Computers and Fluids 161 (2018) 14–22

Contents lists available at ScienceDirect

Computers and Fluids

journal homepage: www.elsevier.com/locate/compfluid

# An *h*-adaptive implicit immersed boundary-lattice Boltzmann flux solver based on JASMIN AMR package

Pan Zhang [a], Zhenhua Xia [b,*], Qingdong Cai [a,*]

[a] State Key Laboratory for Turbulence and Complex Systems, College of Engineering, Peking University, Beijing 100871, China
[b] Department of Engineering Mechanics, Zhejiang University, Hangzhou 310027, China

ARTICLE INFO

*Article history:*
Received 11 March 2017
Revised 19 October 2017
Accepted 8 November 2017
Available online 10 November 2017

*Keywords:*
Immersed boundary-lattice Boltzmann flux solver
Adaptive mesh refinement
JASMIN

ABSTRACT

In order to efficiently simulate flows with complex geometries, an implicit immersed boundary-lattice Boltzmann flux solver (IB-LBFS) combined with an h-adaptive mesh refinement (AMR) technology is proposed and implemented on JASMIN infrastructure. In the present implementation, the original velocity-splitting IB-LBFS is modified by a momentum-splitting during the factional step. Four benchmark problems are used to validate the present method, including the flow over a stationary circular cylinder, the sedimentation of a two-dimensional elliptical particle, the flow around a stationary sphere and the sedimentation of a single sphere. The simulated results are in good agreements with previously published data, which demonstrates the accuracy and the capability of the proposed method in simulating flow problems with stationary or moving boundaries.



## 1. Introduction

In computational fluid dynamics (CFD), there are many different types of numerical methodologies. As one of the various categories, the immersed boundary method (IBM), first proposed by Peskin [1,2], is now very popular in the CFD community, especially for simulating flows with complex geometries and moving boundaries [3,4]. In IBM, Cartesian mesh, which is not required to conform to the boundary geometry, is adopted for the solution of the flow field while discrete Lagrangian points are used for the implementation of non-slip boundary condition. With this numerical configuration, which is in sharp contrast to those used in conventional numerical methods based on the body-conformal structured or unstructured grids, IBM can avoid tedious grid generation process, especially for the three-dimensional (3D) problems with complex geometries, and the remeshing and projecting processes for the flows with moving boundaries. Thanks to these advantages, IBM has drawn considerable attentions in the CFD community in recent years, and many efforts have been devoted to develop efficient IBMs [5–11].

In the miscellaneous implementations of IBMs, either a continuum Navier-Stokes (NS) equation solver [2,12] or the kinetic lattice Boltzmann method (LBM) [13,14] can be employed as the basic flow solver to simulate the flow field. Many combinations of IBM with LBM, i.e. IB-LBM, have been successfully proposed to simulate the flow problems [15–30] in the past dozen years due to the attractive features of LBM. As an alternative, IBM can also be implemented with the recently developed lattice-Boltzmann flux solver (LBFS) [10,11,31], which is in fact a finite volume solver with both viscous and inviscid fluxes evaluated simultaneously by local reconstructions of LBM solutions at each surface.

In most of the applications of IBM, a uniformed Cartesian grid will be used, and this will result in a sharp increase in the grid size. As estimated by Mittal and Iaccarino [4], the grid-size ratio of a Cartesian grid to a body-conformal grid will increase with $Re^{1.5}$ for a three-dimensional problem if the Reynolds number ($Re$) increases. In order to overcome this drawback, adaptive method was introduced to incorporate with IBM [26,30,32–36]. In this work, we are going to present an h-adaptive implicit immersed boundary-lattice Boltzmann flux solver based on JASMIN infrastructure [37]. JASMIN, which is the abbreviation of J parallel Adaptive Structured Mesh applications INfrastructure, is a software infrastructure developed by the Institute of Applied Physics and Computational Mathematics (IAPCM) in 2004. Its main objective is to accelerate the development of parallel programs for large scale simulations of complex applications on parallel computers. Based on this infrastructure, the IB-LBFS method is successfully integrated with an h-adaptive mesh refinement technique (AMR), and four different canonical flow problems, both two-dimensional (2D) and three-dimensional (3D), stationary as well as moving boundary problems, are used to validate the proposed AMR-IB-LBFS method. Furthermore, an alternative momentum-based splitting, which is

* Corresponding authors.
*E-mail addresses:* xiazh@zju.edu.cn (Z. Xia), caiqd@pku.edu.cn (Q. Cai).

https://doi.org/10.1016/j.compfluid.2017.11.003
0045-7930/

based on the conserved macroscale quantities as the AMR technique does, is proposed during the factional step. Our simulation results have demonstrated the accuracy and capability of the AMR-IB-LBFS method in simulating 2D/3D flow problems with stationary/moving boundaries.

