

# PINNによってエネルギー保存則・エントロピー増大則を保つGENERIC系の作用素学習

## Operator Learning of GENERIC Systems Satisfying Energy Conservation and Entropy Increase Using PINNs

徐 百歌<sup>1)</sup> 松原 崇<sup>2)</sup> 谷口 隆晴<sup>3)</sup>

Baige Xu, Takashi Matsubara and Takaharu Yaguchi

<sup>1)</sup>神戸大学大学院理学研究科 数学専攻 (〒 657-8501 神戸市灘区六甲台町 1-1,

E-mail: baigexu@stu.kobe-u.ac.jp)

<sup>2)</sup>大阪大学大学院基礎工学研究科 システム創成専攻 (〒 560-8531 大阪府豊中市待兼山町 1-3,

E-mail: matsubara@sys.es.osaka-u.ac.jp)

<sup>3)</sup>神戸大学大学院理学研究科 数学専攻 (〒 657-8501 神戸市灘区六甲台町 1-1,

E-mail: yaguchi@pearl.kobe-u.ac.jp)

GENERIC formulation is a theoretical framework of non-equilibrium thermodynamics, with systems described by it having a specific structure. In this research, in order to reduce the computational cost of physical simulations for such systems, we propose a method applying the DeepONet, a representative model of neural operator based on deep learning, to the physical simulations of GENERIC systems. In particular, we propose to correct the solutions using Physics-Informed Neural Networks (PINNs) for preserving the laws of energy conservation and entropy increase of GENERIC systems.

**Key Words** : Neural operator, Physics-informed neural network, GENERIC system

### 1. はじめに

Hans Christian Öttinger によって提案された GENERIC (General Equation for the Non-Equilibrium Reversible-Irreversible Coupling, 非平衡の可逆・不可逆結合の一般方程式) 定式化 [1] は, 非平衡熱力学の一つの代表的な理論として, 熱力学系の非平衡状態を記述する一般的な数学的枠組みの一つである. この定式化では, 可逆・不可逆の二つのダイナミックスに関する項が含まれている. また, いくつかの拘束条件下では, 系のエネルギー保存則やエントロピーの増大則などの物理法則が成り立つことが知られている. GENERIC 系は, 具体的には

$$\frac{du}{dt} = L \frac{\partial E}{\partial u} + M \frac{\partial S}{\partial u} \quad (1)$$

という方程式で記述される. ここで,  $u = (q, p)$  で,  $q$  は状態を表す変数であり,  $p$  は一般化運動量と呼ばれる変数である.  $E$  と  $S$  は変数  $u$  に依存する実数値関数であり, 全エネルギーと全エントロピーを表す.  $L$  と  $M$  は  $u$  に依存する演算子または行列である. 特に,  $L$  は歪対称であり,  $M$  は半正定値である. GENERIC 系は縮退条件

$$L \frac{\partial S}{\partial u} = 0, \quad M \frac{\partial E}{\partial u} = 0 \quad (2)$$

が満たされるとき, エネルギー保存則とエントロピー増大則が成り立つ.

GENERIC 系のような物理系のシミュレーションでは, 数値ソルバーが多く利用されている. 例えば, 有限要素法に基づくソルバーは, 構造や流体などの系の研究

によく使われている [2]. 一方, このような系の物理シミュレーションで起こる大きい計算コストに対して, 近年, 深層学習を用いる物理系のモデル化手法や物理現象のシミュレーションが盛んに研究されている. このような方法は, 従来の数値ソルバーに代わる有効な手法として, 精度と速度のバランスをより良く実現することが期待されている [3].

深層学習に基づく微分方程式の数値解法のうち, ニューラル作用素 (neural operator) は, 関数から関数への作用素を学習できるようにニューラルネットワークを一般化したものとしてよく知られている. 例えば, 2019 年に提案された DeepONet は, 任意の非線形連続作用素を学習可能なニューラルネットワークモデルであり, 実験によりその効率と精度が確認され, 特に, シミュレーションを大幅に高速化できることが示されている [4]. 2021 年に提案された Fourier Neural Operator(FNO) は, フーリエ空間の積分カーネルを直接にパラメータ化することで, ゼロショット超解像による乱流モデリングに成功し, 従来の偏微分方程式ソルバーより最大 3 桁の高速化を達成したと報告されている [5]. しかし, 以上のようなデータ駆動型作用素学習法では, 物理系に対して利用した際に, 学習対象の物理法則が失われ, シミュレーションによる予測がうまくできない可能性がある. これに対し, 近年, 科学技術分野に機械学習技術を応用する際の手法として, 物理法則を組み込んだ機械学習手法 (Physics-informed Machine Learning, PIML) が注目を集めている. PIML の一例である Physics-informed Neural Network (PINN) は, 既知の物理法則を事前知識

として組み込んで PDE を数値的に解く [6].

