = 2\phi(r-3) + \frac{1}{4} + \frac{(4-3K)r}{6} - \frac{r^3}{6}, \tag{2.7}$$

$$\phi(r) = 2\phi(r-3) + \frac{5}{8} - \frac{K+r^2}{4}, \tag{2.8}$$

$$\phi(r+1) = -3\phi(r-3) + \frac{1}{4} - \frac{(4-3K)r}{6} + \frac{r^3}{6}, \tag{2.9}$$

$$\phi(r+2) = \phi(r-3) - \frac{1}{16} + \frac{K+r^2}{8} - \frac{(3K-1)r}{12} - \frac{r^3}{12}. \tag{2.10}$$

The new 6-point kernel is a Gaussian-like function, as shown in Figure 1, and it is the first IB kernel developed that has three continuous derivatives. As a comparison, the standard 3-point, 4-point, 6-point kernels and their continuous first derivative are plotted in Figure 2. We notice that the new 6-point kernel is non-negative for all r, whereas the standard 6-point kernel has negative tails.



(a) (b)

Figure 1: (a) The new 6-point kernel compared with the Gaussian with the *same* second moment. (b) The first three derivatives of the new 6-point kernel.

3. Numerical Tests

In this section, we demonstrate a significant improvement in the translational (and rotational!) invariance of our new 6-point kernel. We randomly select 100,000 pairs of Lagrangian markers $\mathbf{X}_1, \mathbf{X}_2$ in a periodic box $[0, 32]^3$ with meshwidth $h = 1$ and compute the 3D generalization of (1.3),

$$\tilde{\delta}(\mathbf{X}_1, \mathbf{X}_2) = \sum_{\mathbf{x} \in g_h} \delta_h(\mathbf{x} - \mathbf{X}_1)\, \delta_h(\mathbf{x} - \mathbf{X}_2), \tag{3.1}$$

where $\mathbf{x}$ denotes a grid point on the Eulerian grid g_h. In Figure 3, we plot $\tilde{\delta}(\mathbf{X}_1, \mathbf{X}_2)$ normalized by the constant C^3 from (1.2), versus the distance $|\mathbf{X}_1 - \mathbf{X}_2|$. The data are binned according to $|\mathbf{X}_1 - \mathbf{X}_2|$, and error-bars showing the maximum, mean and minimum of each bin are overlaid with the data. If an IB kernel were exactly translation-invariant, the plot

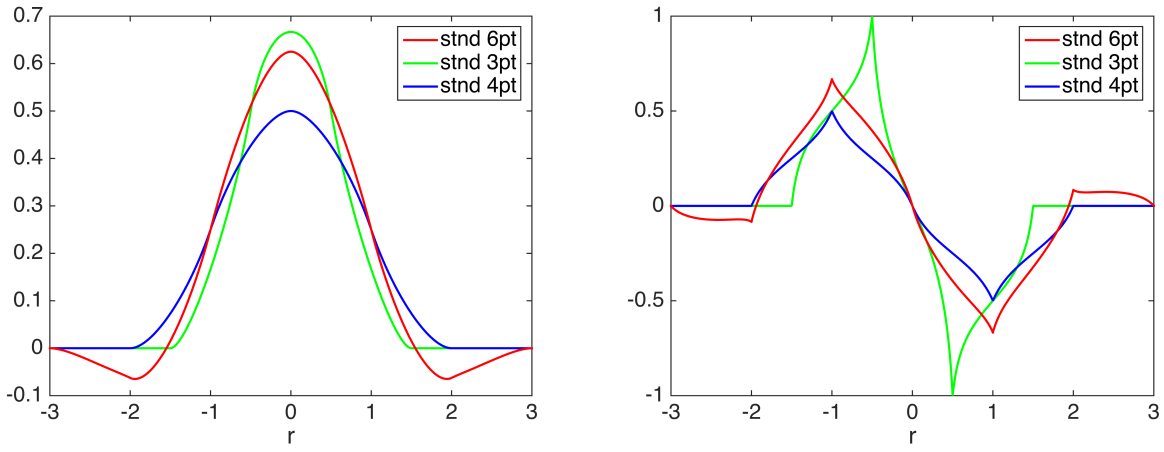


Figure 2: (a) The standard 3-point, 4-point, and 6-point kernels. (b) The first derivatives of the standard 3-point, 4-point, and 6-point kernels.

of $\tilde{\delta}(\mathbf{X}_1, \mathbf{X}_2)$ would be a curve. The spreading pattern in the data around this curve clearly indicates that none of the IB kernels compared here are exactly translation-invariant, but gives a qualitative picture of how close to translational and rotational invariance each kernel is. The data for the new 6-point kernel almost form a curve, while the data of the other kernels have significantly larger deviations from the mean. Moreover, the deviation in the data of the new 6-point kernel is uniform for all distances, but the standard 6-point kernel has a much larger deviation near $r = 2.5$. For a more quantitative comparison, we summarize the maximum standard deviation of all bins for each IB kernel in Table 2. The maximum standard deviation of the new 6-point kernel is an order of magnitude smaller than that of the other kernels. In all respects, the new 6-point kernel gives a significant improvement in grid translational invariance.

