jh210051-MDH

Development of Fast Surrogate for Approximating Large-scale 3D Blood Flow Simulation

Takashi Shimokawabe (The University of Tokyo)

Abstract
Coronary heart disease is a leading cause of death worldwide. The main cause of coronary heart disease is coronary stenosis, which is mainly due to atherosclerosis. Recently, computational fluid dynamics (CFD) has been used to compute the blood flow for patient-specific artery with medical images in diagnosing ischemic stenosis. However, CFD simulation requires a lot of computational resources and time. Therefore, in order to use CFD in clinical practice, it is essential to accelerate CFD analysis. In this project, we will use deep learning to build a fast surrogate for approximating the 3D blood flow simulation. In this year, we have developed a method to predict the results of time-dependent flows by using patch-based convolutional neural network (CNN) inference. By considering time-sequence data as three-dimensional data with two spatial dimensions and one temporal dimension, we apply to them a model consisting of 3D convolutional and de convolutional layers. This method allows us to apply the same neural network architecture to any size of 2D input data.

1. Basic Information

(1) Collaborating JHPCN Centers

Tokyo

(2) Research Areas

- ■ Very large-scale numerical computation
- ■ Very large-scale data processing
- □ Very large capacity network technology
- □ Very large-scale information systems

(3) Roles of Project Members

- Takashi Shimokawabe (The University of Tokyo): Development of a method for predicting large-scale simulation results
- Weichung Wang (National Taiwan University): Development of deep learning, surrogate modelling and algorithm designs
- Naoyuki Onodera (Japan Atomic Energy Agency): Advice and support to apply deep learning to CFD simulations
- Kengo Nakajima (The University of Tokyo): Advice and support for large-scale computations
- Toshihiro Hanawa (The University of Tokyo): Advice and support for large-scale deep learning
- Masashi Imano (The University of Tokyo): Advice and support for using OpenFOAM
- Hayato Shiba (The University of Tokyo): Advice and support of CFD and medical simulations
- Hiromichi Nagao (The University of Tokyo): Advice on machine learning methods
- Hiroya Matsuba (The University of Tokyo): Advice on machine learning execution environment
- Shota Suzuki (The University of Tokyo): Development of CFD simulations
- Takuro Omori (The University of Tokyo): Development of CFD

simulations

- Akira Hatakeyama (The University of Tokyo): Development of CFD simulations
- Cheng-Ying Chou (National Taiwan Normal University): Advice and support of CFD and medical imaging
- Che-Yu Hsu (National Taiwan University Hospital): Advice and support of medical backgrounds and knowledge
- Mei-Heng Yueh (National Taiwan Normal University): Development of computational geometry
- Yuehchou Lee (National Taiwan University): Development of CFD simulations and deep learning

2. Purpose and Significance of Research

Coronary heart disease is a leading cause of death worldwide. The main cause of coronary heart disease is coronary stenosis, which is mainly due to atherosclerosis. Fractional flow reserve (FFR) is defined as the ratio between distal pressure and proximal pressure and has been used as a standard tool to diagnose the severity of coronary stenosis. Recently, computational fluid dynamics (CFD) has been used to compute the blood flow and FFR for patient-specific artery. Some clinical trials demonstrated that the method combining CFD and medical image is better than the method using medical image solely in diagnosing ischemic stenosis. However, this method can be computationally demanding because it may take hours to perform CFD simulation. This drawback may limit the usage of this method in clinic practice. Therefore, it is indispensable to accelerate the process of CFD analysis.

In this study, we will use deep learning to build a fast surrogate for approximating the 3D blood flow simulation. We will also develop a parallelization method to make it possible to apply the deep learning to large-scale geometry. This method divides the large-scale geometry into multiple parts and applies deep learning in parallel to them. This makes it possible to approximate a large-scale 3D blood flow simulation.

