

令和7年度 卒業論文概要					
所 属	物理情報工学科・電子物理工学コース				
学生番号	222C3058	学生氏名	杉本 陸	指導教員	小田部 荘司
論文題目	CUDA を用いた超伝導体内の磁束線運動の並列高速シミュレーション				

1. 緒言

超伝導体における渦糸の生成・消滅や相転移といった動的現象の解明は、基礎物理学のみならず、次世代超伝導デバイスの設計においても極めて重要である。これらの非平衡・非線形現象を記述する有効な理論として時間依存ギンツブルグ・ランダウ（Time Depended Ginzburg Landu: TDGL）方程式が知られているが、その数値シミュレーションは膨大な計算コストを伴う。特に大規模格子系を扱う場合、従来の CPU による逐次計算ではシミュレーションの時間的コストが高いことが問題として知られている。

本研究の目的は、この計算ボトルネックを抜本的に解消することにある。本稿では、GPU が持つ大規模な並列計算能力に着目し、CUDA (Compute Unified Device Architecture) を用いた TDGL 方程式の高速並列シミュレーションプログラムを、2次元と3次元それぞれについて開発した手法と結果について報告する。

2. GPU による並列実装と高速化

この節では 2 次元の超伝導シミュレーションの高速化について述べる。まず高速化の手法について述べ、その結果について考察する。

2.1. GPU による並列実装と高速化

本シミュレーションの高速化は、計算集約的な処理の GPU オフロードと、メモリ及びネットワークにおけるデータ転送を最小化する最適化を行うことによって実現した。開発には Java と CUDA C++を JCuda で連携させ、CPU には全体の制御を、GPU には TDGL 方程式の時間発展計算を、Processing には結果の描画を行うよう役割を分離した。これにより並列性の高い処理は GPU 上で並列実行され、画像描画は従来通り Processing で行うことにより、高速化を実現しつつ CPU の実装と同じ画像の生成を行うことが可能である。

CPU-GPU 間のデータ転送負荷を削減するため、大規模な配列データはシミュレーション開始時に一度だけ GPU へ転送し、以降の計算は GPU 内部で完結させる。結果の描画等、必要な場合にのみデータを CPU へコピーバックする設計である。なお、この GPU 化の過程では、開発効率向上の

ため生成 AI を補助的に活用した。コードの雛形作成に利用したが、出力されたコードの構文エラー等は手動で修正し、実装を完成させた。

2.2. シミュレーション結果と考察

計算速度の向上を評価するため、CPU による逐次計算実装と、GPU による並列計算実装の性能を、同一条件下で比較した。なお、GPU を用いたシミュレーション性能は、GPU デバイス性能によって変動する可能性がある。計算速度は 1 秒あたりの平均ステップ演算数で測定した。その結果と比較を図 1 に示す。また、シミュレーションによって得られた画像の例を図 2 に示す。

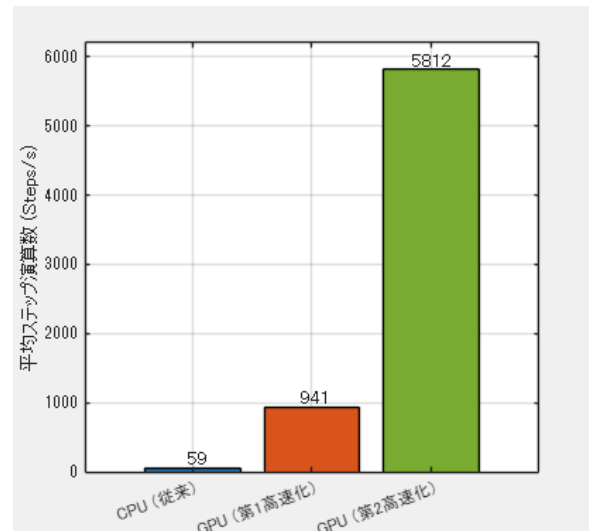


図 1 演算ステップ数毎秒の推移。段階的な高速化により大幅にシミュレーション速度が向上した

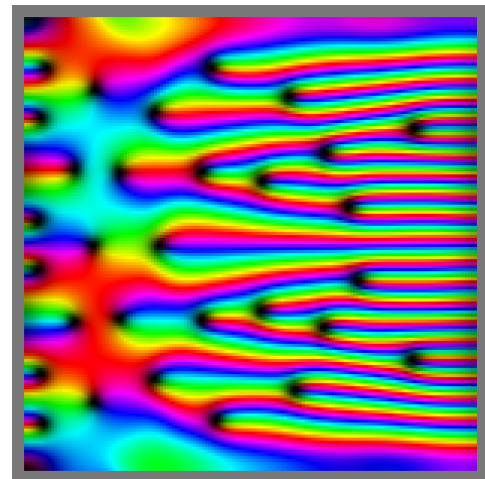


図 2 シミュレーションの様子(2次元)。濃淡が秩序パラメータの大きさ、色は位相情報、黒点は渦糸

シミュレーション結果から、従来の CPU での実装から GPU に移植することで、最終的に約 99 倍の高速化が達成されていることが分かる。第 1 高速化から第 2 高速化においては CPU-GPU 間及びネットワークのデータ転送負荷の軽減を行った。この改善においてもシミュレーションが大幅に高速化されていることから、CPU-GPU 間のデータ転送の負荷がシミュレーションの進行に大きく影響を及ぼしていることが読み取れる。

