The 33rd Annual Conference of the Japanese Society for Artificial Intelligence, 2019

勾配ブースティング回帰木を用いた製造業流体シミュレーションの高速化手法

Application of Gradient Booting regression toward the Computational Fluid Dynamics in the Manufacturing industry

小川　雄太郎[*1]　Yutaro Ogawa
清水　琢也[*1]　Takuya Shimizu
横井　俊昭[*1]　Toshiaki Yokoi

[*1] 株式会社電通国際情報サービス
INFORMATION SERVICES INTERNATIONAL-DENTSU, LTD.

Abstract: A faster calculation of MPS (Moving Particle Semi-implicit) method which is a computational fluid dynamics in the Manufacturing industry is proposed. Proposed method surrogates the semi-implicit part of MPS by the Gradient boosting regression trees using 10 original features as inputs. Finally, we confirmed that the qualitative properties of the proposed method coincide with the conventional MPS method by simulations of the dam-break problem.

1. はじめに

CAE (Computer Aided Engineering) とは製造業における製品および製品部品の設計時に, コンピュータ上で設計対象の to 特性や挙動をシミュレーション (数値計算) し, 最適な設計条件を明らかにする技術である. CAE の例としては, 製品部品の剛性を数値計算し軽くて壊れない部品設計を実現する構造最適化, 製品内を流れる流体の抵抗や翼部品が持つ揚力などを所望の性能が得られる設計するための流体シミュレーションなどが挙げられる.

これら CAE 技術が共通して抱える課題のひとつはシミュレーション時間である. 偏微分方程式を数値計算するため, 高精度にシミュレーションするためには時間方向と空間方向に分解能が必要となり, 計算時間が増大する. CAE を用いて新たなデザインを生み出すためにはこのシミュレーションにかかる時間を短縮し, 様々な設計条件を容易に試せる計算環境を生み出すことが重要となる.

本研究では CAE 技術のなかでも流体シミュレーションに着目する. 近年, 流体シミュレーションの高速化という課題に向けて機械学習を適用する研究が進みつつある. 具体的には従来のナビエ・ストークス方程式に従う物理モデルを, 機械学習で事前に学習させた推論モデルに代替することで CAE の高速化を目指すものである.

例えば Guo らは, 物体の周りを流れる空気の定常状態の流速場を求める際に, ナビエ・ストークス方程式を用いた数値計算を行うのではなく, 物体形状を CNN (Convolutional Neural Network) に入力し, 物体を囲む周辺領域の各座標点での流速を回帰するセマンティックセグメンテーションを応用した手法を提案している [GUO 2016]. また同様のスキームにより, 翼の周りの流速から翼の揚力を求める手法も提案されている [ZHANG 2018]. これらの先行研究は時間発展を伴う数値計算を省略し, 製品が持つ定常状態での特徴を, 物体形状を機械学習モデルに入力して 1 回の推論計算で求める.

また, 時間発展を伴う非定常な状態での数値時間を機械学習で代替する事例として, Fukami らは乱流の時系列変化の計算を機械学習で代替する手法を提案している[FUKAMI 2018].

連絡先：小川雄太郎, 株式会社電通国際情報サービス　技術本部開発技術部, 東京都港区港南 2-17-1, 03-6713-607, ogawa.yutaro@isid.co.jp

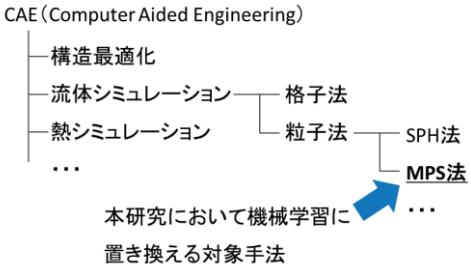

図 1. 本研究で機械学習モデルに置き換える MPS 法の CAE および各種流体シミュレーション手法における位置づけ

Fukami らの手法は計算対象領域内の各座標点における物理量を CNN エンコーダーで圧縮し, 圧縮した特徴量を LSTM に入力して時系列発展したdt後の特徴量を求め, 求めた特徴量を CNN デコーダーで復元してdt後の物理量を計算する.

