Vol.2020-HPC-177 No.4
2020/12/21

深層学習による混相流の時間発展シミュレーション結果の予測手法の検討

長谷川 敦[1,a)]　下川辺 隆史[2,b)]

概要：流体における時間発展シミュレーションは工業分野において多く取り入れられている．しかし，従来の流体シミュレーションでは多大な計算時間やリソースを要するといった課題を抱えている．そこで，本研究では混相流流体シミュレーションの時間発展解析について深層学習を用いた予測を可能とすることを目的としてネットワークの開発を行う．ネットワークの学習に用いるデータセットは，OpenFOAM を利用して，混相流問題である二次元ダムブレイクシミュレーションにより生成する．ダムブレイクとは水槽中で堰き止められた水が流れる様子ついて扱う流体問題である．本研究では，入力の数フレームから連続した次の数フレームを予測して出力するようなネットワークモデルの開発を行い，生成したデータセットを用いて学習・予測を行いその結果を評価しモデルの精度について検討する．

キーワード：数値流体力学，畳み込みニューラルネットワーク，OpenFOAM, 非定常流

Examination of prediction method of time evolution simulation result of multiphase flow by deep learning

Abstract: Time evolution simulations in fluids are widely used in the industrial field. However, conventional fluid simulations have problems that are required a large amount of calculation time and resources. Therefore, in this work, we develop a network to make it possible to predict the time evolution analysis of multiphase flow fluid simulation using deep learning. The data set used for learning our network model is generated by two-dimensional dam break simulation, which is a multiphase flow problem, using OpenFOAM. A dam break is a fluid problem that deals with the flow of dammed water in a case. In this work, we develop and train a network model that predicts the next few consecutive frames from several frames of input with the generated data set. Then we evaluate the result and examine the accuracy of the trained model.

Keywords: Computational fluid dynamics, Convolutional neural network, OpenFOAM, Time-dependent flow

1. 研究背景と目的

数値流体力学は工学分野において非常に広く利用されている手法であり，特に非定常流に関する計算は重要な役割を占めている．流体の時間変化による振る舞いを見ることは，流体に関わる構造の設計や現象の解明に繋がるため欠かせない操作である．しかし，数値流体力学における解析の問題点として流体力学の支配方程式であるナビエ・ストークス方程式やオイラー方程式を解く際に偏微分方程式を解く必要がある．その際，計算する格子点の数に応じて計算リソースが膨大になってしまうという点や圧力の計算において離散化したポアソン方程式を解く際にループ計算が必要となり多大な計算時間を要するといった問題点がある．こうした問題の解決法として計算時間を要する一部の計算プロセスを深層学習を用いることで高速化する研究 [8] [7] もある．そこで本研究では，時間発展する混相流のシミュレーション結果を深層学習によって高速に予測する手法を開発することを目的とする．

2. 混相流解析

本研究では，混相流における流体シミュレーションの

[1] 東京大学大学院工学系研究科電気系工学専攻
[2] 東京大学情報基盤センター
[a)] hasegawa@cspp.cc.u-tokyo.ac.jp
[b)] shimokawabe@cc.u-tokyo.ac.jp

ⓒ 2020 Information Processing Society of Japan

ツールとして OpenFOAM [2] を用いる．OpenFOAM は工業的に広く利用されており，オープンソースであることから様々な企業や研究所で活用され改良され続けている．

混相流解析とは気体と液体などといった複数の相が混ざり合った流れ（混相流）についての解析である．今回は気体と液体の混相流である気液二相流についての解析を行う．流体における数値計算では支配方程式について離散化を行い計算機を用いて結果を算出する．

2.1 支配方程式

流体の運動を記述する支配方程式としてナビエ・ストークス方程式がある．また，ナビエ・ストークス方程式から粘性項をなくした完全流体についての運動を記述したオイラー方程式がある．数値流体力学では，これらの式について時間および空間について細かく分割して計算することで近似解を導く．本研究では等温非混和性・非圧縮性の混相流を対象とした解析をしている．そこで対象とするべき基礎方程式は以下の 2 つとなる．

$$\nabla \cdot \boldsymbol{u} = 0 \tag{1}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot (\boldsymbol{u}\boldsymbol{u}) = -\frac{1}{\rho}\nabla p + \nabla \cdot \left[\mu \left\{\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T\right\}\right] + \frac{1}{\rho}\boldsymbol{b} \tag{2}$$