3. Significance as JHPCN Joint Research Project

In this project, we are developing a fast surrogate that approximates 3D blood flow simulation using deep learning and a parallelization method of the surrogate for applications with large-scale geometry. Since we perform a large number of CFD simulations with large-scale geometry to generate training data sets and train deep neural networks (DNN) with these data sets to build the surrogate, a lot of computational resources are indispensable to realize this proposal. We use the lattice Boltzmann method (LBM) as CFD solver. Since LBM can achieve high performance on GPUs, we have exploited Reedbush-H/L and Wisteria/BDEC-01 (Wisteria) to generate the training data sets. We have utilized Oakforst-PACS, Reedbush-H/L and Wisteria for training DNN, since the deep learning frameworks we use can achieve high performance with Xeon Phi, which are installed on Oakforst-PACS, and GPUs, which are installed on both Reedbush-H/L and Wisteria. Due to queue configurations, we have used Reedbush-H for developing DNN models and Reedbush-L for long-time running for training. After the end of Reedbush-H/L's operation, we have used mainly Wisteria. This project is being carried out by collaborative research by blood flow experts, CFD experts, the experts of large-scale deep learning, and HPC experts. Therefore, implemented as a JHPCN joint research project, this project has been able to effectively carry out collaborative research and achieve research results.

4. Outline of Research Achievements up to FY2020

In our previous research, we have developed a method to predict the simulation results in large computational domains in 2D. This method combines neural network inference and boundary exchange. It exploits neural network to predict the simulation results of each subdomain and exchanges boundary between neighbor subdomains to maintain consistency in them. This method allows us to apply the same neural network architecture to any size of input data. We performed steady-state flow simulations with objects of simple shapes by using LBM. Using these results as training data sets, we trained the our neural networks. By using these networks with boundary exchange, our proposed method successfully predicted the results of a large-scale 2D steady flow.

We originally used Chainer as the framework for deep learning. However, since the development of Chainer was discontinued, we replaced it with PyTorch. We have introduced Horovod, which enables us to train the convolutional neural network (CNN) models on multiple GPUs. We have extended the 2D method to predict large-scale 3D steady flow simulations. Unlike the results of 2D prediction, the results of 3D prediction by the proposed method still have a large error at the boundary regions of subdomains.

5. Details of FY2021 Research Achievements

In the first half of this year, as originally planned, we have improved our code of LBM used for generating data sets to train deep neural network models. We have introduced the D3Q27 model and the cumulant LBM in our code, replacing the D3Q19 model, in order to be able to compute with higher accuracy and stability. We have also been investigating ways to generate complex geometries like blood vessels using Python's trimesh library.

The original plan was to extend the prediction method developed for two dimensions to three dimensions. However, the development of a 3D prediction method has not yet been completed and its results still have a large error at the boundary regions of subdomains. It is difficult to make predictions in 3D using a dataset of the geometry of actual blood vessels. Therefore, in the second half of this year, we began development of a prediction method for time-dependent flow, which was originally planned to be started after the development of the prediction method for 3D steady flow was completed. Since time-dependent flow must be considered in blood flow simulations in order to obtain more accurate results, this development itself is a necessary technical component for the overall progress of this project.

In this section, we describe the overview of the proposed prediction method for 2D simulations of time-dependent flow.

5.1 Dataset

We explain the data set of the neural network used in our method.

First, we place one or two cylinders or triangular polygons in a 1024 × 1024 computational domain and simulate a fluid with a Reynolds number of 100 flowing from negative to positive in the x direction by the lattice Boltzmann method (LBM). The number, type, size, and position of the cylinders or polygons to be placed are changed to perform several time-dependent flow simulations. The developed network model uses the computational results with a 64 × 64 region of multiple timesteps as input and predicts the subsequent frames with a 64 × 64 region of multiple timesteps. To create the dataset, we run 24 sets of LBM simulations. Three frames at 100-step intervals are stored as inputs after a certain time step, and three frames at 100-step intervals following the inputs are stored as prediction targets. From a computational domain of 1024 × 1024, we mechanically cut out 200 pieces of 64 × 64 regions allowing for some overlap of regions and without considering the arrangement of objects. In order to use the signed distance function (SDF), which represents the distance to the geometry, as the input, we also prepare the SDF data corresponding to each 64 × 64 computational domain. Since the fluid is time-varying, training data are extracted for multiple time steps from a single simulation execution. To enable data augmentation and to make the dataset independent of the direction of fluid flow, the rotated and flipped results of the LBM simulations are also added to the dataset, resulting in a final dataset of 65,021 for training and 27,867 for evaluation.

