

深層学習と境界交換を用いた複数領域にまたがる 定常流のシミュレーション結果の予測

畑山 そら^{1,a)} 下川辺 隆史^{2,b)} 小野寺 直幸^{3,c)}

概要: 流体シミュレーションにおいて広く用いられる数値流体力学解析には計算機のメモリ消費量と計算時間に課題があり、近年それらを解決すべく深層学習を用いた従来の数値計算結果を高速に予測する研究が行われているが、予測が可能な領域の大きさが限定される。そこで本研究では、深層学習を用いた定常流体のシミュレーション結果の推論と境界条件の交換を用い、より大きな領域を予測可能な手法を提案する。

キーワード: 数値流体力学, 格子ボルツマン法, 符号付き距離関数

Steady flow prediction across multiple regions using deep learning and boundary exchange

SORA HATAYAMA^{1,a)} TAKASHI SHIMOKAWABE^{2,b)} NAOYUKI ONODERA^{3,c)}

Abstract: Computational fluid dynamics analysis, which is widely used in fluid dynamics simulations, has problems in terms of computer memory consumption and computation time. In recent years, researches have been carried out to solve these problems by using deep learning to rapidly predict the results of conventional numerical calculations, but the size of the region that can be predicted is limited. Therefore, in this study, we use deep learning to simulate the results of steady-state fluid using an exchange of inference and boundary conditions, we propose a method that can predict larger regions.

Keywords: Computational fluid dynamics, Lattice Boltzmann method, Signed distance function

1. 研究背景と目的

数値流体力学解析は流体の解析手法として広く知られており、速度、密度、圧力の時間変化がない流体である定常流のシミュレーションにおいてもこの数値流体力学解析が用いられる。しかし、数値流体力学解析では、解析の過程で流体の支配方程式であるナビエ・ストークス方程式等の偏微分方程式を解く必要があることから計算機の計算メモリを大幅に消費してしまうという問題点がある。また、定常流のシミュレーションに用いた際には、流体が定常状態

に達するまで計算を続けなければならないことから、計算時間を大幅に要する。そこで本研究では、深層学習を用いることで、定常流体のシミュレーション結果をより高速に算出することを目的とする。

2. 定常流のシミュレーション手法

本研究では、数値流体力学解析を用いた定常流のシミュレーション結果を深層学習の教師データとして用いて学習を行い、深層学習モデルを作成することで、シミュレーションを高速化している。数値流体力学解析は、流体の支配方程式であるナビエ・ストークス方程式、オイラー方程式等を離散近似し、計算機を用いて解析する手法である。数値流体力学解析の代表的な手法としては有限体積法、有限要素法、有限差分法等が挙げられる。本研究では、大規

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

³ 日本原子力研究開発機構

a) hatayama@cspp.cc.u-tokyo.ac.jp

b) shimokawabe@cc.u-tokyo.ac.jp

c) onodera.naoyuki@jaea.go.jp

模計算を効率的に実行できると近年注目されている格子ボルツマン法 [2], [5], [6], [7], [8], [10] (Lattice Boltzmann Method, 以下 LBM とよぶ) を用いる。

2.1 格子ボルツマン法の基礎方程式

LBM は、流体を微視的にとらえて粒子の集まりとみなし、粒子の移流と衝突の運動を支配するボルツマン方程式を流れ場の解析に適用した数値流体力学解析手法である。ただし、LBM は個々の粒子のふるまいを扱うのではなく粒子の分布関数の時間発展方程式を解くもので、基本的にはこの分布関数は連続的な関数である。LBM は Boltzmann 方程式の速度空間を有限個の離散速度で表すことで

$$\frac{\partial f_i(t, x)}{\partial t} + c_{i\alpha} \frac{\partial f_i(t, x)}{\partial x_\alpha} = \Omega_i(f) \quad (1)$$

と書き換える。ここで f_i は粒子の分布関数をあらわし、添え字の i は粒子の持つ有限個の速度、つまり粒子の種類の数に対応しており、 α は直角座標を表す。 f_i は時間 t と位置 x (格子の位置を位置ベクトルで表している) の関数で実数値を取る。すなわち f_i は、各格子点におけるある速度の粒子の数を表している。右辺は衝突を表す項で、一般には分布関数 f の非線形関数である。一般には衝突項を単純化した次の離散化 BGK (Bhatnagar-Gross-Krook) 方程式

$$f_i(x + e_i \Delta t, t + \Delta t) = f_i(x, t) - \tau^{-1} (f_i(x, t) - f_i^{\text{eq}}(x, t)) \quad (2)$$

