Proceedings of the ASME 2021
Fluids Engineering Division Summer Meeting
FEDSM2021
August 10-12, 2021, Virtual Conference, Online

FEDSM2021-65599

AN IMPROVED LEVEL-SET-BASED IMMERSED BOUNDARY RECONSTRUCTION METHOD FOR COMPUTING BIO-INSPIRED UNDERWATER PROPULSION

Yu Pan, Haibo Dong, Wei Zhang
Department of Mechanical and Aerospace Engineering,
University of Virginia
Charlottesville, VA, USA

ABSTRACT

The immersed boundary method (IBM) has been widely employed to study bio-inspired underwater propulsion which often involves the high Reynolds number, complex body morphologies and large computational domain. Due to these problems, the immersed boundary (IB) reconstruction can be very costly in a simulation. Based on our previous work, an improved level-set-based immersed boundary method (LS-IBM) has been developed in this paper by introducing the narrow-band technique. Comparing with the previous LS-IBM, the narrow-band level-set-based immersed boundary method (NBLS-IBM) is only required to propagate the level set values from the points near the boundaries to all the points in the narrow band. This improvement reduces the computational cost from $O((L_D/\Delta x)^3)$ *to* $O(k(L_D/\Delta x)^2)$. *By simulating a steady-swimming Jackfish-like body, the consistency and stability of the new reconstruction method in the flow solver have been verified. Applications to a dolphin-like body swimming and a shark-like body swimming are used to demonstrate the efficiency and accuracy of the NBLS-IBM. The time for reconstructions shows that the reconstruction efficiency can increase up to 64.6% by using the NBLS-IBM while keeping the accuracy and robustness of the original LS-IBM. The vortex wake of the shark-like body in steady swimming shows the robustness, fastness and compatibility of the NBLS-IBM to our current flow solver.*

Keywords: Level set, immersed boundary method, narrow band, bio-inspired underwater propulsion

NOMENCLATURE

L_D	the characteristic length
Δx	mesh size
k	number of cells in the narrow band
u_i	the velocity component
p	the pressure
ρ	the fluid density
Re	Reynold number
St	Strouhal number
φ	level set function
R	searching radius

1. INTRODUCTION

The immersed boundary method was first proposed by Peskin[1] to simulate cardiac blood flow problems. After introducing the discrete forcing method[2], immersed boundary methods have been widely used to study bio-inspired flows, including insect flight, fish swimming, and biomedical problems, often involving complex body morphologies and moving boundaries. For such flows, the unstructured surface mesh is employed to present the body shape, and the interface between the solid body and the fluid at each time step is detected by the immersed boundary reconstruction process, which directly determines the efficiency and quality of the simulations[3]. For some bio-inspired underwater propulsion problems, the geometry of the immersed bodies can be complex with sharp concave and convex parts. It is challenging to identify the interior and exterior nodes for complex body morphologies, and incorrectly labeling the fluid and solid nodes can ruin the simulation. Besides, in a real fish swimming simulation, the Reynolds number can reach $O(10^6)$ or above, which requires dense body mesh and fine-enough Cartesian mesh, and the computation domain can be extremely large. This results that the computational cost for IB reconstruction at each step is expensive. Thus, a fast, accurate and versatile IB reconstruction method is necessary for studying the bio-inspired underwater propulsion.

Mittal et al. [3] employed a direct searching method to determine fluid and solid nodes, which reconstructed the interface by sweeping through all the triangular body meshes and finding all Cartesian points near the boundary. The exterior (fluid) and interior (solid) points can be separated according to the sign of the dot-product of the vector extending from the centroid of a triangular element to its closest node with the surface normal of the element. The computational cost of this method is in the order of $O((L_D/\Delta x)^5)$[4], where L_D is the characteristic length and Δx is the minimum mesh size. The

computational consumption will be very high when the Cartesian mesh is dense or the computation domain is large. In addition, Senocak et al.[5] found that the method may fail when the differences in size between triangular surface elements are too large. To improve the robustness and efficiency of detecting immersed boundaries, Deng et al.[4] proposed a level-set-based fast reconstruction method for complex moving boundaries with the 3D finite-difference sharp-interface immersed boundary method, and we call it LS-IBM in this work. In their work, the immersed boundary was identified by calculating the signed distance value, stored in the level set function, of grid points near the solid boundaries, and propagating the values to the surrounding domain. The computational cost was reduced from $O((L_D/\Delta x)^5)$ to $O((L_D/\Delta x)^3)$ comparing with the standard IB method. Then, Wang et al.[6] further developed this method to handle the bio-inspired models with both solid body and membrane.