本研究では, DeepONet の GENERIC 系に応用し, また, PINN によって方程式の解を補正することで, 物理法則を保ち, かつ, 効率的なシミュレーション手法を提案する.

## 2. 準備: DeepONet

$G$  を入力関数  $u$  をとる非線形作用素とし,  $G(u)$  を対応する出力関数とする. ここで, 出力関数は実数値関数, すなわち,  $G(u)$  の領域内の任意の点  $y$  に対して, 出力  $G(u)(y)$  は実数であるとする. DeepONet は,  $u$  と  $y$  の 2 つを入力として受け取り,  $G(u)(y)$  を出力できるように設計され, 学習される. ただし, 関数を入力とする作用素の学習が目的であるものの, 関数を計算機で扱うためには, 入力関数を離散化する必要がある. そこで, 実際には, 入力関数  $u$  は有限個の位置点集合  $\{x_i\}_{i=1}^m$  における値  $[u(x_1), u(x_2), \dots, u(x_m)]^\top$  として与えられる. DeepONet の構造は, いくつか提案されているが, その中の一つである, スタックされていない (unstacked) DeepONet は, 1 つの trunk network と 1 つの branch network をもつ. Trunk network は  $y$  を入力とし,  $[t_1, t_2, \dots, t_p]^\top \in \mathbb{R}^p$  を出力とする. Branch network は  $[u(x_1), u(x_2), \dots, u(x_m)]^\top$  を入力とし, スカラー  $b_k \in \mathbb{R}, k = 1, 2, 3, \dots, p$  を出力とする. 最終的出力はそれらの内積

$$G(u)(y) \approx \sum_{k=1}^p b_k t_k \quad (3)$$

を取ることによって得られる. 実務的には, 学習の汎化誤差を低減させるために, バイアス項  $b_0 \in \mathbb{R}$  を学習可能な変数として追加する:

$$G(u)(y) \approx \sum_{k=1}^p b_k t_k + b_0. \quad (4)$$

また,  $G(u)(y)$  は  $u$  を条件とする  $y$  の関数とみなすことができるため, DeepONet の損失関数

$$\mathcal{L} = \int \|G_{\text{model}}(u)(y) - f_{\text{data}}(y)\|^2 dy \quad (5)$$

は次のように近似できる:

$$\mathcal{L} \approx \sum_{y_i} \|G_{\text{model}}(u)(y_i) - f_{\text{data}}(y_i)\|^2. \quad (6)$$

## 3. 提案手法

本講演では, DeepONet を用いて GENERIC 系の時間発展を予測する手法を提案する. モデル化対象の系は GENERIC 系であり, 以下の微分方程式に従うと仮定する:

$$\begin{cases} \frac{d}{dt} \mathbf{u}(t) = L \frac{\partial E}{\partial \mathbf{u}} + M \frac{\partial S}{\partial \mathbf{u}} =: \mathbf{g}(\mathbf{u}(t)) \\ \mathbf{u}(0) = \mathbf{u}_0. \end{cases} \quad (7)$$

ここで,  $\mathbf{u}_0 \in \mathbb{R}^k$  は初期条件を表しており,  $\mathbf{u} : [0, T] \rightarrow \mathbb{R}^k$  は方程式の解である. 近似したい  $G$  は, 入力  $\mathbf{u}_0$  を

出力  $\mathbf{u}$  に移す作用素である. つまり,  $\mathbf{g}$  を上のように定義すると,  $G\mathbf{u}_0$  は以下を満たす:

$$(G\mathbf{u}_0)(t) = \mathbf{u}_0 + \int_0^t \mathbf{g}((G\mathbf{u}_0)(\tau)) d\tau. \quad (8)$$

