Joint Usage/Research Center for Interdisciplinary Large-scale Information Infrastructures
Final Report for JHPCN Joint Research of FY 2023

jh230033

Developing AI-Assisted high performance fluid simulation codes

Shinya Maeyama（NIFS）

We aim at establishing AI-assisted HPC fluid simulations. As a proof of principle, we will demonstrate simple model cases in a scalable manner to production applications. For AI-assisted fluid simulations, we focus on two important targets: modeling of subgrid-scale (SGS) dynamics and data-assimilation (DA) procedures. We demonstrated a deep learning-based SGS model which allows the large eddy simulation with 1/10 of grid points compared to direct simulations. A diffusion model-based DA model exhibits better accuracy than a conventional method when the model is biased. For HPC, we have established a performance portable implementation of the Local Ensemble Transform Kalman Filter (LETKF) using senders/receivers while exploiting the performance gain by overlapping file I/O, communications and computations.

1. Basic Information

(1) Collaborating JHPCN Centers

Tokyo

Tokyo-Tech

(2) Theme Area

Large scale computational science area

(3) Research Areas

Very-large-scale numerical computation

Very-large-scale data processing

(4) Project Members and Their Roles

Project representative Shinya Maeyama performs the local plasma turbulence simulations. Yuuichi Asahi works to develop a machine learning / deep learning (ML/DL) model to extract features from fluid simulation data. Julien Bigot works on the in-situ data analysis of GYSELA. Xavier Garbet works for theoretical analysis of non-local transport processes. Virginie Grandgirard gives the advice for the large scale plasma turbulence simulation. Kevin Obrejan gives the advice for the optimization of mini-apps. Thomas Padioleau gives the advice for state-of-the-art GPU implementations of mini-apps. Takashi Shimokawabe gives advice for large scale simulation and deep learning models.
Hayato Shiba gives the advice for deep learning models. Keisuke Fujii contributes on data-driven analysis. Naoyuki Onodera gives the advice for the large scale LBM simulation. Yuta Hasegawa gives the optimization on GPUs. Prof. Watanabe comments on characteristics of local transport processes. Yasuhiro Idomura gives advice for the large scale simulations. Prof. Aoki gives advices about the usage of TSUBAME3.0.

2. Purpose and Significance of the Research

In this project, we aim at establishing AI-assisted HPC fluid simulations. As a proof of principle, we will demonstrate simple model cases in a scalable manner to production applications. We will focus on two important targets: modeling of sub-grid-scale dynamics and data-assimilation procedures. We will establish the strategy to loosely couple HPC fluid simulation codes and AI models in a flexible manner. We are planning to publish our codes on GitHub which serve as working examples.

Deep learning models to enhance simulations

In the material science, researchers explore new horizons with AI-assisted MD simulations which cannot be reached otherwise. There is also a room to enhance fluid simulations with deep learning models, for example by modelling subgrid-scale (SGS) dynamics and data-assimilation (DA) procedures. We aim at demonstrating the potential of deep learning models to enhance fluid simulations using simple chaotic mini-applications.

Flexible workflow for HPC fluid codes

We will establish a baseline implementation for asynchronous execution of fluid simulation codes while keeping the performance portability. Exploring the strategy to minimize the communication and I/O overheads by asynchronous execution will contribute to the computational dynamics (CFD) community.

3. Significance as JHPCN Joint Research Project

To establish AI-assisted HPC fluid simulations, three key technologies are needed: high performance implementation of fluid codes, dedicated deep learning models and coupling technologies between the two. For high performance implementation, we have established performance portable approach with directives, libraries and language based approaches under French-Japanese collaboration. In addition, we have developed surrogate models of fluid simulations involving image and sequential data. The French group has already succeeded to manage the on-memory simulation data from Dask scripts with PDI library (developed by Dr. Bigot). By coupling these experiences, we can achieve AI-assisted high performance fluid simulations.

