1J-04

# 電位・電界シミュレーションの NVIDIA Ampere アーキテクチャへの最適化

小西 秀策 †
岡山理科大学大学院工学研究科情報工学専攻 †

上嶋 明 ‡
岡山理科大学工学部情報工学科 ‡

## 1 はじめに

Poisson 方程式を用いた電位・電界シミュレーションは, 半導体の動作解析や落雷の予測などの分野に応用されている. この手法には低コストで詳細な解析データを得られる利点がある一方で, 膨大な計算量を要するという課題が存在する. そのため, 高速化手法として GPGPU(General Purpose Computing on GPUs) を用いた実装 [1] が提案されている. 本研究では, Ampere アーキテクチャ採用 GPU である NVIDIA A100 を用いて電位・電界シミュレーションの高速化を図る. そして, プロファイリングに基づき Ampere アーキテクチャに合わせた最適化を行い, 先行研究のプログラムと比較してどの程度性能が向上したかについての評価を行う.

## 2 電位・電界シミュレーション

### 2.1 Poisson 方程式

電磁気学の分野において, Poisson 方程式は静電ポテンシャルの記述に用いられる. 与えられる電荷の分布を $\rho$ としたとき, 静電ポテンシャル $\phi$ は以下の Poisson 方程式を満たす.

$$\Delta\phi = -\frac{\rho}{\epsilon_0} \tag{1}$$

本研究では座標系を 2 次元とし, 電位は $x$,$y$ の関数で $z$ 方向は考慮しないものとする. この場合, 式 (1) は以下のように表すことが可能である.

$$\frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} = -\frac{\rho}{\epsilon_0} \tag{2}$$

### 2.2 計算手順

本シミュレーションのプログラムの流れを図 1 に示す. まず, 電位を格納する配列と電荷を格納する配列を確保し, 領域全体を 0 で初期化する. 次に, 電荷密度の設定を行う. 本プログラムにおいては, 電荷の数を 10, 密度を $1.0 \times 10^{-8}[\mathrm{C/m^2}]$ とする. そして, 領域端を除く範囲で電位の計算を繰り返す. その際, 前回計算した電位との残差を全領域に対して求め, 残差の最大値が収束判定係数以下となるまで処理を繰り返す. 最後に, 電界を算出し電位とともに出力する.

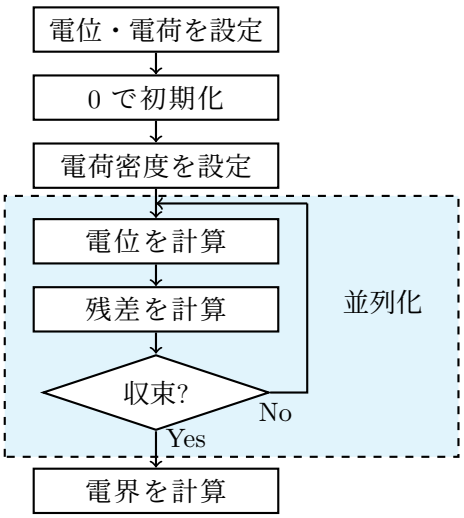


図 1 シミュレーションの流れ

## 3 Ampere アーキテクチャ向けの実装

### 3.1 明示的な warp 同期

Kepler 世代向けのコードを NVIDIA A100 で動作させるにあたって, いくつかの部分を Ampere 世代の仕様に合わせた実装に変更する必要がある. 従来のコードでは warp の組み込み関数の`__shfl_xor()` および`__any()` を使用していたが, Volta 世代以降においてこれらは非推奨となっている. Kepler 世代以前では warp 内の 32 スレッドの演算が同時に実行されることから暗黙の同期が保証されていたが, Volta 世代以降では warp 内の各スレッドが個別にスケジューリングが可能であり, これらが全て同時に実行される保証はない. よって, Volta 世代より追加された warp 同期用の新たな組み込み関数を仕様するか, Cooperative Groups によってスレッドを明示的に同期する必要がある [2][3]. 本研究では,warp 同期用の組み込み関数を使用した場合および Cooperative Groups を使用した場合についての評価を行う.

### 3.2 Async Copy

Ampere アーキテクチャでは, `memcpy_async()` 関数を用いることにより, グローバルメモリからシェアードメモリへレジスタを介さずに直接データを転送することが可能である. `memcpy_async()` 関数の実装には Cooperative Groups によるもの, Barrier API によるもの, Pipeline API によるものがある. 本研究では, Cooperative Groups による実装を用いて性能評価を行う [4].