(1) は連続の式，(2) はナビエ・ストークス方程式である．ここで，$\boldsymbol{u}$ は流速ベクトル，ρ は密度，p は圧力 μ は粘性係数，$\boldsymbol{b}$ は体積力ベクトルを表す．

2.2 有限体積法

OpenFOAM において有限体積法が用いられている．有限体積法とは偏微分方程式の離散化手法の一つであり，計算対象となる領域を格子状に分割して，それぞれの格子における流体の支配方程式を積分することを繰り返し，全領域で保存則を満たすような離散化された式を導く. 有限体積法における格子の例を**図 1** に示す．この手法では構造格子，非構造格子ともに使用することができ，複雑な形状に対しても適用が可能であるという利点から現在の流体解析における主流な離散化手法の一つとなっている. 有限体積法では通常の差分法と違い計算格子において，スカラー量とベクトル量をそれぞれ別の定義点を取る.OpenFOAM では有限体積法における計算手法のうち，速度と圧力の解法については SIMPLE 法と PISO 法のハイブリッド解法である PIMPLE [4] 法が利用されている. この手法によ って最終的な状態のみを求める定常ではなくその経過を時間を追って見ることができる非定常解析を行うことができる. また，PISO 法に比べ大きな時間差を取ることができる反面，各時間ステップでの反復計算を適切に行う必要があるため非常に多くの演算時間を要する．

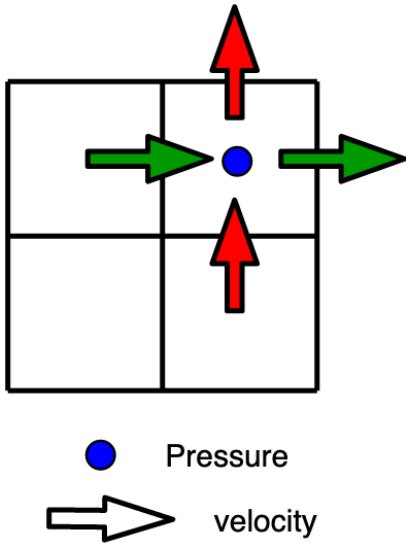


図 1 FVMgrid

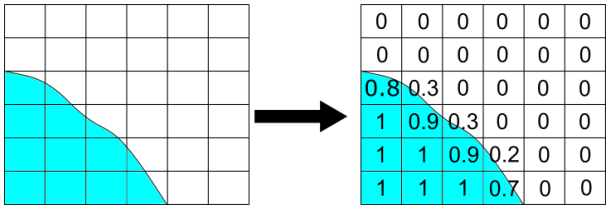


図 2 VOF

2.3 VOF 法

本研究において混相流の問題を扱う際のソルバーとして InterFoam を利用する．InterFoam では流体体積 (VOF:Volume of Fluid) 法を利用している．VOF 法とは自由表面についての流れの解析手法で界面捕獲法の一つである．界面捕獲法は界面の移動を界面関数を用いて定義し，固定メッシュにおける界面の位置を計算によって得る方法である．VOF 法の例を**図 2** に示す．一般的に VOF 法における移流方程式は次の式で表される．

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot (\alpha \boldsymbol{u}_{\mathrm{water}}) = 0 \tag{3}$$

(3) において α は流体の体積率を示し，0 から 1 の範囲で変化する．しかし，OpenFOAM では上記の式で生じる数値拡散を抑えるために平均流速を用いた修正項を追加することでより鮮明な界面を表現することができる．

3. Convolution Neural Network

本研究では深層学習の中でも特に畳み込みニューラルネットワーク (CNN:Convolutional Neural Network) [5] を用いてネットワークモデルを作成し学習を及び予測を行う．CNN は画像認識や音声テキスト変換，合成などに利用されており，他にも座標空間にマッピングされたものに対しても有効である．例として AlphaGO [6] や，ポーカー [3] などへの利用例がある．CNN の一般的な例を**図 3** に示す．CNN で主な層として畳み込み層とプーリング層がある．

