Journal of Computational Physics 316 (2016) 139–144

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Short note

# A Gaussian-like immersed-boundary kernel with three continuous derivatives and improved translational invariance

Yuanxun Bao *, Jason Kaye, Charles S. Peskin

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

ARTICLE INFO

*Article history:*
Received 2 October 2015
Received in revised form 28 March 2016
Accepted 10 April 2016
Available online 14 April 2016

*Keywords:*
Immersed boundary method
Fluid–structure interaction
Discrete delta function
Immersed-boundary kernel
Translational invariance

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.



## 1. Introduction

The immersed boundary (IB) method was originally proposed 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 $\sin(x)/x$. 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

* Corresponding author.
*E-mail addresses:* billbao@cims.nyu.edu (Y. Bao), jkaye@cims.nyu.edu (J. Kaye), peskin@cims.nyu.edu (C.S. Peskin).

http://dx.doi.org/10.1016/j.jcp.2016.04.024
0021-9991/

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 $\phi(r)$,

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

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 $\phi$ 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 $\mathcal{C}^1$ 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 $\phi$ is required for computational efficiency.

The function $\phi(r)$ is constructed by requiring a subset of the following moment conditions:

(i) Zeroth moment: $\sum_j \phi(r-j) = 1$,

(ii) Even–odd: $\sum_{j\ \text{even}} \phi(r-j) = \sum_{j\ \text{odd}} \phi(r-j) = \frac{1}{2}$,

(iii) First moment: $\sum_j (r-j)\,\phi(r-j) = 0$,

(iv) Second moment: $\sum_j (r-j)^2\phi(r-j) = K$, for some constant $K$,

(v) Third moment: $\sum_j (r-j)^3\phi(r-j) = 0$.

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 $\delta_h$ 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 $\delta_h$ 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 $\phi$ being compactly supported [2]. The sum-of-squares condition does give some information about the coupling of $\phi$, 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 condition. It was first introduced in an adaptive IB method using the staggered-grid discretization [4]. The standard 6-point kernel

**Table 1**
Common immersed-boundary kernels with their properties and moment conditions they satisfy. ✓: the kernel satisfies the moment condition; ✗: the kernel does not satisfy the moment condition. In the second moment column, the value of the second moment constant $K$ is given when the second moment condition is satisfied. The regularity column shows the number of continuous derivatives each IB kernel has.

| IB kernel | Support $r_s$ | Even–odd | Zeroth moment | First moment | Second moment | Third moment | Sum of squares | Regularity |
|---|---|---|---|---|---|---|---|---|
| Standard 3-point | 1.5 | ✗ | ✓ | ✓ | ✗ | ✗ | $\frac{1}{2}$ | $\mathcal{C}^1$ |
| Smoothed 3-point | 2 | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ | $\mathcal{C}^2$ |
| Standard 4-point | 2 | ✓ | ✓ | ✓ | ✗ | ✗ | $\frac{3}{8}$ | $\mathcal{C}^1$ |
| Smoothed 4-point | 2.5 | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ | $\mathcal{C}^2$ |
| Standard 6-point | 3 | ✓ | ✓ | ✓ | 0 | ✓ | $\frac{67}{128}$ | $\mathcal{C}^1$ |
| New 6-point | 3 | ✓ | ✓ | ✓ | $\frac{59}{60}-\frac{\sqrt{29}}{20}$ | ✓ | $\approx .326$ | $\mathcal{C}^3$ |

(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, as shown in Fig. 3, it has a larger deviation from translational invariance near $d=2.5$ even compared to 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 $\mathcal{C}^3$ and significantly improves translational invariance compared with other IB kernels. The construction of an IB kernel with a positive and constant second moment $K$ was originally motivated by the important physical implication of the second moment in particle suspensions, namely it is associated with the quadrupole correction in the Faxén relation for a rigid sphere in an arbitrary Stokes flow [6]. The result that the new 6-point kernel has three continuous derivatives is unexpected, however, this makes the new kernel more generally useful. By applying a smoothing technique to the standard IB kernels, Yang et al. [7] developed a family of $\mathcal{C}^2$ IB kernels whose first derivative satisfies up to the second moment condition for the derivative. They showed that these derivative moment conditions are intrinsically linked to the error of force spreading in the IB scheme, and IB kernels that satisfy these conditions can significantly reduce non-physical spurious oscillations of force spreading in moving-boundary problems. The derivative of the new 6-point kernel can be shown to satisfy up to the third moment condition. We will also include the smoothed 3-point and 4-point kernels in the comparison of translational invariance in section 3. Liu and Mori developed a MATLAB routine that automatically generates all the standard IB kernels as well as many others [3]. We have also made our MATLAB codes for generating the new 6-point kernel available at https://github.com/stochasticHydroTools/IBMethod.

