

高信頼度な PINN に向けた 境界条件の厳密な付与方法と動的重みの併用

Reliable PINN by Exact Boundary Condition Imposition
and Dynamic Weight Tuning Method

出口翔大¹⁾ 浅井光輝²⁾
Shota Deguchi and Mitsuteru Asai

¹⁾九州大学大学院 工学府土木工学専攻 日本学術振興会 特別研究員 DC2
(〒 819-0395 福岡県福岡市西区元岡 744 番地, E-mail: deguchi@doc.kyushu-u.ac.jp)

²⁾九州大学工学研究院 社会基盤部門 准教授
(〒 819-0395 福岡県福岡市西区元岡 744 番地, E-mail: asai@doc.kyushu-u.ac.jp)

We present an approach to improve the accuracy of the inverse analysis in PINN by imposing the boundary conditions exactly and by introducing the dynamic weight tuning method. In standard PINN formulations, the boundary conditions are considered by the loss function. In this study, we introduce the distance function to impose them exactly. In addition, we apply the dynamic weight tuning to improve the accuracy of the inverse analysis. The effectiveness of the proposed method is demonstrated by numerical experiments.

Key Words : physics-informed neural network, boundary condition imposition, inverse analysis

1. 緒言

Neural Network を用いて偏微分方程式を (数値的に) 解く取り組みは, 1990 年代から行われている [1,2,3]. 近年の計算機の高速度化・大容量化, および機械学習ライブラリの普及により, 機械学習モデルの応用例は様々な分野で展開されている. 一方, 機械学習モデルの発展に伴い, モデルに説明性を求める動向が高まっており [4,5,6], 大量のデータに基づくモデル (purely data-driven) に代わり, 事前知識を導入したモデル (prior knowledge-informed) の開発が盛んである [7]. 物理問題への適用においても同様の動きが見られ, 前述の手法に類似した形で既知の物理法則を学習に利用するモデルが複数提案されている (例えば, [8,9,10]). 特に, PINN (Physics-Informed Neural Network) [11] は順解析・逆解析の両者に適用可能という特徴を持ち, 希少データを活用する問題・分野などで応用されている [12,13].

著者らはこれまでに PINN を用いて非圧縮性流体の流れ場・パラメータを逆解析する技術を開発してきた [14, 15]. しかし, 従来の PINN では損失関数を介して境界条件の影響を導入していたため (soft (BC) imposition), これを厳密に満足することが困難であった. 対処として,
(1) soft imposition の影響を強める方法 [9,16,17,18]
(2) 複数の Neural Network を用意し, それぞれに境界条件と支配方程式を学習させる方法 [8,19,20]
(3) 境界までの距離を表現する関数を用意・利用して境界条件を課す方法 [21,22,23]

などが挙げられる. ただし, (1) と (2) の方法は, 境界条件の合致の精度が最適化手法の性能に依存する. また, これらは Neural Network による近似に留まるため, 境界条件の厳密な満足は保証されない. 最適化手法や Neural Network の性能に依存しない方法として, 本研

究では (3) の方法を採用する. これには, 境界までの距離を表現する距離関数を用意する必要があるが, 幅広い問題に対応するためには, 解析領域の形状に柔軟に対応可能な関数を定義する必要がある. そこで, R-function [24,25] に基づく距離関数を導入することで, 境界条件を厳密に満足した (hard (BC) imposition) PINN を構築する [23]. ただし, 逆解析に適用する際には, 観測データの情報を取り込むための損失項は必須である. 一方, 損失項が複数存在する場合, 各項に乗ずる重みを適切に設定する必要があるため, 著者らが提案する動的重み付け法 [18] を導入する.

2. PINN: Physics-Informed Neural Network

モデル問題として, 以下の Poisson 方程式を考える.

$$-\nabla \cdot (\kappa \nabla u) = f \quad \text{in } \Omega \quad (1)$$

$$u = g_D \quad \text{on } \Gamma_D \quad (2)$$

$$\mathbf{n} \cdot \nabla u = g_N \quad \text{on } \Gamma_N \quad (3)$$

ただし, $\Omega \subset \mathbb{R}^d$ は有界領域, $\Gamma (= \partial\Omega)$ は Ω の境界である. Γ_D は Dirichlet 境界, Γ_N は Neumann 境界であり, $\Gamma = \bar{\Gamma}_D \cup \bar{\Gamma}_N$, $\Gamma_D \cap \Gamma_N = \emptyset$, $\Gamma_D \neq \emptyset$ とする. また, \mathbf{n} は Γ 上の外向き単位法線ベクトルである. κ, f は, それぞれ, 既知の拡散係数 (定数とする), ソースである.

