

Boundary element analysis of the multi-capsule flow using an ultra-high speed GPGPU computation

D. Matsunaga¹, Y. Imai¹, T. Ishikawa¹ & T. Yamaguchi²

¹*Department of Bioengineering and Robotics, Tohoku University, Japan*

²*Department of Biomedical Engineering, Tohoku University, Japan*

Abstract

A capsule is a liquid drop enclosed by a thin membrane. Understanding the mechanics of capsule suspension is important in a variety of applications. However, computational time is still a barrier against successful simulations for dense suspension of capsules. We have developed an ultra-high speed computation method based on Graphics Processing Unit (GPU) computing for simulating multi-capsule flow.

In the case of using single GPU, the performance achieves 1.20 TFlops that corresponds to a 550-fold speed up compared with a CPU core. We also developed an algorithm of multi GPUs that hide data communication by overlapping with computational time. Since our method enables us to simulate from one to a few hundred capsules in a realistic computational time, the method could be a powerful solution for simulating the capsule suspension.

1 Introduction

A capsule is a liquid drop enclosed by a deformable membrane. Biological cells and membrane bounded drugs can be classified into capsules. To simulate the motion of capsules, a variety of numerical methods, such as boundary element (BE) method and immersed boundary (IB) method, have been employed. The BE method, which is used in the present work, is the most accurate method to simulate the motion of capsules, since the method directly treats jump condition at the membrane without any interpolation. Various studies have been conducted to simulate the detailed motion of capsules using this method, such as deformation



of single capsule in shear flow [1] and interaction of two capsules [2]. Recently, Walter *et al.* [3] proposed a coupling method of the BE method with the finite element (FE) method of membrane mechanics to improve the stability of the simulation. However, the analyzing scale of pervious studies was still limited in few capsules, because of heavy computational load of BE method.

To overcome this issue, we have developed a full-GPU implementation of this coupling method [4]. A GPU is a high performance processor that consist of many streaming processors and large-bandwidth memory. GPU computing has been used for accelerating a variety of numerical simulations in recent years [5]. We achieved over 1.0 TFlops that corresponds to 550-fold speedup compared with a CPU core. We also extended the method into multi GPU, by developing the algorithm that hides data communication time. The multi GPU ensured good scaling and achieve maximum 18.4 TFlops when using 16 GPUs [4].

2 Numerical method

2.1 Problem statement

Consider M capsules with radius a suspended in a Stokes flow with undisturbed flow \underline{v}^∞ . The viscosity of internal and external fluid are consider as same viscosity μ and density ρ , for simplicity. The membrane mechanics and the fluid mechanics are coupled using boundary condition on the capsule surface.

2.2 Governing equations

2.2.1 Membrane mechanics

In the present study, the capsule membrane is modeled as two dimensional sheet that can neglect the effect of bending rigidity. When the inertia is neglected, the equilibrium equation can be described, by applying virtual work principal, as:

$$\int_{S_m} \hat{\underline{u}} \cdot \underline{q} dS = \int_{S_m} \hat{\underline{\epsilon}} : \underline{T} dS, \quad (1)$$

where S_m is the surface of capsule m , $\hat{\underline{u}}$ is the virtual displacement, \underline{q} is the load on the membrane, $\hat{\underline{\epsilon}}$ is the virtual strain, and \underline{T} is the Cauchy tension of the membrane. The tension \underline{T} of the membrane can be described by using strain energy function w as

$$\underline{T} = \frac{1}{\det \underline{F}} \underline{F} \frac{\partial w}{\partial \underline{e}} \underline{F}^T, \quad (2)$$

where \underline{F} is the surface deformation gradient tensor and \underline{e} is the Green-Lagrange strain tensor.