In this work, we further improve the efficiency of immersed boundary reconstruction by developing a narrow band level-set-based immersed boundary method (NBLS-IBM) and apply it to some underwater propulsion problems. For the previous LS-IBM[4], the level set values of other grid points in the computational domain are obtained by propagating the signed distance values of points in the immediate vicinity of the immersed boundary. The computational cost of this operation is $O((L_D/\Delta x)^3)$, and the operation is required at every time step in a simulation. However, to identify the solid and fluid nodes, we just need to update the level set values near the immersed boundary at the current time step to detect the interface, and directly use the level set values passed on from the precious time step for other grid points. Thus, we employ the narrow-band technique[7] to speed up the boundary reconstruction process. By building a thin band around the zero-level set, the computations are only performed on the points in the band, which can reduce the computation cost from $O((L_D/\Delta x)^3)$ to $O(k(L_D/\Delta x)^2)$[8], where k is the number of cells in the narrow band. In the meantime, the narrow band LS-IBM (NBLS-IBM) inherits the accuracy and robustness of LS-IBM demonstrated in Ref. [4]. Next, we present the narrow band level-set-based immersed boundary method in detail and apply it to underwater propulsion problems including Jackfish swimming, dolphin swimming and shark swimming with complex moving boundaries.

2. METHODOLOGY

2.1 Sharp-interface immersed-boundary-based flow solver

The governing equations considered in this work are 3-D unsteady Navier-Stokes equations for a viscous incompressible flow, written in an indicial form as,

$$\frac{\partial u_i}{\partial x_i} = 0,\ \frac{\partial u_i}{\partial t} + \frac{\partial u_i u_j}{\partial x_j} = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \nu\frac{\partial^2 u_i}{\partial x_i \partial x_j} \quad (1)$$

where the u_i are the velocity components, p is the pressure, and ρ and ν denote the fluid density and the kinematic viscosity respectively.

A finite-difference Cartesian-grid sharp-interface immersed boundary method[3] is employed to solve the above eqautions. In this method, the complex moving boundaries are conducted on stationary Cartesian grids, which greatly reduces the computational cost comparing to the body-conformal methods. The equations are discretized in space by using a second-order central difference scheme. A fractional step method is employed to obtain second-order accuracy in time for these unsteady equations. A second-order, Adams-Bashforth scheme is employed for the convection terms, while the diffusion terms are discretized using an implicit Crank-Nicolson scheme. Besides, a multi-dimensional ghost-cell method is used to impose boundary conditions on the immersed boundary precisely. This method has been successfully applied to simulate biological fish swimming[9, 10], fish-like swimming[11-14] and other flapping propulsions[15-19]. More details about the method can be found in Ref.[3].

2.2 A fast narrow-band level-set based IB reconstruction

The computation complexity of the previous LS-IB reconstruction method is $O((L_D/\Delta x)^3)$. Even though this method greatly increases the reconstruction speed compared to the direct searching method in Ref. [3], the IB reconstruction process will be time-consuming and inefficient when the body mesh and domain mesh are dense, or the computational domain is large. It limits applying the method to high Reynolds number underwater propulsions or the complex swimming motion in a large domain.

To improve the efficiency of LS-IB reconstruction for complex moving boundaries, we introduce the narrow band technique. In the LS-IB reconstruction method, the level set values are updated for each grid points in the computation domain after calculating the signed distance values of grid points in the immediate vicinity of the solid boundary. The central idea of the narrow method is to build a thin band around the zero-level set, that is, the solid interface, and to update the level set values at the Cartesian grids only in this band, which thus reduces the computational complexity of propagating level set values to $O(k(L_D/\Delta x)^2)$.

