

## 2B06 Physics-informed neural networkを用いた圧縮性流体现象のモデル化

○水野裕介，三坂孝志，古川慈之（産業技術総合研究所）

Modeling compressible fluid phenomena using physics-informed neural networks  
Yusuke Mizuno, Takashi Misaka and Yoshiyuki Furukawa (AIST)

Key Words : Physics-informed neural network, Compressible Flow, Shock Wave

### Abstract

An oblique shock wave problem was modeled by using a physics-informed neural network (PINN) based on two-dimensional steady compressible Euler equations and an equation of state. The conditions are a uniform flow at Mach number 2.0 passing through the slanted wall with an angle of 10 degrees. The estimation results of the PINN model showed good agreement with the theoretical solution. The PINN model demonstrated high accuracy in capturing the discontinuities.

### 1. はじめに

圧縮性流体の現象は，航空宇宙，製造や医療など様々な分野で生じ，その現象の可視化や推定技術の確立が求められている．これまでに計算機技術の向上とともに数値流体解析 (CFD) が用いられてきた．CFDでは差分近似などを用いて物理現象を可視化できるが，そのモデルや計算条件によって得られる解が異なる場合がある．また近年では，機械学習を用いた推定技術も確立されつつある．機械学習の出力結果は，学習データの影響を踏まえた結果を得られるが，物理法則に従わない場合がある．

PINN (physics-informed neural network)<sup>(1)</sup>は物理法則モデルの方程式と初期・境界条件を損失関数として定義し，その制約が満たされるようにニューラルネットワーク (NN) で学習することで，方程式の解を推定する機械学習を用いたサロゲートモデルである．PyTorch<sup>(2)</sup>などのPythonライブラリの自動微分機能を用いることで，差分近似を用いずに現象を推定/可視化することができる．さらに学習データが無い場合でも，物理モデルに基づいた解を推定できる．圧縮性流体の現象をPINNでモデル化した研究<sup>(3)(5)</sup>では，衝撃波のような不連続な分布を推定できるモデルが提案されたが，事前にCFDや実測値などのデータが必要となる場合がある．

本研究では，圧縮性流体の現象をPINNモデルだけで，推定できる技術の確立を目的とする．本発表では，2次元斜め衝撃波問題を対象にPINNの推定結果を理論解と比較検討を行うことで，PINNモデルの精度検証を行う．

### 2. 解析モデル

損失関数の評価に用いる偏微分方程式 (PDE) は，2次元定常Euler方程式とする．また，理想気体の状態方程式も損失関数の評価に用いる．

$$\begin{aligned} F_1(x, y) &= \frac{\partial \rho u}{\partial x} + \frac{\partial \rho v}{\partial y} = 0 \\ F_2(x, y) &= \frac{\partial}{\partial x}(\rho u^2 + P) + \frac{\partial}{\partial y}(\rho uv) \\ &= 0 \\ F_3(x, y) &= \frac{\partial}{\partial x}(\rho uv) + \frac{\partial}{\partial y}(\rho v^2 + P) \\ &= 0 \\ F_4(x, y) &= \frac{\partial}{\partial x}((E + p)u) \\ &\quad + \frac{\partial}{\partial y}((E + p)v) = 0 \\ F_5(x, y) &= P - \rho RT = 0 \end{aligned} \quad (1)$$

ここで， $\rho$ ， $\mathbf{U}=(u, v)$ ， $P$ は，それぞれ密度，流速，圧力とする．また， $R$ は気体定数とする． $E$ は単一体積あたりの全エネルギーで，

$$E = \rho(C_v T + 1/2(u^2 + v^2)) \quad (2)$$

として， $C_v$ と $T$ は定積比熱と温度とする．

PINNモデルのNNの入力値を空間座標 $x$ と $y$ として，出力値を密度と流速，温度，圧力とする (図1)． $Loss_{PDE}$ は，計算領域にランダム配置される学習点数 $N$ において，式(1)のPDEと状態方程式から，

$$Loss_{PDE} = MSE_{PDE} = \sum_{i=1}^5 \frac{1}{N} \sum_{j=1}^N |F_i(x_j, y_j)|^2 \quad (3)$$