## 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 $\phi$ 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, $\phi$ being $\mathcal{C}^1$ 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). Analogous to the unexpected consequence that the standard 4-point kernel is $\mathcal{C}^1$ when we only assume continuity in its construction, we verify a posteriori that the new 6-point kernel is $\mathcal{C}^3$ (see Fig. 1(b)). We emphasize that *only* the choice of $K$ given by (2.1) makes $\phi(r)\in\mathcal{C}^3$, and that this is the smallest value of $K$ for which $\phi(r)\geq 0$ for all $r$. For other choices of $K$, $\phi$ is only $\mathcal{C}^1$, and there are some choices of $K$ for which $\phi$ does not exist. The formula for the new 6-point kernel is given by

$$\beta(r) = \frac{9}{4}-\frac{3}{2}(K+r^2)r+(\frac{22}{3}-7K)r-\frac{7}{3}r^3, \tag{2.2}$$

$$\gamma(r) = -\frac{11}{32}r^2+\frac{3}{32}(2K+r^2)r^2+\frac{1}{72}\left((3K-1)r+r^3\right)^2+\frac{1}{18}\left((4-3K)r-r^3\right)^2, \tag{2.3}$$

$$\phi(r-3) = \frac{-\beta(r)+\operatorname{sgn}\left(\frac{3}{2}-K\right)\sqrt{\beta^2(r)-112\gamma(r)}}{2\alpha}, \tag{2.4}$$

$$\phi(r-2) = -3\phi(r-3)-\frac{1}{16}+\frac{K+r^2}{8}+\frac{(3K-1)r}{12}+\frac{r^3}{12}, \tag{2.5}$$

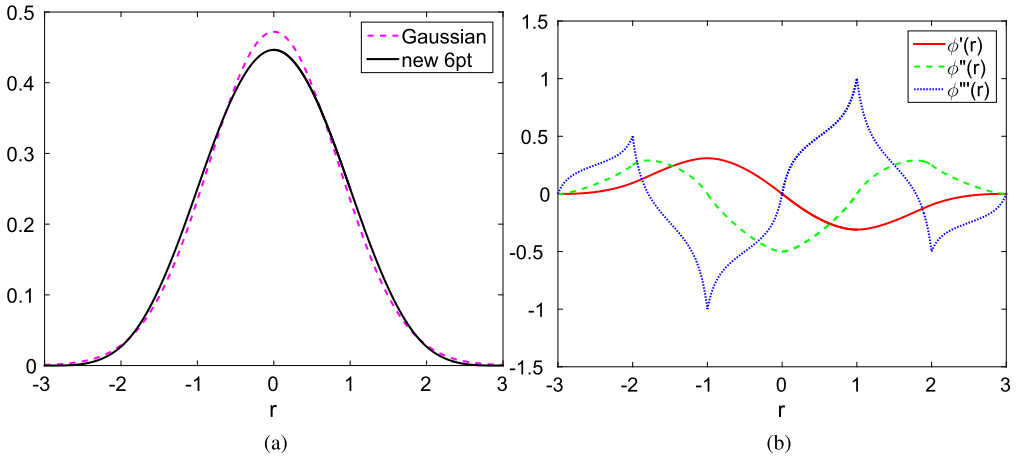


**Fig. 1.** (a) The new 6-point kernel compared with the Gaussian with the *same* second moment. (b) The first three derivatives of the new 6-point kernel.

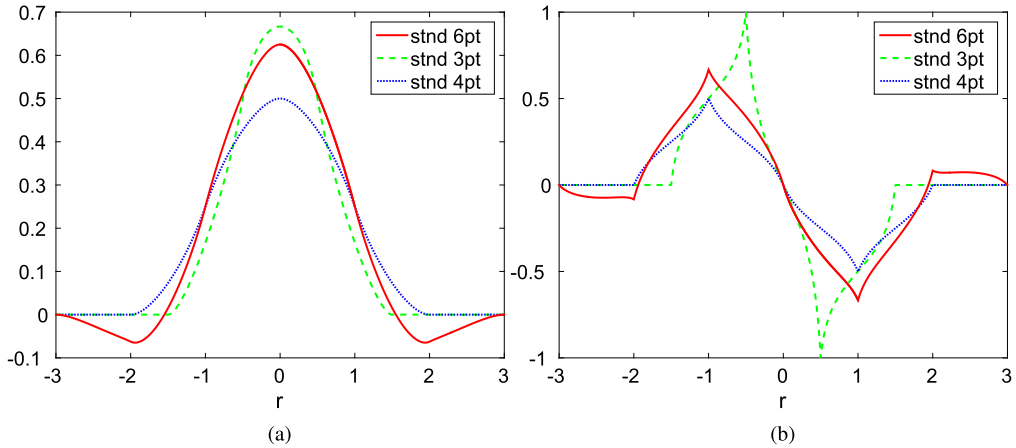


**Fig. 2.** (a) The standard 3-point, 4-point, and 6-point kernels. (b) The first derivatives of the standard 3-point, 4-point, and 6-point kernels.