が用いられる。右辺の衝突項に現われる f_i^{eq} は局所平衡分布関数で、 τ は分布関数が衝突により局所平衡分布関数に近づく時間の尺度で、緩和時間係数と呼ばれる。

2.2 格子形状

本研究では、LBM の格子形状として 2 次元 9 速度速度モデル (D2Q9) を用いる。D2Q9 における局所平衡分布関数を式 3 に示す。

$$f_i^{\text{eq}}(x, t) = w_i \rho \left[1 - \frac{3}{2} (u)^2 + 3 (e_i \cdot u) + \frac{9}{2} (e_i \cdot u)^2 \right] \quad (3)$$

ここで、 ρ , u はそれぞれ流体の巨視的な密度と速度を、 e_i は仮想粒子の速度を表す。また、係数 w_i は、異なる速度方向に対する重み係数であり、 $i = 0$ に対して $4/9$, $i = 1-4$ に対して $1/9$, $i = 5-8$ に対して $1/36$ である。粒子は図 1 に示すように静止粒子を含め 9 種類である。

2.3 計算条件

本研究では、LBM を用いて、レイノルズ数 20 で流体が領域中を x 軸の正方向に流れる状況を計算する。LBM の境界条件としては、バウンズバック、鏡面反射など様々な

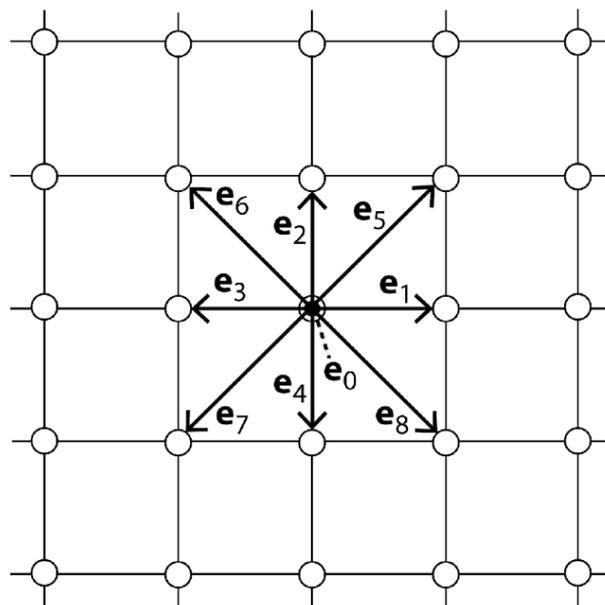


図 1 Rectangular 2D lattice illustrating the D2Q9 scheme for the LBM.

条件が挙げられるが、本研究では x 軸の流出側のみを流出境界、その他をディリクレ境界としている。流出境界とは計算領域内部から外部への流出を想定した境界条件である。また、ディリクレ境界とは境界上の流速を一意に与える境界条件であり、本研究では x 軸の正方向に速度 1, y 軸方向は速度 0 とする。

3. 符号付き距離関数による物体表現

CNN による物体周りの定常流の予測において、物体までの距離を表す符号付き距離関数 (Signed Distance Function, 以下 SDF) は予測精度の性能向上に有効であることが知られている [9]。SDF を用いた計算に際して、まず必要となるのがレベルセット関数の計算結果である。 (i, j) を領域上の任意の点とすると、 $\Omega \subset R^2$ において、幾何学的形状 Z を以下のように定義する。

$$Z = \{(i, j) \in R^2 : f(i, j) = 0\} \quad (4)$$

ここで、 f はレベルセット関数であり、 (i, j) が境界上の場合 $f(i, j) = 0$, (i, j) が境界の内部の場合 $f(i, j) < 0$, (i, j) が境界の外部の場合 $f(i, j) > 0$ となる。方向成分を有する距離関数 $D(i, j)$ は、レベルセット関数 $f(i, j)$ を用いて以下のように定義される。

$$D(i, j) = \min_{(i', j') \in Z} |(i, j) - (i', j')| \text{sign}(f(i, j)) \quad (5)$$