とする． $Loss_{BC}$ は，境界上の学習点数 $N_{BC}$ と各出力変数

$N_{VAR}$ （密度，流速，温度，圧力）において，

$$\begin{aligned} Loss_{BC} &= MSE_{BC} \\ &= \sum_{i=1}^{N_{VAR}} \frac{1}{N_{BC}} \sum_{j=1}^{N_{BC}} |f_{NN,i}(x_j, y_j) \\ &\quad - f_{BC,i}(x_j, y_j)|^2 \end{aligned} \quad (4)$$

とする．よって，全体の損失 $Loss$ は，

$$Loss = Loss_{PDE} + Loss_{BC} \quad (5)$$

で定義される．最適化手法と活性化関数はAdam optimizer<sup>(6)</sup>/L-BFGS-B optimizer<sup>(7)</sup>とTanhを用いた．解析コードは，Python(NumpyとPyTorch)言語で構築し，CUDAを用いてGPU上で計算を行った．

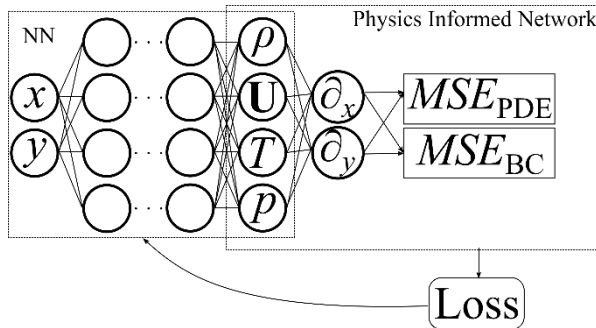


図1 PINNモデルのネットワーク構造

### 3. 解析条件

PINNでモデル化する流れ場は，斜度10度の壁面上をMach数2の一樣流が流れる斜め衝撃波問題とする（図2）． $x=0$ 上はDirichlet境界条件で一樣流を， $y=0.0$ 上の $x=0-0.5$ でSymmetric境界条件を， $x=0.5-1.5$ でSlip-wall境界条件を課した．学習点は計算空間に $N=16384$ 点を，境界上の学習点は $N_{BC}=250$ 点をランダムに配置した．学習は初めにAdam optimizerで行った後，L-BFGS-B optimizerで関数変化の許容誤差が $1 \times 10^{-9}$ 以下になるまで行った．

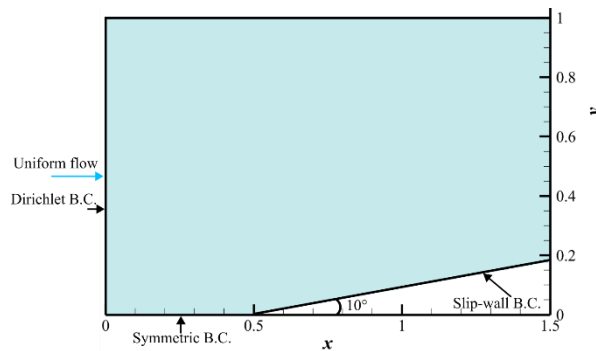


図2 流れ場と境界条件の概略図

### 4. 解析結果

図3にエポックに対するLossの履歴を示す．ここで，エポックは，オプティマイザーが実行するステップを意味する．Lossは初めに急激に減少した後，緩やかに減少する．L-BFGS-B optimizerに切り替わると再び大きく減少した後，許容誤差の条件を満たして，学習を終了した．