PINN は, まず問題に対する解を MLP (Multi-Layer Perceptron) により近似する [26,27]. 入力を $\mathbf{x} (\in \mathbb{R}^{f^{(0)}})$, 出力を $\hat{\mathbf{y}} (\in \mathbb{R}^{f^{(L)}})$ とする L 層の MLP において, 第 l ($= 1, 2, \dots, L$) 層での順伝播 $\mathbf{z}^{(l)} (\in \mathbb{R}^{f^{(l)}})$ は以下の通りである.

$$\mathbf{z}^{(l)} = \sigma^{(l)} (\mathbf{W}^{(l)} \mathbf{z}^{(l-1)} + \mathbf{b}^{(l)}) \quad (4)$$

ただし, $\mathbf{z}^{(0)} = \mathbf{x}$, $\mathbf{z}^{(L)} = \hat{\mathbf{y}}$ であり, $\sigma^{(l)}(\cdot)$ は活性化関数

と呼ばれる、要素ごとに作用する非線形写像である。ただし、 $\sigma^{(L)}(\cdot)$ は恒等写像とする。 $\mathbf{W}^{(l)} (\in \mathbb{R}^{f^{(l)} \times f^{(l-1)}})$, $\mathbf{b}^{(l)} (\in \mathbb{R}^{f^{(l)}})$ は第 l 層の重み、バイアスである。ここで、式 (1) の解 $u(\mathbf{x})$ を MLP $\hat{u}(\mathbf{x}; \boldsymbol{\theta})$ により近似し ($\boldsymbol{\theta} = \{\mathbf{W}^{(l)}, \mathbf{b}^{(l)}\}_{l=1}^L$)、以下の損失関数を定義する。

$$\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}_{\text{PDE}}(\boldsymbol{\theta}) + \lambda_{\text{BC}} \mathcal{L}_{\text{BC}}(\boldsymbol{\theta}) \quad (5)$$

ただし、 λ_{BC} は \mathcal{L}_{PDE} に対する \mathcal{L}_{BC} の相対的な重要度を示す重みである。 \mathcal{L}_{PDE} , \mathcal{L}_{BC} は以下の通りである。

$$\mathcal{L}_{\text{PDE}}(\boldsymbol{\theta}) = \frac{1}{N_{\text{PDE}}} \sum_{i=1}^{N_{\text{PDE}}} |-\nabla \cdot (\kappa \nabla \hat{u}(\mathbf{x}_i; \boldsymbol{\theta})) - f(\mathbf{x}_i)|^2 \quad (6)$$

$$\begin{aligned} \mathcal{L}_{\text{BC}}(\boldsymbol{\theta}) = & \frac{1}{N_{\text{BC},D}} \sum_{i=1}^{N_{\text{BC},D}} |\hat{u}(\mathbf{x}_i; \boldsymbol{\theta}) - g_D(\mathbf{x}_i)|^2 \\ & + \frac{1}{N_{\text{BC},N}} \sum_{i=1}^{N_{\text{BC},N}} |\mathbf{n}(\mathbf{x}_i) \cdot \nabla \hat{u}(\mathbf{x}_i; \boldsymbol{\theta}) - g_N(\mathbf{x}_i)|^2 \end{aligned} \quad (7)$$

なお、逆解析に適用する際には、観測データの情報を取り込むため、損失関数を以下のように改める。

$$\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}_{\text{PDE}}(\boldsymbol{\theta}, \hat{\kappa}) + \lambda_{\text{BC}} \mathcal{L}_{\text{BC}}(\boldsymbol{\theta}) + \lambda_{\text{Data}} \mathcal{L}_{\text{Data}}(\boldsymbol{\theta}) \quad (8)$$

ただし、 $\hat{\kappa}$ は κ の推定値、 λ_{Data} は $\mathcal{L}_{\text{Data}}$ の重要度を示す重みである。 $\mathcal{L}_{\text{Data}}$ は以下のように定義する。

$$\mathcal{L}_{\text{Data}}(\boldsymbol{\theta}) = \frac{1}{N_{\text{Data}}} \sum_{i=1}^{N_{\text{Data}}} |\hat{u}(\mathbf{x}_i; \boldsymbol{\theta}) - u(\mathbf{x}_i)|^2 \quad (9)$$

ただし、 $u(\mathbf{x}_i)$ は \mathbf{x}_i における観測データである。

3. 高精度化手法

(1) 境界条件の厳密な付与

式 (5) では、損失関数を介して境界条件の影響を考慮している。この方法は soft imposition と呼ばれ、境界条件を厳密に満足することは困難である。先行研究において、soft imposition を用いて十分な精度を得た例もあるものの [11,28]、問題によっては解の精度や収束性が著しく低下する場合もあることが知られている [21,23]。本研究では、R-function [24,25] に基づく距離関数を用いて、境界の表現、および境界条件の付与を行う [23]。

a) R-function による距離関数

本研究では、 Ω 上で定義される、 Γ までの距離を表現する関数として、その厳密な距離を示す関数を EDF (Exact Distance Function) と呼び Φ と記すこととする。また、その近似的な距離を示す関数を ADF (Approximate Distance Function) と呼び ϕ と記す。

まず、EDF の特性として、以下が挙げられる [25]。

(A) $\Phi(\mathbf{x}) = 0$ at \mathbf{x} on Γ

(B) $\partial_\nu \Phi(\mathbf{x}) = 1$ at \mathbf{x} on Γ

(C) $\partial_\nu^m \Phi(\mathbf{x}) = 0$ at \mathbf{x} on Γ for any $m \geq 2$

ただし、 $\partial_\nu(\cdot)$ は (\cdot) 方向の方向微分、 ν は Γ 上の内向き単位法線ベクトルとする ($\nu = -\mathbf{n}$)。ADF は EDF の近似であるため、可能な限り EDF と同様の特性を持つことが望ましい。具体的には、上記の特性 (A), (B) については ADF も同様に満足することを要求し、特性 (C) については幾つかの高階微分がゼロとなることを要求する。すなわち、ADF が備えるべき特性は以下である。