$$\phi(r-1) = 2\phi(r-3) + \frac{1}{4} + \frac{(4-3K)r}{6} - \frac{r^3}{6}, \tag{2.6}$$

$$\phi(r) = 2\phi(r-3) + \frac{5}{8} - \frac{K+r^2}{4}, \tag{2.7}$$

$$\phi(r+1) = -3\phi(r-3) + \frac{1}{4} - \frac{(4-3K)r}{6} + \frac{r^3}{6}, \tag{2.8}$$

$$\phi(r+2) = \phi(r-3) - \frac{1}{16} + \frac{K+r^2}{8} - \frac{(3K-1)r}{12} - \frac{r^3}{12}. \tag{2.9}$$

Note that, in the formula presented above, $r \in [0,1]$. The new 6-point kernel is a Gaussian-like function, as shown in Fig. 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 Fig. 2. We notice that the new 6-point kernel is non-negative for all $r$, whereas the standard 6-point kernel has negative tails.

## 3. Numerical tests

In this section, we demonstrate a significant improvement in the translational invariance[1] of our new 6-point kernel. We randomly select $10^5$ 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),

[1] The test that we use actually checks for rotational invariance at the same time, since it involves the Euclidean distance between a pair of markers, and not merely the vector from one marker to the other.

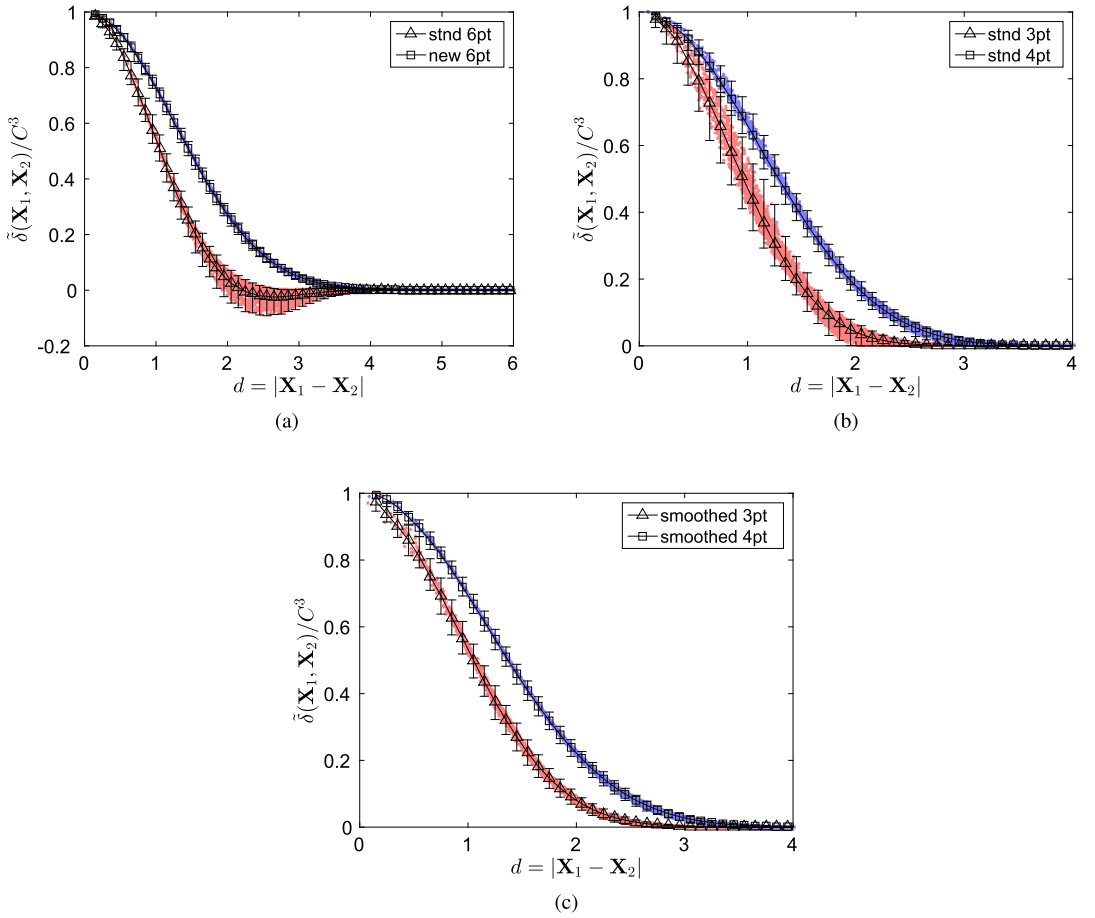


**Fig. 3.** Normalized $\tilde{\delta}(\mathbf{X}_1,\mathbf{X}_2)$ is plotted versus $d=|\mathbf{X}_1-\mathbf{X}_2|$ for $10^5$ pairs of randomly selected Lagrangian markers. The data are binned according to $d=|\mathbf{X}_1-\mathbf{X}_2|$ and error-bars showing the maximum, mean and minimum of each bin are overlaid with the data. The deviation in the data gives a quantitative measure of translational 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 standard IB 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. (c) The smoothed 3-point kernel vs. the smoothed 4-point kernel. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

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

where $\mathbf{x}$ denotes a grid point on the Eulerian grid $g_h$. In Fig. 3, we plot $\tilde{\delta}(\mathbf{X}_1,\mathbf{X}_2)$ normalized by the constant $C^3$ from (1.2), versus the distance $d=|\mathbf{X}_1-\mathbf{X}_2|$. The data are binned according to $d=|\mathbf{X}_1-\mathbf{X}_2|$, 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 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 invariance each kernel is. The data of the new 6-point kernel and the smoothed 4-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 $d=2.5$. 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 standard IB kernels, and is about half of the deviation of the smoothed 4-point kernel. In terms of computational cost, we summarize the computation time of $\tilde{\delta}(\mathbf{X}_1,\mathbf{X}_2)$ for $10^5$ pairs of Lagrangian markers using the kernels we have compared. The timings are based on simulations performed on a desktop with Intel Core i7-4770 CPU 3.40 GHz under the MATLAB environment. The main cost of using an IB kernel in spreading/interpolation depends on its support size. In Table 2, the 6-point kernels