ⓒ 2020 Information Processing Society of Japan

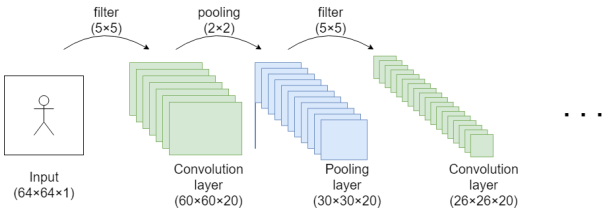


図 3 An example of CNN

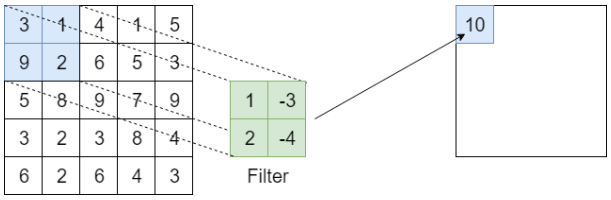


図 4 An example of convolution layer

畳み込みに用いられるフィルタを学習し，プーリング層において特徴量の抽出を行うことで目的とするデータを表現できるようなネットワークに近づける．畳み込み層における操作の例を**図 4** に示す．畳み込み層では，入力に対してフィルタを掛け合わせる．入力データに対してフィルタを通して出力されたものが次の層への入力となる．プーリングでは畳み込みを行った後の出力から特徴的な量だけを取り出す操作が行われる．プーリング層にも種類があり最大プーリング層では，あるフィルタサイズのうちの最大値を取り出し圧縮する．

4. CNN を用いた流体シミュレーション結果の時間変化の予測

本研究では，指定した空間における非定常な流体の解析を OpenFOAM を用いて行い，その結果を構築したネットワークモデルで学習することでより高速なシミュレーション結果の予測が可能となるかの検討を行う．ここでは，予測に使用するデータセット，ネットワークモデルについて説明する．

4.1 概要

本研究における概要を説明する．今回行った研究の概要を**図 5** に示す．本研究では，CNN によって混相流における流体の流れを，流体の運動に関するパラメータから連続した次のタイムステップにおけるパラメータの予測を行う．今回用いたデータセットは格子法によって生成されたデータであることから，各格子データを二次元配列として受け取れることに着目し，畳み込みの操作と非常に親和性が高いことから CNN をネットワークモデルとして採用した．今回は二次元における解析を行い，計算領域を 64×64 とする．

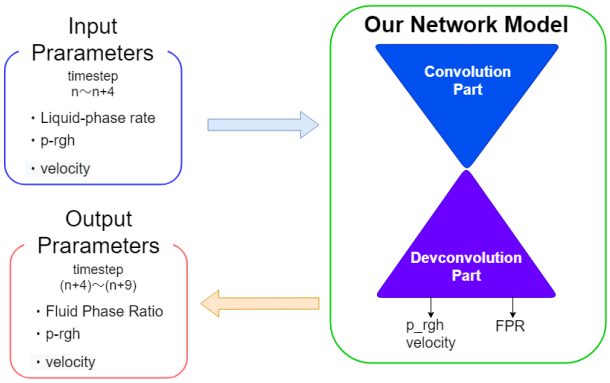


図 5 Our convolutional neural network learns to generate fluid simulations in next 5steps from 5steps parameters

4.2 データセット

本研究では混相流における流体の流れの問題のうちダムブレイク流れについて取り扱う．学習及び予測に用いるデータセットとして OpenFOAM によって生成したシミュレーションデータを用いる．OpenFOAM において格子法を用いて計算されたデータであることを利用して二次元配列に時間軸を足した 3 次元データセットとする．その際に利用するパラメータとして液相率，二次元方向に分割した流速，圧力場の静水圧との差分をそれぞれ指定している．また，ダムブレイク流れにおける水柱の大きさの調整を行いシミュレーションケースを複数用意する．O 水柱の幅を 8〜24 格子分，高さを 4〜40 格子分それぞれ間 4 格子となるように変化させる．合計 50 個分のシミュレーションケースを用意し，そのうち時間幅として 0〜2 秒を 0.02 秒毎にタイムステップとして出力したものから入力データ 5，正解データ 5 とした連続した 10 個のタイムステップを抜き出していく．こうして $64 \times 64 \times 10$ の 4 パラメータのデータを一つのデータセットとしたものを学習用 400，評価用 100 の合計 500 として用意する．データセットの生成には，東京大学情報基盤センターにあるスーパーコンピュータ Oakforest-PACS を利用している．OpenFOAM で 1 つのダムブレイク流れのシミュレーションを実行した場合，今回の計算条件においては平均 78.14 秒要した．