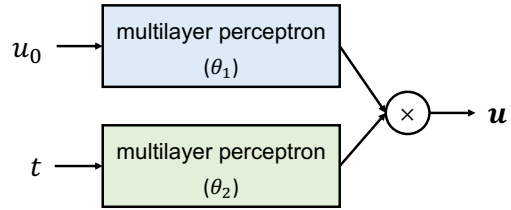


図-1 DeepONet を用い GENERIC 系の学習モデル

図-1 に示すように, branch network の入力  $\mathbf{u}_0 = [u_1(0), u_2(0), \dots, u_k(0)]^\top$  は  $u$  の初期値のサンプル値を並べたベクトルであり, trunk network の入力時刻を表す実数  $t$  である. 損失関数は平均二乗誤差 (MSE) で計算し, 次のように表すことができる:

$$\mathcal{L}_1 = \frac{1}{KP} \sum_{i,j=1}^{K,P} (G_{\text{model}}(\mathbf{u}^{(i)})(t_j) - G_{\text{data}}(\mathbf{u}^{(i)})(t_j))^2, \quad (9)$$

ここで,  $K$  はデータセットのサイズであり,  $P$  は評価したい時刻  $t$  の個数である.  $G_{\text{model}}$  は DeepONet として表された作用素であり,  $G_{\text{data}}$  は学習対象の作用素である.

次に, GENERIC 系の物理性質を保つため, すなわちエネルギーの保存則とエントロピー増大則を満たすためには, DeepONet の学習で得られた方程式の解を補正する必要がある. 解の補正を担うネットワークでは, 図-2 が示すように,  $\mathbf{u}_0$  と多層パーセプトロンのパラメータ  $\theta_1$  を固定とし,  $t$  を入力とする trunk network を学習し直すことで解を補正する. 解  $\mathbf{u}$  が GENERIC 系の物理

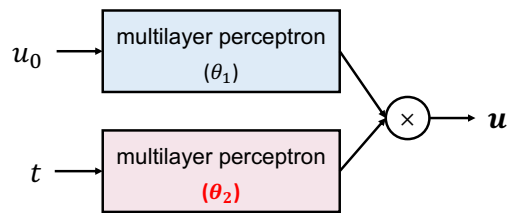


図-2 PINN を用いて修正された GENERIC 系の学習モデル

法則を満たすように, PINN を利用し, 損失関数  $\mathcal{L}_1$  に GENERIC 方程式自身からの乖離量

$$\begin{aligned} \mathcal{L}_2 &= \int_0^T \left\| \frac{\partial \mathbf{u}_\theta}{\partial t} - (L \nabla E(\mathbf{u}_\theta) + M \nabla S(\mathbf{u}_\theta)) \right\|^2 dt \\ &\approx \sum_{t_j} \left\| \frac{\partial \mathbf{u}_\theta}{\partial t_j} - (L \nabla E(\mathbf{u}_\theta) + M \nabla S(\mathbf{u}_\theta)) \right\|^2 \end{aligned} \quad (10)$$

を加える. 学習を行う際には, まず, 両方のネットワークのパラメータ  $\theta_1, \theta_2$  を,  $\mathcal{L}_1$  を最小化するように学習

し、その後、trunk network のパラメータである  $\theta_2$  のみを動かして  $\mathcal{L}_2$  を最小化するように学習する。

数値実験の結果については、発表当日に報告する。

**謝辞:** 本研究は JST CREST JPMJCR1914, JST さきがけ JPMJPR21C7, JSPS 科研費 20K11693 の助成を受けている。

## 参考文献

- [1] H.C. Öttinger: GENERIC integrators: structure preserving time integration for thermodynamic systems, *Journal of Non-Equilibrium Thermodynamics*, Vol.43, pp.89-100, 2018.
- [2] O. C. Zienkiewicz, R. L. Taylor, P. Nithiarasu: The finite element method for fluid dynamics, *Butterworth-Heinemann*, 2013.
- [3] X. Li, Y. Zhou, Z. Pan, et al.: Partial order pruning: for best speed/accuracy trade-off in neural architecture search, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.
- [4] L. Lu, P. Jin, G.E. Karniadakis: DeepONet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators, *arXiv preprint*, arXiv:1910.03193, 2019.
- [5] Z. Li, M. Kovachki, K. Azizzadenesheli, et al: Fourier neural operator for parametric partial differential equations, *arXiv preprint*, arXiv:2010.08895, 2020.
- [6] 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*, 378, pp.686-707, 2019.