

Unity によるマイクロマグネティクスシミュレーションの可視化

LI JIAQING[†] 成見 哲[‡]
 電気通信大学 情報理工学研究科^{†‡}

1 はじめに

自然現象の解析や分析は工業製品の開発に欠かせないものであり、日頃から盛んに行われている。リアルタイムに可視化しながらシミュレーションできれば、解析サイクルを早く出来る。画像処理を担当する Graphics Processing Unit(GPU)に汎用計算を行わせる GPGPU が近年流行しており[1]、シミュレーションした結果をそのまま描画にも使用することで可視化しながらシミュレーションできる。

本研究では、Unity におけるマイクロマグネティクスシミュレーションのリアルタイム可視化システムを開発する。Unity の機能を使って、Compute shader(cs)を用いてシミュレーションを高速化し、Shader プログラムを用いて描画する。NVIDIA の GPU だけではなく、他のブランドの GPU、あるいは CPU 内蔵 GPU でシミュレーションを実行することができる。これにより、学生はスマートフォンなどのデバイスで手軽にシミュレーションができ、教育用途に役立つ。

2 マイクロマグネティクスシミュレーション

マイクロマグネティクスとは、磁石内部に現れる原子磁気モーメント[式(1)]によって作られる磁化構造や、その動的な変化を扱う分野である。

$$\dot{\vec{M}} = -\gamma \vec{M} \times \vec{H} + \frac{\alpha}{M} (\vec{M} \times \dot{\vec{M}}) \quad (1)$$

ここで \vec{M} は磁気モーメント、 $\dot{\vec{M}}$ は \vec{M} の時間微分、 \vec{H} は磁界、 γ と α は定数である。

\vec{H} は外部磁界 (\vec{H}^{EXT})、一軸異方性エネルギーによる磁界 (\vec{H}^A)、交換エネルギーによる磁界 (\vec{H}^E)、静磁界 (\vec{H}^D) を用いて以下のように表される。

$$\vec{H} = \vec{H}^{EXT} + \vec{H}^A + \vec{H}^E + \vec{H}^D \quad (2)$$

このうち静磁界の計算量は計算点の数を n とすると $O(n^2)$ であり、計算量が膨大である。そのため、リアルタイムでシミュレーションするには GPU での加速が不可欠である。

3 先行研究

CUDA でマイクロマグネティクスシミュレーションを行い、グラフィック API(OpenGL)を用いて可視化を行っている研究は存在している[2]が OpenGL で可視化プログラムを構築するのは手数がかかる。また、CUDA は NVIDIA の GPU だけをサポートし、別のブランドの GPU を使ったり、スマートフォンを使うと動作しないという問題がある。また、Unity と Compute shader でクロスシミュレーションを行う研究が存在している[3]。さらに、Compute shader は CUDA に負けない速度をもち、ボリュームレイキャスティングで CUDA より速いと報告されている[4]。本研究では、グラフィック API の Compute shader を用いて計算を加速している。

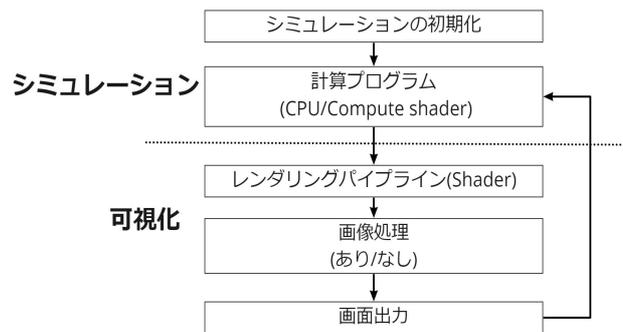


図 1: 可視化システムの概要

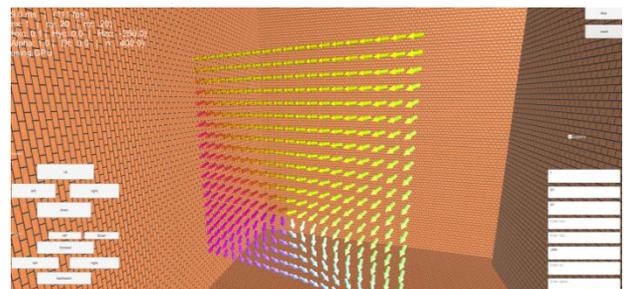


図 2: リアルタイム可視化システムの外見

Visualization of micromagnetics simulations with Unity
[†]JIAQING LI, Graduate School of Informatics and Engineering,
 The University of Electro-Communications
[‡]Tetsu NARUMI, Graduate School of Informatics and Engineering,
 The University of Electro-Communications

4 シミュレーションシステムの概要

図1は本システムでの処理の流れを示している。図2はアプリの外観である。本システムは主にシミュレーション計算プログラムと可視化プログラムの二つの部分がある。

シミュレーションプログラムでは、C#スクリプトでCPUベースのシミュレーションプログラムあるいはCompute shaderプログラムでGPUベースのシミュレーションプログラムを実行し、計算する。Compute shaderを選ぶとGPGPUにより、シミュレーションを高速に行う。