4.3 時間発展予測モデル

図 6 に本手法で用いたネットワークの構造を示す．この図において，入力と出力はその 3 次元の大きさとチャンネル数，畳み込み層の引数は，フィルターの大きさと出力のチャンネル数である．ネットワークの入力として各層は畳み込み層とバッチノーマライゼーション層，逆畳み込み層からなり，5 度畳み込みを行い，同じだけ逆畳み込みを行うようなネットワークとなっている．また，活性化関数にモデルの中間では双曲線関数 (tanh:Hyperbolic tangent function) を用いて，ネットワークの最後ではチャンネルで分岐を行い tanh とシグモイド関数を用いている．また，

通常 CNN のネットワーク構造において多くの場合，各画像のパートからより代表的な特徴量を抽出するための操作として大抵の場合プーリング層が入れられる．そこで本研究でもプーリング層を入れた場合の比較を行う．プーリング層として最大プーリング層を利用する．このネットワークの入出力として連続した 10 個のタイムステップのうち前半 5 ステップのデータを入力することで後半 5 ステップの予測が可能となることを見込んだ．また，入力するパラメータが液相率，静水圧との差分の圧力，流速が 2 次元であり，それぞれをチャンネルとして区分している．各パラメータをネットワークに入力する際，0〜1 の範囲もしくは −1〜1 の範囲にデータの分布が収まるようにスケーリングを行っている．本手法で用いたネットワークモデルは，Wiewel,Becher, および Thuerey らが発表したモデル [10] や Byungsoo Kim, Vinicius C. Azevedo，および Nils Thuerey らの研究で用いられているネットワークモデル [1] の一部であるオートエンコーダを参考にモデルの作成を行っている．ディープラーニング用フレームワークとして PyTorch を使用し，バッチ数を 64，重みの初期化に正規分布，学関数習率を 0.005, 損失関数に平均自乗誤差 (MSE:Mean Squared Error) を用いる．

5. 学習および予測結果

プーリング層があるモデルとないモデルそれぞれで学習を行った．学習には東京大学情報基盤センターにあるスーパーコンピュータ Reedbush-L を利用している．学習にはエポック数を 2000 とし計算時間として全体で 3382.14 秒，1 エポックあたり 1.69 秒かかった．また，MSE による Loss の大きさは 6.20×10^{-5} となった．

表 1 Comparison results of trained network with multi-GPU

Number of GPUs	Loss	Average calculation time(s)
1	6.20×10^{-5}	1.69
2	1.06×10^{-4}	1.00
4	1.43×10^{-4}	0.48

5.1 マルチ GPU による並列計算

本研究ではマルチ GPU による並列計算を実行することで学習の高速化を図っている．今回用いた Reedbush-L では GPU として NVIDIA Tesla P100(Pascal) を搭載している．機械学習のフレームワークとして使用する PyTorch で並列計算を実行するにあたり Horovod [9] を利用した．各ノードあたりの GPU 搭載数は 4 であり，GPU 数が 2 つ，4 つの場合の計算速度と Loss の精度の比較を行った．GPU 数を 1,2,4 と変化させたときの平均計算時間と MSE における Loss の大きさを表 1 に示す．2GPU では 2008.61 秒，4GPU では 962.17 秒となった．表より，GPU 数倍近くの高速化がなされていることがわかる．なお，ここで学習したモデルはプーリング層を持たないネットワークモデルである．

5.2 プーリング層を含むネットワークの学習結果

プーリング層が本研究において有用であるかを確認するため，プーリング層を含んだ場合におけるネットワークの学習を行う．プーリング層を含んで各 GPU 数毎に学習を行った結果を表 2 に示す．エポック数や学習率などについてはプーリング層を入れない場合と同じ条件で学習を行った．結果からプーリング層を入れた場合における特徴量の正確な抽出などは特に望めず，最大プーリング層における操作の分だけ時間がかかっていることが確認できる．

