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
簑島　敬　　海洋研究開発機構 付加価値情報創生部門 数理科学・先端技術研究開発センター

著者
銭谷　誠司*[1]
*[1] 神戸大学 都市安全研究センター 特命准教授

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 の並列計算部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，HLLD 法，LHLLD 法

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。

本課題メンバーの一人は、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 [3] 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 12 本の研究論文に利用されてきた。

最近、OpenMHD はコア部分を CUDA Fortran で再実装することで NVIDIA 社の GPU 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 の並列計算部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. 2021 年度の成果

2021 年度の実質利用時間は年度末の 1 ヶ月であった。そこで時間の制約上、ES4-GPU 上で GPU 並列コードを動作させるためのいくつかの準備を行なった。例えば、コードを整理する過程で GPU 版の使用メモリを軽減させることに成功した。

さらに、最近提案された数値解法である LHLLD 法 [4] を部分実装し、OpenMHD および OpenMHD-GPU で試験的に利用できるようにした。図 1 に、その検証結果を示す。これはケルビン・ヘルムホルツ不安定の渦の線形成長問題を、旧来の HLLD 法（実線）および新しい LHLLD 法（記号）で、解像度を変えて計算したものである。このうち、いちばん上の赤の実線が基準結果である。旧手法では数値粘性が強く働くため、低解像度の計算（緑および青の実線）で渦の成長が抑えられている。新手法では、数値粘性が軽減されてより渦が成長しやすくなることが期待されているが、今のところ期待した改善が見えず、新旧手法でほぼ同じ結果を得ている（図 1）。原因は不明であるが、OpenMHD 側の問題のために LHLLD 法本来の精度が出ていない可能性があるため、引き続きテストを進める予定である。

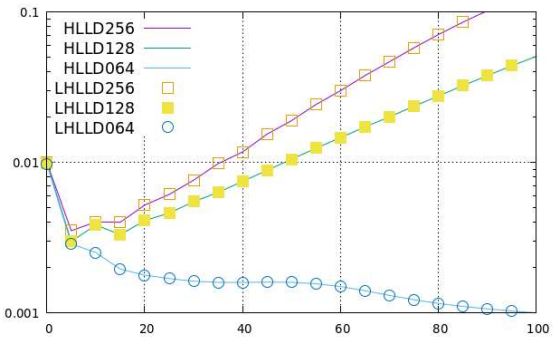

図 1 ：横軸は時間、縦軸は max(|v_y|)で、KH 不安定の線形モードの成長を簡易的に示す。

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
[3] Takahiro Miyoshi, Kanya Kusano, “A multi-state HLL approximate Riemann solver for ideal magnetohydrodynamics”, Journal of Computational Physics 218, 315 (2005)
[4] Takashi Minoshima, Takahiro Miyoshi, “A low-dissipation HLLD approximate Riemann solver for a very wide range of Mach numbers,” Journal of Computational Physics 446, 110639 (2021)

Development of GPU-ready Plasma Simulation Code

Project Representative
Takashi Minoshima Center for Mathematical Science and Advanced Technology, Research Institute for Value-Added-Information Generation, Japan Agency for Marine-Earth Science and Technology

Authors
Seiji Zenitani *[1]
*[1]Research Center for Urban Safety and Security, Kobe University

We develop an open-source magnetohydrodynamic (MHD) simulation code “OpenMHD” for solar and space plasma researches. The code has been ported to the NVIDIA GPU architecture recently. We are preparing to improve the parallel communication module of 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.

One of the project members (SZ) has 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 [3]. It is written in modern Fortran and is parallelized with MPI and OpenMP. As of 2021, the code was used in 12 research papers.

We have recently ported OpenMHD to the NVIDIA GPU architecture using the 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 2021

Unfortunately, since we applied for ES4 very late, we only had one month in FY 2021. Since our time was limited, we did several preparations to run multi-node GPU code on ES4-GPU. For example, we managed to reduce the memory footprint of OpenMHD-GPU.

In addition, we partially implemented a latest numerical solver, an LHLLD solver [4]. One can test the new solver in OpenMHD and OpenMHD-GPU by setting a variable. Figure 1 presents our test results. We have calculated the linear evolution of the Kelvin-Helmholtz instability with the HLLD (solid lines) and LHLLD (symbols) solvers. The top red line is a reference result. The HLLD solver fails to reproduce the linear evolution of the KH vortex due to numerical dissipation for the low-resolution runs (the green and blue lines). The LHLLD solver is designed to reduce the numerical dissipation --- thus, we expect rapid growth of the KH vortex even in the low-resolution cases. However, as shown in Figure 1, we see that the LHLLD results and the HLLD results are very similar. At this point, we fail to see improvements by the LHLLD solver. This is quite probably due to our implementation of the LHLLD solver in the OpenMHD code. We are investigating this problem now.

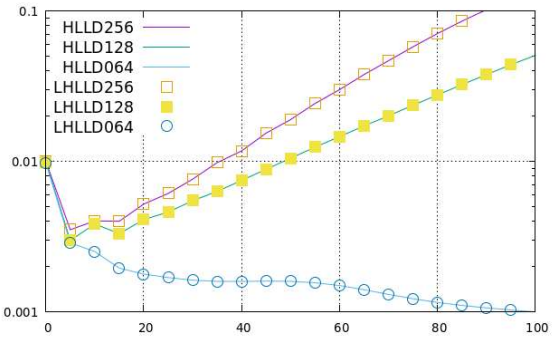

Fig. 1: Time development of $\max(|v_y|)$, which is an ad-hoc proxy of the evolution of the fastest-growing mode of the KH instability.

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
[3] Takahiro Miyoshi, Kanya Kusano, “A multi-state HLL approximate Riemann solver for ideal magnetohydrodynamics”, Journal of Computational Physics 218, 315 (2005)
[4] Takashi Minoshima, Takahiro Miyoshi, “A low-dissipation HLLD approximate Riemann solver for a

very wide range of Mach numbers," Journal of Computational Physics 446, 110639 (2021)