

GPU 対応・宇宙流体シミュレーションコードの開発

課題責任者

銭谷 誠司 神戸大学 都市安全研究センター

著者

銭谷 誠司*¹*¹ 神戸大学 都市安全研究センター

太陽・宇宙空間プラズマ研究のための公開・磁気流体シミュレーションコード「OpenMHD」の GPU 対応を図る。ES4-GPU の実機上で OpenMHD-GPU の改良と安定化を図り、その成果を公開することによって、当該分野のシミュレーション研究の基盤となることを目指す。

キーワード：磁気流体シミュレーション，並列計算，GPGPU

1. 概要

太陽コロナや宇宙空間の複雑なプラズマ現象を理解するために、磁気流体 (MHD) シミュレーションは重要な役割を果たす。しかし、最新の数値解法を取り入れながら、現代の複雑な計算機環境で動作するコードを研究者個人が開発するのは困難になってきている。こうした中、研究グループ内で用いる共有コード、あるいはコミュニティに公開された公開コードは重要な意味を持つ。

本課題メンバーは、自らのシミュレーション研究のために開発した並列 MHD コードを「OpenMHD」と命名して公開し、継続的にコードの改良を続けている [1, 2]。OpenMHD は modern fortran で書かれた時間・空間 2 次精度の有限体積コードで、HLLD タイプの数値流束解法を採用しており、MPI および OpenMP を用いた超並列計算に対応している。OpenMHD はこれまで、累計 13 本の研究論文に利用されてきた。

最近、OpenMHD は主要コードを CUDA Fortran で再実装することで NVIDIA 社の GPU (Graphic Processing Unit) でも動作するようになった。今後は並列 GPU 環境でより大規模の計算にチャレンジできるよう、コードのさらなる改良を進める予定である。本課題では、ES4-GPU の実機上で OpenMHD-GPU の改良と安定化を図り、その成果を公開することによって、国内外の当該分野のシミュレーション研究に寄与することを目指す。

2. 2022 年度の成果

ES4 の GPU ノードおよび複数の HPC センターの GPU 計算機を利用して、OpenMHD-GPU のシングルノードおよび MPI 複数ノード環境でのテストを行い、ノウハウを蓄積した。

シングルノードでは、OpenMHD-GPU は十分に安定動作することがわかった。図 1 は、OpenMHD 開発版の性能をさまざまな実行環境でベンチマークした結果である。AMD 社のマルチコア CPU (EPYC プロセッサ 7282 : 16 コアおよび 7452 : 32 コア) および NVIDIA 社の GPU (K80, P100, … A100) でテスト問題の実行時間を比較した。ノードあたり (≒電力あたり) で考えると、GPU は EPYC プロセッサより有意に高速であることがわかる。特に ES4 の A100 は圧倒的な速度を誇っており、中規模な計算であれば短時間

で大量の計算をこなすことができる。また、NVIDIA 社のローエンド GPU の Tesla K80 や P100 は、Google のクラウドサービス Google Colaboratory (以降、Colab) で提供されているものである。我々は 2022 年 9 月にオンライン開催された第 14 回国際宇宙シミュレーションスクール (ISSS-14) で、OpenMHD-GPU を Colab で動作させる実習コースを提供した。

一方、複数ノード環境では、国立天文台および核融合科学研究所の GPU 環境では OpenMHD-GPU を MPI 実行できるものの、ES4 の MPI-GPU 環境では起動時にプログラムが異常終了する。この問題の究明にしばらく取り組んだが、年度内には解決できなかった。これは将来の課題である。

最後に、OpenMHD の非 GPU 部の改良にも取り組んだ。MPI 通信部のバッファ領域で MPI データ型を使ってみたが、有意な性能向上は得られなかった。さらに OpenMP の記述を工夫してスレッド数が多い場合の安定性を向上させた。海外の利用者からの問い合わせに応じて、コードの英語ドキュメントを大幅に増やしたところである。2023 年度以降も引き続き、OpenMHD の改良とドキュメントの拡充を進めていく。

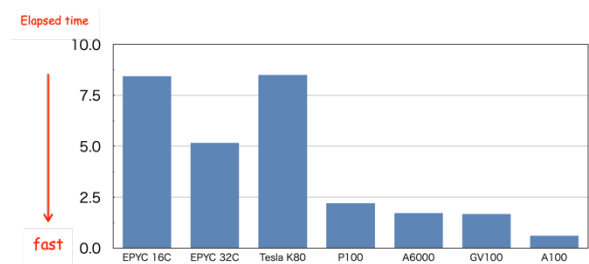


図 1：磁気リコネクションテスト問題のベンチマークテスト。各実行環境 (横軸) での実行時間 (分; 縦軸) を示す。

参考文献

- [1] Seiji Zenitani, “Magnetohydrodynamic structure of a plasmoid in fast reconnection in low-beta plasmas: Shock-shock interactions”, Physics of Plasmas 22, 032114 (2015)

