Fast Calculation of Hydrodynamic Interaction among Particles in the Stokes Flows

Kengo Ichiki *)

Graduate School of Human and Environmental Studies, Kyoto University, Kyoto 606-8501

(Received)

This work presents an efficient computational method of hydrodynamic interaction among spherical particles in low-Reynolds-number flows using real-space multipole expansion in Cartesian coordinate, iterative method, and fast multipole method.

§1. Introduction

The aim of this work is to present an efficient computational method of hydrodynamic interaction among spherical particles in low-Reynolds-number flows. A standard method called Stokesian Dynamics [1] has two main difficulties: the heavy computational load which is scaled by $O(N^3)$ with the number of particle N, and no systematic way to improve approximation. These will overcome in terms of multipole expansion, iterative scheme and fast multipole method.

§2. Formulation and Results

We use simple derivatives in Cartesian coordinate rather than harmonic functions in spherical coordinate. This has advantage of implementation on computer; the formulation might be lengthy, but simple and fast to calculate.

Let us consider many-body problem of mobility relation in the Stokes flow. Using the Taylor expansion at the center of particles, we obtain systematic formulation of mobility problem where velocities are presented by derivatives of mobility tensor and force moments.

To demonstrate this expansion scheme, we consider two-body problems with the point-force approximation with Oseen tensor. Forces are given as $\boldsymbol{F} = (1, 1, 1)$. The separation between the expansion-points is denoted as r. Figure 1 shows the errors for various separations r's and orders of expansion p's.

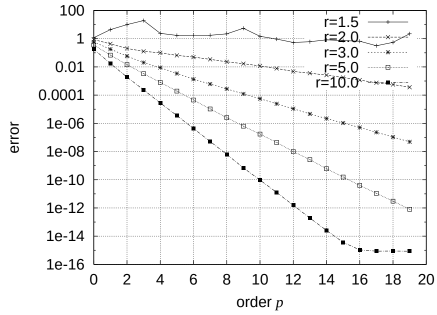

Fig. 1. Error with separation r for order p's.

We need to calculate inverse in Stokesian Dynamics where many-body resistance

*) E-mail address: ichiki@kona.jinkan.kyoto-u.ac.jp

matrix R is approximated as $\mathsf{R} = \mathsf{M}^{-1} + \mathsf{L}$, where M and L are mobility tensor with low order in expansion and lubrication tensor which contains contribution of higher order [1]. This inversion takes into account many-body effect in way of method-of-reflections which is one of iterative method [2]. In turn, the application of iterative method which consists of the following matrix-vector product $\boldsymbol{U} = \mathsf{M} \cdot \boldsymbol{F}$ is the first step to improve the calculation time which is scaled as $O(N^2)$ [*] .

Further improvement on speed is achieved by the fast multipole method (FMM) [4]. The point of FMM is to expand not only with the position of force but also with that of velocity [**] .

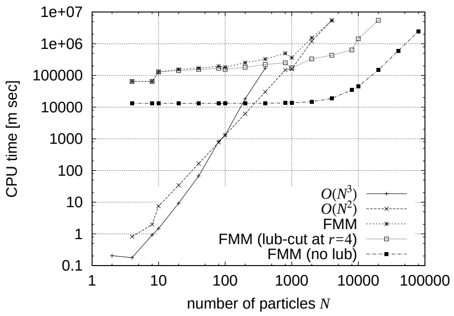

Fig. 2. CPU time with number of particles N for various schemes.

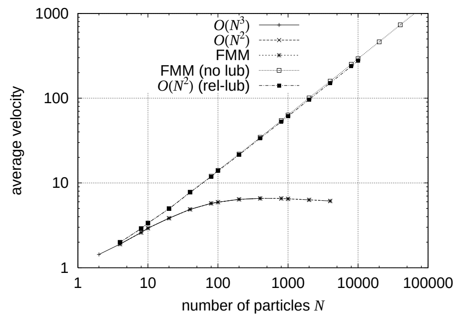

Fig. 3. Average velocity of particles.

We calculate mobility problem by various methods with the configuration of simple-cubic lattice. Figure 2 shows CPU time. The Stokesian Dynamics is labeled as $O(N^3)$ and method with iterative scheme Orthomin is labeled as $O(N^2)$. For system with more than 1000 particles, method with FMM is comparable with $O(N^2)$. Introducing cut-off length on lubrication makes the scheme efficient, which is labeled as FMM (lub-cut).

Figure 3 shows the average velocity. The average velocity should be scaled as $N^{2/3}$. This shows that the lubrication in the original Stokesian Dynamics fails. Excluding collective motion on calculation of lubrication [5] helps this situation.

Acknowledgments

The author would like to thank Prof. John F. Brady for fruitful discussions on the subject. This work is supported by Japan Society for the Promotion of Science.

References

1) J. F. Brady and G. Bossis, Annu. Rev. Fluid Mech. **20**, 111 (1988).
2) K. Ichiki and J. F. Brady, submitted to Phys. Fluids.
3) R. Weiss, *Parameter-Free Iterative Linear Solvers* (Akademie Verlag, Berlin, 1996).
4) L. Greengard and V. Rokhlin, J. Comput. Phys. **73**, 325 (1987).
5) B. Cichocki, M. L. Ekiel-Jeżewska, E. Wajnryb, J. Chem. Phys. **111**, 3265 (1999).

[*] For iterative schemes, see [3] for example.
[**] For details, see the original paper [4].