しかしながら, これまで機械学習を用いた流体シミュレーション高速化手法は種々提案されているが, 製造業における重要な流体シミュレーション手法のひとつである「MPS 法 (Moving Particle Semi-implicit Method)」について, 機械学習による高速化が実現されていないという課題が存在する.

ここで, 本研究で着目する MPS 法について図 1 を参照し解説する. 流体シミュレーションはナビエ・ストークス方程式に基づき物理現象を数値計算するが, 大きく「格子法」と「粒子法」に計算手法が分類される. 格子法とは対象領域を格子状にメッシュで分割し, その格子点上の物理量を数値計算する手法である. いわゆる差分法や有限要素法であり, 乱流の再現などが得意である. 一方で, 粒子法は流体を仮想粒子の集まりとして扱い, 各粒子の時間発展に伴う移動を計算する手法である. この粒子は仮想的な流体の塊を想定しており, 流体の分子とは異なる. 粒子法は対象流体に対して計算用のメッシュを作る必要がないので, 大変形や自由表面を伴う系, 剛体と流体の連成解析系などのシミュレーションが得意である. 例えば自動車エンジンでは, 駆動時にピストンが上下運動しており, そのピストン内の燃焼ガスを閉じ込める目的とピストンを冷却する目的でオイルを噴出させている. このオイルがどのように挙動し, 所望の閉じ込め・冷却動作をしているかをシミュレーションする際などに粒子法が使用される [越塚 2014].

本研究で着目する MPS 法は粒子法の一種である. 粒子法の計算アルゴリズムには MPS 法と SPH 法 (Smoothed Particle Hydrodynamics Method) が存在する. SPH 法は銀河系の衝突など宇宙物理学におけるシミュレーションのために提案された手法であり [LUCY 1977], 近年では流体シミュレーションにも応用されている. ただし元来圧縮性流体（速度が早い, マッハ数 0.3 以上）を取り扱うために提案された手法であったため, 非圧縮性流体（流速がマッハ数 0.3 以下）を想定していない. そこで非圧縮性流体を粒子法として計算するために Koshizuka らによって MPS 法が提案された [KOSHIZUKA 1995] [越塚 2014].

MPS 法と SPH 法は流体の非圧縮性以外にも SPH 法が陽解法（fully explicit）で計算を行うのに対して, MPS 法は半陰的解法（semi-implicit）で計算するという違いがある. MPS 法ではナビエ・ストークス方程式の粘性項, 重力・外力項は陽解法で, 圧力勾配項は陰解法で計算する.

MPS 法は非圧縮性流体を高精度に粒子法でシミュレーションすることができ, 製造業の製品設計時に広く活用されている. しかしながら MPS 法の計算時間を短縮するための機械学習手法の提案はこれまで行われていない. SPH 法においては, ランダムフォレスト回帰木を用いた機械学習手法が提案されているが [JEONG 2015], Jeong らの手法はただちに MPS 法に流用できるものではない.

そこで本研究では製造業の流体シミュレーションで重要な MPS 法を機械学習で高速化することを目的に研究を進める. 以下, MPS 法の計算手法の概要を説明し, その後提案する機械学習手法を解説する. 最後に提案手法の妥当性を流体シミュレーションのトイプロブレムであるダム崩壊問題で検証する.

2. MPS 法

2.1 MPS 法の概要

図 2 に MPS 法の計算アルゴリズムの概要を示す. 本稿では紙面の都合上概要の解説となるため, 詳細については [越塚 2014] などを参照されたい.

はじめにナビエ・ストークス方程式の粘性項, 重力・外力項を, 計算ステップkでの仮想粒子（以下, 粒子と呼ぶ）の速度$\boldsymbol{u}_i^k$, および粒子の位置$\boldsymbol{r}_i^k$を用いて計算する. 添え字iは粒子番号を示す. 粘性項, 重力・外力項は陽解法で解き, 粒子の時刻dt後の仮速度$\boldsymbol{u}_i^*$と仮位置$\boldsymbol{r}_i^*$を計算する (①, ②). MPS 法において粘性項$(\nabla^2\boldsymbol{u}_i)^k$の計算は, 対象とする粒子$i$の近傍粒子の位置や速度に対してその相対距離に応じて重みを与える重み関数

$$w(\boldsymbol{r}_j-\boldsymbol{r}_i)=\begin{cases}\left(\dfrac{r_e}{|\boldsymbol{r}_j-\boldsymbol{r}_i|}\right)-1 & (|\boldsymbol{r}_j-\boldsymbol{r}_i|<r_e)\\ 0 & (|\boldsymbol{r}_j-\boldsymbol{r}_i|\geq r_e)\end{cases}$$