3. 3次元への拡張

この節では第 2 節を踏まえ、3 次元の超伝導シミュレーションの高速化について述べる。まず高速化の手法について述べ、その結果について考察する。

3.1. 3次元シミュレーションの手法

高速化の手法は 2 次元と同様であり、計算集約部分の GPU オフロードを行う。しかし、3 次元系では計算格子数が劇的に増加するため、単なる演算時間の増加に加え、CPU-GPU 間のデータ転送及び Processing への TCP 通信帯域がボトルネックとして顕在化することが予想される。本研究では、計算の進行を阻害しない範囲で、可視化および解析に必要なデータのみを転送するよう、適切なステップ幅を設定し、実用的な演算速度の維持を図った。

3.2. 3次元シミュレーションの結果と考察

3 次元シミュレーションにおいても、2 次元と同様に GPU 実装による劇的な高速化が確認された。具体的な計算速度を 1 秒あたりの平均ステップ演算数で比較すると、CPU 実装が約 0.469 にとどまったのに対し、GPU 実装では約 89.6 を記録した。これは CPU と比較して約 191 倍の高速化に相当する。シミュレーションの様子を図 3 に示す。

2 次元の場合は約 99 倍の高速化であったのに対し、3 次元においては 191 倍という飛躍的な向上が見られた。この要因として、計算規模の増大により GPU の並列演算コアがより効率的に稼働し、計算密度が向上したことで、データ転送等の遅延に繋がる影響が相対的に低下したことが考えられる。

4. 高速化によるメリット

シミュレーションの性能向上により、従来は困難であった長時間のシミュレーションや、多数の物理パラメータの組み合わせを試行する大規模なパラメータ探索が、現実的な時間で実行可能となった。加えて、パラメータを変更しながら、その影響が系のダイナミクスに与える効果を効率

的に確認できる。この対話性の獲得は、効率化だけでなく直感的理解を促す教育的観点からも有益である。

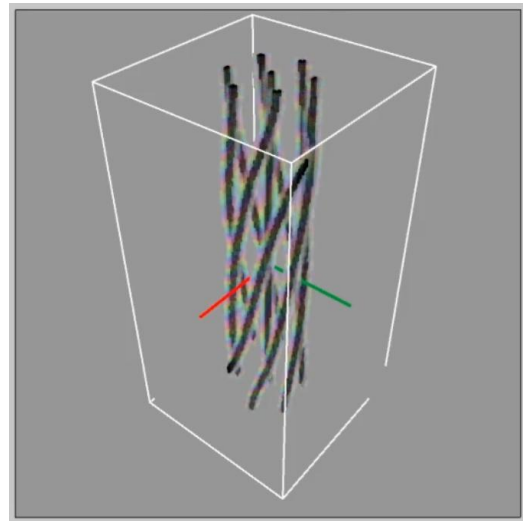


図 3 シミュレーションの様子(3次元). 超伝導体内に量子渦が生成され、ねじれながら動いている

5. 結言

本研究では、計算コストが課題となっていた TDGL 方程式の数値シミュレーションに対し、CUDA を用いた GPU 並列化による高速化手法を提案、実装した。Java と JCuda を用いて計算集約的な処理を GPU へオフロードし、メモリ転送を最適化することで、CPU による従来手法からの大幅な性能向上を図った。

性能評価の結果、2 次元シミュレーションにおいては約 99 倍、より計算負荷の高い 3 次元シミュレーションにおいては約 191 倍の高速化を達成した。特に 3 次元系においてより高い加速率が得られたことは、大規模数値計算における GPU 並列処理の優位性を強く裏付けるものである。本実装により実現したリアルタイムかつ対話的なシミュレーション環境は、複雑な超伝導ダイナミクスの解明に向けたパラメータ探索を効率化するだけでなく、教育用ツールとしても極めて高い有用性を持つといえる。

参考文献

- [1] K. Arita *et al.*, *Physica C* **662** (2024) 1354522.
- [2] T. Matsuno, E.S. Otabe, Y. Mawatari, *Journal of the Physical Society of Japan* **89** (2020) 054006.

研究実績

杉本陸 他、「CUDA を用いた超伝導体動力学の並列高速シミュレーション」、2025 年度電子情報通信学会九州支部学生会講演会、福岡工業大学、令和 7 年 9 月 17 日