5.2 Neural network model

Figure 1 shows the structure of the neural network used in this method. The inputs to the neural network are the fluid density, the velocities in the x and y directions, and the SDF representing the object shapes for three time frames in a 64 × 64 region. The output is a three-frame prediction of the fluid density and velocities in the x and y directions. The output should be the predictions for the three time steps following the input data. The network used in this method consists of two major structures: the first half consists of multiple 3D convolutional layers, and the second half consists of multiple 3D deconvolutional layers. In order to predict the time dependent flow of the fluid, these time sequence data are considered as three-dimensional data, consisting of two dimensions in the x and y directions with a time axis as the third dimension. These data are trained in a network structure with 3D convolutional and deconvolutional layers. The tanh function is used as the activation function, and batch normalization is introduced in all layers except the input and output layers. In order to improve the prediction accuracy, skip connections used in U-net are introduced between a convolutional layer and its corresponding deconvolutional layer. The network model is implemented using PyTorch.

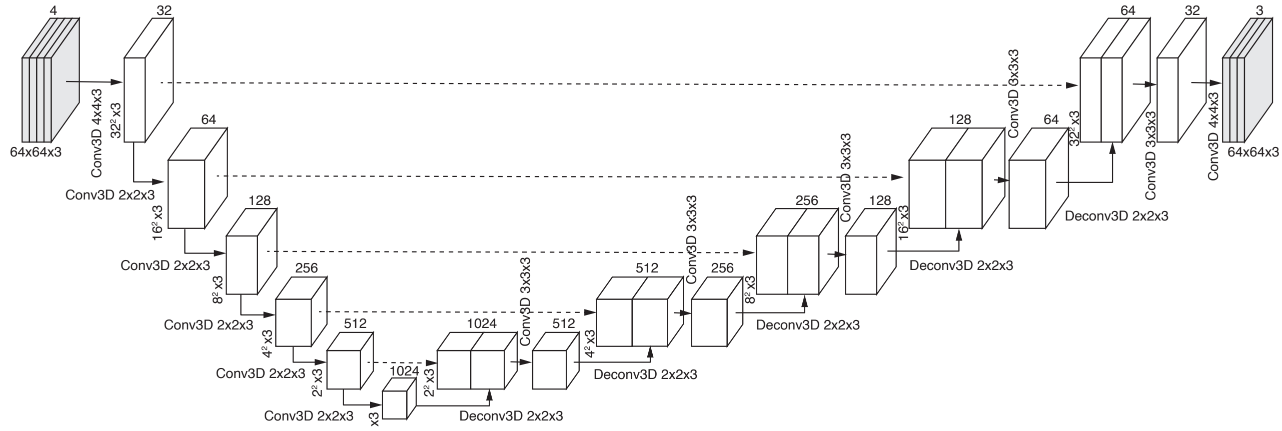

Figure 1: Network architecture for 2D simulations of time-dependent flow.

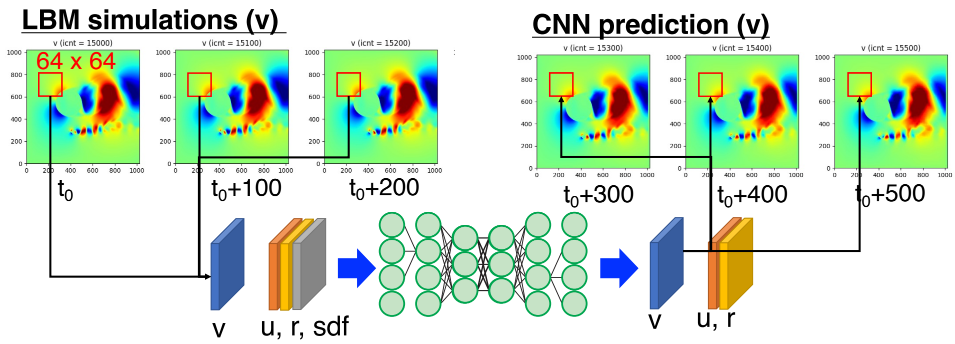

Figure 2: Prediction procedure for time-dependent flow with 3 frames of input and subsequent 3 frames of output. Patch-based CNN inference is used for prediction.