ここで、 $D(i, j)$ は方向成分を有する距離関数であり、幾何学的形状 Z の境界から点 (i, j) までの最短距離を測定する。図 2 に符号付き距離関数の例を示す。

4. 畳み込みニューラルネットワーク

畳み込みニューラルネットワーク (Convolutional Neural

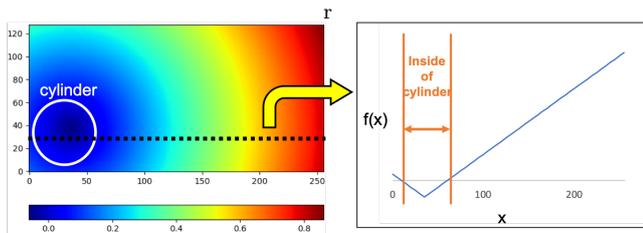


図 2 SDF representation of a circle shape (zero level set) in a 256x128 (Left). The graph of cross-section $y = 30$ (Right).

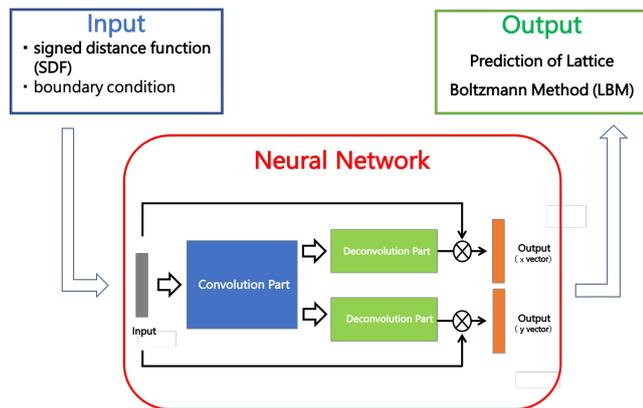


図 3 Steady flow prediction using convolutional neural network.

Networks, 以下 CNN) は畳み込み層を含んだニューラルネットワークであり, 幾何学的表現や画像認識分野での有効性が認められている. 例としては, FlowNet[3] は一対の画像からオプティカルフローを予測することを可能にし, Eigen, Puhersch, および Fergus ら [4] は, 単一の画像からその深度を推定することを可能にした. 本研究では CNN を用いたモデルを作成し, それを用いて定常流のシミュレーション結果の予測を行っている.

ニューラルネットワークによる学習を行う際には, 予め学習の挙動を制御するハイパーパラメータを設定する. ハイパーパラメータの設定を誤ると訓練データへの過剰な適合を起こしたり, 学習によるモデルの改善が停滞するといった問題が起こるため, ハイパーパラメータには適切な値を選定する必要がある. 本研究ではハイパーパラメータの最適化手法として TPE (Tree-structured Parzen Estimator) [1] を用いる.

5. CNN を用いた LBM の計算結果の予測

本研究では, 特定のサイズの領域を流れる定常流のシミュレーション結果の予測モデルを作成し, このモデルと領域の境界交換を併用することで, より広範囲のサイズの予測を可能にする. ここでは, まず CNN を用いた特定のサイズの領域の予測モデルについて説明する.

5.1 概要

本手法では, CNN で柱体まわりの LBM による定常流の

計算結果を予測する. 計算領域を 256×128 とし, 任意の場所に柱体をおく, 流体は x 方向負から正に向けて流れ, レイノルズ数は 20 とする. 図 3 に本手法の概要を示す.

5.2 ネットワーク

図 4 に本手法で用いたニューラルネットワークの構造を示す. 本手法ではニューラルネットワークの入力として, SDF による物体形状と境界における流速を既知として境界条件を与え, 出力として計算領域全域の LBM の計算結果の予測を得る. LBM の計算結果のうち, 定常流の速度を予測し, x, y 方向を別に出力させている. ネットワークのうち全結合より前の前半部分では x, y 方向で共通とし, 後半部分ではそれぞれの方向ごとにネットワーク構造を持つ. また, 入力の SDF から各格子点における物体の有無を取得し, これを CNN による予測に Mask として適用することで, 物体が存在しない流体領域の速度予測結果のみを最終的な予測結果としている. 既知とする境界部の幅は任意に指定でき, 本研究では 1 に設定した.

