
Solving {Non-linear} Oscillator ODE via Physics-informed Neural Network(PINN) {Constrained by Energy Conservation on DeepXDE} with {Noisy} Small Data

卢凯良[†], 苏雨蒙[†], 毕卓^{*}

*xxx@163.com, 上海建桥学院信息技术学院

1

Abstract

应用PINN方法求解典型线性/非线性谐振ODE，以标准测试对比PINN相对于标准数值积分、常规神经网络方法的优缺点^{h1-@`y'}。相较于常规神经网络，PINN方法的原理是将物理定理（例如本文中的能量守恒）以及ODE及其初始边界条件作为惩罚项加入到损失函数中。我们关注于在训练过程中使用尽可能少的数据量的可能性，还要注意增加的罚函数项及其调参对于训练时长的影响。测试结果表明，当问题的非线性较弱时，非常少量的真实采集训练数据^{6ffΓ2*}加上少量的配置点数据就足以预测解。然而，在强非线性问题中情况有所不同，在部分或整个时间积分区间上训练数据的先验知识是必要的。

1 Introduction

1.1 Why PINN: Where Neural Networks can be Improved

凭借爆炸式增长的可用数据和计算资源，深度学习已在图像识别、下棋等多个任务中取得了超越人类的表现 [LBH15, SHS⁺18]。其成功首先仰赖于大数据，却也存在过拟合和泛化性能差的问题 [BG21]以及可理解性问题 [CY⁺22]。另一方面，在现实的复杂科学和工程实践中（例如物理、生化、医学等），数据通常通过昂贵的实验或大规模仿真计算产生，数据往往是稀疏的且带有噪声；有时获取数据的成本高得令人望而却步，甚至根本无法获得大量实验数据 [KKL⁺21, YLRK20, CLS21]。在这种小数据环境下，大多数神经网络方法（包括卷积/递归神经网络）都缺乏鲁棒性，无法提供收敛保

^{††} 为共同一作。(The first two authors contributed equally to this work.)

证 [RPK19], 因此我们不可避免地要面对在部分信息下预测或决策的挑战。好的消息是, 在这些系统建模案例中, 存在着大量的先验知识 (即前人智慧的结晶), 而这些信息在传统的神经网络中尚未得到充分利用; 深度神经网络由于表达能力强, 从而忽略了利用先验信息和知识, 却也导致了过于强调和依赖数据以及算力。

Raissi等人 [RPK19]提出了在神经网络的学习过程中利用物理信息和先验知识的想法, 即物理信息神经网络 (Physics-informed/inspired Neural Network, PINN), 这是一类新的通用函数近似器 (继承了神经网络的强表达能力), 能够编码支配给定数据集的任何基本物理定律 (这经常用微分方程来描述), 即利用我们对世界的观察、经验、物理或数学理解所产生的先验知识来提高学习算法性能 (包括减少数据冗余、改进泛化性能、提高计算效率、增加可解释性和鲁棒性等) 的过程。

在结构化、模块化的神经网络学习架构中, 将物理信息和知识融入网络拓扑结构或计算流程作为模型先验, 具体实现主要分为硬约束(Hard Constraints) [LPY+21]和软约束(Soft Constraints)两种。硬约束一般通过将物理知识或定律 (如微分方程、对称性、守恒律等) [WJX+22]以及边界、初始条件直接嵌入作为神经网络架构或求解过程的一部分来确保它们被严格遵守, 这需要通过特定的网络设计或训练策略来近似实现。本文所讨论的软约束则是通过将物理残差项或正则化项加入损失函数来间接实现的, 较易实现, 这种方法可被视为多任务学习的一种特定用例。软约束允许在满足物理定律或条件时有一定的灵活性或容忍度, 但需要通过仔细调整损失函数中的各项权重来平衡预测性能和物理准确性。

将这些先验作为物理残差项或正则化项, 可使所求解问题的可接受解的解空间限制在可管理的范围内 (例如, 在不可压缩流体动力学问题中, 可以摒弃任何违反质量守恒原理的非现实流动解 [KKL+21])。作为回报, 将这种结构化信息编码到学习算法中, 可以放大算法所看到的数据的信息含量, 使其能够快速引导自己找到正确的解决方案, 并在只有少量 (监督的或标注) 训练示例的情况下也能很好地泛化——这是PINN的关键特性。更进一步地, 这些先验知识或约束条件可以产生更多可解释的学习方法, 这些方法在数据不完善 (如缺失或噪声值、异常值等) 的情况下仍能保持稳健, 并能提供准确且物理上一致的预测, 甚至能用于外推任务。