を使用して

$$(\nabla^2\boldsymbol{u}_i)^k=\frac{2d}{\lambda^0 n^0}\sum_{j\neq i}(\boldsymbol{u}_j^k-\boldsymbol{u}_i^k)w(\boldsymbol{r}_j^k-\boldsymbol{r}_i^k)$$

と計算される. ここでr_eは粒子の近傍とみなす影響半径, νは動粘性係数, dは対象としている計算領域の次元をそれぞれ示す. λ^0は初期状態での 1 粒子の影響範囲内にある近傍粒子との距離の 2 乗の重み平均値を示し, 次式で計算される.

$$\lambda^0=\frac{\sum_{j\neq i}|\boldsymbol{r}_j^0-\boldsymbol{r}_i^0|^2 w(\boldsymbol{r}_j^0-\boldsymbol{r}_i^0)}{\sum_{j\neq i}w(\boldsymbol{r}_j^0-\boldsymbol{r}_i^0)}$$

n^0は粒子数密度の基準値であり, 以下の通りに計算される.

$$n^0=\sum_{j\neq i}w(\boldsymbol{r}_j^0-\boldsymbol{r}_i^0)$$

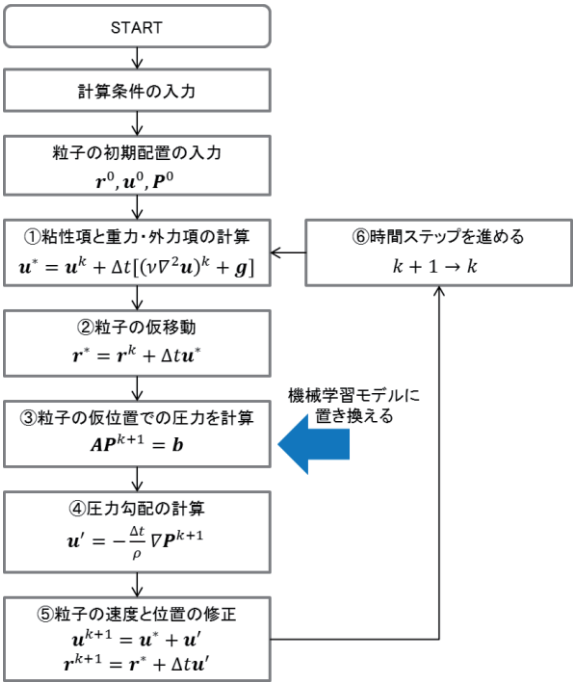

図 2 MPS 法概要および本研究で機械学習に置き換えるパート

次に粒子の仮位置において圧力$\boldsymbol{P}^{k+1}$を, 流体の非圧縮性条件から導かれる陰解法形式の式 $\boldsymbol{A}\boldsymbol{P}^{k+1}=\boldsymbol{b}$ を解いて求める (③). $\boldsymbol{A}, \boldsymbol{b}$ の各要素は粒子の仮位置から代数演算で求まり,

$$a_{ij}=\begin{cases}-\dfrac{1}{\rho^0}\dfrac{2d}{\lambda^0 n^0}w(\boldsymbol{r}_j^*-\boldsymbol{r}_i^*) & (j\neq i)\\ \dfrac{1}{\rho^0}\dfrac{2d}{\lambda^0 n^0}\displaystyle\sum_{j\neq i}w(\boldsymbol{r}_j^*-\boldsymbol{r}_i^*) & (j=i)\end{cases}$$

$$b_i=\frac{1}{\Delta t^2}\frac{n_i^*-n_0}{n_0}$$

として計算される. ここでρ^0は流体の密度を示し, n_i^*は$n_i^*=\sum_{j\neq i}w(\boldsymbol{r}_j^*-\boldsymbol{r}_i^*)$ で計算される粒子の仮位置での粒子密度を示す. この圧力の計算式は陰解法形式であり, 陽的に計算することができない. そのためガウスの消去法や共役勾配法など, 連立一次方程式の任意の解法で解く. なお, 粒子の周囲に粒子が密に存在しない液面や飛沫の粒子の圧力P_i^{k+1}は 0 とする.