本手法で用いたネットワークは Guo, Li, および Iorio らの研究 [9] にて提案されたものを参考としている. 先行研究では入力に SDF のみを用いているのに対し, このネットワークを応用して複数領域にまたがる定常流を予測するため, 本研究では, SDF に加えて計算領域の境界条件を入力として用いている.

学習用フレームワークに PyTorch を, ハイパーパラメータ最適化フレームワークに Optuna を用い, 活性化関数は ReLU, 初期化は Xavier, 最適化は Adam を用いる.

5.3 データセット

本手法で用いたニューラルネットワークのデータセットについて説明する.

まず 1024×1024 の領域に 1 つまたは 2 つの柱体を配置し, LBM により左方から右方へと流体 (レイノルズ数 20) が流れる計算を行う. 配置する柱体の個数, 種類, 大きさ, 位置を変えながら, 複数の定常流計算を行う. 柱体の種類は角柱 (角数: 3-7), 円柱の計 6 種で, 種類ごとに大きさが異なる 32 個を用いる. 領域への柱体の配置は, 1 種の柱体を 1 つ配置したパターンと, ランダムに選択した 2 種の柱体を 1 つずつ配置するパターンを用意する. 2 つの柱体を配置することで柱体同士が流れに影響を与える場合の計算結果がデータセットに含まれるようにしている.

次にそれぞれの 1024×1024 の領域の LBM の計算結果から, 21 個の 256×128 の領域を切り出す. このとき, 切り出す領域はお互いに重ならないようにする. 入力として, SDF を用いるため, 切り出された 256×128 の領域に対応する SDF のデータも合わせて用意する. こうして, 256×128 の領域の LBM 計算結果と SDF の組みを学習用として 3528, 評価用として 1176 の合計 4704 用意する.

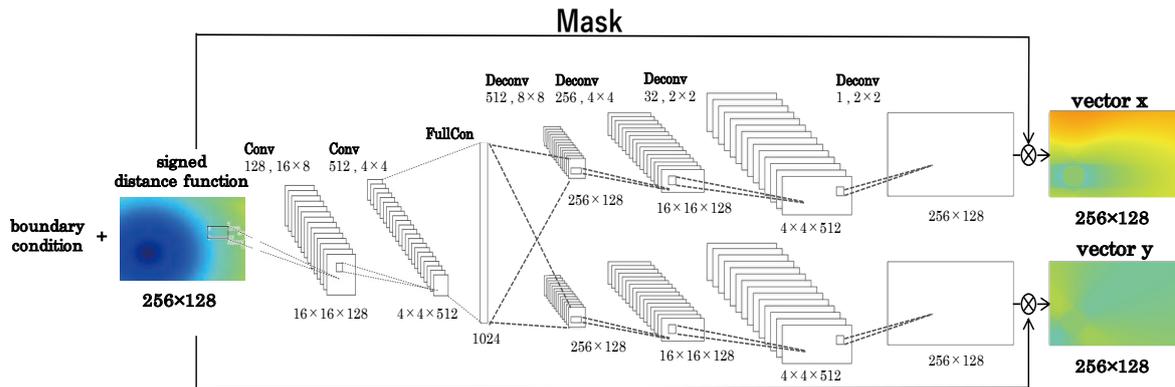


図 4 Network architecture for 2D geometry.

5.4 結果

学習は東京大学情報基盤センターに設置されたスーパーコンピュータ Reedbush-L を用いた。Reedbush-L は 1 ノードあたり NVIDIA Tesla P100 GPU を 4 基搭載しており、本学習は 4 ノード 16GPU を用いた。200 エポックの学習に対し、ハイパーパラメータの探索を 60 回行った結果、計算時間は 119.3 分を要した

5.4.1 予測結果

図 5 により、提案手法による予測が高精度で行われていることが分かる。Reedbush に搭載された 1GPU (Tesla P100) を用いると、LBM の計算時間は定常状態に達するまでに 70.6 秒 (124000step) を要した。一方で、CNN を用いた提案手法による予測には 1 GPU で 0.6 秒を要したため、提案手法により計算時間を 99%短縮することが可能である。図 5 に学習済みモデルを用いた予測の結果を示す。

5.5 複数 GPU によるニューラルネットワークの学習の並列化

本研究では CNN のモデルの作成に伴い、計算時間の短縮のため複数 GPU による並列計算を行った。ここでは並列化に用いた GPU 数の変化に伴う学習時間の変化について述べる。