5.3 Training model and predicted results for time-dependent flow

For the training of this model, we use the Data/Learning Nodes (Aquarius) of the Wisteria/BDEC-01 supercomputer system installed at the Information Technology Center, the University of Tokyo. The Data/Learning node group (Aquarius) consists of 45 computation nodes with Intel 3rd generation Xeon scalable processors (Ice Lake) and NVIDIA A100 Tensor core GPUs. Each node is equipped with 8 GPUs. We use one node with 8 GPUs for training. We use Xavier for initialization, Adam for optimization, and Horovod as a framework for distributed deep learning training. We use the mean squared errors of the ground truth and predicted values of the density and velocity fields at each grid point for a loss function. We use, in addition, the mean squared error of the ground truth and predicted values of their spatial gradients. Computational time was 386 minutes for training the proposed neural network with 1500 epochs.

The neural model targets a 64×64 region. However, in order to predict the results of a large-scale simulation, it is necessary to be able to predict the results for a computational domain with an arbitrary size. Therefore, we exploits patch-based CNN inference for the prediction for the entire computational domain as shown in Figure 2. The entire computational domain is divided into a large number of 64×64 regions with each patch region overlapping. We apply the CNN inference to each 64×64 region. At each grid point, since several predicted values are obtained from the inference of the neural

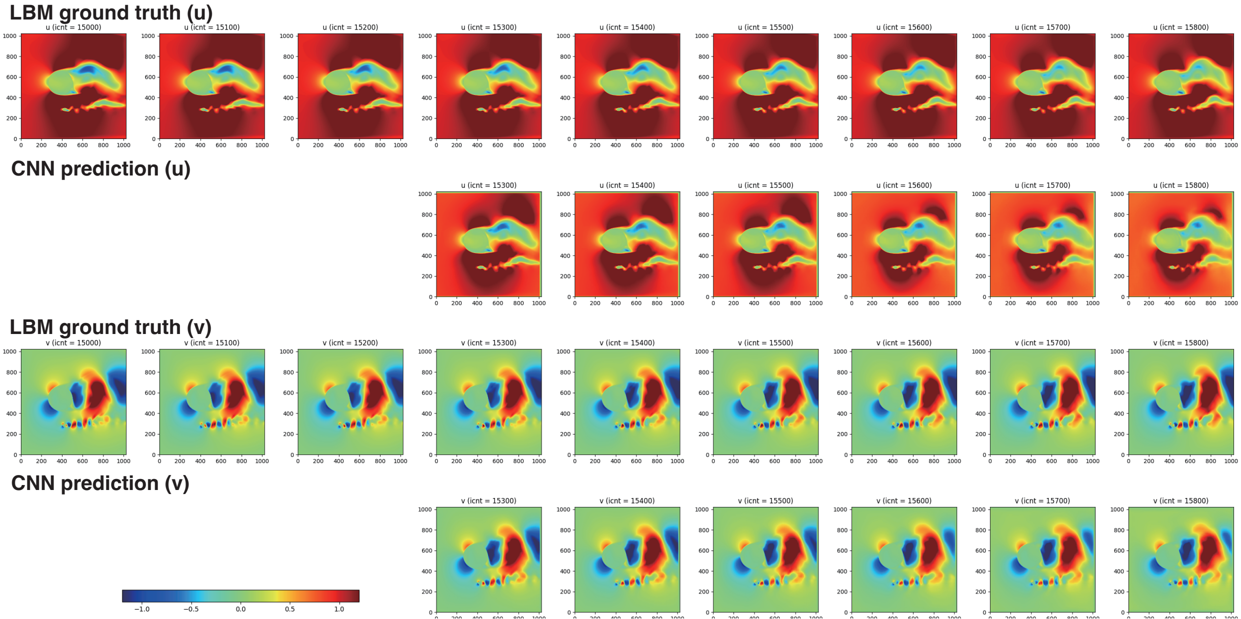

Figure 3: Prediction results over an entire computational domain using the patch-based CNN inference. The LBM ground truth and the CNN prediction of the velocities in the x and y directions (i.e., u and v) are shown. The images from 15000 to 15200 time steps are the input data generated by the LBM, and the images from subsequent time steps are the prediction results by the CNN.

network, the average of these values is used as the final prediction result. In this study, three frames for every 100 steps of the simulation are used as input, and the following three frames are predicted. To predict further results, the prediction results obtained by the neural network are used as input to predict the next three frames. By repeating this process, the prediction is advanced in time.