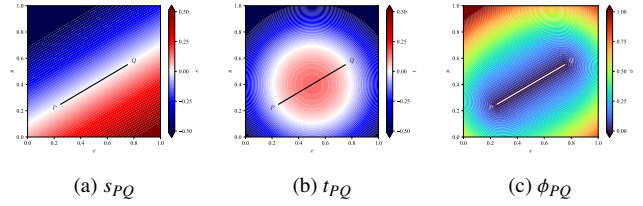


図-1: 線分 S_{PQ} に対する各関数

(a) $\phi(\mathbf{x}) = 0$ at \mathbf{x} on Γ

(b) $\partial_\nu \phi(\mathbf{x}) = 1$ at \mathbf{x} on Γ

(c) $\partial_\nu^m \phi(\mathbf{x}) = 0$ at \mathbf{x} on Γ for some $m \geq 2$

このとき、 ϕ は m 次まで正規化されている、という [24]。以降、これを強調するとき、 $\phi = \phi^{(m)}$ と書くこととする。

初めに、ある 1 つの線分に対する距離関数を考える。点 $P (= \mathbf{x}_P)$ と点 $Q (= \mathbf{x}_Q)$ により定義される線分 S_{PQ} に対し、以下の符号付き距離関数 s_{PQ} を定義する。

$$s_{PQ}(\mathbf{x}) = \frac{1}{\|\mathbf{x}_{PQ}\|_2} s'_{PQ}(\mathbf{x}) = \frac{1}{\|\mathbf{x}_{PQ}\|_2} (\mathbf{n}_{PQ} \cdot (\mathbf{x} - \mathbf{x}_P)) \quad (10)$$

ただし、 s'_{PQ} は点 P と点 Q により定義される直線 S'_{PQ} の陰関数表示、 $\|\mathbf{x}_{PQ}\|_2 (= \|\mathbf{x}_Q - \mathbf{x}_P\|_2)$ は S_{PQ} の長さ、 $\mathbf{n}_{PQ} (= \mathbf{x}_{PQ} / \|\mathbf{x}_{PQ}\|_2)$ は S_{PQ} の単位法線ベクトルである。また、以下のトリミング関数 t_{PQ} を定義する。

$$t_{PQ}(\mathbf{x}) = \frac{1}{\|\mathbf{x}_{PQ}\|_2} \left(\left(\frac{\|\mathbf{x}_{PQ}\|_2^2}{2} \right)^2 - \|\mathbf{x} - \mathbf{x}_C\|_2^2 \right) \quad (11)$$

ただし、 $\mathbf{x}_C (= (\mathbf{x}_P + \mathbf{x}_Q) / 2)$ は S_{PQ} の中点である。最後に、線分 S_{PQ} を、 $s_{PQ} = 0$ と $t_{PQ} \geq 0$ の積集合として解釈し [24,25]、任意の点 \mathbf{x} から S_{PQ} までの距離 (ADF) $\phi_{PQ} (= \phi_{PQ}(s_{PQ}, t_{PQ}))$ を以下のように定義する [29]。

$$\phi_{PQ}(\mathbf{x}) = \left(s_{PQ}^2 + \left(\frac{(s_{PQ}^4 + t_{PQ}^2)^{1/2} - t_{PQ}}{2} \right)^2 \right)^{1/2} \quad (12)$$

以上より、点 $P(0.25, 0.25)$ と点 $Q(0.75, 0.55)$ に対して定義される s_{PQ} , t_{PQ} , ϕ_{PQ} を図-1 に示す。同図 (c) より、線分 S_{PQ} 上で $\phi_{PQ} = 0$ 、かつ S_{PQ} 以外の点で $\phi_{PQ} > 0$ となっていることが確認できる。

次に、境界 Γ が複数の線分から構成される場合を考える ($\Gamma = \bigcup_{i=1} \Gamma_i$)。境界 Γ_i までの距離を示す ADF を ϕ_i とすると ($\phi_i = 0$ on Γ_i)、境界 Γ までの距離 ϕ は以下のように得られる [25]。

$$(\phi^{(m)}) = \phi = \frac{\prod_i \phi_i}{(\sum_i \phi_i^m)^{1/m}} = \frac{1}{(\sum_i \phi_i^{-m})^{1/m}} \quad (13)$$

ただし、 $m (\in \mathbb{Z}_+)$ は ϕ の正規化の次数である。以上より、単位正方形領域 $\Omega = [0, 1]^2$ に対して定義される ADF ϕ 、および EDF Φ を図-2 に示す。同図より、正規化の次数 m の増加に伴い ϕ が Φ に漸近することが確認できる ($\phi \xrightarrow{m \rightarrow \infty} \Phi$ [25])。 m の選び方には任意性があるが、先行研究 [25,30] では $m = 2$ 程度が広く用いられている。本研究では、 $m = 1, 2, 4, 8$ を用いて事前検証を行った。