---

Optimization of Potential and Electric Field Simulations to the NVIDIA Ampere Archtecture

† Shusaku Konishi, Gradudte School of Engineering, Okayama University of Science
‡ Akira Uejima, Faculty of Engineering, Okayama University of Science

## 4 プロファイリングに基づいた最適化

### 4.1 Nsight Compute によるプロファイリング

NVIDIA は，同社の GPU における GPU カーネルのプロファイリングを行うためのツールとして，Nsight Compute を提供している．本研究では，Ampere アーキテクチャにおいて電位・電界シミュレーションの GPU カーネルのプロファイリングを行い，それに基いた最適化を行ったコードの実行時間を評価する．

### 4.2 FMA 化の禁止

CUDA では，コンパイルの際に浮動小数点の乗算・加算命令のペアが存在する場合は FMA(Fused Multiply-Add: 融合積和演算) 命令に置き替えられる．FMA は 1 回の命令で乗算と加算を行えるため，一般的には個別に乗算と加算を行うよりも高速といわれている [5]．一方で，今回扱う Poisson 方程式のカーネルにおいてはコンパイラによる FMA 化によって演算量が増加する問題があるため，本研究では加算部分を`__fadd_rn()` に，乗算部分を`__fmul_rn()` に置き替えることにより FMA 化を禁止したコードを作成して評価を行う．

## 5 評価

### 5.1 評価方法

評価に用いた計算機を表 1 に示す．収束判定周期を 100 とし，それぞれのプログラムで格子サイズを変化させながら実行し，実行時間の比較を行う．Kepler 世代向けのコードをコンパイル・実行した場合 (Kepler モード) と，Volta 世代および Ampere 世代で追加された機能を用いた場合でそれぞれ比較を行う．更に，プロファイリングに基づいて最適化を行った場合との比較も行う．

表 1　計算機環境

| | |
|---|---|
| CPU | AMD EPYC 7413<br>24 cores, 2.65GHz |
| GPU | NVIDIA A100<br>6912 cores<br>1065MHz<br>80GB |
| OS | Linux 5.4 |
| コンパイラ | gcc 9.4.0<br>nvcc 11.6 |

### 5.2 実行時間

#### 5.2.1 Ampere 向け実装の評価

A100 GPU にて，Kepler モードと明示的な同期を用いたコード，Cooperative Groups を用いたコード，および Async Copy の各実装を用いたコードを用いて評価を行う．図 2 に実行時間を示す．従来のコードを Kepler モードで実行した場合と，明示的な warp 同期を用いたコードの間の実行時間の差は 1% 程度にとどまった．一方で，Cooperative Groups を用いた場合 5% 程度実行時間が長くなり，Async Copy 使用時には更に 1% 程度実行時間が長くなった．

#### 5.2.2 プロファイリングに基づいた最適化の評価

明示的な同期を用いたコードを元に，Poisson 方程式の加算部分を`__fadd_rn()` に，乗算部分を`__fmul_rn()` に置き替えたコードをコンパイルし，元のコードとの実行時間の比較を行う．

`__fadd_rn()` および`__fmul_rn()` によって FMA 化を禁止したコードでは，格子サイズが 1408 のときに元のコードの 34% 程度の実行時間に短縮されることが確認できた．

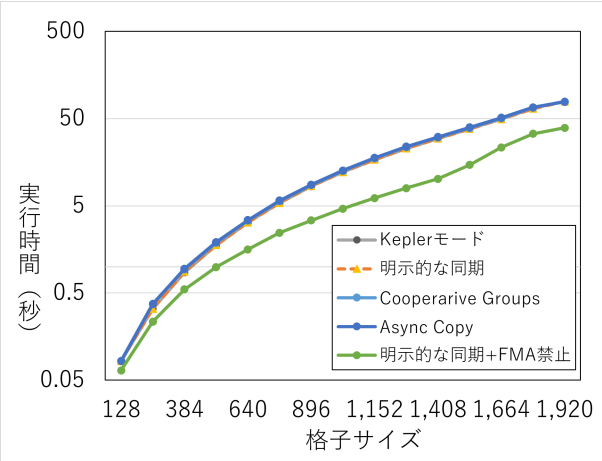


図 2　実行時間

## 6 おわりに

Kepler 世代 GPU 向け電位・電界シミュレーションを NVIDIA A100 GPU を対象に移植し，性能評価を行った。暗黙の同期を明示的な同期に置き換えることや，グローバルメモリからシェアードメモリへの非同期コピーによる性能向上は見られなかった．一方で，Poisson 方程式による電位の計算部分の FMA 化を禁止することにより，最大で実行時間を 1/3 程度にすることが可能であることがわかった．今後の課題として，複数 GPU 環境への最適化などが挙げられる．

## 参考文献

[1] 松原 翼, 長尾 栄作, 上嶋 明, 尾崎 亮, 小畑 正貴 : GPGPU による電位・電界シミュレーションの並列化, 情報処理学会 第 77 回全国大会講演論文集, pp. 1-73–1-74, 2015.

[2] 小西 秀策, 上嶋 明 : NVIDIA A100 GPU における電位・電界シミュレーションの性能評価, 情報処理学会 第 85 回全国大会講演論文集, pp. 1-35–1-36, 2023.

[3] 三木 洋平 : Volta 世代の GPU における重力ツリーコードの性能評価, 情報処理学会研究報告, Vol.2018-HPC-166, No.6, 2018.

[4] 三木 洋平 : NVIDIA A100 における重力ツリーコードの性能評価, 情報処理学会研究報告, Vol. 2021-HPC-180, No. 24, pp. 1-7, 2021.

[5] NVIDIA : Floating Point and IEEE 754 Compliance for NVIDIA GPUs (2023).