2.2.2 Fluid mechanics

We assume the particle Reynolds number is much smaller than unity, and then the velocity field is described by the boundary integral formulation of the Stokes flow



over the surface of the capsules. The fluid velocity of the given observation point \underline{x} can be expressed as:

$$\underline{v}(\underline{x}) = \underline{v}^\infty(\underline{x}) - \frac{1}{8\pi\mu} \sum_m^M \int_{S_m} \underline{J}(\underline{x}, \underline{y}) \cdot [\underline{\sigma}(\underline{y})] \cdot \underline{n}(\underline{y}) dS(\underline{y}), \quad (3)$$

where $[\underline{\sigma}] \cdot \underline{n}$ is the jump of viscous traction. The Green's function \underline{J} is

$$\underline{J}(\underline{x}, \underline{y}) = \frac{1}{|\underline{r}|} + \frac{r_i r_j}{|\underline{r}|^3}, \quad (4)$$

where $\underline{r} = \underline{x} - \underline{y}$.

2.2.3 Boundary conditions

The membrane velocity of a material point matches the fluid velocity at the material point with the kinematic condition:

$$\frac{d\underline{x}}{dt} = \underline{v}(\underline{x}). \quad (5)$$

Also, the dynamic condition leads to the balance between the load on the membrane and the viscous traction jump as:

$$\underline{q} = [\underline{\sigma}] \cdot \underline{n}. \quad (6)$$

2.3 Discretization

The capsule surface is discretized with N_E triangular elements and N_N nodes. In the present study, we follow the discretized method developed by Walter *et al.* [3]. The membrane mechanics is solved by discretizing the equation (1) to finite element (FE) matrix. Fluid velocity is calculated by discretizing equation (3) using Gaussian quadrature method as

$$\underline{v}(\underline{x}) = \underline{v}^\infty(\underline{x}) - \frac{1}{8\pi\mu} \sum_m^{N_E} \sum_e^e \sum_{gp}^{N_{gp}} \underline{J}(\underline{x}, \underline{y}) \cdot \underline{q}(\underline{y}) \omega_{gp} \Delta S_e, \quad (7)$$

where N_{gp} is the number of Gaussian points, w_{gp} is the weight of Gaussian quadrature and ΔS_e is the surface area of the element e . Since the fluid velocity is couple with membrane velocity at the surface, as defined in equation (5), the membrane velocity of a given capsule node \underline{x}_i can be computed by substituting \underline{x}_i into equation (7). The membrane position is updated using 2nd order Runge-Kutta method.

3 Single GPU implementation

3.1 Implementation and validation

Computer unified device architecture (CUDA) is used for the implementation on GPU computing. GPU consists of few hundreds of cores that the device can handle



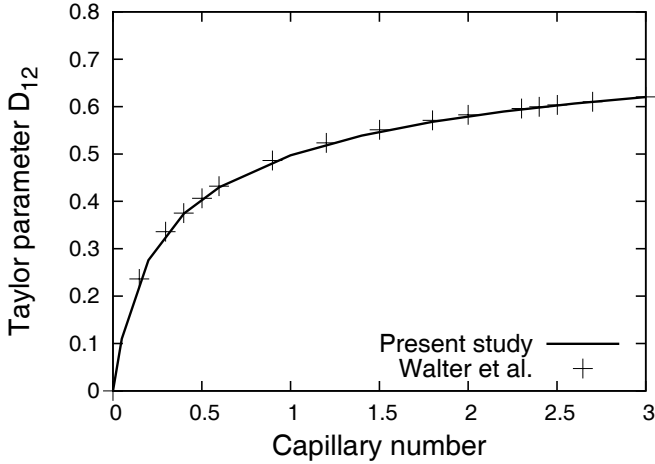


Figure 1: Validation of capsule deformation in simple shear flow.

multiple operation at once. The equation (7) is solved in parallel by making i -th GPU thread to be responsible for calculating the membrane velocity of x_i . We also parallelize the FE matrix solution using GPU-implemented BiCGStab solver.

We check the capsule deformation in simple shear flow, to validate the simulation result. The deformation of the capsule is measured by Taylor parameter D_{12} :

$$D_{12} = \frac{|L_1 - L_2|}{L_1 + L_2}, \quad (8)$$

where L_1 and L_2 is the lengths of the principal axes of the capsule in the shear plane. Figure 1 shows the deformation of capsules following Skalak ($C=1$) law [6]. Capillary number Ca is the non-dimensional variable defined as

$$Ca = \frac{\mu \dot{\gamma} a}{G_s} \quad (9)$$

where $\dot{\gamma}$ is the shear rate and G_s is the surface shear modulus. Since the result has good agreement with previous study [3], the accuracy of our GPU program is ensured.