求めた圧力$\boldsymbol{P}^{k+1}$を使用してナビエ・ストークス方程式の圧力勾配項を以下のように計算する.

$$\nabla P_i^{k+1}=\frac{d}{n^0}\sum_{j\neq i}\frac{P_j^{k+1}-\hat{P}_i^{k+1}}{|\boldsymbol{r}_j^*-\boldsymbol{r}_i^*|^2}(\boldsymbol{r}_j^*-\boldsymbol{r}_i^*)w(\boldsymbol{r}_j^*-\boldsymbol{r}_i^*)$$

ここで$\hat{P}_i^{k+1}$は粒子番号iの粒子の圧力を近傍粒子との圧力差に負圧が生まれないように補正したものである [越塚 2014]. この圧力勾配項による速度変化を計算し (④), 仮速度と仮位置を修正する (⑤). 以上の①から⑤の計算を繰り返すことで, 時系列発展を求める (⑥).

2.2 MPS 法の高速化における課題

MPS 法の計算の高速化において課題となるパートは, 粒子の仮位置での圧力の計算 (③) である. その他のパートは陽解法であり, さらに対象粒子の近傍の情報のみを利用するため, 解析領域を分割して計算機を並列させ高速化が可能である（ただし区画をまたがる粒子の移動情報はやりとりする）. しかしパート③は連立 1 次方程式を任意の解法で解いて求める陰解法の部分であり, 並列計算が困難である. そのため MPS 法の高速化の課題は圧力を求める陰解法部分にある.

3. 勾配ブースティング回帰木を用いた MPS 法の改良の提案

本研究では, MPS 法において圧力$\boldsymbol{P}^{k+1}$を求める陰解法部分を機械学習の回帰モデルに置き換え, 計算機が並列状態のままで圧力の計算を可能にし, 流体シミュレーションを高速化する手法を提案する. 提案手法では粒子ごとに特徴量を計算して回帰モデルに入力し, 各粒子の圧力P_i^{k+1}を出力する. 圧力計算以外は従来の MPS 法と同様である.

提案手法の回帰モデルに使用する特徴量について解説する. 提案手法は各粒子の影響半径内にある近傍粒子の仮位置の情報を使用して 10 個の特徴量を作成する.

特徴量 1 を影響半径r_e内に存在する粒子N_0個との距離とする. 特徴量 1 はN_0次元のベクトルであり, 各要素は昇順に配置する. 特徴量 1 のN_0個の要素F_{1j}^iは次式で定義される.

$$F_{1j}^i = \left|\boldsymbol{r}_j^* - \boldsymbol{r}_i^*\right| \qquad \left(\left|\boldsymbol{r}_j - \boldsymbol{r}_i\right| < r_e\right)$$

添字iは圧力を求める対象の粒子を, jはその近傍粒子を示す.

特徴量 2–5 は圧力の計算式で使用する行列$\boldsymbol{A}, \boldsymbol{b}$の要素に関連する特徴量であり, N_0次元のベクトルとする. 各特徴量のN_0個の要素は次式で定義される.

$$F_{2j}^i = -\frac{1}{a_{ii}\rho^0}\frac{2d}{\lambda^0 n^0} w\left(\boldsymbol{r}_j^* - \boldsymbol{r}_i^*\right)$$

$$F_{3j}^i = -\frac{1}{a_{ii}\rho^0}\frac{2d}{\lambda^0 n^0} w\left(\boldsymbol{r}_j^* - \boldsymbol{r}_i^*\right) n_j^*$$

$$F_{4j}^i = -\frac{1}{a_{ii}\rho^0}\frac{2d}{\lambda^0 n^0} w\left(\boldsymbol{r}_j^* - \boldsymbol{r}_i^*\right) n_j^{**}$$

$$F_{5j}^i = \frac{1}{a_{ii}\Delta t^2}\frac{n_j^* - n_0}{n_0}$$

ただしn_j^{**}は粒子番号jの粒子の仮位置$\boldsymbol{r}_j^*$での相対粒子密度であり, $n_j^{**} = \frac{n_j^* - n^0}{n^0}$ として計算する. 各要素の順番は特徴量 1 で昇順にした近傍粒子j の順番に合わせる.