図 6 に、並列化に用いた GPU 数と、提案手法におけるモデルの学習が基準となる Loss の値に達するまでにかかった時間との関係性を示す。なお、測定には東京大学情報基盤センターに設置されたスーパーコンピュータ Reedbush を用い、Loss の基準値として 0.00001 を定めた。

図 6 により、並列化に用いた GPU 数が増えるほど学習の収束にかかる時間は短縮され、Loss が変動しやすくなることが確認できる。

図 7 に、並列化に用いた GPU 数と、1 エポックの学習にかかる時間の関係性を示す。図 7 により、GPU 数が同じであればノード数が異なってもほぼ同程度の速度で計算が可能であることが確認できる。

6. 境界交換を用いた LBM の予測範囲の拡大

CNN を用いた LBM の計算結果の予測手法は、学習に用いたデータセットの領域サイズでの計算結果しか予測することができない。そこで、本研究では、これを発展させ、特定の領域サイズのデータセットにより学習されたモデルを、その領域サイズよりも広範囲の領域の計算結果の予測に用いる手法を提案する。

6.1 予測方法

本手法では、まず、予測領域を 256×128 の小領域に分割し、それぞれの小領域を 5 の手法を用いて予測する。ただし、隣接する小領域は境界部が重複している。次に、重複する境界部分の予測結果を隣接する領域で共有したうえで、これを境界部分の既知の入力として用いることで再度予測を行う。この「予測と共有」という操作を繰り返すことで、やがて小領域の境界部が連続となり、予測領域全域の予測結果が得られる。本手法は、予測領域内部の物体がいかなる形状であっても適用でき、物体の有無にも依存しない。また、小領域を予測するモデルの学習に境界を分断する物体のパターンも用いているため、小領域の境界部に物体が存在し、境界が分断される状況にも適用可能である。

図 8 に 256×344 に提案手法を用いた際の領域の分割方法を示す。図 8 にあるように、本手法では予測対象の領域を、複数の 256×128 の小領域に分割する。以降、図 8 にしたがって各小領域を Part1-3 と呼称する。

図 9 に本手法における LBM 計算結果の予測手順を示す。図 9 の Step1 にあるように、まず隣接していない小領域 Part1, 3 の予測を行う。次に Step2 の通り Part2 の予測を行う。この時、Part1, 3 で出力された予測結果の中で Part2 の境界部分にあたる部分の予測結果を Part2 の入力 (境界条件) として用いる。Part2 の予測結果の出力を確認した後、今度は、Part2 の予測結果の中で Part1, 3 の境界部分にあたる部分の予測結果を Part1, 3 の入力 (境界条