	Standard 3-point	Standard 4-point	Standard 6-point	New 6-point
maximum std. dev.	0.0428	0.0168	0.0296	0.0042

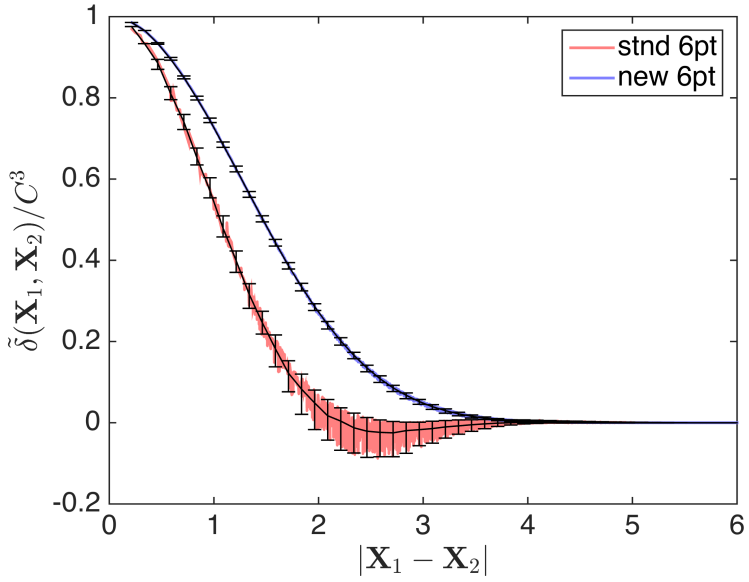
Table 2: Maximum standard deviation of $\tilde{\delta}(\mathbf{X}_1, \mathbf{X}_2)$ over all bins for various IB kernels.

4. Conclusion

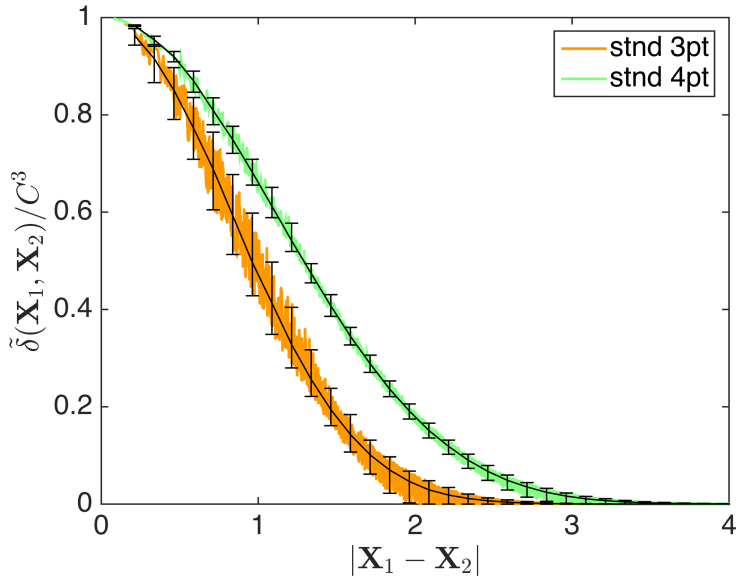
In this note, we have presented a new 6-point immersed-boundary kernel that is used for force spreading and velocity interpolation in the immersed boundary method. The new 6-point kernel is distinguished from other existing IB kernels by its nonzero second-moment constant K. The special choice of $K = \frac{59}{60} - \frac{\sqrt{29}}{20}$ leads to a 6-point IB kernel that is C^3 and features substantially improved translational and rotational invariance compared with the existing standard IB kernels. We believe that, in certain applications, the improved grid invariance and regularity of the new 6-point kernel will be worth its extra computational cost.

5. Acknowlegements

We thank Aleksandar Donev for many enlightening discussions on this work, and in particular for his suggestion to use the nonzero second moment condition as a postulate of the new 6-point IB kernel.



(a)



(b)

Figure 3: $\tilde{\delta}(\mathbf{X}_1, \mathbf{X}_2)$ is plotted versus $|\mathbf{X}_1 - \mathbf{X}_2|$ for 100,000 pairs of randomly selected Lagrangian markers. The data are binned according to $|\mathbf{X}_1 - \mathbf{X}_2|$ and error-bars showing the maximum, mean and minimum of each bin are overlaid with the data. The deviation in the data gives a quantitative measure of translational and rotational invariance of an IB kernel. The standard deviation in the data for the new 6-point kernel (blue) is an order of magnitude smaller than for the other kernels (a) The standard 6-point kernel vs. the new 6-point kernel. (b) The standard 3-point kernel vs. the standard 4-point kernel.

[1] C. S. Peskin, Numerical analysis of blood flow in the heart, J. Computational Phys. 25 (3) (1977) 220–252.
[2] C. S. Peskin, The immersed boundary method, Acta Numerica 11 (2002) 479–517.
[3] Y. Liu, Y. Mori, Properties of discrete delta functions and local convergence of the immersed boundary method, SIAM J. Numer. Anal. 50 (6) (2012) 2986–3015.
[4] A. M. Roma, C. S. Peskin, M. J. Berger, An adaptive version of the immersed boundary method, Journal of Computational Physics 153 (2) (1999) 509 – 534.
[5] J. M. Stockie, Analysis and computation of immersed boundaries, with application to pulp fibres, ProQuest LLC, Ann Arbor, MI, 1997, thesis (Ph.D.)–The University of British Columbia (Canada).