The level set function is smooth [20], recording the signed distance values of any grid point to the interface between the solid body and the fluid, and is denoted as φ. In this work, $\varphi = 0$ is the interface, $\varphi < 0$ denotes the fluid and $\varphi > 0$ represents the solid body. The LS-IBM described in the previous study[4] contains four steps. The difference between the current method and the previous LS-IBM is mainly at the third step. For the completeness of the paper, we still list all four steps here and highlight the changes.

Step 1: Go through all the elements of the immersed boundary, determine their positions by checking their coordinates along the x, y and z grid lines of the Cartesian mesh. In the above process, the neighbor grid points in the immediate vicinity of the solid boundary have been searched and recorded with the corresponding elements. In Figure 1, the circles and squares represent the fluid points and the solid points respectively. And the neighbor points are labeled by the solid red circles and squares. The computational cost of this step is $O((L_D/\Delta x)^3)$.

Step 2: For each neighbor point, find the closest element and calculate the signed distance value (level set value) to the element. The signed distance value of other Cartesian points within the researching area in Figure 1 are also calculated. Figure 1 marked the searching area by the dashed lines and the searching points by hollowed red circle and square. In this step, the updated level set values may be incorrect, which can jeopardize the reconstruction of the interface and crash the simulation. In case of that, we check the level set values one by one and fix the failed point by averaging the level set values around the point.

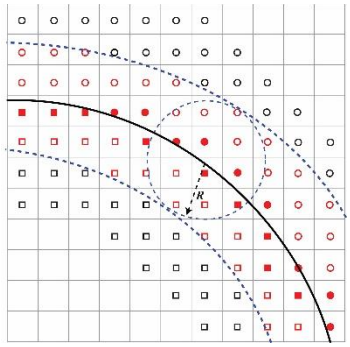

FIGURE 1: SCHEMATIC OF SEARCHING THE NEIGHBOR POINTS. THE DARK SOLID LINE DENOTES THE IMMERSED BOUNDARY AND THE SEARCHING AREA IS BORDERED WITH THE BLUE DASHED LINES. THE CIRCLES AND SQUARES REPRESENT THE FLUID POINTS AND THE SOLID POINTS RESPECTIVELY.

Step 3: Use the labeled neighbor points to build a narrow band and propagate level set values to other points in this band. In this work, we set the width of the band as four; that is, the level set values propagate forward and backward for two grids in each direction, respectively. The narrow band is the area bordered by the dashed blue line in Fig. 2. For other points in the narrow band, the calculation of level set values is as follows,

$$\begin{cases} \varphi^n_{I,J,K} = \frac{sum\varphi^u}{m} + sgn(sum\varphi^u) \times dist\varphi \\ sum\varphi^u = \sum_{i=I-1}^{I+1}\sum_{j=J-1}^{J+1}\sum_{k=K-1}^{K+1}\varphi^u_{i,j,k} \\ dist\varphi = \sqrt{\left(\overline{x^u} - x_{I,J,K}\right)^2 + \left(\overline{x^u} - x_{I,J,K}\right)^2 + \left(\overline{x^u} - x_{I,J,K}\right)^2} \\ max\{|I - I_0|, |J - J_0|, |K - K_0|\} \le 2 \end{cases} \quad (2)$$

Here $\varphi^n_{I,J,K}$ is the level set value at the gird (I,J,K) that is to be updated by the neighboring grids with the updated level set value $\varphi^u_{i,j,k}$, m is the number of cells in the summation. $\bar{\cdot}$ denotes the average center of the updated grids around the new point. (I_0, J_0, K_0) is the neighbor point in the immediate vicinity of the solid boundary, and $max\{|I - I_0|, |J - J_0|, |K - K_0|\} \le 2$ represents that the band width is four.

Step 4: Use the level set value to decide the status of each grid point. As mentioned before, $\varphi > 0$ represents the solid points, and $\varphi < 0$ denotes the fluid points.

In the initial time step of a simulation, the IB reconstruction is conducted by directly calculating the level set value of grid points in the whole domain. Thus, the initialized level set function stores accurate level set values for all grid points. After that, by building a narrow band around the IB, the level set values are just to be updated for the points in this band at the current time step. The level set values of grid points outside the narrow band can be directly passed on from the previous step. In this way, the computational cost of the third step is reduced to $O(k(L_D/\Delta x)^2)$. Then, the fluid/solid grid status can be decided from the sign of the level set value.