are about 3–4 times more expensive than the standard 4-point kernel in our comparison, as they communicate with 192 nearby grid points in spreading/interpolation in 3D, while a 4-point kernel only communicates with 64 nearby grid points. The smoothed 3-point and 4-point kernels are more expensive than their standard counterparts in that they have wider supports as shown in Table 1. In all respects, the new 6-point kernel achieves a significant improvement in grid translational invariance with a modest increase in computational cost.

**Table 2**
Maximum standard deviation of $\tilde{\delta}(\mathbf{X}_1, \mathbf{X}_2)$ over all bins for various IB kernels, and the computation time for computing $\tilde{\delta}(\mathbf{X}_1, \mathbf{X}_2)$ for $10^5$ markers.

| | Standard 3-point | Smoothed 3-point | Standard 4-point | Smoothed 4-point | Standard 6-point | New 6-point |
|---|---|---|---|---|---|---|
| maximum std. dev. | 0.0428 | 0.0212 | 0.0168 | 0.0083 | 0.0296 | 0.0042 |
| computation time | 6.34 s | 9.52 s | 9.01 s | 12.06 s | 31.19 s | 30.86 s |

## 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 invariance compared with the existing standard IB kernels. Recently, we have successfully applied the new 6-point kernel to a new IB method with an exactly divergence-free interpolated velocity field (manuscript in preparation), in which derivatives of the discrete delta function are involved, and have achieved $10^3$–$10^5$ times improvements in volume conservation of the IB method. We believe that, in many other applications in which derivatives of the IB kernel are needed, the improved grid invariance and regularity of the new 6-point kernel will be worth its extra computational cost.

## Acknowledgements

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. Y. Bao was supported in part by the Air Force Office of Scientific Research under grant number FA9550-12-1-0356, as well as the U.S. Department of Energy Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under Award Number DE-SC0008271. J. Kaye was supported in part by the National Science Foundation under grants NSF DMS-1115341 and DMS-1016554.

## References

[1] C.S. Peskin, Numerical analysis of blood flow in the heart, J. Comput. Phys. 25 (3) (1977) 220–252.
[2] C.S. Peskin, The immersed boundary method, Acta Numer. 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, J. Comput. Phys. 153 (2) (1999) 509–534.
[5] J.M. Stockie, Analysis and computation of immersed boundaries, with application to pulp fibres, PhD thesis, The University of British Columbia (Canada), ProQuest LLC, Ann Arbor, MI, 1997.
[6] J.F. Brady, R.J. Phillips, J.C. Lester, G. Bossis, Dynamic simulation of hydrodynamically interacting suspensions, J. Fluid Mech. 195 (1988) 257–280.
[7] X. Yang, X. Zhang, Z. Li, G.-W. He, A smoothing technique for discrete delta functions with application to immersed boundary method in moving boundary simulations, J. Comput. Phys. 228 (20) (2009) 7821–7836.