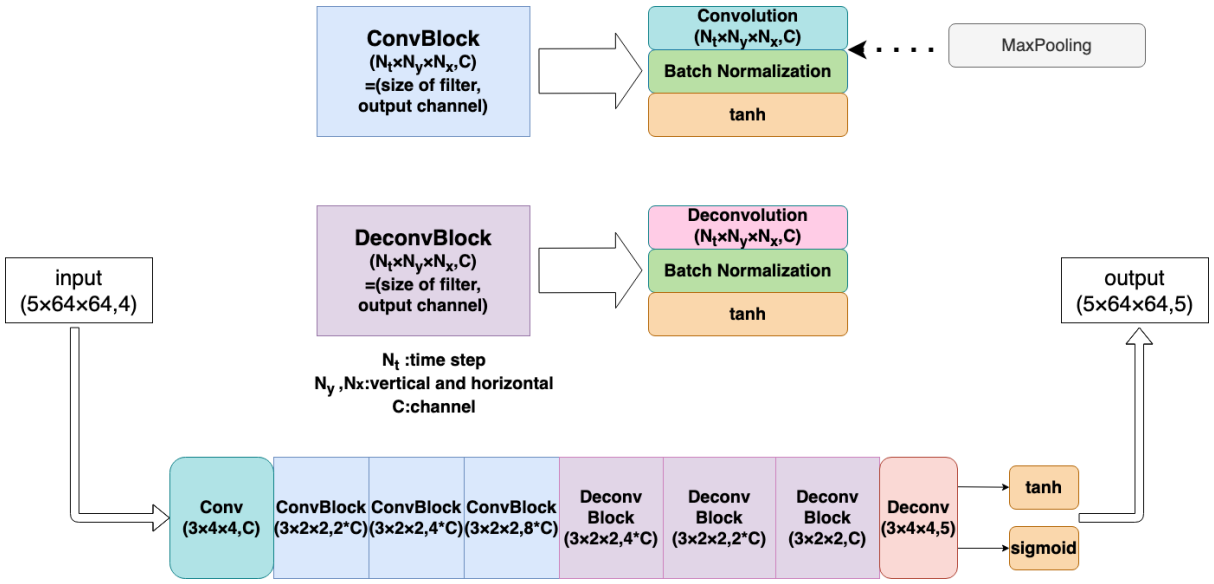


図 6 Time evolution convolutional neural network

表 2 Comparison results of trained network involving Pooling layers

Number of GPUs	Loss	Average calculation time(s)
1	6.70×10^{-5}	2.07
2	1.06×10^{-4}	1.00
4	1.65×10^{-4}	0.61

5.3 学習済みモデルによる予測結果

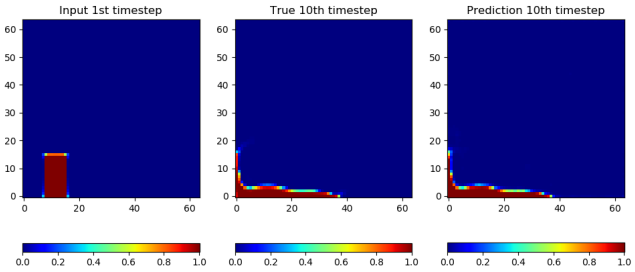
今回提案したモデルを学習したものからダムブレイク流れのシミュレーションを予測した結果を**図 7** に示す. 図 7 において，入力が前半 5 ステップ，出力および正解データが後半 5 ステップの計 10 ステップのうち，一番左の画像は入力の 1 ステップ目，真ん中の画像は正解データの 10 ステップ目，一番右の画像は予測データの 10 ステップ目をそれぞれ表している．また，予測に要した時間は 10 個のシミュレーションケースに対して 3.30 秒であった．今回提案したモデルにおける予測結果の図からは，視覚的に大きな誤差は確認出来ないほどであることがわかった．