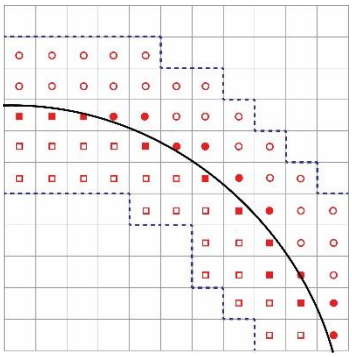

FIGURE 2: SCHEMATIC OF PROPAGATING THE LEVEL SET VALUES TO ALL POINTS IN THE NARROW BAND.

3. RESULTS AND DISCUSSION

In this section, the consistency and stability of the new reconstruction method to the flow solver is firstly demonstrated by comparing the hydrodynamics of Jackfish-like body in steady swimming simulated by using the NBLS-IBM based flow solver and the LS-IBM based flow solver. Then, the efficiency of the new method will be analyzed by comparing the reconstruction time of NBLS-IBM and that of LS-IBM for a dolphin-like body and a shark-like body with different background Cartesian meshes. Besides, with the NBLS-IBM, the vortex wake of a steady-swimming shark is also presented to show the computability, reliability and effectiveness of the new IB reconstruction method in the current flow solver.

3.1 Jackfish steady swimming

To demonstrate that the consistency and stability of NBLS-IBM in the current flow solver, we simulate the steady swimming of a Jackfish-like body by using NBLS-IBM based flow solver and the previous LS-IBM based flow solver separately. The Reynolds number is set as $Re = 2{,}000$ to reduce the computational cost, and the Strouhal number is fixed at 0.5. Figure 3 shows the time history of thrust, which is produced by caudal fin (CF), and drag, which works on the trunk (TK), of a Jackfish in steady swimming. In this paper, the hydrodynamic forces are normalized by using the following formula,

$$C_F = \frac{F}{1/2\rho U^2 S} \quad (3)$$

where $\rho = 1$ is the fluid density in the solver, $U = 0.5$ is the swimming velocity and S denotes the area of caudal fin. From Figure 3, it can be seen that results calculated by these two solvers are completely matching, which demonstrates the

consistency and stability of the narrow-band level-set method in the flow solver.

