GPU ccelerated Fast Multipole Methods for Dynamic N-body Simulation

Qi Hu, Nail A. Gumerov, Ramani Duraiswami
University of Maryland Institute of Advanced Computer Study
Email: huqi,gumerov,ramani@umiacs.umd.edu

Many physics based simulations can be efficiently and accurately performed using particle methods which focus computational resources at the location of sources or discontinuities (particles), and evaluation of relevant fields at locations of interest. These particle methods result in the so-called N-body problem, in which the sum of N kernel functions Φ centered at N source locations $\mathbf{x}_i$ with strengths q_i are evaluated at M receiver locations $\{\mathbf{y}_j\}$ in $\mathbb{R}^d$ (see Eq. 1),

$$\phi(\mathbf{y}_j) = \sum_{i=1}^{N} q_i \Phi(\mathbf{y}_j - \mathbf{x}_i), \quad j = 1, \ldots, M, \quad \mathbf{x}_i, \mathbf{y}_j \in \mathbb{R}^d, \tag{1}$$

The N-body problem also arises in interpolation using implicit functions, in simulation of molecular and stellar dynamics, RBF interpolation and other areas of interest to the computational fluid dynamic communities.

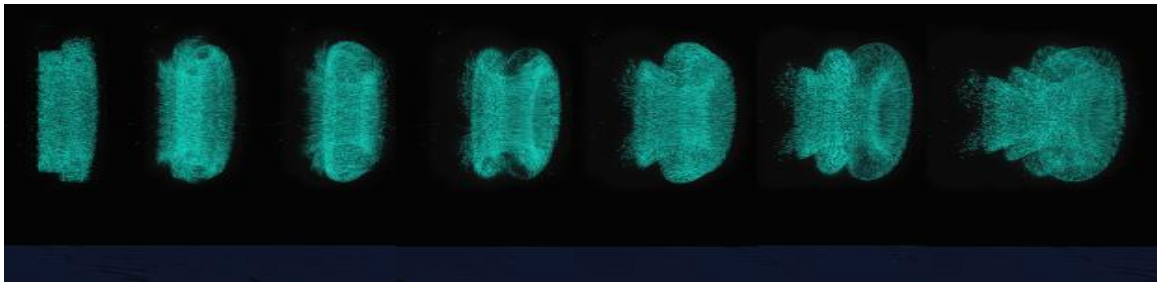

Figure 1: The vortex ring leapfrog

Fast and accurate N-body simulations are the goal of this paper. Recently, as tremendous advances of graphics processors (GPU), it is possible to perform large size simulations for high fidelity on a single desktop. In this paper, the Fast Multipole Method (FMM) has been proposed for the case of dynamic N-body simulations using vortex methods on graphics processors. Such approximation algorithm reduces its $O(NM)$ cost per time step to $O(N + M)$ but can achieve any precision which is controlled by the truncation number p. Our major contributions include providing efficient data structures implemented on GPUs, and a novel parallel formulation of the FMM on GPUs to address this problem. It shows the capacity of running million size dynamic N-body simulations on a single GPU. Our FMM translations and expansions use real number representations as opposed to the usual complex spherical harmonic based representation usually provided. This allows for GPU computation efficiency. We adapt the baseline FMM, which computes the scalar $1/r$ Coulomb kernel, to compute the vector Green's function kernel (the Biot-Savart kernel). As an example application, we simulate the interactions between vortex rings and 1 million fluid particles on a single desktop (vortex methods with the smoothing kernel) which shown in Figure 1. Except for the initial setup, our approach process all the computations and updates on GPU without expensive communications between CPU and GPU.

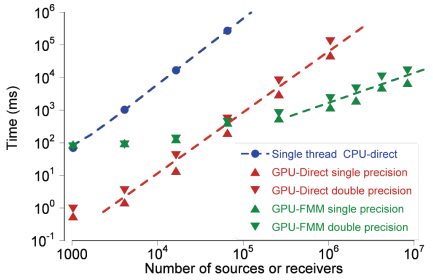

Figure 2: FMM performance comparisons

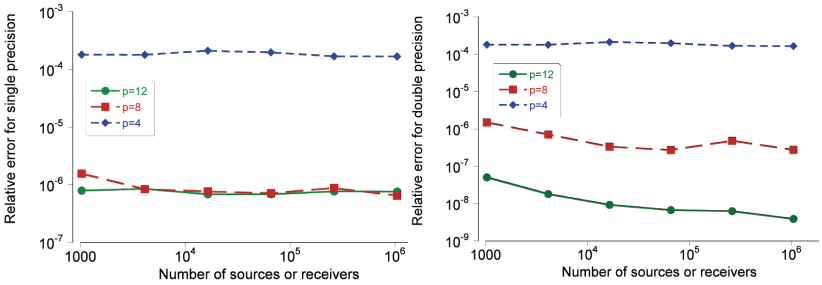

Figure 3&4: FMM relative error for single and double precision

[1] and [2] both developed the GPU-based FMM. However, in these implementations, the data structures were built on the CPU, which is too expensive for dynamic problems. [3] developed a CPU-GPU-Hybrid *treecode* to accelerate the computation, but its overall performance does not outperform the implementation presented in [1]. Recently, [4] achieves the state of the art performance by developing the hybrid FMM algorithm on the heterogeneous architectures, which is only for the scalar Coulomb kernel and has not been applied to vortex methods yet.

Based on our performance evaluation, by using GPU parallel algorithm, the data structure overheads are reduced up to 0.1 sec as opposed to 2-8 sec required for CPU for million size problems, which provides substantial savings if particle positions need to be regenerated every time step. The total running times analysis of FMM on different sizes and its error analysis are summarized in Figure 2 and Figure 3&4 respectively. We also show that even though the baseline FMM algorithm is called three times, since we combine them together to reduce the GPU global memory access, the full FMM computation time of Biot-Savart kernel is not tripled but even less than doubled compared with the baseline FMM running time.

References

[1] N. A. Gumerov and R. Duraiswami, "Fast multipole methods on graphics processors," *J. Comput. Phys.*, vol. 227, pp. 8290–8313, September 2008.

[2] R. Yokota, T. Narumi, R. Sakamaki, S. Kameoka, S. Obi, and K. Yasuoka, "Fast multipole methods on a cluster of GPUs for the meshless simulation of turbulence," *Computer Physics Communications*, vol. 180, no. 11, pp. 2066–2078, 2009.

[3] M. J. Stock and A. Gharakhani, "Toward efficient GPU-accelerated n-body simulations," in *46th AIAA Aerospace Sciences Meeting and Exhibit, AIAA 2008-608*, January 2008.

[4] Q. Hu, N. A. Gumerov, and R. Duraiswami, "Scalable fast multipole methods on distributed heterogeneous architectures," in *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '11, (New York, NY, USA), pp. 36:1–36:12, ACM, 2011.