次に，あるタイムステップにおける正解と予測の誤差を視覚化するため，横軸に正解データにおける値，縦軸に正解データにおける値としてそれぞれの格子についてプロットしたグラフを各パラメータ毎に**図 8** に示す．図中，近似直線を表した式と相関係数を表示しており，PointID は計算領域を横に 4 分割して各領域を表したものである．左下を 0 として横方向を先として番号を増やしていき，右上を 4095 となるようにしている．この結果において正確に予測が出来ている場合グラフの傾きはより 1 に近づくものとなるが，それぞれどのパラメータにおいてもばらつきがある．また，データのスケーリングを行っていた圧力や速度に関しては液相率と比較して相関係数が低くなっていることがわかる．

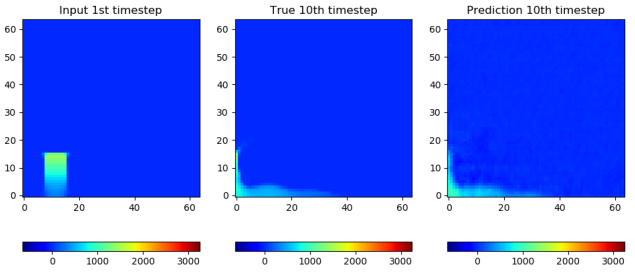
また，あるシミュレーションケースにおいて，ある格子点の時間的変化を最初の 5 つのタイムステップから次のタイムステップの予測を行い，また予測した結果からその先の予測を行うことを繰り返し，シミュレーションの最後まで予測を行った場合に得られた結果を**図 9** に示す．同じシミュレーションケース，および格子における予測結果をそれぞれのパラメータについてグラフにしている．このグラフは，計算領域において，64×64 の格子の左下を原点としたときの右側に 8 格子上に 3 格子移動した座標，つまり (8,3) におけるデータについて予測を行い正解データとの比較を行っている．グラフの結果から時間変化予測においてより正確な予測は出来ていないが概形としては近いものとなっている．図 8，図 9 について Cheng M. らの研究 [11] で用いたグラフを参考に作成を行った．

6. まとめと今後の展望

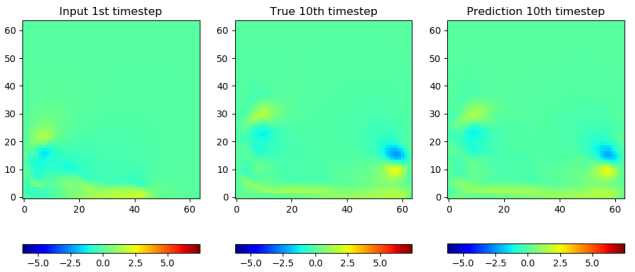
本研究では流体シミュレーションにおける非定常解析の高速化を目的として，CNN を利用したダムブレイク流れの



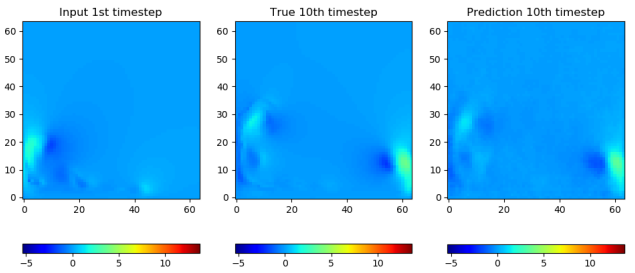
(a)Liquid-phase rate



(b)Difference between pressure and hydrostatic pressure



(c)Horizontal velocity

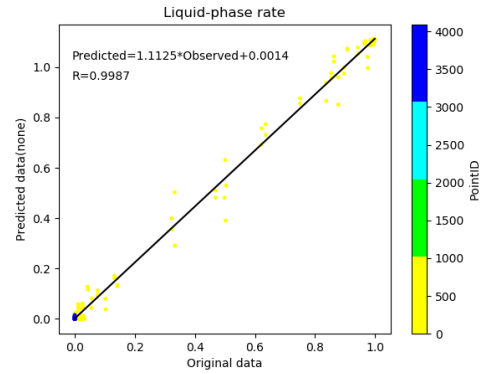


(d)Vertical velocity

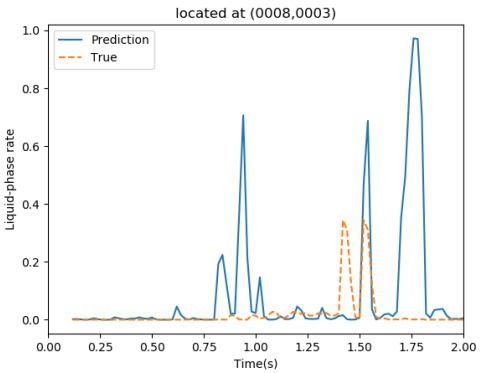
図 7 Fluid flow images of network input,predicted result and true data