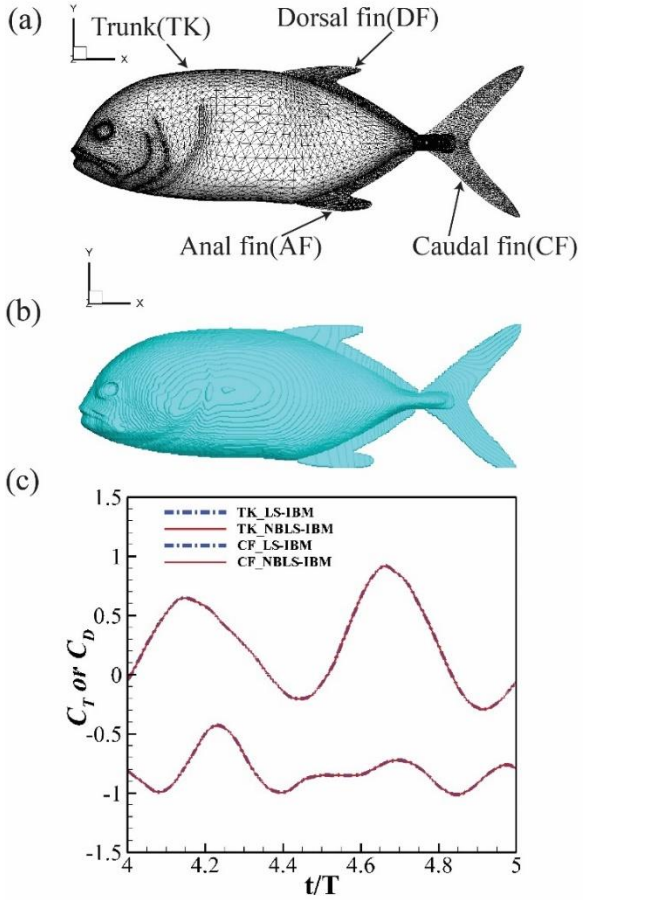

FIGURE 3: (A) UNSTRUCTURED MESH ON A JACKFISH-LIKE BODY, (B) THE CALCULATED ZERO-LEVEL-SET BODY SHAPE, AND (C) THE TIME HISTORY OF THRUST AND DRAG ON THE JACKFISH BODY CALCULATED BY USING NBLS-IBM FLOW SOLVER AND PRECIOUS LS-IBM FLOW SOLVER SEPARATELY. HERE, FOR CONVENIENCE, ONLY THE THRUST PRODUCED BY CAUDAL FIN AND DRAG ON THE FISH TRUNK ARE PRESENTED.

3.2 Dolphin-like body reconstruction

The reconstruction of a dolphin-like body in steady swimming is applied to demonstrate the higher efficiency and reliability of the NBLS-IBM. A comparison of reconstruction times for the dolphin-like body has been conducted by applying LS-IBM and NBLS-IBM to different Cartesian meshes. The total number of surface elements of the dolphin body is 59,968, and the unstructured surface mesh is shown in Fig. 4(a). The Cartesian grids are 16.6 million, 48.7 million, and 146.2 million, respectively, and the corresponding smallest Cartesian mesh dimensions are 0.028, 0.021, and 0.015.

The times for reconstructing the immersed boundary at one time step for each case are list in Table 1. We use the Δt to denote the time saved by the NBLS-IBM comparing to the LS-IBM, and use the ratio of Δt to the time for IB reconstruction by using LS-IBM to represent to what extent the IB reconstruction efficiency has been improved.

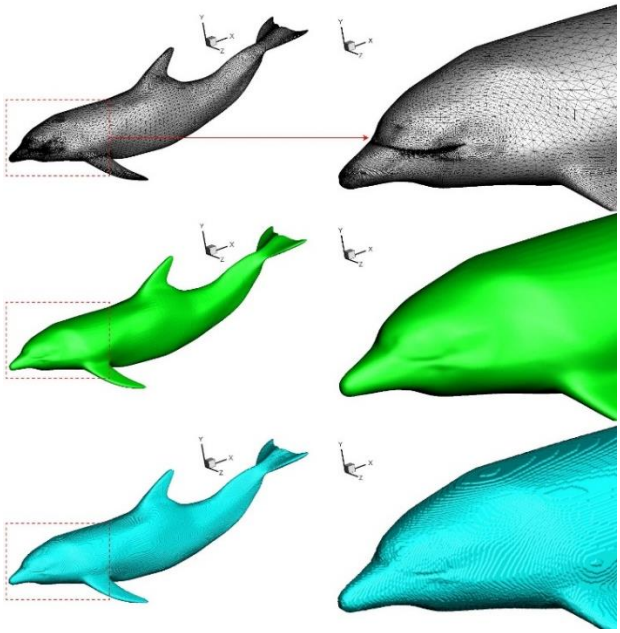

FIGURE 4: THE IB RECONSTRUCTION PROCESS OF A DOLPHIN-LIKE BODY, (A) UNSTRUCTURED MESH ON A DOLPHIN-LIKE BODY, (B) THE CALCULATED ZERO-LEVEL-SET BODY SHAPE, AND (C) THE RECONSTRUCTED IMMERSED BOUNDARY BUILT ON THE CARTESIAN MESH.

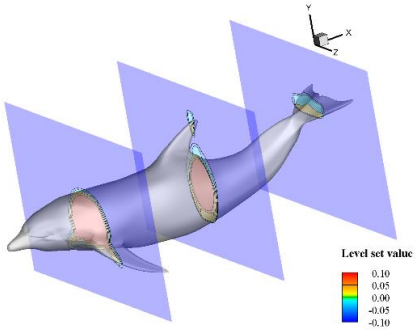

FIGURE 5: THE LEVEL-SET CONTOUR SLICES ARE AT THE FRONT, MIDDLE AND TAIL OF A DOLPHIN-LIKE BODY.

