GPU-based parallel computing in large-scale rigid-particle suspensions modeling

Zhe Chen*[1]

[1]Courant Institute of Mathematical Sciences, New York University

February 2019

1 Introduction

Modeling of colloidal suspensions is an interesting topic and modeling of active suspensions has especially become very popular nowadays. For large number of suspension particles, large-scale collective behavior can be observed and is what people desire to study most. However, it's really hard to simulate large-scale system in complex fluid since good scaling is hard to reach. Here, I'm going to introduce an approach to studying suspensions of rigid particles using Pycuda[2] which implements parallel computing on GPU. And this is our group's work, which is public on Github[1], and I'm currently starting to work on this. This method is developed to solve modeling of passive and active rigid-particle in stokes flow efficiently with good scaling. In this method, a rigid multi-blob method is used to model rigid particles with arbitrary geometry. The main computational cost of this method is the product of blob-blob mobility matrix and a vector. For free boundary problem, Rotne-Prager-Yamakawa tenser combined with fast multipole method can be reached analytically, which helps it scales in linear in the number of particles. However, with situation where there's a no-slip wall at the bottom of this system which is common in real life, RPY tensor approach is not reachable, thus we switch to GPU parallel computing, where this product is computed by direct summation. In our work an asymptotic quadratic scaling in the number of particles is reached.[1]

The main idea of accelerating this product is that direct summation on GPU is more efficient than CPU based asymptotically-optimal FMMs(even with multicore acceleration) for systems of less than hundreds of particles. And the commutation between CPU and GPU is only particles' location. Thus, it is just perfect to use GPU paralleling acceleration to reach scaling of O(N*log(N)),

*zc1291@nyu.edu

[1] https://github.com/stochasticHydroTools/RigidMultiblobsWall

which is even better than O(N) method on CPU because of paralleling. Moreover, we should notice that FMMs method is only reachable for free boundary problem. For problem with a wall, RPY tensor can not be got and GPU paralleling is the better way.

With this method, we can model a system as large as millions of blobs on our group's computer(Titan X and Pascal). In general, this is super helpful in large scale complex fluid modelling though it's just trivial application of GPU paralleling computing, which is interesting.

References

[1] Florencio Balboa Usabiaga, Bakytzhan Kallemov, Blaise Delmotte, Amneet Bhalla, Boyce Griffith, and Aleksandar Donev. Hydrodynamics of suspensions of passive and active rigid particles: a rigid multiblob approach. *Communications in Applied Mathematics and Computational Science*, 11(2):217–296, 2017.

[2] Andreas Klöckner, Nicolas Pinto, Yunsup Lee, Bryan Catanzaro, Paul Ivanov, and Ahmed Fasih. Pycuda and pyopencl: A scripting-based approach to gpu run-time code generation. *Parallel Computing*, 38(3):157–174, 2012.