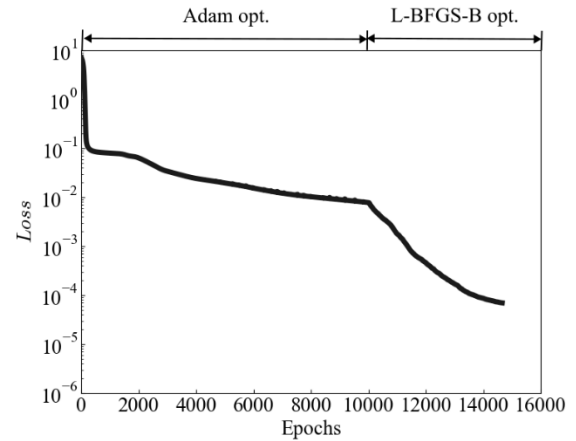


図3 学習履歴

学習し終えたPINNモデルより，等間隔直交格子（ $300 \times 200$ ）の点で出力変数の値を推定した．

図4に $x=1$ における $y$ 軸方向の密度分布を示す．ここで， $y < 0.09$ は壁面内部とする．赤色の実線で示す理論値に対して，黒色の丸で示すPINNのプロットは同じ分布を示す．特に，衝撃波のような不連続な分布を精度良く捉えた．

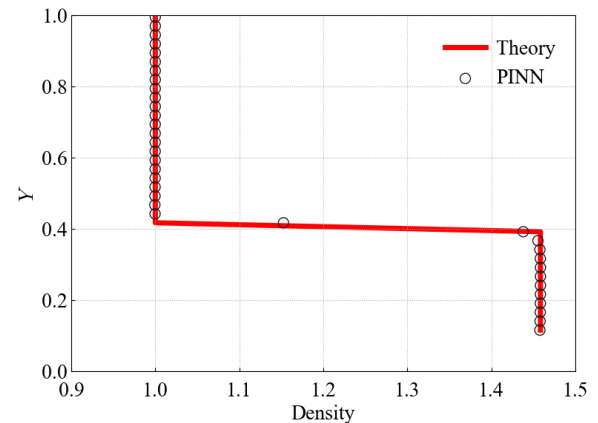


図4  $x=1$ における $y$ 軸方向の密度分布（赤線：理論解，黒丸：PINN）

PINNの推定結果と理論解の相対誤差を出力変数ごとに導出した（表1）．変数ごとに精度を検証した結果においても，精度良く提案したPINNモデルは推定できていることを示した．

表1 理論解との相対誤差

$\rho$	$u$	$v$	$T$	$P$
0.021	0.007	0.119	0.009	0.034

5. まとめ

PINNで斜め衝撃波の流れ場をモデル化し、理論解と比較を行った。PINNモデルで推定した結果は、理論解と同じ結果を示した。特に不連続な分布も精度良くとらえることができた。

参考文献

(1) Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. 378, 686–707 (2019).

(2) Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: PyTorch: An Imperative Style, High-

Performance Deep Learning Library.

(3) Mao, Z., Jagtap, A.D., Karniadakis, G.E.: Physics-informed neural networks for high-speed flows. Comput. Methods Appl. Mech. Eng. 360, 112789 (2020).

(4) Jagtap, A.D., Mao, Z., Adams, N., Karniadakis, G.E.: Physics-informed neural networks for inverse problems in supersonic flows. J. Comput. Phys. 466, 111402 (2022).

(5) Ferrer-Sánchez, A., Martín-Guerrero, J.D., de Austri-Bazan, R.R., Torres-Forné, A., Font, J.A.: Gradient-annihilated PINNs for solving Riemann problems: Application to relativistic hydrodynamics. Comput. Methods Appl. Mech. Eng. 424, 116906 (2024).

(6) Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization.

(7) Byrd, R.H., Lu, P., Nocedal, J., Zhu, C.: A Limited Memory Algorithm for Bound Constrained Optimization. SIAM J. Sci. Comput. 16, 1190–1208 (1995).