[2] <https://github.com/zenitani/OpenMHD>

Development of GPU-ready Plasma Simulation Code

Project Representative

Seiji Zenitani Research Center for Urban Safety and Security, Kobe University

Authors

Seiji Zenitani *1

*1Research Center for Urban Safety and Security, Kobe University

We develop an open-source magnetohydrodynamic (MHD) code “OpenMHD” for solar and space plasma simulations. The code has been ported to NVIDIA’s GPU architecture recently. We are testing and improving OpenMHD-GPU on ES4-GPU, and then we will make the code publicly available in future.

Keywords : MHD simulation, Parallel computing, GPGPU, HLLD, LHLLD

1. Introduction

Magnetohydrodynamic (MHD) simulation plays a key role in predicting complex phenomena in solar and space plasma environments. Meanwhile, it has become difficult for researchers to maintain a simulation code, which employs up-to-date numerical schemes and which runs on the latest computing platforms. Owing to this, there are growing demand for a shared simulation code in a research group or a publicly-available code for a community.

We have been developing a parallel MHD code “OpenMHD” over years [1,2]. OpenMHD is a second-order finite-volume code with HLLD-type MHD flux solver. It is written in modern Fortran and is parallelized with MPI and OpenMP. As of 2022, the code was used in 13 research papers.

We have recently ported OpenMHD to NVIDIA’s GPU (Graphics Processing Unit) architecture using CUDA Fortran language. We further improve our code, so that we can explore larger-scale problems on multiple GPU nodes. In this project, we will extensively test and improve the parallel communication module in OpenMHD-GPU on ES4-GPU. Then we will make our code publicly available on the Internet.

2. Progress in 2022

We have used the GPU nodes at JAMSTEC (ES4) and two other HPC centers, in order to test OpenMHD-GPU in single- and multiple GPU environments.

OpenMHD-GPU is stable enough on a single GPU. Figure 1 compares the elapsed times of a test problem in various environments: on AMD EPYC processors (EPYC 7282: 16 cores and 7452: 32 cores) and NVIDIA GPUs (Tesla K80, P100, ... A100). Here, performance per node is similar to performance per power. We see that the code runs significantly faster on GPUs than on EPYC processors. In particular, ES4’s A100 looks very promising. Tesla K80 and P100 are provided by Google’s free cloud service, Google Colaboratory (Colab in short). We have managed to use OpenMHD-GPU on Colab. Using OpenMHD-GPU on Colab, we offered a hands-on course at the 14th International School for Space Simulation (ISSS-14). On the

other hand, even though OpenMHD-GPU runs on multiple GPUs at National Astronomical Observatory of Japan and National Institute for Fusion Science, it abnormally stops at startup on the multiple GPU nodes at ES4. As of April 2023, we have not yet solved this problem; this is left for our future work.

We have also worked on the non-GPU part of OpenMHD. We tested a new MPI datatype in the communication part, but this does not significantly improve the performance. Furthermore, we have improved the stability when we use many OpenMP threads. Lastly, we have provided key documents in English. In FY 2023, we continue to improve the code and expand documentations.

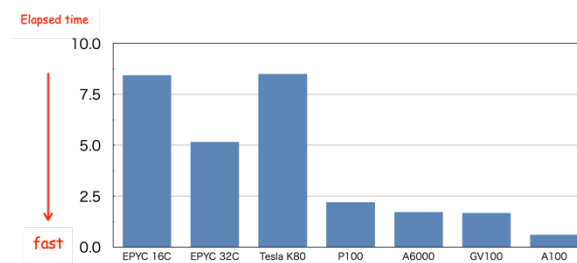


Figure 1: A benchmark test of a magnetic reconnection problem in OpenMHD. Elapsed times (vertical axis) are compared in various CPU/GPU systems (in minutes; horizontal axis).

References

- [1] Seiji Zenitani, “Magnetohydrodynamic structure of a plasmoid in fast reconnection in low-beta plasmas: Shock-shock interactions”, *Physics of Plasmas* 22, 032114 (2015)
- [2] <https://github.com/zenitani/OpenMHD>