The rest part of this paper is organized as follows. Section 2 will be devoted to introduce the AMR-IB-LBFS method, including a brief review on IB-LBFS in Section 2.1, an introduction on the structured adaptive mesh refinement (SAMR) technique in Section 2.2 and a summary on AMR-IB-LBFS in Section 2.3. The numerical results and discussions will be presented in Section 3. Four test cases will be simulated, including the flow around a stationary 2D circular cylinder in Section 3.1, the sedimentation of a 2D elliptical particle in Section 3.2, the flow around a stationary sphere in Section 3.3 and the sedimentation of a single sphere in Section 3.4. Finally, the conclusions will be drawn in Section 4.

## 2. AMR-IB-LBFS

### 2.1. Brief review on IB-LBFS

In this section, a brief review on the IB-LBFS method will be presented. For an incompressible viscous flow, the governing equations of mass and momentum conservation can be written as [31]

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = 0 \tag{1}$$

$$\frac{\partial \rho \boldsymbol{u}}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}\boldsymbol{u}) = -\nabla p + \nabla \cdot [\mu(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T)] + \boldsymbol{f} \tag{2}$$

where $\rho$, $\boldsymbol{u}$, $p$ and $\mu$ are the fluid density, flow velocity, pressure and dynamic viscosity, respectively. $\boldsymbol{f}$ is the restoring force that is used to reproduce the effect of boundary, and it is determined by IBM. In this paper, we will review the two-dimensional situation, the three-dimensional IB-LBFS is similar and can be found in the paper by Wang et al. [11].

For a two-dimensional flow, a finite-volume method (FVM) together with a fractional step method are adopted to discrete Eqs. (1) and (2), and they can be written as [10]

$$\frac{1}{\Delta t}\begin{pmatrix} \rho^{n+1} - \rho^n \\ \rho^{n+1}\boldsymbol{u}^* - \rho^n \boldsymbol{u}^n \end{pmatrix} = -\frac{1}{h}\sum_k \boldsymbol{G}_k \tag{3}$$

$$\frac{\rho^{n+1}\boldsymbol{u}^{n+1} - \rho^{n+1}\boldsymbol{u}^*}{\Delta t} = \boldsymbol{f} \tag{4}$$

where $\Delta t$ and $h$ are the time step and mesh size, while $\boldsymbol{G}_k$ is combination of viscous and inviscid fluxes. In IB-LBFS, $\boldsymbol{G}_k$ will be estimated based on the LBM. The readers can refer to references [10,31] for more details on the flux estimation. In this paper, we will give more description on the detailed implementation of IBM.

Differently from the velocity-based splitting used in Eqs. (3) and (4), here we are using the momentum-based splitting in the realization of the fractional step method. We denote $\boldsymbol{w} \equiv \rho \boldsymbol{u}$, Eqs. (1) and (2) can also be discretized as follows,

$$\frac{1}{\Delta t}\begin{pmatrix} \rho^{n+1} - \rho^n \\ \boldsymbol{w}^* - \boldsymbol{w}^n \end{pmatrix} = -\frac{1}{h}\sum_k \boldsymbol{G}_k \tag{5}$$

$$\frac{\boldsymbol{w}^{n+1} - \boldsymbol{w}^*}{\Delta t} = \boldsymbol{f} \tag{6}$$

From Eq. (5), the conserved quantities $\rho^{n+1}$ and $\boldsymbol{w}^*$ can be obtained, and the conserved quantity $\boldsymbol{w}^{n+1}$ can be acquired through Eq. (6) after the boundary condition enforced IBM [10] is implemented. Denoting $\delta \boldsymbol{w} = \boldsymbol{w}^{n+1} - \boldsymbol{w}^*$, the fluid momentum can be corrected through

$$\boldsymbol{w}^{n+1} = \boldsymbol{w}^* + \delta \boldsymbol{w} \tag{7}$$

and Eq. (6) can be rewritten as

$$\boldsymbol{f} = \frac{\delta \boldsymbol{w}}{\Delta t}. \tag{8}$$

In order to satisfy the non-slip boundary condition, the fluid velocity on the boundary point must equal the boundary velocity $\boldsymbol{U}_B$ at the same position. The fluid velocity on the boundary point can be obtained through interpolation of $\boldsymbol{w}^{n+1}$ and $\rho^{n+1}$ using the smooth delta function, and this relation can be written as

$$\frac{\sum_{ij} \boldsymbol{w}^{n+1}(\boldsymbol{x}_{ij}) D(\boldsymbol{x}_{ij} - \boldsymbol{X}_B^l)}{\sum_{ij} \rho^{n+1}(\boldsymbol{x}_{ij}) D(\boldsymbol{x}_{ij} - \boldsymbol{X}_B^l)} = \mathbf{U}_B^l(\boldsymbol{X}_B^l), \tag{9}$$