特徴量 6, 7 はN_0次元のベクトルであり, 次式で定義される.

$$F_{6j}^i = n_j^*$$

$$F_{7j}^i = n_j^{**}$$

特徴量 8–10 はスカラーであり, 次式で定義される.

$$F_8^i = \frac{1}{a_{ii}\Delta t^2}\frac{n_i^* - n_0}{n_0}$$

$$F_9^i = n_i^*$$

$$F_{10}^i = a_{ii}$$

以上により, 特徴量ベクトルの要素数は合計で, $(7N_0 + 3)$個となる. なお各特徴量において, 影響半径r_e内に存在する粒がN_0個に満たない場合は, 値 0 を代替として特徴量の要素とする.

提案手法の枠組みにおいて回帰手法はとくに制限されないが, 本研究では勾配ブースティング回帰木を使用し, 勾配ブースティング回帰木の LightGBM [KE 2017] を使用した.

4. ダム崩壊問題における実験と考察

4.1 流体シミュレーションの条件

提案手法の妥当性を定性的に確認するために, ダム崩壊問題を対象に MPS 法および提案手法による流体シミュレーションを実施した. シミュレーション次元は $d = 2$, シミュレーション範囲を囲む縦横の壁の長さは1.0 [m] および0.6 [m], 水柱の大きさは縦横0.5 [m] および0.25 [m] の範囲に含む粒子と設定した. 水柱の位置は右端, 左端, 中央の 3 条件を用意した. 粒子の直径は$l_0 = 0.01$ [m], タイムステップは$dt = 0.001$ [s], シミュレーション時間は$T = 4$ [s] までとした. 粒子の影響半径r_eについては, 粘性項で粒子径l_0の 3.1 倍, その他の項で 2.1 倍とした.

粒子を仮位置に移動させる際に, 粒子径l_0 の 0.5 倍以下の距離まで近づいた場合には衝突と判定し, 反発率 0.2 の剛体衝突処理を行った. 流体は水を想定し, 密度 1000 [kg/m^3], 動粘性係数$\nu = 1.0 \times 10^{-6}$ とした. その他本稿 MPS 法の解説で省略したパラメータの設定は次の通りである. 自由表面粒子の判定基準には, 初期整列状態の密度n^0の 0.97 倍を使用した. 数値計算の安定性を保つために, 圧力の陰解法部分において用いられる緩和係数γは0.2 とし, 圧縮率は$\alpha = 0.45 \times 10^{-9}$とした.

4.2 機械学習モデルの学習および推論条件

提案手法の学習および推論の条件を記載する. 学習には水柱の位置が右端の条件での MPS 法のシミュレーション結果を使用した. 訓練データとして時刻 0 [s] から時刻 2 [s] までの全粒子の情報を 0.01 [s] 間隔で 200 スナップショット分使用した. 特徴量は$N_0 = 28$ とする 199 要素とした. 検証データは訓練データに対して 0.005 [s] だけ後ろにずらした時刻で, 同じく 200 スナップショット分の全粒子のデータを使用した. LightGBM の実装には [MS LightGBM] を使用した. ハイパーパラメータの設定は, ブースティングする回帰木の最大数 n_estimators を 30000 とし, その他はデフォルト設定のままとした. LightGBM の学習時に, 損失計算に二乗誤差関数を使用して検証データによる early stopping を適用した. LightGBM はブースティング手法であり, 設定した最大数まで回帰木を学習して訓練データに過学習するため, 検証データによる early stopping が必要となる.

提案手法の定性的な動作確認のために, 水柱の位置を左端および中央にした条件で流体シミュレーションを実施し, MPS 法による結果と提案手法の結果を比較した.