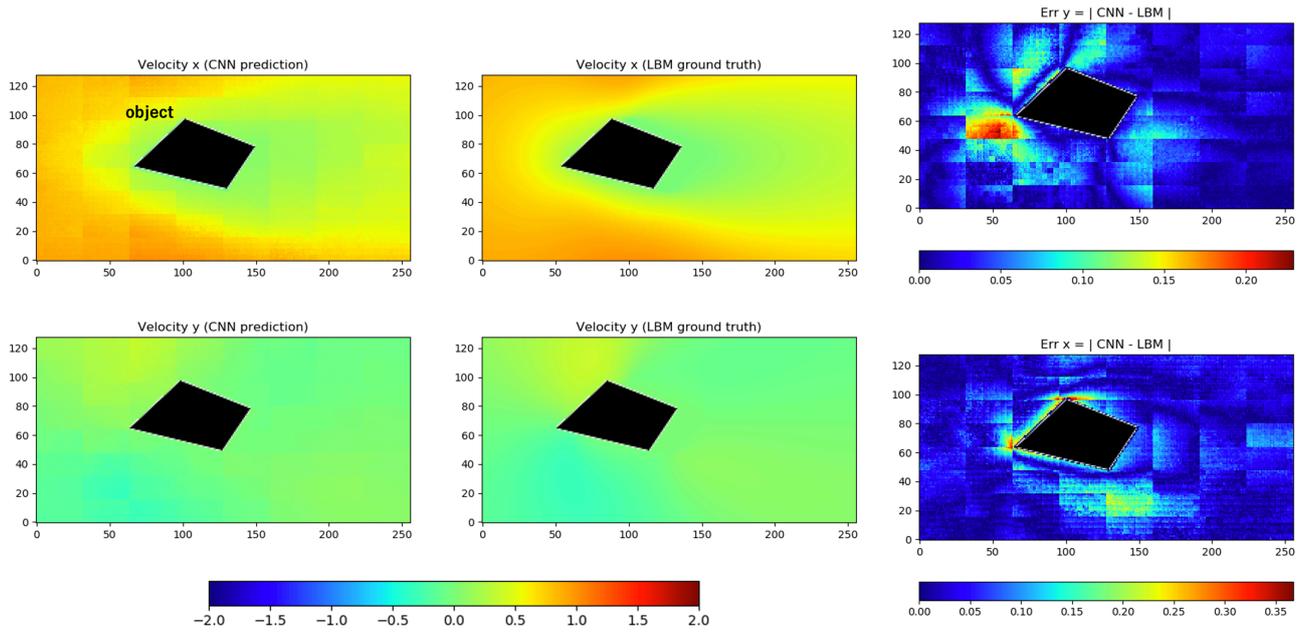


図 5 Prediction results of a single area. The first row is the velocity x results, and the second row is the velocity y results. The first column is the CNN Predictions, the second column is the LBM ground truths, and the third column is the error between the ground truth and the prediction.

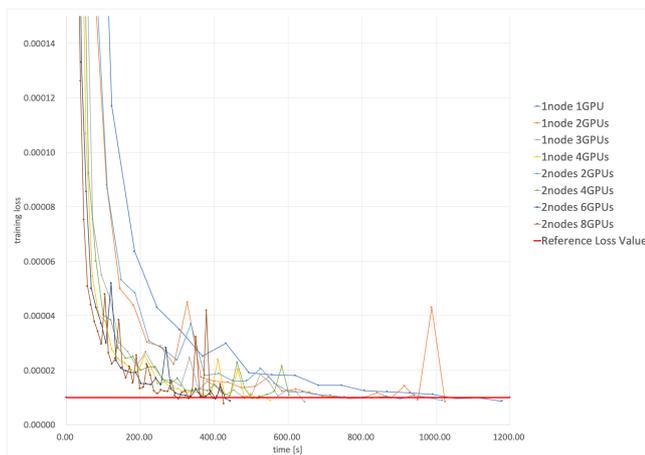


図 6 Convergence time of Loss as the number of GPUs changes. Red line means reference Loss value.

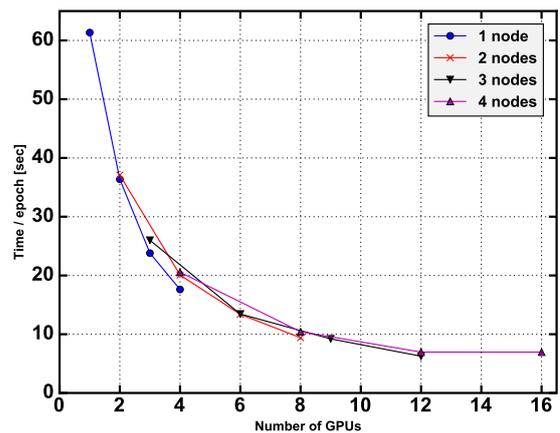


図 7 Weak scaling results in terms of computational time/epoch on Reedbush-L.

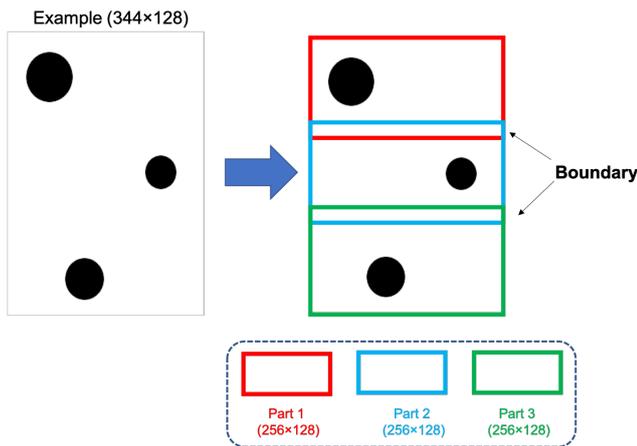


図 8 The division of the extended area used in the proposed method (344x128).

