

## 格子 QCD 計算のための Julia コードの GPU による高速化

### GPU acceleration of a Julia code for lattice QCD

大野浩史

筑波大学計算科学研究センター

#### 1. 研究目的

本研究では、Julia 言語による格子 QCD 計算パッケージ JuliaQCD の GPU 高速化を行い、開発効率と高性能計算を両立した格子 QCD 計算環境の構築を目指した。格子 QCD 計算では、大規模なモンテカルロシミュレーションや大規模疎行列の反復解法が必要となるため、GPU を用いた高速化が不可欠である。一方で、既存の高性能コードは Fortran や C/C++ を用いて高度に最適化されており、初学者にとって参入障壁が高いという課題がある。

Julia は高い記述性と高性能を両立した言語であり、JuliaQCD は既に MPI 並列化を含む基本的な格子 QCD 計算機能を備えている。本研究では、このパッケージを GPU に対応させることで、初学者でも扱いやすく、かつ最先端研究にも利用可能なポータブルな格子 QCD 計算環境を整備することを目的とした。

#### 2. 研究成果の内容

本年度は、JuliaQCD の GPU 化を進めた。GPU 実装には Julia の GPU プログラミング基盤である JACC.jl を用い、NVIDIA GPU および AMD GPU の双方に対応した実装を行った。これにより、特定ベンダーに依存しない性能ポータブルな格子 QCD コードの基盤を整備した。GPU 化に伴い、格子 QCD 計算において計算時間の大部分を占める Dirac 演算子やゲージ場演算等が GPU 上で計算可能となった。

コードの性能を調べるため、GPU を用いない単純な MPI 並列計算と単一 GPU 計算に対し、Wilson Dirac 演算子を用いたベンチマークテストを実施した。その結果、MPI 並列計算は strong scaling、weak scaling の双方において良好な並列性能を示した。さらに、 $32^3 \times 64$  格子上での計算において、単一 GPU 計算は、Pegasus、Miyabi-G、および Sirius 上でそれぞれ、単一プロセス計算と比較して約 180 倍、約 95 倍、約 260 倍の高速化を実現した (図 1)。

これらの結果は、JuliaQCD が GPU 環境においても高い性能を発揮できることを示している。また、Julia 言語による高い開発効率を維持しながら、大規模 GPU 計算に適用可能な実装を実現できた点は重要な成果である。

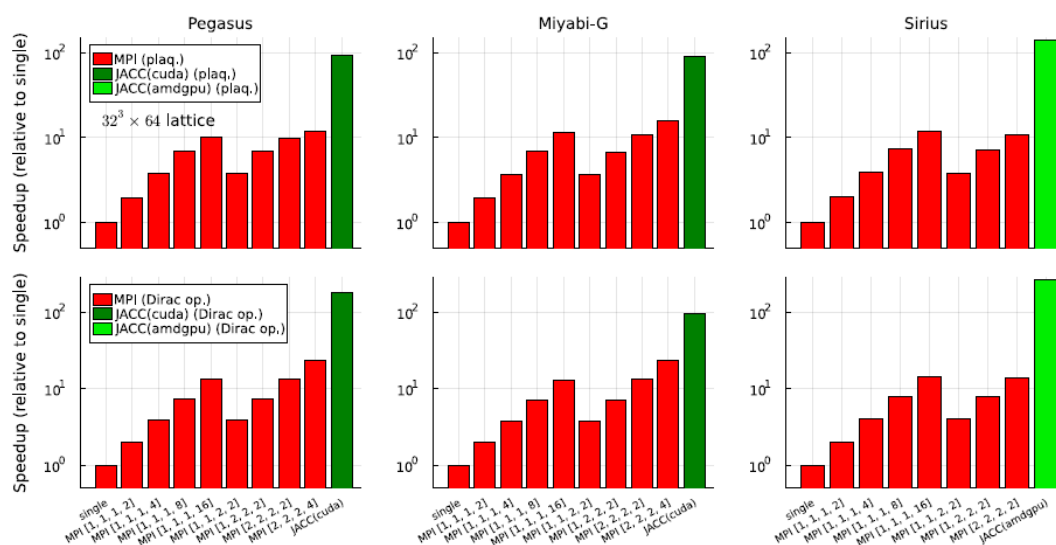


図 1: MPI 並列計算の strong scaling 性能および単一 GPU 性能 [2]

### 3. 学際共同利用プログラムが果たした役割と意義

本研究では、学際共同利用プログラムの計算資源を活用することで、JuliaQCD の GPU 化と大規模性能評価を効率的に進めることができた。GPU 化によって、JuliaQCD の性能は飛躍的に向上し、従来よりも大規模な格子サイズに対する計算が現実的な時間で実行可能となった。これにより、Julia ベースの格子 QCD 計算コードを実際の研究用途に適用できる可能性が大きく広がった。