予測の有用性について検討を行った．結果として，シミュレーション結果の画像としては正解画像と比較したとき視覚的に差があまりないということがわかった．しかし，数値的な誤差には改善の余地があり，精密な流体データを必要とする場合は現在のモデルによる予測では不十分である．学習時間については GPU による高速化が可能であることがわかった．しかし精度が少し落ちてしまうこともあり精

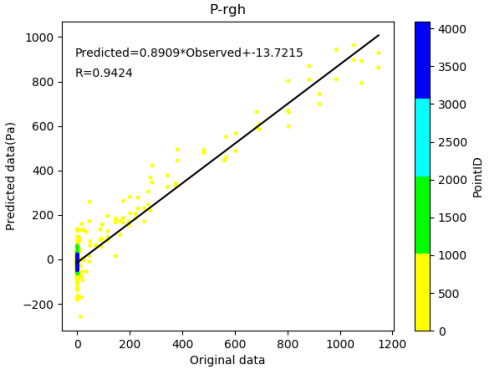
ⓒ 2020 Information Processing Society of Japan



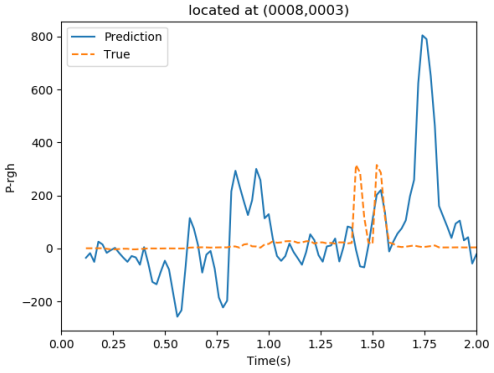
(a)Liquid-phase rate



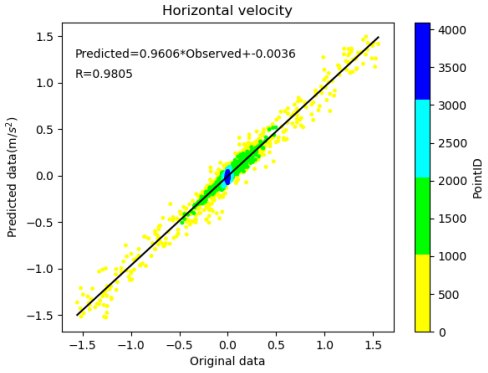
(a)Liquid-phase rate



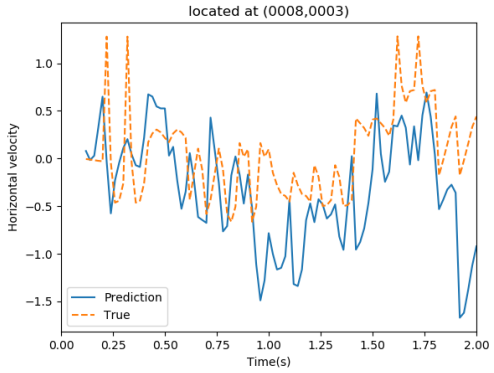
(b)Pressure without hydrostatic pressure



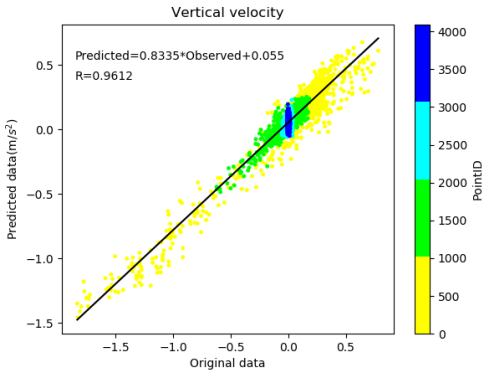
(b)Pressure without hydrostatic pressure