TABLE 1. THE IB CONSTRUCTION TIME FOR EACH CASE IN DOLPHIN SWIMMING.

Cartesian grids (million)		**16.6**	**48.7**	**146.2**
Time(sec) for IB reconstruction	LS-IBM	7.99	21.45	97.76
	NBLS-IBM	3.90	11.13	35.49
Time-saving		51.19%	48.11%	63.70%

Table 1 shows that the time saved by the NBLS-IBM changes from 48.1% to 63.7% for different numbers of the Cartesian mesh. It demonstrates the narrow band level-set

method is much more efficient than the original LS-IBM. Next, we present the unstructured mesh on a dolphin-like body, the calculated zero-level-set body shape, and the reconstructed immersed boundary of the case with 146.2 million Cartesian mesh in Fig. 4. Figure 5 shows three slices of the level set contour around the dolphin-like body at the front, middle and tail of the body. From Fig. 4 and Fig. 5, it can be seen that the immersed boundary can be precisely detected by using the NBLS-IBM.

3.3 Shark-like body reconstruction

In this part, we tested the boundary reconstruction times for a shark-like body with 107K triangular surface elements, as shown in Fig. 6(a), immersed in the Cartesian mesh 24.8M, 48M, 97.6M and 196.4M, respectively. The corresponding smallest Cartesian mesh dimensions for each case are 0.021, 0.016, 0.013, and 0.010. The results are listed in Table 2. From Table 2, it is found that the maximum time-saving is 64.6% and is reached when the number of the Cartesian mesh is 48M, which shows the higher efficiency of the NBLS-IBM comparing to the original LS-IBM. It also shows that the matching between the body mesh and the Cartesian mesh may affect the reconstruction speed, which will be explored in the future. Like the case of the dolphin-like body, we also present the IB reconstruction process for the shark-like body including the unstructured mesh on the body surface, the zero-level-set body shape and the reconstructed immersed boundary built on the Cartesian mesh in Fig. 6. Besides, the three slices at the front, middle and tail of the shark-like body of the level set contour around the body are shown in Fig. 7.

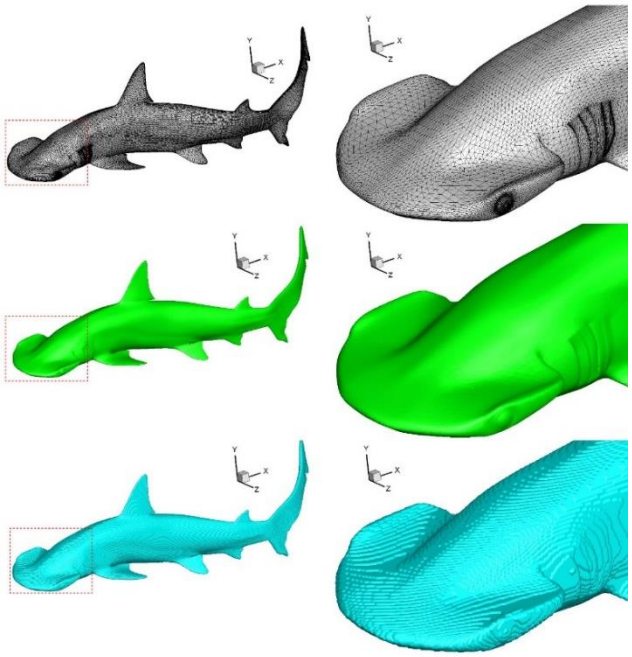

FIGURE 6: THE IB RECONSTRUCTION PROCESS OF A SHARK BODY, (A) THE UNSTRUCTURED MESH ON A SHARK-LIKE BODY, (B) THE CALCULATED ZERO-LEVEL-SET BODY SHAPE, AND (C) THE RECONSTRUCTED IMMERSED BOUNDARY BUILT ON THE CARTESIAN MESH.