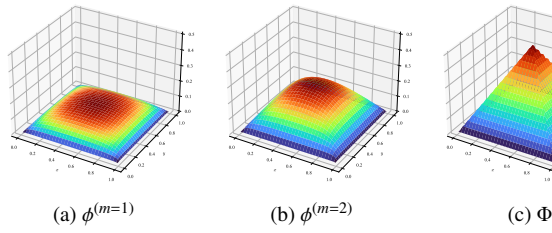


図-2: 単位正方形領域における距離関数

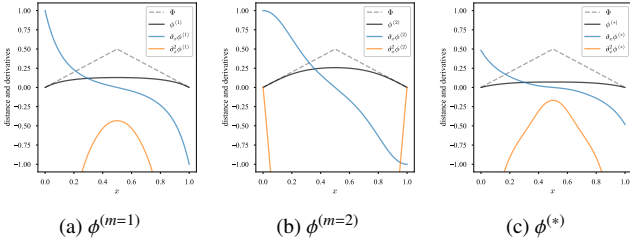


図-3: 各距離関数, および 1 階・2 階導関数

結果, $m = 1, 2$ はほぼ同等の精度・収束性を示したが, $m = 4, 8$ では学習が不安定となった. 以上の検討, および先行研究 [23] に倣い, 以降の数値実験では $m = 1$ を用いる. ここで, Γ 上でゼロとなる距離関数の候補として, 以下も考えられる.

$$(\phi^{(*)}) = \phi = \prod_i \phi_i \quad (14)$$

式 (13) と式 (14) の相違点は $(\sum_i \phi_i^m)^{1/m}$ で除するか否かであるが, 式 (14) は前述の特性 (b) $\nabla_\nu \phi(\mathbf{x}) = 1$ at \mathbf{x} on Γ を満たさない (図-3 参照). 本研究では混合境界値問題への適用を考えるため, 単位法線ベクトルの定義のし易さから, 式 (13) を用いる. 複数の先行研究 [1,2,21,22] で式 (14) に類似した距離関数が用いられているが, 同様の理由から混合境界値問題への適用には不適と考える.

b) 距離関数を用いた境界条件の付与

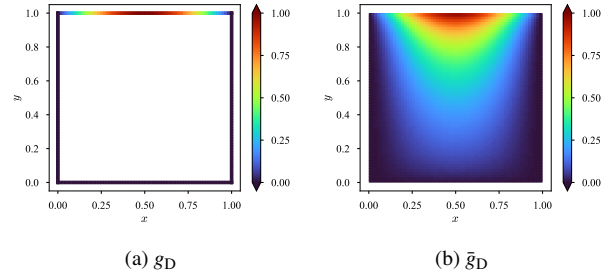
ここでは, Dirichlet 条件について議論する (Neumann 条件も同様に, 距離関数を用いて考慮するが, 詳細は文献 [23,31,32] に譲る). soft imposition では MLP 出力そのものを近似解として採用するが ($u \approx I(\hat{u}) := \hat{u}$), hard imposition では以下を近似解とする.

$$u \approx \mathcal{P}(\hat{u}) := \tilde{u} := \bar{g}_D + \phi_D \hat{u} \quad (15)$$

ただし, \bar{g}_D は Dirichlet 条件の内挿関数 ($\bar{g}_D|_{\Gamma_D} = g_D$) である. ϕ_D は Γ_D までの距離を示す ($\phi_D|_{\Gamma_D} = 0$) から, \tilde{u} は Dirichlet 条件を厳密に満たす. ここで, Dirichlet 境界 Γ_D が $\Gamma_{D,i}$ ($i = 1, 2, \dots, n$) から構成されており, Dirichlet 条件 $g_{D,i}$ と各境界までの距離 ϕ_i が既知とする. このとき, 内挿関数 \bar{g}_D を以下のように定める [24].

$$\bar{g}_D = \sum_i w_i g_{D,i} \quad (16)$$

$$w_i = \frac{\phi_i^{-\mu_i}}{\sum_j \phi_j^{-\mu_j}} = \frac{\prod_{j:j \neq i} \phi_j^{\mu_j}}{\sum_k \prod_{j:j \neq k} \phi_j^{\mu_j}} \quad (17)$$

図-4: Dirichlet 条件 g_D と内挿関数 \bar{g}_D

ただし, μ_i は Γ_i における補間の次数であり, \bar{g}_D は Γ_i 上で $\mu_i - 1$ 回連続微分可能となる [24]. 例として,

$$\Gamma_1 = \{(x, 1) \mid 0 \leq x \leq 1\} \quad (18)$$

$$\Gamma_2 = \{(1, y) \mid 0 \leq y \leq 1\} \quad (19)$$

$$\Gamma_3 = \{(x, 0) \mid 0 \leq x \leq 1\} \quad (20)$$

$$\Gamma_4 = \{(0, y) \mid 0 \leq y \leq 1\} \quad (21)$$

で定められる領域 Ω に対して, $\partial\Omega = \Gamma_D = \bigcup_{i=1}^4 \Gamma_i$, $g_{D,1} = \sin(\pi x)$, $g_{D,2} = g_{D,3} = g_{D,4} = 0$ としたときの g_D と \bar{g}_D を図-4 に示す (μ_i は全て 1 とした). 同図より, 境界上でのみ定義されている $g_D : \Gamma_D \rightarrow \mathbb{R}$ が, $\bar{g}_D : \bar{\Omega} \rightarrow \mathbb{R}$ として領域内部にも拡張されていることが確認できる.