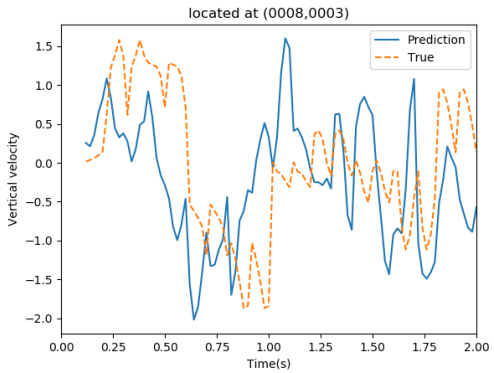
(c)Horizontal velocity



(c)Horizontal velocity



(d)Vertical velocity



(d)Vertical velocity

図 8 Differences and correlation coefficients of each fluids parameters between predicted data and the original data

図 9 Comparison of the temporal varidation of each parameters at (8,3)

ⓒ 2020 Information Processing Society of Japan

度と速度ともに最適な条件の模索が必要である．また，予測時間に関しては OpenFOAM ではひとつのケースに対して平均 78.14 秒かかっていたのに対して，CNN では 1 シミュレーションケースあたり 10 ステップの予測を行った時間が 0.33 秒であることから OpenFOAM の予測分 100 タイムステップに換算した場合の 3.30 秒と比較して高速化に成功していると考えられる．今後の展望として現在構築している CNN に対してより流体シミュレーションにおける表現が可能なモデルを構築したいと考えている．また，現在は損失関数として MSE を用いているが，流体の運動を考慮したときにそれぞれの関係をより明確にするような関数の定義することが望まれる．

謝辞 本研究の一部は科学研究費補助金 課題番号 19H05662，課題番号 20K21787，および，学際大規模情報基盤共同利用・共同研究拠点，および，革新的ハイパフォーマンス・コンピューティング・インフラから支援を頂いた．記して謝意を表す．

参考文献

[1] Kim, Byungsoo, et al.: *Deep fluids: A generative network for parameterized fluid simulations*, Computer Graphics Forum. pp. 59-70. (2019).
[2] Jasak, Hrvoje, Aleksandar Jemcov, and Zeljko Tukovic.:*OpenFOAM: A C++ library for complex physics simulations.*, International workshop on coupled methods in numerical dynamics. Vol. 1000. IUC Dubrovnik Croatia (2007).
[3] Yakovenko, Nikolai, et al.:*Poker-CNN: a pattern learning strategy for making draws and bets in poker games.*, arXiv preprint arXiv:1509.06731 (2015).
[4] Ferziger, Joel H., Milovan Perić, and Robert L. Street.:*Computational methods for fluid dynamics*, Vol. 3. Berlin: springer (2002).
[5] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton.:*Imagenet classification with deep convolutional neural networks.*, Communications of the ACM 60.6 (2017).
[6] Silver, David, et al.:*Mastering the game of Go with deep neural networks and tree search.*, nature 529.7587 pp.484-489.(2016).
[7] Tompson, Jonathan, et al.:*Accelerating eulerian fluid simulation with convolutional networks.*, International Conference on Machine Learning. PMLR (2017).
[8] Yang2016, Cheng, Xubo Yang, and Xiangyun Xiao.:*Data‐driven projection method in fluid simulation.*, Computer Animation and Virtual Worlds 27.3-4 pp.415-424.(2016).
[9] Sergeev, Alexander, and Mike Del Balso.:*Horovod: fast and easy distributed deep learning in TensorFlow.*, arXiv preprint arXiv:1802.05799 (2018).
[10] Wiewel, Steffen, Moritz Becher, and Nils Thuerey.:*Latent space physics: Towards learning the temporal evolution of fluid flow.*,Computer Graphics Forum. Vol. 38. No. 2. (2019).
[11] Cheng M., Fang F.,Pain C.C. and Navon I.M.:*An advanced hybrid deep adversarial autoencoder for parameterized nonlinear fluid flow modelling.* Computer Methods in Applied Mechanics and Engineering:372.113375 (2020).

ⓒ 2020 Information Processing Society of Japan