The multi platforms offered by JHPCN framework are essential to establish the performance portable implementations over different devices including CPUs and NVIDIA/AMD GPUs. In addition, the large storage offered by JHPCN is necessary for deep and/or machine learning studies.

4. Outline of Research Achievements up to FY2022 (Only for continuous projects)

This is a new project.

5. Details of FY2023 Research Achievements

Deep learning models to enhance simulations

We have developed two simulation codes enhanced with AI: sub-grid-scale (SGS) model and data assimilation (DA) model. A deep learning based sub-grid-scale (SGS) model has been developed which extracts the effects of small-scale fluctuations from sequential turbulence simulation data. Based on the projection operator method, we have constructed the linear (so-called Mori-Zwanzig (MZ) projection operator) and nonlinear (using neural networks) projection operators from the data.

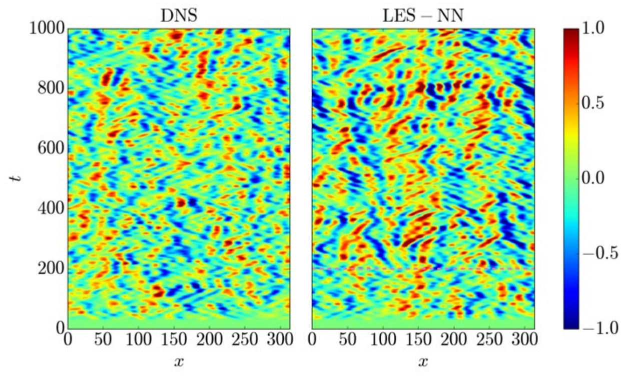


Fig. 1 Spatio-temporal evolution of fluctuations with DNS and LES. We apply the low-path filter to the DNS data for comparison. In LES-NN model, we first run DNS up to t=200 and LES-NN model is enabled.

Figure 1 shows the spatio-temporal evolution of 1D Kuramoto-Sivashinsky, with direct numerical simulations (DNS) and LES-NN models. LES-NN model employs the SGS operator trained on a large amount of DNS data. We employed the Transformer architecture to model the SGS operator from DNS sequential data. In the LES model, we only use 1/10 of grid points in DNS wherein the scale of instability source and dissipation scale are not even resolved. We also confirmed good agreements of time average of energy spectral with DNS, LES (MZ) and LES (NN) models.

Secondly, we have developed an ensemble data assimilation (DA) method using pseudo ensembles generated by denoisng diffusion probabilistic model (DDPM). We update the simulation state in the framework of the Local Ensemble Transform Kalman Filter (LETKF) using these ensembles. As a proof-of-concept, we perform an observing system simulation experiment (OSSE) using the Lorenz96 model as a minimal chaotic system. We train an observation-guided diffusion model to generate simulation results close to incomplete observation data. Thanks to the variance in generated ensembles, our proposed method displays better performance than the well-established ensemble data assimilation method LETKF when the simulation model is biased (the force term F in Lorenz96 is different).

Figure 2 shows the Hovmoller diagrams of “Nature” run with F=5 and DA simulations with F=8. When the model is biased, DA with LETKF fails due to the so-called filter divergence issue (when the variances of ensembles is too small, the filter becomes overconfident around an incorrect state and thus the subsequent observations are ignored). Our method exhibits better accuracy, since the generated ensembles by DDPM always track the observations and the variance in ensembles is kept due to the properties of DDPM. This work has been presented in the international workshop [1].