(2) 動的重み付け法

上述の手法に従えば境界条件を満たす近似解を得ることができるため, soft imposition で必要とされた式 (7) を損失関数から取り除くことができる. しかし, 逆解析への適用では式 (9) を導入して観測データを取り込む必要がある. すなわち, hard BC imposition を用いた場合の損失関数は以下のように改められる.

$$\tilde{\mathcal{L}}(\theta) = \tilde{\mathcal{L}}_{\text{PDE}}(\theta, \hat{\kappa}) + \lambda_{\text{Data}} \tilde{\mathcal{L}}_{\text{Data}}(\theta) \quad (22)$$

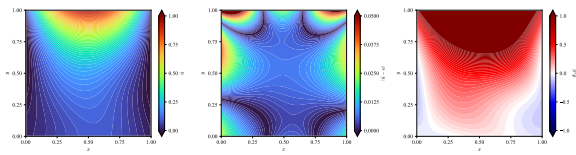
ただし, $\tilde{\mathcal{L}}_{\text{PDE}}$, $\tilde{\mathcal{L}}_{\text{Data}}$ は式 (6), 式 (9) における \hat{u} を \tilde{u} に置き換えたものである. 損失関数が複数の項から構成される場合, 各項の重要度が同様となるよう重みを調整すべきであるが [17,33], 適切な重みを事前に設定することは困難である. そこで本研究では, 著者らが提案した動的重み付け法 (dynamic normalization) を併用する. 詳細は文献 [18,34] に譲るが, n epoch 目のパラメータ更新において, $\tilde{\mathcal{L}}_{\text{Data}}$ に乗ずる重みを以下のように更新する.

$$\lambda_{\text{Data}}^{(n)} = \frac{\|\nabla_{\theta} \mathcal{L}_{\text{PDE}}(\theta^{(n)})\|_2}{\|\nabla_{\theta} \mathcal{L}_{\text{Data}}(\theta^{(n)})\|_2} \quad (23)$$

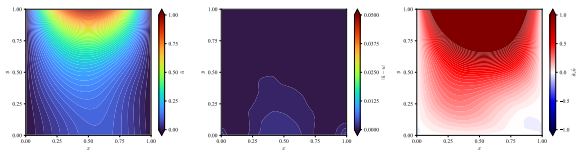
$$\hat{\lambda}_{\text{Data}}^{(n)} = \beta \hat{\lambda}_{\text{Data}}^{(n-1)} + (1 - \beta) \lambda_{\text{Data}}^{(n)} \quad (24)$$

$$\tilde{\lambda}_{\text{Data}}^{(n)} = \frac{\hat{\lambda}_{\text{Data}}^{(n)}}{1 - \beta^n} \quad (25)$$

ただし, 式 (22) の λ_{Data} を式 (25) の $\tilde{\lambda}_{\text{Data}}$ に置き換える. なお, $\beta (\in [0, 1))$ は指数減衰率であり, 経験的に $\beta \geq 0.9$ の値を用いることが多い [18,34]. また, 特に式 (23) の計算負荷が大きいことから, その緩和のため, 重みの更新は $\tau (\geq 1)$ epoch ごとに実行する.



(a) soft imposition (左: 近似解 \hat{u} , 中央: 参照解との差分 $|\hat{u} - u_{\text{FDM}}|$, 右: y 方向への微分 $\partial_y \hat{u}$)



(b) hard imposition (左: 近似解 \tilde{u} , 中央: 参照解との差分 $|\tilde{u} - u_{\text{FDM}}|$, 右: y 方向への微分 $\partial_y \tilde{u}$)

図-5: Poisson 方程式における soft imposition と hard imposition の比較

(3) 正値性の要求

一般的な PINN の逆解析では、物理パラメータの推定値（例えば、 $\hat{\kappa}$ ）を重みやバイアスと同様に更新する。これは、本来正値である κ の推定値 $\hat{\kappa}$ が負値を取り得ることを認めており、学習によって $\hat{\kappa}$ が正値（すなわち、真値）に収束することを期待している、一種の soft imposition と言える。本研究では $\hat{\kappa}$ に対して以下の変換を施すことで、正値性を要求する。

$$\tilde{\kappa} := h_+(\hat{\kappa}) \quad (26)$$

ただし、 $h_+ : \mathbb{R} \rightarrow \mathbb{R}_+$ は正値を返す関数である。本研究では、式 (26) を positivity enforcement と呼ぶこととし、 $h_+(\cdot)$ には指数関数 $\exp(\cdot)$ を用いる ($\mathbb{R} \rightarrow \mathbb{R}_+$ で全単射)。