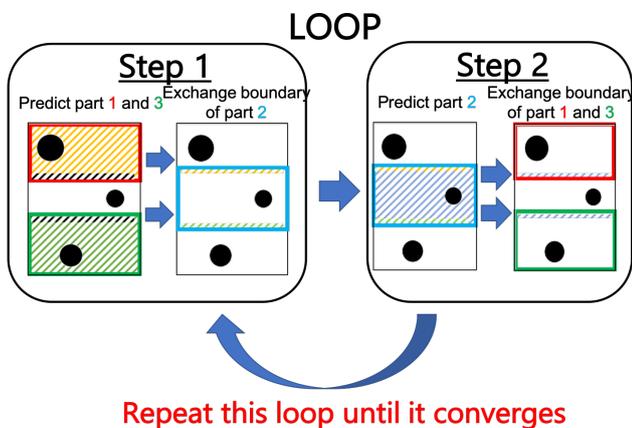


図 9 Small area convergence method with boundary exchange used in the proposed method.

件)として用いて Part1, 3 の予測を行う。これを 1 ループとし、以後このループをを隣接する小領域の境界が収束する、すなわち元々独立であった小領域の予測結果が連続になるまで繰り返すことで、複数の小領域で構成された予測領域全域の予測を実現する。なお、予測領域全域の境界部は既知としている。

本項では説明を簡潔にすべく領域を y 軸方向に 3 分割した例を用いたが、分割数は x 軸方向、 y 軸方向共に任意に指定することが可能である。 x 、 y 軸両方向に分割を行った際には、本項で述べた y 軸方向での分割同様、 x 、 y 軸方向奇数番目の小領域を先に予測し、次に偶数番目を予測する。そして隣接する領域同士の境界部の共有とそれを用いた再予測のループをを繰り返すことで、予想領域全域の予測を行う。

6.2 結果

図 10 に、 1021×1017 の領域に物体を置き、レイノルズ数 20 の流体を x 軸正方向に流した際に、 x 、 y 方向の流速を本手法を用いて予測した結果、流速の正解 (LBM)、お

よび予測と正解との誤差を示す。予測結果は 0 ループの結果と最終ループ (今回は 149) の結果を示している。また境界幅は 1 とし、 x 軸方向に 4、 y 軸方向に 8 の計 32 の小領域で分割して予測を行った。

図 10 より、 x 、 y 方向のいずれの流速の予測についても、初めのループでは大きく正解と異なるが、境界交換によって正解に近づくことが確認できる。また、最初と最後のループを比較すると誤差は減っているが、物体近辺と領域の継ぎ目が比較的誤差が大きいが確認できる。

7. まとめと今後の課題

本研究では定常流のシミュレーションの高速化を目的とし、CNN を用いた LBM の結果の予測を行った。その結果、図 5 から明らかであるように高精度の予測を実現し、計算時間を 99 % 短縮することを可能にした。次に、先に提案した手法の予測領域を境界部の交換により拡大することを提案し、境界部が収束することを確認した。今後の展望としては、後者の提案手法を用いた際に誤差が大きかった物体近辺と領域の継ぎ目の精度向上や、三次元領域への拡張が望まれる。

謝辞 本研究の一部は科学研究費補助金・基盤研究 (S) 課題番号 19H05662 「(計算+データ+学習) 融合によるエクサスケール時代の革新的シミュレーション手法」、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] Bergstra, J. S., Bardenet, R., Bengio, Y. and Kégl, B.: Algorithms for hyper-parameter optimization, *Advances in neural information processing systems*, pp. 2546–2554 (2011).
- [2] Chopard, B. and Droz, M.: Cellular Automata Modeling of Physical Systems, *Cambridge University Press* (1998).
- [3] Daberkow, D. and N., D.: New Approaches to Conceptual and Preliminary Aircraft Design: A Comparative Assessment of a Neural Network Formulation and a Response Surface Methodology. (1998).
- [4] H. Fang, M. Rais-Rohani, Z. L. and Horstemeyer., M.: A comparative study of metamodeling methods for multi-objective crashworthiness optimization., *In Advances in neural information processing systems*, (2014).
- [5] Qian, Y. H., S. S. and Orszag, S.: “Recent Advances in Lattice Boltzmann Computing, *Annual Reviews of Computational Physics. III, D. Stauffer ed. World Scientific*, pp. pp.195–242 (1995).
- [6] Rothman, D. H. and S., Z.: Lattice-Gas Cellular Automata, Cambridge U.P. (1997). (1997).
- [7] S., C., “LatticeDoolen G.D. : Lattice Boltzmann method for fluid flows, *Annual Reviews of Fluid Mechanics*, pp. pp. 329–364 (1998).
- [8] Wolf-Gladrow, D.: Lattice-Gas Cellular Automata and Lattice Boltzmann Models, *Lecture Notes in Mathematics, Springer* (2000).