3.2 Result

For the performance benchmark, we simulate the capsule (single capsule for $N_N = 2562$, $N_E = 5120$) flow by changing the number of capsules M from 1 to 64. Figure 2 shows the speedup ratio of GPU computation (nVIDIA Geforce GTX 580) comparing with a CPU core (Intel Core i7-930). The speedup ratio increase with the increase in capsule number M . The figure shows single precision

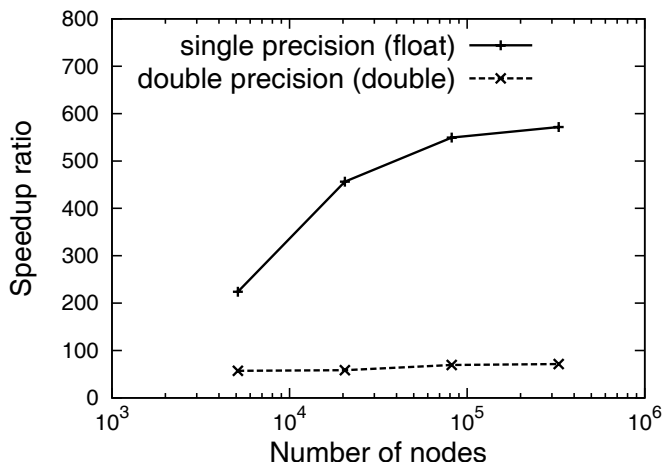


Figure 2: Speedup ratio by using GPU (nVIDIA Geforce GTX 580) compared with CPU (Intel Core i7 930).

operation shows maximum 550-fold speedup, while double precision achieve 70-fold speedup. The performance of whole computation *Flops* is estimated by

$$Flops = \frac{FLOP}{t} \quad (10)$$

where t is the total computational time, and $FLOP$ is the total number of floating-points operations. In the simulation of single precision, GPU achieves maximum 1.20 TFlops, while CPU (Intel Core i7-930) reaches only 2.10×10^{-3} TFlops. In the case of double precision, GPU achieves maximum 0.14 TFlops, while CPU reaches 1.95×10^{-3} TFlops. Although the performance of CPU do not differ in single and double precision, GPU performance decrease when using double precision.

4 Multi GPU implementation

4.1 Implementation

Consider simulation of M capsules on K number of GPUs. Before starting simulation in multi GPUs, the capsules are divided into a K groups of M/K capsules and make each GPU to be responsible for one capsule group, to equalize the load balance.

Since the FE matrix solution is closed for each capsule, data communications among GPU are not need in this procedure. During the integral procedure of BE method, by contrast, each GPU has to communicate data to complete the surface integral of equation (7). Data communication among GPUs are executed in

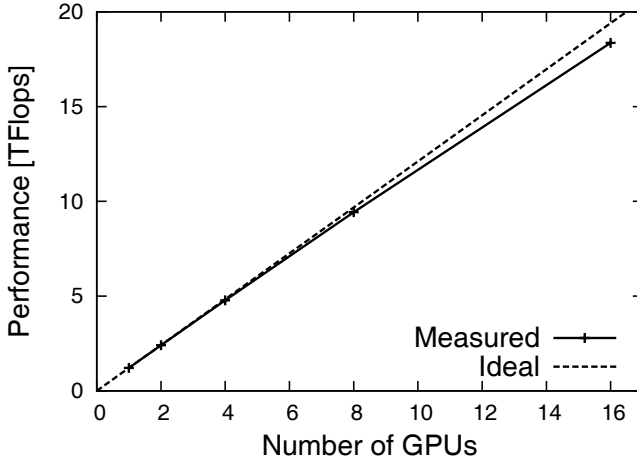


Figure 3: Performance of multi GPU up to 16 GPUs when computing 256 capsules.

following three steps. First the each GPU transfer data to their host CPU memory (device to host communication). Next CPU exchange the capsule data to other CPUs by using MPI library. Finally, each CPU transfer the capsule data to their GPU memory (host to device communication).