4. 数値実験

(1) 順解析: Poisson 方程式

まず、式 (1)–(3) の Poisson 方程式を対象とし、境界条件の取り扱いに関する比較を行う。式 (18)–(21) の境界を持つ単位正方形領域 Ω に対して、 $\Gamma_D = \Gamma_1 \cup \Gamma_2 \cup \Gamma_4$, $\Gamma_N = \Gamma_3$ とし、ソース f , Dirichlet 条件 g_D , Neumann 条件 g_N を以下のように定める。

$$f(\mathbf{x}) = \sin(2\pi(x+y)) \quad (27)$$

$$g_D(\mathbf{x}) = \begin{cases} \sin(\pi x) & \text{on } \Gamma_1 \\ 0 & \text{on } \text{otherwise} \end{cases} \quad (28)$$

$$g_N(\mathbf{x}) = 0 \quad \text{on } \Gamma_N \quad (29)$$

以上の条件で有限差分法 (FDM: Finite Difference Method) により得た数値解 u_{FDM} を参照解とする。

幅 $f^{(l)} = 64$, 深さ $L = 3$ の MLP を用いた結果を図-5 に示す。同図より、soft imposition \hat{u} と hard imposition \tilde{u} では、近似解自体の概形は同様であるものの、特に境界付近での誤差の生じ方に大きな相違があることが確認できる。具体的には、図-5 の中央列を比較すると、soft imposition では Dirichlet 境界 $\Gamma_D = \Gamma_1 \cup \Gamma_2 \cup \Gamma_4$ で Dirichlet 条件 g_D が満たされていないことが顕著に確認できる。一方、hard imposition では、 Γ_D 上で $g_D = \tilde{u}$ が

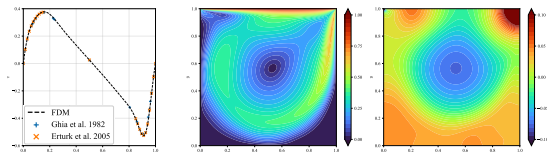


図-6: Cavity 流れ ($\text{Re} = 1,000$) における参照解 (左: 鉛直速度, 中央: 速度ノルム, 右: 圧力)

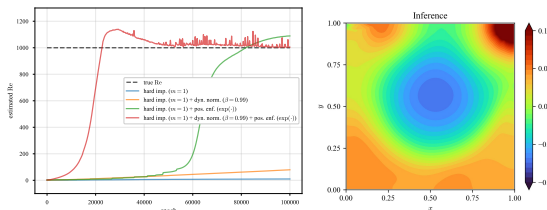


図-7: Cavity 流れ ($\text{Re} = 1,000$) における逆解析結果 (左: Reynolds 数の逆解析結果, 右: 提案法での圧力分布の逆解析結果)

確認できる。また、図-5 の右列より、soft imposition では Neumann 境界 $\Gamma_N = \Gamma_3$ で $\mathbf{n} \cdot \nabla \hat{u} = g_N = 0$ が満たされていない。他方、hard imposition では、 Γ_N 上で $\mathbf{n} \cdot \nabla \tilde{u} = 0$ となっており、Neumann 条件も正しく考慮できている。参照解との相対 ℓ^2 誤差 ϵ を比較すると、

- soft imposition: $\epsilon = 4.47 \times 10^{-2}$
- hard imposition: $\epsilon = 2.31 \times 10^{-3}$

であり、1 オーダー程度の精度向上が確認できる。

(2) 逆解析: Navier-Stokes 方程式

非圧縮性流体の逆解析を対象として、hard imposition と動的重み付け法、および positivity enforcement の併用による効果を検証する。問題設定として、2次元の Cavity 流れを考える。支配方程式は、以下の（無次元化した）非圧縮性 Navier-Stokes 方程式である。

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega \quad (30)$$

$$(\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla p + \frac{1}{\text{Re}} \nabla^2 \mathbf{u} \quad \text{in } \Omega \quad (31)$$

ただし、 $\mathbf{u} = (u, v)^T$ は速度、 p は圧力、 Re は Reynolds 数である（本研究では $\text{Re} = 1,000$ ）。逆解析への適用を検討するため、観測データとして利用する参照解を作成した。これには FDM を適用し、変数配置に Arakawa B 型格子 [35], 空間スキームに 3 次精度風上差分法 [36,37] と 2 次精度中心差分法, 時間スキームに射影法 [38] を用いた。得られた数値解は先行研究 [39,40] と概ね良く合致しており（図-6）、学習に利用するには十分な精度であると考えられる。

本問題は、空間中に無作為に配置した $N_{\text{Data}} (= 256)$ 点の速度の観測データから、当該流れ場の Reynolds 数、および圧力場を逆解析するものとする。幅 $f^{(l)} = 64$, 深さ $L = 5$ の MLP を用いた結果を図-7 に示す。図-7 の左図より、hard imposition の導入、および hard imposition と動的重み付け法の併用では、Reynolds 数の推定値が真値に収束していない。hard imposition と positivity enforcement の併用では、80,000 epoch 程度で真値に収束

しつつあるが、安定するには至っていない。一方、hard imposition と動的重み付け法、および positivity enforcement の併用では、20,000 epoch 程度で真値に収束し始め、50,000 epoch 程度で安定している。また、図-7 の右図より、各手法を併用した結果、観測データを与えていない圧力場も精度良く逆解析できている。