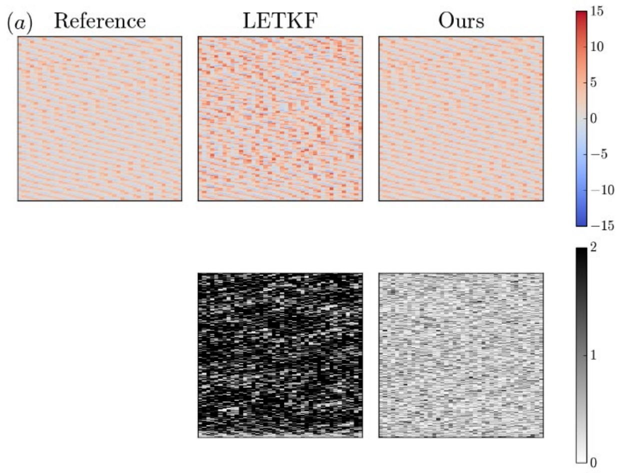


Fig. 2 Hovmoller diagrams and errors with F=5 case. The upper row includes the Hovmoller diagrams of reference, LETKF, and our method. The bottom row includes the absolute errors between reference and DA.

Flexible workflow for HPC fluid codes

We have investigated the use case of C++ senders/receivers to achieve asynchronous performance portable applications. Many scientific simulations consist of different kind of tasks including computations, communications, and file I/O. For better scalability, it is quite important to execute these tasks concurrently to minimize these costs. As a case study, we implement a 2D turbulence simulation code with the LETKF using C++ senders/receivers. In OSSE, we confirm that LETKF is appropriately implemented (See Fig. 3).

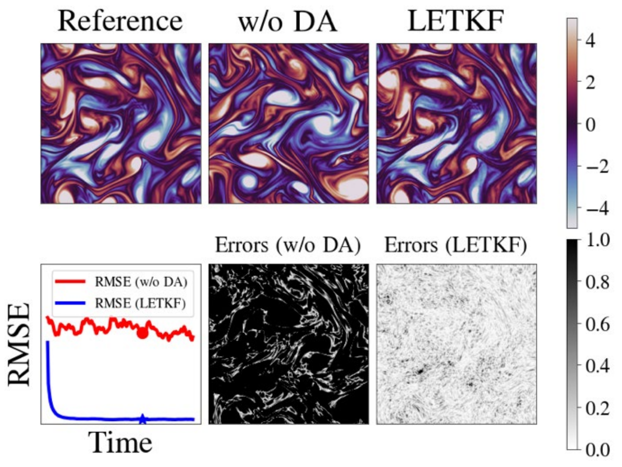
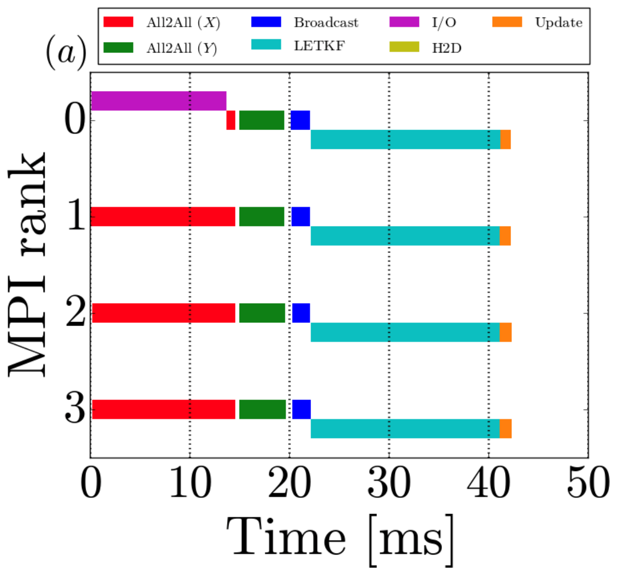


Fig. 3 Impact of DA on 2D isotropic turbulence. The top and bottom rows contain vorticities and errors from reference. As seen from left bottom panel, the error from reference immediately drops with LETKF.

files followed by MPI communications and dense matrix operations on CPUs. We demonstrate the 30% of performance gain on A100 with the introduced concurrency. From the Gantt chart in Fig. 4, we can confirm that file I/O and MPI communications are overlapped with computations in LETKF. This work has been presented in the international workshop [2].