If the communication time could not be neglected with the computational time, the communication time would be the overhead of this parallelization. To avoid the overhead, we hide the communication time by developing algorithm to execute the data communication and computation concurrently.

4.2 Result

In the present study, we use 16 machines each consists of CPU (Intel Xeon E5620) and GPU (nVIDIA GTX 580). All the machines is connected to others by 10Gbps QDR-Infiniband. Figure 3 shows the performance of multi GPU when computing 256 capsules, and the figure shows performance achieves 18.4 TFlops when using 16GPUs. If we use k number of GPUs, we can expect k times larger performance than single GPU case. From the figure, there are only slight decrease from ideal performance that it can be conclude that our algorithm have good scaling .

By using 16 GPUs, we can calculate 256 (Figure 4) in only 1.7 days (for 10,000 iterations).

5 Conclusion

We have developed a GPU implementation method to simulate capsule flow simulation under Stokes flow. By single GPU, we achieve 1.20 TFlops



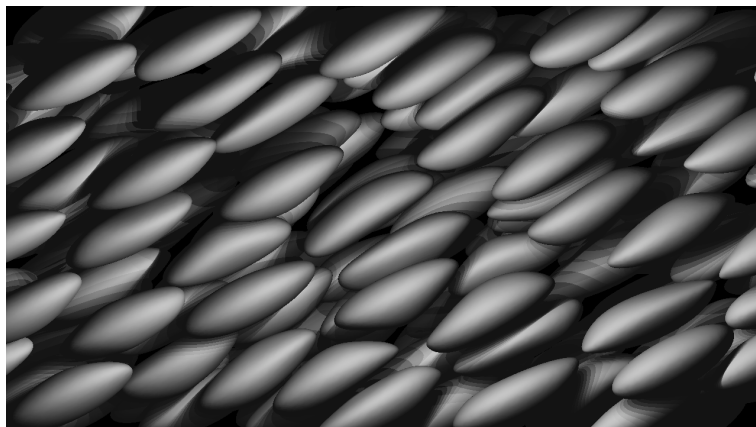


Figure 4: Simulation of 256 capsules in simple shear flow.

performance that correspond to 550 times faster than CPU. We also developed a multi GPU implementation method, which hides the data communication by computational time. The performance have successfully demonstrated the scalability, reaching the performance of 18.4 TFlops when simulating 256 capsules (256×2562 nodes) in 16 GPUs. By using our method, 16 GPUs can solve the interactions of 256 capsules in 1.7 days (for 10,000 iterations).

Although BE method have been treated as one of the most accurate method to model dynamics of suspended capsules, the computational load had been major issue. Since our method enable us to simulate from one to few hundred capsules in a realistic computational time, the method would become powerful solution to simulate the capsule suspension.

References

- [1] Pozrikidis, C., Finite deformation of liquid capsules enclosed by elastic membranes in simple shear flow. *Journal of Fluid Mechanics*, **297**, pp. 123–152, 1995.
- [2] Lac, E. & Barthès-Biesel, D., Hydrodynamic interaction between two identical capsules in simple shear flow. *Journal of Fluid Mechanics*, **573**, pp. 149–169, 2007.
- [3] Walter, J., Salsac, A.V., Barthès-Biesel, D. & Le Tallec, P., Coupling of finite element and boundary integral methods for a capsule in a Stokes flow. *International Journal for Numerical Methods in Engineering*, **83(7)**, pp. 829–850, 2010.
- [4] Matsunaga, D., Imai, Y., Omori, T., Ishikawa, T. & Yamaguchi, T., A full gpu implementation of the boundary element method for simulating capsule suspension. *Journal of Computational Physics*, submitted.

- [5] Shimokawabe, T., Aoki, T., Ishida, J., Kawano, K. & Muroi, C., 145 Tflops performance on 3990 GPUs of TSUBAME 2.0 supercomputer for an operational weather prediction. *Procedia Computer Science*, **4(0)**, pp. 1535–1544, 2011. Proceedings of the International Conference on Computational Science, ICCS 2011.
- [6] Skalak, R., Tozeren, A., Zarda, R. & Chien, S., Strain energy function of red blood cell membranes. *Biophysical Journal*, **13(3)**, pp. 245–264, 1973.