5. 結言

本研究では、PINN における境界条件の取り扱いに関する議論を行った。まず、Poisson 方程式を対象に、損失関数を介して境界条件を考慮する soft imposition [11], 境界条件を満たすよう近似解を修正する hard imposition [23] を比較した。本検討により、hard imposition を導入することで、損失関数を利用せずに境界条件を満たすことができることを確認した。また、非圧縮性 Navier-Stokes 方程式を対象に、hard imposition を用いた物理パラメータ (Reynolds 数)、および未観測の物理量 (圧力場) の逆解析を行った。その結果、hard imposition を導入するのみでは物理パラメータの逆解析が困難であることが確認された。hard imposition に加え、動的重み付け法 [18,34], および物理パラメータの符号を考慮する positivity enforcement を導入することで、推定値が正しく真値に収束することを確認した。また、同手法ではデータを与えていない圧力場の逆解析も高精度に実施可能であることを示した。本研究に示した境界条件の取り扱いは複雑な形状の境界に対しても適用可能であるため [23,24,32], 今後はより一般的な問題への拡張を検討する計画である。

謝辞: 本研究は、JSPS 科研費 JP23KK0182・JP23K17807・JP23H01662・JP22H03601・JP23KJ1685, JST 次世代研究者挑戦的研究プログラム JPMJSP2136, および九州大学数理・データサイエンス教育研究センターの支援を受けた。ここに記して謝意を表する。

参考文献

- [1] I.E. Lagaris, A. Likas, and D.I. Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. *IEEE Transactions on Neural Networks*, Vol. 9, No. 5, pp. 987–1000, 1998.
- [2] I.E. Lagaris, A.C. Likas, and D.G. Papageorgiou. Neural-network methods for boundary value problems with irregular boundaries. *IEEE Transactions on Neural Networks*, Vol. 11, No. 5, pp. 1041–1049, 2000.
- [3] Kevin Stanley McFall and James Robert Mahan. Artificial neural network method for solution of boundary value problems with exact satisfaction of arbitrary boundary conditions. *IEEE Transactions on Neural Networks*, Vol. 20, No. 8, pp. 1221–1233, 2009.
- [4] Amina Adadi and Mohammed Berrada. Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, Vol. 6, pp. 52138–52160, 2018.
- [5] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Benoit, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, Vol. 58, pp. 82–115, 2020.
- [6] Ribana Roscher, Bastian Bohn, Marco F. Duarte, and Jochen Garcke. Explainable machine learning for scientific insights and discoveries. *IEEE Access*, Vol. 8, pp. 42200–42216, 2020.
- [7] Laura von Rueden, Sebastian Mayer, Katharina Beckh, Bogdan Georgiev, Sven Giesselbach, Raoul Heese, Birgit Kirsch, Julius Pfrommer, Annika Pick, Rajkumar Ramamurthy, Michal Walczak, Jochen Garcke, Christian Bauckhage, and Jannis Schuecker. Informed machine learning - a taxonomy and survey of integrating prior knowledge into learning systems. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 35, No. 1, pp. 614–633, 2023.
- [8] Jens Berg and Kaj Nyström. A unified deep artificial neural network approach to partial differential equations in complex geometries. *Neurocomputing*, Vol. 317, pp. 28–41, 2018.
- [9] Weinan E and Bing Yu. The deep ritz method: A deep learning-based numerical algorithm for solving variational problems. *Communications in Mathematics and Statistics*, Vol. 6, No. 1, pp. 1–12, 03 2018.
- [10] Liyao Lyu, Zhen Zhang, Minxin Chen, and Jingrun Chen. Mim: A deep mixed residual method for solving high-order partial differential equations. *Journal of Computational Physics*, Vol. 452, p. 110930, 2022.
- [11] M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, Vol. 378, pp. 686–707, 2019.
- [12] Amirhossein Arzani, Jian-Xun Wang, and Roshan M. D’Souza. Uncovering near-wall blood flow from sparse data with physics-informed neural networks. *Physics of Fluids*, Vol. 33, No. 7, p. 071905, 07 2021.
- [13] Zhao Chen, Yang Liu, and Hao Sun. Physics-informed learning of governing equations from scarce data. *Nature Communications*, Vol. 12, No. 1, p. 6136, 10 2021.
- [14] 出口翔大, 柴田洋佑, 浅井光輝. 予測に物理的意味を付与した機械学習 PINNs による誤差を含む教師データからのパラメータ推定. 土木学会論文集 A2 (応用力学), Vol. 77, No. 2, 2021.
- [15] 出口翔大, 柴田洋佑, 浅井光輝. 空間特徴量抽出を援用した PINNs によるパラメータ逆解析の効率化. 土木学会論文集, Vol. 79, No. 15, pp. 22–15011, 2023.
- [16] Colby L. Wight and Jia Zhao. Solving allen-cahn and cahn-hilliard equations using the adaptive physics informed neural networks. *Communications in Compu-*