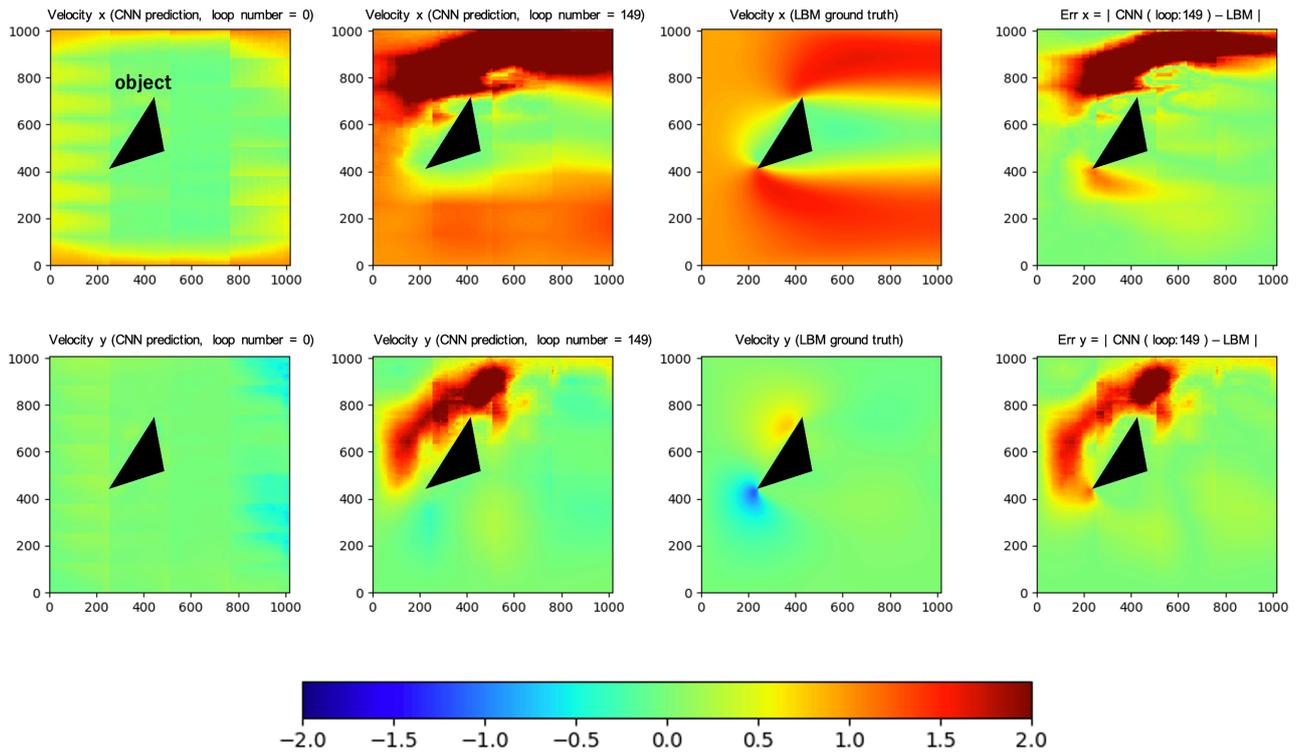


図 10 Prediction results of the extended area. The first row is velocity x results, and the second row is velocity y results. The first column from the left is the results of the proposed method (loop number = 0, first loop), the second column is the results of the proposed method (loop number = 0, final loop), the third column is the LBM ground truth, and the fourth column is the error between the ground truth and the prediction.

- [9] Xiaoxiao Guo, Wei Li, F. I.: Convolutional Neural Networks for Steady Flow Approximation Xiaoxiao (2016).
- [10] 葛原道久, 高田尚樹, 片岡武, : 格子気体法, 格子ボルツマン法. コロナ社 (1999).