4.3 結果および考察

訓練データおよび検証データの行列サイズは$(238237,199)$, および(238214, 199) となった. 流体シミュレーション中に自由粒子と判定された粒子は圧力が 0 となり計算には使用されないため, 訓練データおよび検証行列の行数はその分減少する. LightGBM で学習させた結果, 回帰木の数は 867 本で early stopping された. 学習の結果, 圧力P_i^{k+1}の決定係数は訓練データが 0.70, 検証データが 0.52 であった. 図 3 に水柱の位置を左端および中央にしたシミュレーション結果を示す. それぞれにおいて, 上段が MPS 法による結果を, 下段が提案手法による結果を示す. 図 3 上側において時刻 0 [s] で左端にあった水柱が, 徐々に右方向へと崩壊し水が流れていく様子が時刻 0.3 [s] である. MPS 法と提案手法の挙動がほぼ一致していることが分かる. さらに時刻 0.9 [s] においては右端の壁にぶつかり水がはね返るが, こちらも挙動が一致していることが分かる.以降, 水の塊は左右の壁にぶつかることを繰り返し, 時刻 3.2 [s] での左端にぶつかったタイミングの挙動も一致している. また水柱が中央の場合も MPS 法と提案手法で挙動がほぼ一致している.

今回の実験において, 訓練データには水柱右端条件で時刻が 0.0 [s] から 2.0 [s] までのデータを使用したが, 時刻 2.0 [s] 以降かつ水柱の位置が違う条件でも定性的にうまく機械学習モデルが汎化されていることが明らかとなった. 以上により, 提案手法をダム崩壊問題でシミュレーションして, 定性的な挙動が従来の MPS 法の結果と一致していることを確認でき, MPS 法の陰解法による圧力計算パートを, 勾配ブースティング回帰木を用いた機械学習モデルに代替することが可能と確認された.