- tational Physics*, Vol. 29, No. 3, pp. 930–954, 2021.
- [17] Sifan Wang, Yujun Teng, and Paris Perdikaris. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM Journal on Scientific Computing*, Vol. 43, No. 5, pp. A3055–A3081, 2021.
 - [18] Shota Deguchi and Mitsuteru Asai. Dynamic & norm-based weights to normalize imbalance in back-propagated gradients of physics-informed neural networks. *Journal of Physics Communications*, Vol. 7, No. 7, p. 075005, 2023.
 - [19] Chengping Rao, Hao Sun, and Yang Liu. Physics informed deep learning for computational elastodynamics without labeled data, 2020.
 - [20] Hailong Sheng and Chao Yang. Pfn: A penalty-free neural network method for solving a class of second-order boundary-value problems on complex geometries. *Journal of Computational Physics*, Vol. 428, p. 110085, 2021.
 - [21] Luning Sun, Han Gao, Shaowu Pan, and Jian-Xun Wang. Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data. *Computer Methods in Applied Mechanics and Engineering*, Vol. 361, p. 112732, 2020.
 - [22] Lu Lu, Raphaël Pestourie, Wenjie Yao, Zhicheng Wang, Francesc Verdugo, and Steven G. Johnson. Physics-informed neural networks with hard constraints for inverse design. *SIAM Journal on Scientific Computing*, Vol. 43, No. 6, pp. B1105–B1132, 2021.
 - [23] N. Sukumar and Ankit Srivastava. Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks. *Computer Methods in Applied Mechanics and Engineering*, Vol. 389, p. 114333, 2022.
 - [24] V.L. Rvachev, T.I. Sheiko, V. Shapiro, and I. Tsukanov. Transfinite interpolation over implicitly defined sets. *Computer Aided Geometric Design*, Vol. 18, No. 3, pp. 195–220, 2001.
 - [25] Arpan Biswas and Vadim Shapiro. Approximate distance fields with non-vanishing gradients. *Graphical Models*, Vol. 66, No. 3, pp. 133–159, 2004.
 - [26] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, Vol. 2, No. 5, pp. 359–366, 1989.
 - [27] Moshe Leshno, Vladimir Ya. Lin, Allan Pinkus, and Shimon Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, Vol. 6, No. 6, pp. 861–867, 1993.
 - [28] Ehsan Kharazmi, Zhongqiang Zhang, and George E.M. Karniadakis. hp-vpinns: Variational physics-informed neural networks with domain decomposition. *Computer Methods in Applied Mechanics and Engineering*, Vol. 374, p. 113547, 2021.
 - [29] Vadim Shapiro and Igor Tsukanov. Implicit functions with guaranteed differential properties. In *Proceedings of the Fifth ACM Symposium on Solid Modeling and Applications*, SMA '99, p. 258–269, New York, NY, USA, 1999. Association for Computing Machinery.
 - [30] Daniel Millán, N. Sukumar, and Marino Arroyo. Cell-based maximum-entropy approximants. *Computer Methods in Applied Mechanics and Engineering*, Vol. 284, pp. 712–731, 2015. Isogeometric Analysis Special Issue.
 - [31] Vadim Shapiro and Igor Tsukanov. The architecture of sage – a meshfree system based on rfm. *Engineering with Computers*, Vol. 18, No. 4, pp. 295–311, 2002.
 - [32] I. Tsukanov, V. Shapiro, and S. Zhang. A meshfree method for incompressible fluid dynamics problems. *International Journal for Numerical Methods in Engineering*, Vol. 58, No. 1, pp. 127–158, 2003.
 - [33] Franz M. Rohrhofer, Stefan Posch, Clemens Gößnitzer, and Bernhard C. Geiger. Data vs. physics: The apparent pareto front of physics-informed neural networks. *IEEE Access*, Vol. 11, pp. 86252–86261, 2023.
 - [34] 出口翔大, 浅井光輝. 支配方程式による制約を加えた機械学習 PINN に対する動的重み付け法. 計算工学講演会論文集, Vol. 28, pp. 176–181, 05 2023.
 - [35] Akio Arakawa and Vivian R. Lamb. Computational design of the basic dynamical processes of the ucla general circulation model. In *General Circulation Models of the Atmosphere*, Vol. 17 of *Methods in Computational Physics: Advances in Research and Applications*, pp. 173–265. Elsevier, 1977.
 - [36] T. KAWAMURA and K. KUWAHARA. *Computation of high Reynolds number flow around a circular cylinder with surface roughness*.
 - [37] Tetuya Kawamura, Hideo Takami, and Kunio Kuwahara. Computation of high reynolds number flow around a circular cylinder with surface roughness. *Fluid Dynamics Research*, Vol. 1, No. 2, p. 145, dec 1986.
 - [38] Alexandre J. Chorin. Numerical solution of the navier-stokes equations. *Mathematics of Computation*, Vol. 22, No. 104, pp. 745–762, 1968.
 - [39] U Ghia, K.N Ghia, and C.T Shin. High-re solutions for incompressible flow using the navier-stokes equations and a multigrid method. *Journal of Computational Physics*, Vol. 48, No. 3, pp. 387–411, 1982.
 - [40] E. Erturk, T. C. Corke, and C. Gökçöl. Numerical solutions of 2-d steady incompressible driven cavity flow at high reynolds numbers. *International Journal for Numerical Methods in Fluids*, Vol. 48, No. 7, pp. 747–774, 2005.