where, $l = 1, \cdots, N_p$ is the index of the $N_p$ Lagrangian points, $\boldsymbol{X}_B^l = (X_B^l, Y_B^l)$ is the coordinate of the $l$th Lagrangian point, $\boldsymbol{x}_{ij} = (x_{ij}, y_{ij})$ is the coordinate of the Eulerian point, and $D(\boldsymbol{x}_{ij} - \boldsymbol{X}_B^l) = \delta(\frac{x_{ij}-X_B^l}{h})\delta(\frac{y_{ij}-Y_B^l}{h})$ is the smooth delta function. $\delta(r)$ is the one-dimensional delta function, and can be defined as

$$\delta(r) = \begin{cases} \frac{1}{4}\left(1 + \cos\left(\frac{\pi r}{2}\right)\right), & |r| \le 2, \\ 0, & |r| > 2, \end{cases} \tag{10}$$

substituting Eq. (7) into Eq. (9),

$$\sum_{ij} \delta \boldsymbol{w}(\boldsymbol{x}_{ij}) D(\boldsymbol{x}_{ij} - \boldsymbol{X}_B^l) = \mathbf{U}_B^l(\boldsymbol{X}_B^l) \sum_{ij} \rho^{n+1}(\boldsymbol{x}_{ij}) D(\boldsymbol{x}_{ij} - \boldsymbol{X}_B^l) - \sum_{ij} \boldsymbol{w}^*(\boldsymbol{x}_{ij}) D(\boldsymbol{x}_{ij} - \boldsymbol{X}_B^l) \tag{11}$$

Note that $\delta \boldsymbol{w}$ can also be estimated through the boundary momentum correction $\delta \boldsymbol{w}_B$, that is, $\delta \boldsymbol{w}(\boldsymbol{x}_{ij}) = \sum_l \delta \boldsymbol{w}_B(\boldsymbol{X}_B^l) D(\boldsymbol{x}_{ij} - \boldsymbol{X}_B^l)$, Eq. (11) can be rewritten as follows

$$a_{lk} x_k = b_l. \tag{12}$$

Here $x_k = \delta \boldsymbol{w}_B(\boldsymbol{X}_B^k)$ is the unknown quantity, while $a_{lk} = \sum_{ij} D(\boldsymbol{x}_{ij} - \boldsymbol{X}_B^l) D(\boldsymbol{x}_{ij} - \boldsymbol{X}_B^k)$ and $b_l = \mathbf{U}_B^l(\boldsymbol{X}_B^l) \sum_{ij} \rho^{n+1}(\boldsymbol{x}_{ij}) D(\boldsymbol{x}_{ij} - \boldsymbol{X}_B^l) - \sum_{ij} \boldsymbol{w}^*(\boldsymbol{x}_{ij}) D(\boldsymbol{x}_{ij} - \boldsymbol{X}_B^l)$ are known. It is easy to verify that the matrix coefficient $a_{lk}$ is a banded sparse symmetric matrix and strictly diagonal dominant. In the present work, the Jacobi iteration is applied to solve the system of equations. After the above system of equations are solved, the flow field can be updated accordingly. The forces and torques exerted on the body by the surrounding fluid can be estimated by using the expressions in reference [10], where the internal mass effects of moving objects were included [38].

It should be emphasized that the present momentum-based splitting is different from the velocity-based splitting. In the velocity-based splitting, the fluid velocity is the main quantity all through the implementation of IBM. Only the interpolation of velocity is needed if one wants to get the flow field, but another interpolation of density should be performed if one also needs to accurately estimate the force exerted on the object. In the present method, we do not have to calculate the fluid velocity but use the interpolations of momentum and density to get the velocity on the Lagrangian points all through the implementation of IBM (In the implementation of LBFS, we can also use the momentum and density without calculate the velocity). In general, the error $\varepsilon$ between these two methods

$$\varepsilon(\boldsymbol{X}_B^l) \equiv \frac{\sum_{ij} \boldsymbol{w}(\boldsymbol{x}_{ij}) D(\boldsymbol{x}_{ij} - \boldsymbol{X}_B^l)}{\sum_{ij} \rho(\boldsymbol{x}_{ij}) D(\boldsymbol{x}_{ij} - \boldsymbol{X}_B^l)} - \sum_{ij} D(\boldsymbol{x}_{ij} - \boldsymbol{X}_B^l) \frac{\boldsymbol{w}(\boldsymbol{x}_{ij})}{\rho(\boldsymbol{x}_{ij})} \tag{13}$$

is not zero, especially when the present smooth function $D$ is used. However, the local density variations $\rho'$ at Eulerian points around