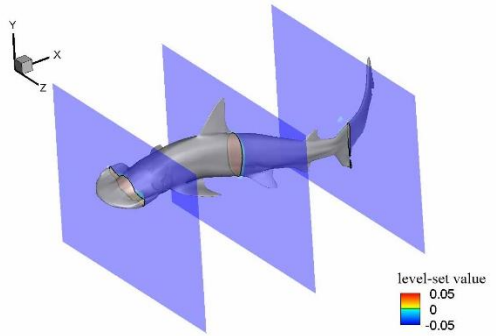

FIGURE 7: THE LEVEL-SET CONTOUR SLICES ARE AT THE FRONT, MIDDLE AND TAIL OF A SHARK-LIKE BODY.

TABLE 2. THE IB CONSTRUCTION TIME FOR EACH CASE IN SHARK SWIMMING.

Cartesian grids (Million)		**24.8**	**48**	**97.6**	**196.4**
Time(sec) for IB reconstruction	LS-IBM	8.71	9.52	32.47	42.94
	NBLS-IBM	3.08	5.45	12.74	29.37
Time-saving		64.64%	42.75 %	60.76 %	31.60 %

3.4 Shark steady swimming

In this part, we simulate the shark-like body steady swimming to demonstrate the robustness and fastness of the NBLS-IBM in the current flow solver. The Reynolds number is set as $Re = 8{,}000$ to reduce the computational cost, and the Strouhal number is 0.8. Besides, to generate fine-enough grids with limited memory, a block-wise octree-like structured adaptive mesh refinement (AMR) technique is employed in the solver[4]. The OpenMP parallelization is also applied to improve the calculation speed. Figure 8 shows the vortex wake generated by the shark-like body in steady swimming. The vortex structures are visualized by using the iso-surface with the λ_2-criterion at two different values ($|\lambda_2| = 4, 12$).

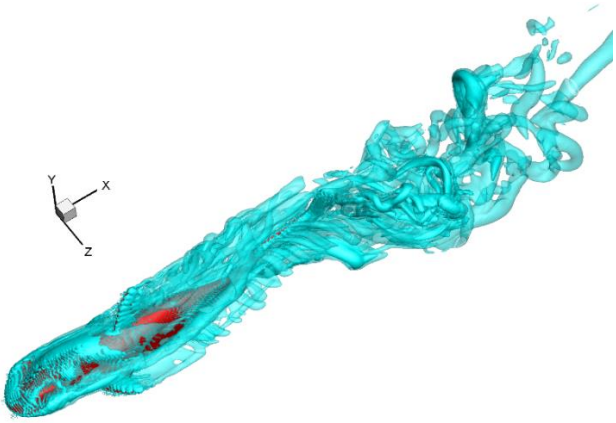

FIGURE 8: THE VORTEX WAKE OF A SHARK-LIKE BODY IN STEADY SWIMMING.

4. CONCLUSION

In this work, we propose a narrow-band level-set immersed boundary method to speed up the IB reconstruction process. Comparing with the original LS-IBM in Ref. [4], the current method only needs to propagate the level set value from the neighbor points near the boundary to other points in the narrow band when identifying the solid and fluid nodes at each time-step. The introduction of the narrow band technique reduces the computational cost of IB reconstruction from $O((L_D/\Delta x)^3)$ to $O(k(L_D/\Delta x)^2)$. The consistency and stability of the new reconstruction method for the flow solver was demonstrated by comparing the hydrodynamics of a Jackfish-like body in steady swimming simulated by using the NBLS-IBM based solver and LS-IBM solver separately. Then, we tested the reconstruction efficiency of NBLS-IBM by analyzing the reconstruction time for a dolphin-like body and a shark-like body with different Cartesian meshes. The results show that the efficiency of IB reconstruction is greatly improved by introducing the narrow band method and increases up to 64.6%. The detected immersed boundaries shown in Fig. 4 and Fig. 6 demonstrate that the accuracy of IB reconstruction can be maintained at the same time. Finally, the vortex wake of the shark-like body in steady swimming proves the stability and fastness of NBLS-IBM when applying to the current flow solver.

ACKNOWLEDGEMENTS

This work was supported under Grant Nos. ONR MURI N0014-14-1-0533 and NSF CNS-1931929.