In Figure 3, the neural network predicts the results of 15,300, 15,400, and 15,500 time steps using three frames of 15,000, 15,100, and 15,200 time steps as input. By repeatedly applying the CNN inference spatially and temporally, the neural network predicts the subsequent three frames of 15,600, 15,700, and 15,800 time steps. In other words, the prediction results are obtained every 100 steps between 15,300 and 15,800 time steps. This figure shows the results of the LBM ground truth of the velocity in the x direction (u) and its prediction by the neural network, and those of the velocity in the y direction (v) and its prediction by the neural network from the top. Although only the simulation results of one frame per 100 steps are used, it can be seen that the proposed neural network is able to predict the simulation results by LBM well.

6. Progress during FY2021 and Future Prospects

In this year, as originally planned, we have improved the code for the lattice Boltzmann method (LBM) used to generate datasets for training deep neural network models. The D3Q27 model and cumulant LBM were introduced into the code in place of the D3Q19 model for higher accuracy and stability. We have started development of a method for predicting time-dependent flow, which was scheduled to begin after the development of the method for predicting 3D steady flow was completed. This development itself is a necessary technical component to the overall progress of this project, since blood flow simulations must need time-dependent flow in order to obtain more accurate results.

The research plan for the next year is as follows. We will continue to improve our methods for predicting the results of 3D large-scale steady-state flow simulations. We have found that the patch-based CNN inference developed this year for predicting time-dependent flows is one of the effective way. By using this patch-based CNN inference, the accuracy of the prediction for the steady flow simulations may be improved. After we are able to predict 3D steady flow results by using deep learning, we will establish a method for predicting a large-scale 3D steady flow of the blood flow simulation.

7. List of Publications and Presentations

Please note that items in status of "to be submitted/presented" and "submitted" cannot be included.

(1) Journal Papers (Refereed)

None.

(2) Proceedings of International Conferences (Refereed)

[1] Yuuichi Asahi, Sora Hatayama, Takashi Shimokawabe, Naoyuki Onodera, Yuta Hasegawa and Yasuhiro Idomura, "AMR-Net: Convolutional Neural Networks for Multi-resolution Steady Flow Prediction", The 2nd Workshop on Artificial Intelligence and Machine Learning for Scientific Applications, IEEE Cluster 2021, Online, Sep. 2021.

(3) International conference Papers (Non-refereed)

[2] Shota Suzuki, Takashi Shimokawabe, "Acoustic simulation using lattice Boltzmann method by multi-GPU parallel computing", International Conference on High Performance Computing in Asia-Pacific Region (HPCAsia) 2022, Online, Jan. 2022.(poster)

[3] Akira Hatakeyama, Takashi Shimokawabe, "Multi-GPU computing of moving boundary flow using lattice Boltzmann method", International Conference on High Performance Computing in Asia-Pacific Region (HPCAsia) 2022, Online, Jan. 2022.(poster)

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

[4] 鈴木翔太，下川辺隆史，"格子ボルツマン法に基づく GPU を用いた音響解析"，第 26 回計算工学講演会，オンライン開催，2021 年 5 月．（優秀講演表彰受賞）

[5] 朝比祐一，畑山そら，下川辺隆史，小野寺直幸，長谷川雄太，井戸村泰宏，"機械学習による細分化格子に基づく 2 次元定常流予測"，第 26 回計算工学講演会，オンライン開催，2021 年 5 月．

[6] 鈴木翔太，下川辺隆史，"埋め込み境界法を適用した格子ボルツマン法に基づく 3 次元音響解析"，オープン CAE シンポジウム 2021，オンライン開催，2021 年 12 月．

[7] 鈴木 翔太，下川辺 隆史，“格子ボルツマン法によるインピーダンス境界を用いた音響解析手法の構築”，日本音響学会 2022 年春季研究発表会，オンライン開催，2022 年 3 月.
[8] 下川辺隆史，“深層学習による流体シミュレーション結果予測”，第 41 回計算数理工学フォーラム，オンライン開催，2022 年 3 月.（招待講演）

(5) Published library and relating data
None.

(6) Other (patents, press releases, books and so on)
None.