可視化プログラムでは、前述のシミュレーションプログラムで計算された結果をUnityのリアルタイムレンダリングパイプラインに送り、shaderプログラムを実行し、それぞれの色と方向が付けられた矢印ポリゴンを描画する。最後に、ポストプロセスプログラムにより、描画された画像を処理し、矢印ポリゴンをより観察しやすくすることができる[図3]。

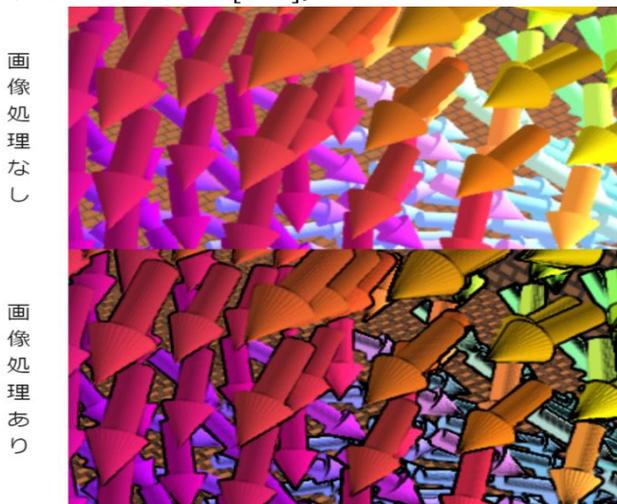


図3：画像処理のあり/なしの比較

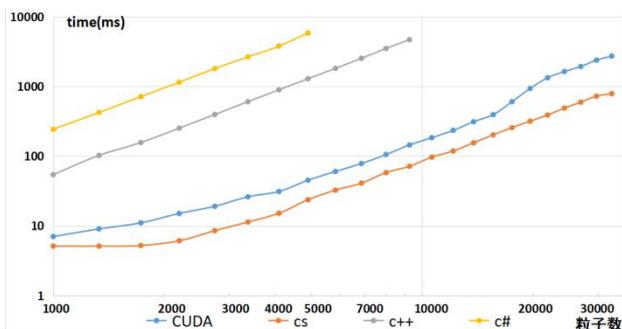


図4：性能比較1

表1：各スマートフォンでの性能(フレームレート)

機種	Pixel3a	GalaxyZ Flip3	xiaomi mi9t pro	Xperia XZ1
発売年	2019	2021	2019	2017
性能	8fps	34fps	27fps	20fps

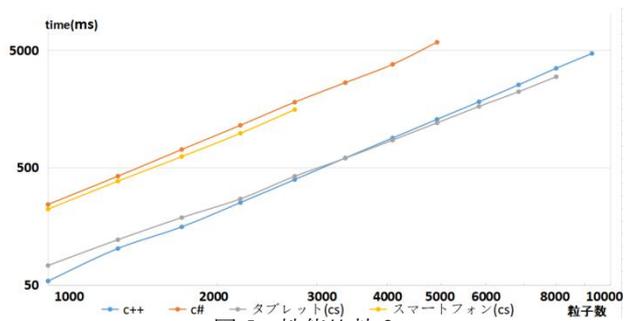


図5：性能比較2

5 性能評価

図4は同じデスクトップPCで計算方法を変えた時のフレームの計算時間を示している。Compute shaderはCUDAより2倍、CPUより60倍速い。CPUのC++はC#より3倍くらい速いが、粒子数が1000を超えるとリアルタイムでシミュレーションすることができなくなる。

図5は比較的性能の低いスマートフォン(OPPO AX7)とタブレット(Fire HD 10)の計算時間をPCのCPUと比較したものである。モバイル端末でもGPUを使えばデスクトップPCのCPU並の性能が出る。

表1は粒子数が1152の場合にスマートフォンでの性能を比較したものである。比較的新しいスマートフォンを用いればリアルタイムのシミュレーションが可能であることが分かる。

6 おわりに

Unityでマルチプラットフォーム対応のマイクロマグネティクスのリアルタイム可視化システムを開発した。本システムは他のGPGPU API(CUDAなど)で実装されるシステムに負けない速度をもち、スマートフォンでスムーズに実行できる。あらかじめドライバーをインストールする必要もなく、学生はスマートフォンなどのデバイスで手軽にシミュレーションができ、教育用途に役立つ可能性がある。

参考文献

- [1] Amos Egela, Lorenzo Pattelli, Giacomo Mazzamuto, Diederik S. Wiersma, Uli Lemmer, "CELES: CUDA-accelerated simulation of electromagnetic scattering by large ensembles of spheres," *Journal of Quantitative Spectroscopy & Radiative Transfer*, vol.199, pp.103-110, 2017
- [2] 清水陽介, マイクロマグネティクスシミュレーションの可視化, 電気通信大学情報理工学研究所 修士論文, 2012
- [3] Hongly Va, Min-Hyung Choi, Min Hong, "Real-Time Cloth Simulation Using Compute Shader in Unity3D for AR/VR Contents", *Multidisciplinary Digital Publishing Institute*, vol.11, issue.17, no.8255, 2021
- [4] Francisco Sans, Rhadamés Carmona, "A Comparison between GPU-based Volume Ray Casting Implementations: Fragment Shader, Compute Shader, OpenCL, and CUDA", *CLEI ELECTRONIC JOURNAL*, vol.20, no.2, 2017