REFERENCES

[1] Peskin, C. S., 2003, "The immersed boundary method," Acta Numerica, 11, pp. 479-517.
[2] Mittal, R., and Iaccarino, G., 2005, "Immersed Boundary Methods," Annual Review of Fluid Mechanics, 37(1), pp. 239-261.
[3] Mittal, R., Dong, H., Bozkurttas, M., Najjar, F., Vargas, A., and Von Loebbecke, A., 2008, "A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries," Journal of computational physics, 227(10), pp. 4825-4852.
[4] Deng, X., Han, P., Wang, J., and Dong, H., "A level set based boundary reconstruction method for 3-D bio-inspired flow simulations with sharp-interface immersed boundary method," Proc. 2018 Fluid Dynamics Conference, p. 4163.
[5] Senocak, I., Sandusky, M., DeLeon, R., Wade, D., Felzien, K., and Budnikova, M., 2015, "An Immersed Boundary Geometric Preprocessor for Arbitrarily Complex Terrain and Geometry," Journal of Atmospheric and Oceanic Technology, 32(11), pp. 2075-2087.
[6] Wang, J., Han, P., Deng, X., and Dong, H., "A Versatile Level Set Based Immersed Boundary Reconstruction for Bio-Inspired Flow Applications," Proc. AIAA Scitech 2020 Forum, p. 2235.
[7] Adalsteinsson, D., and Sethian, J. A., 1995, "A fast level set method for propagating interfaces," Journal of computational physics, 118(2), pp. 269-277 0021-9991.
[8] Sethian, J. A., 1999, Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science, Cambridge university press.
[9] Liu, G., Ren, Y., Dong, H., Akanyeti, O., Liao, J. C., and Lauder, G. V., 2017, "Computational analysis of vortex dynamics and performance enhancement due to body–fin and fin–fin interactions in fish-like locomotion," Journal of Fluid Mechanics, 829, pp. 65-88.
[10] Wang, J., Wainwright, D. K., Lindengren, R. E., Lauder, G. V., and Dong, H., 2020, "Tuna locomotion: a computational hydrodynamic analysis of finlet function," Journal of the Royal Society Interface, 17(165), p. 20190590.
[11] Han, P., Lauder, G. V., and Dong, H., 2020, "Hydrodynamics of median-fin interactions in fish-like locomotion: Effects of fin shape and movement," Physics of Fluids, 32(1), p. 011902.
[12] Khalid, M. S. U., Wang, J., Dong, H., and Liu, M., 2020, "Flow transitions and mapping for undulating swimmers," Physical Review Fluids, 5(6), p. 063104.
[13] Pan, Y., and Dong, H., 2020, "Computational analysis of hydrodynamic interactions in a high-density fish school," Physics of Fluids, 32(12), p. 121901.
[14] Pan, Y., Han, P., Huang, J., and Dong, H., 2020, "Effect of Formation Pattern on Schooling Energetics in Fish-Like Swimming," Fluids Engineering Division Summer Meeting, American Society of Mechanical Engineers, p. 003T005A046.
[15] Li, C., and Dong, H., 2016, "Three-dimensional wake topology and propulsive performance of low-aspect-ratio pitching-rolling plates," Physics of Fluids, 28(7).
[16] Dong, H., Liang, Z., and Harff, M., 2009, "Optimal settings of aerodynamic performance parameters in hovering flight," International Journal of Micro Air Vehicles, 1(3), pp. 173-181.
[17] Pan, Y., Wang, J., and Dong, H., "Study on the passive pitching mechanism of different forms of flapping motion in turning flight," Proc. AIAA Aviation 2019 Forum, p. 3435.
[18] Wang, J., Xi, J., Han, P., Wongwiset, N., Pontius, J., and Dong, H., 2019, "Computational analysis of a flapping uvula on aerodynamics and pharyngeal wall collapsibility in sleep apnea," Journal of Biomechanics, 94, pp. 88-98.
[19] Bode-Oke, A. T., Zeyghami, S., and Dong, H., 2018, "Flying in reverse: kinematics and aerodynamics of a dragonfly in backward free flight," J R Soc Interface, 15(143).
[20] Sussman, M., Smereka, P., and Osher, S., 1994, "A level set approach for computing solutions to incompressible two-phase flow," Journal of Computational physics, 114(1), pp. 146-159.