さらに、本研究で整備した GPU 環境は、機械学習を用いた格子 QCD 研究にも大きく貢献した。特に、機械学習によるゲージ固定の高速化の研究では、大規模格子サイズに対するニューラルネットワークの学習および推論を GPU 上で高速に実行できるようになった。その結果、小さい格子サイズで学習したモデルが大きな格子サイズにも有効であることを確認し、従来手法よりも良好な収束性を示す結果を得た。これは、格子 QCD における AI 技術活用の可能性を示す成果である。この研究成果は Lattice 2025 において報告した[1]。

以上のように、学際共同利用プログラムは、高性能計算技術と格子 QCD、さらに AI 技術を融合した研究を推進する上で重要な役割を果たした。

### 4. 今後の展望

今後は、現在実装済みの単一 GPU 実装をさらに発展させ、MPI と GPU を組み合わせたマルチ GPU 対応を進める予定である。特に、大規模格子 QCD シミュレーションでは複数 GPU を用いた計算が不可欠であり、通信と計算のオーバーラップや GPU-aware MPI の活用などを含めた最適化を行う予定である。

5. 成果発表

(1) 学術論文

- [1] H. Hsiao, B. J. Choi, H. Ohno and A. Tomiya, “A Machine Learning Approach for Lattice Gauge Fixing,” PoS **LATTICE 2025**, 041 (2026).
- [2] A. Tomiya, Y. Nagai and H. Ohno, “JuliaQCD: Performance-portable lattice QCD kernels in Julia,” The Proceedings of the JuliaCon Conferences, Submitted.

(2) 学会発表

- [3] A. Tomiya, Y. Nagai and H. Ohno, “JuliaQCD: Portable lattice QCD package in Julia language,” Conference on the Julia programming language (JuliaCon) 2025, David Lawrence Hall at the University of Pittsburgh, Pennsylvania, USA, July 22 – 26, 2025
- [4] 大野浩史、富谷昭夫、永井佑紀、「JuliaQCD における GPU プログラミングとその性能評価」、「富岳」成果創出加速プログラム 超大規模格子 QCD による新物理探索と次世代計算に向けた AI 技術開発 総括研究会、ビジョンセンター田町、東京、2025 年 9 月 9 日 – 11 日
- [5] H. Hsiao, B. J. Choi, H. Ohno and A. Tomiya、「Machine Learning Approaches for Lattice Gauge Fixing」、「富岳」成果創出加速プログラム 超大規模格子 QCD による新物理探索と次世代計算に向けた AI 技術開発 総括研究会、ビジョンセンター田町、東京、2025 年 9 月 9 日 – 11 日
- [6] H. Hsiao, B. J. Choi, H. Ohno and A. Tomiya、「Lattice Gauge Fixing through a Machine Learning Approach」、日本物理学会第 80 回年次大会、広島大学東広島キャンパス、広島、2025 年 9 月 16 日 – 19 日
- [7] H. Hsiao, B. J. Choi, H. Ohno and A. Tomiya、「Lattice Gauge Fixing through a Machine Learning Approach」、R7 年度学術変革 A 「学習物理学」領域会議、東京大学本郷キャンパス山上会館、東京、2025 年 9 月 24 日 – 25 日 (ポスター)
- [8] H. Ohno, “Machine Learning Applications to Lattice QCD,” 10th International Symposium of Quantum Beam Science at Ibaraki University, Mito Campus, Ibaraki University, Mito, Ibaraki, Japan, October 9 – 10, 2025 (invited)
- [9] H. Hsiao, B. J. Choi, H. Ohno and A. Tomiya, “A Machine Learning Approach for Lattice Gauge Fixing,” The 42nd International Symposium on Lattice Field Theory (Lattice 2025), Tata Institute of Fundamental Research, Mumbai, India, November 2 – 8, 2025
- [10] H. Hsiao, B. J. Choi, H. Ohno and A. Tomiya、「A Machine Learning

Approach for Lattice Gauge Fixing」、「富岳成果創出加速プログラム」基礎科学合同シンポジウム 2025、御茶ノ水ソラシティカンファレンスセンター、東京、2026年1月13日 - 15日

- (3) その他  
特になし。

使用計算機	使用計算機に○	配分リソース※		
		当初配分	移行*	一般利用による追加
Pegasus	○	4,000		
Miyabi-G	○	13,500		
Miyabi-C				
※配分リソースについてはノード時間積をご記入ください。 *バジェット移行を行った場合、「+2000」「-1000」のように記入				