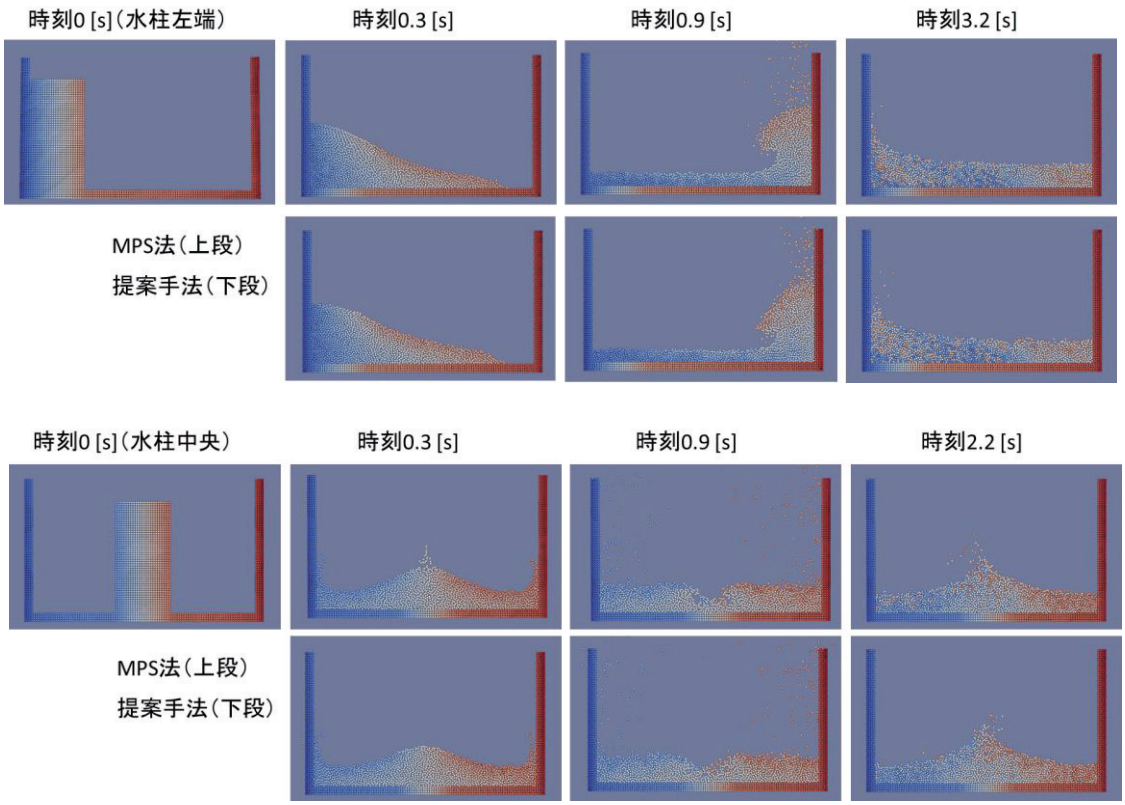

図 3 提案手法と従来手法の流体シミュレーション結果の定性的比較

本研究では MPS 法において，論理的には高速化に貢献する手法を提案した．今後，実際に高速化にどの程度寄与するのか，粒子数を変化させながらその定量的特徴を確認していく必要がある．MPS 法の陰解法パートの計算時間は共役勾配法を用いた場合，粒子数の 1.5 乗に比例する．一方で提案手法の推論時の計算時間は粒子数の 1.0 乗に比例するため，高精度に計算したい場合や大規模な計算など，粒子数が増加するに従い提案手法の高速化が貢献すると期待される．

また本手法の精度を向上させるためには，特徴量の吟味が必要である．現在の 10 個の特徴量のうち寄与度の高いもののみを残すことや，さらなる特徴量の改良が考えらえる．また LightGBM のハイパーパラメータをチューニングしていないため，パラメータチューニングによる精度向上も期待される．

本研究では定性的な挙動の確認として，流体シミュレーションのトイプロブレムであるダム崩壊問題を使用した．今後，実際に製造業のエンジン設計など，実ビジネスで使用する流体シミュレーションの条件と規模において，提案手法がどの程度定性的および定量的に従来手法と一致するのか確認が必要である．

また提案手法の運用方法の検討も実ビジネスへの導入に重要となる．例えば流体が水やガソリンなど異なる条件の場合に，機械学習モデルで違いを吸収できるようにするのか，流体の種類ごとに訓練データを用意するのか検討が必要である．さらに訓練データの作り方も検討する必要がある．なんらかのトイプロブレムの結果を使用するのか，それとも実際の実験条件において 1 度 MPS 法を実施して訓練データを作成するのかなど，実運用に向けた導入方法を検討する必要があると考えられる．

5. おわりに

本研究では製造業の流体シミュレーションで重要な MPS 法の一部を勾配ブースティング回帰木による機械学習を用いたアルゴリズムに代替し，流体シミュレーションを高速化する手法を提案した．提案手法の定性的挙動をダム崩壊問題で確認し，提案手法の結果が従来手法と同様の挙動を示すことを確認した．

参考文献

[GUO 2016] GUO, Xiaoxiao; LI, Wei; IORIO, Francesco. Convolutional neural networks for steady flow approximation. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2016. p. 481-490.

[ZHANG 2018] ZHANG, Yao; SUNG, Woong Je; MAVRIS, Dimitri N. Application of Convolutional Neural Network to Predict Airfoil Lift Coefficient. In: 2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference. 2018. p. 1903.

[FUKAMI 2018] FUKAMI, Kai; KAWAI, Ken; FUKAGATA, Koji. A synthetic turbulent inflow generator using machine learning. arXiv preprint arXiv:1806.08903, 2018.

[LUCY 1977] LUCY, Leon B. A numerical approach to the testing of the fission hypothesis. The astronomical journal, 1977, 82: 1013-1024.

[越塚 2014] 越塚誠一; 柴田和也; 室谷浩平. 粒子法入門. 2014.

[KOSHIZUKA 1995] KOSHIZUKA, Seichii. A particle method for incompressible viscous flow with fluid fragmentation. Comput. Fluid Dyn. J., 1995, 4: 29.

[JEONG 2015] JEONG, SoHyeon, et al. Data-driven fluid simulations using regression forests. ACM Transactions on Graphics (TOG), 2015, 34.6: 199.

[KE 2017] KE, Guolin, et al. Lightgbm: A highly efficient gradient boosting decision tree. In: Advances in Neural Information Processing Systems. 2017. p. 3146-3154.

[MS LightGBM] https://github.com/Microsoft/LightGBM