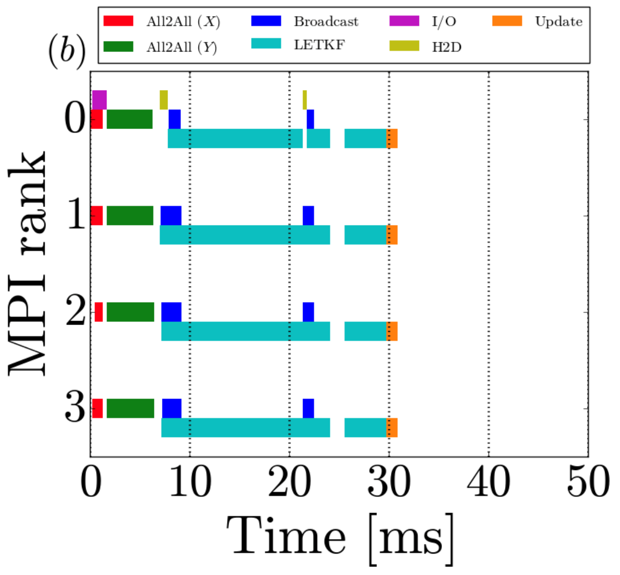


Fig. 4 Gantt diagrams for the LETKF solver (8 ensembles, shown up to rank3) on NVIDIA A100 GPUs with (a) Synchronous and (b) Asynchronous executions. In (b), file I/O (purple) and MPI communications (red, green) are overlapped with computations (cyan).

6. Self-review of Current Progress and Future Prospects

Deep learning models to enhance simulations

In 2023, we have successfully developed DL based SGS and DA models in 1D simulations. As a preparation to extend the developed methods to 2D simulations, we have developed a 2D Hasegawa-Wakatani turbulence code in Python and Kokkos. The code is based on 2D spectral method relying on FFTs. As a subproduct, we have developed kokkos-fft, which is published in kokkos organization [3]. We will investigate the applicability of developed methods to 2D Hasegawa-Wakatani turbulence, which remains as a future work. We will keep our implementation in a scalable manner to illustrate a pathway to the production codes. We are planning to implement the simulation code in C++ and the AI code in Python, which are coupled using PDI library for flexibility.

Flexible workflow for HPC fluid codes

We have explored the performance portable and asynchronous implementation in C++ “senders/receivers” that is proposed for C++26. As planned, we have re-implemented our mini-apps with “senders/receivers” and compared the performance with other programming models. We confirmed that the performance of “senders/receivers” is comparable with other programming models like C++ parallel algorithm (stdpar) and thrust. We have also completed the SYCL implementation which gives roughly the same performance as “senders/receivers”. The asynchronous capability of SYCL is currently under investigation. This remains as a future task. We aim to establish a baseline implementation for asynchronous execution of fluid simulation codes while keeping the performance portability.

7. List of Publications and Presentations

(1) Journal Papers (Refereed)

(2) Proceedings of International Conference Papers (Refereed)

[1] Y. Asahi, Y. Hasegawa, N. Onodera, T. Shimokawabe, H. Shiba, Y. Idomura, "Generating observation guided ensembles for data assimilation with denoising diffusion probabilistic model", SynS & ML Workshop @ ICML 2023 (7/28, Hawaii, US).

(3) Presentations at International conference (non-refereed)

[2] Y. Asahi, Y. Hasegawa, T. Padioleau(+), A. Millan, J. Bigot(+), V. Grandgirard(+), and K. Obrejan(+), "Performance portability of Ensemble Kalman Filter using C++ senders/receivers", 2023 Open Accelerated Computing Summit (10/4-5, Online)

(4) Presentations at domestic conference (non-refereed)

(5) Published open software library and so on.

[3] A shared memory FFT for the Kokkos ecosystem
https://github.com/kokkos/kokkos-fft

(6) Other (patents, press releases, books and so on)