1.2 PINN as a Complement or Alternative to Numerical Discretization

常微分方程 (ODE) 与偏微分方程 (PDE) 是表达科学和工程领域自然规律的重要工具, 现实世界的物理系统可以使用基于不同领域特定的假设和简化的微分方程来建模, 这些模型可以用来近似这些系统的行为 [GKLS24], 前者精准刻画一维动态, 后者处理多维复杂变化, 已成为构建自然规律的数学基础, 推动了对自然现象及其蕴含的科学问题的深入理解和预测。解常/偏微分方程的常见代表性数值离散化求解方法有Runge-Kutta方法和有限元方法(FEM)。从可理解性的角度来看, 由于底层近似的简单性, 这些方法都得到了很好的理解: 具有现成的误差估计以及收敛性和稳定性保证。

通过数值离散化求解微分方程 (ODE/PDE) 以模拟各类场问题方面取得了巨大进步, 理论基础完备, 具有高效率、高精度、稳定性好的优势, 但仍有一些严重限制其应用

的瓶颈问题——包括维度灾难问题，网格生成仍然很复杂，在合并实验数据方面存在困难、仍然无法将噪声数据无缝纳入现有算法，而且无法解决由参数化PDE控制的高维问题等等 [KKL+21]。比如，有限元方法的一个明显的缺点是，它通常依赖于空间离散（空间网格或大多项式基），会遭遇维度诅咒问题：在三维空间，已经较难使用，遑论更高维问题；而且，某些非线性和非光滑的偏微分方程很难离散，例如，由于行为需要解决在一个非常细的网格，一般的非光滑行为，或奇点。由于有限元是一种基于网格的求解器，要在某些不规则域上求解，就必须采用具有设计和求解挑战性的定制方法 [GKLS24]。此外，解决逆问题（例如，用于推断功能材料中的材料性质或发现反应输运中缺失的物理学）往往是非常昂贵的，而且需要复杂的公式、新的算法和复杂的计算机代码；并且，通过传统的方法解决现实中缺失、间隙或噪声边界条件的物理问题目前是不可能的 [KKL+21]。

深度学习方法（包括PINN [RPK19]、深度利兹方法[]、深度伽辽金方法[]等）在解决高维问题方面（例如具有高分辨率的图像分类[]、）取得了巨大成功，最近已成为各种偏微分方程数值解的常用方法。它们有潜力克服上述数值离散化经典方法所面临的一些挑战：①通过利用自动微分[]，消除了离散化的需要，从而具有不依赖于网格的优势。②神经网络能够表示比有限元基更一般的功能，可克服维度灾难 [PMR+16]，为求解高维PDE提供了一个很有前途的方向。③虽然与有限元等数值求解器相比训练神经网络（非凸优化问题）可能变得需要计算，但它在评估新数据点时（即泛化能力）是非常有效和高效的。④其集成数据和知识的能力和灵活性为处理知识不完善或有限数据的问题提供了一个可扩展的框架。然而训练深度神经网络需要大量数据，而相关科学问题，如前所述，并不总能获得这些数据。

其中，PINN作为一种潜在的颠覆性技术，将物理信息和知识融入网络拓扑结构或计算流程中，PINN可以遵守任何对称性、不变性或守恒性原则，这些原则源于由一般时变和非线性偏微分方程建模的、支配观测数据的物理定律 [RPK19]。这种简单而强大的构造使我们能够解决计算科学中的各种问题，从而开发出新的数据高效和物理信息学习机、新的常/偏微分方程数值解算器，以及新的数据驱动模型反演和系统识别方法。需要强调的是，PINN并非用于取代FEM：PINN在解决本节所述的假定问题和逆问题等方面特别有效；而对于不需要任何数据同化的正向问题，现有的基于网格的数值求解器目前的性能优于PINN [KKL+21, GKLS24]。表 1 中归纳了求解微分方程的三类方法（即数值离散化、常规神经网络和PINN方法）的优缺点。

1.3 Advantages and Application Scenarios of PINN

PINN具有在小数据环境中实现强泛化的独特优势。通过强制或嵌入物理学，神经网络模型被有效地限制在一个较低维度的流形，因此只需少量数据就能进行训练；PINN不仅能进行内插，还能进行外推。

必须强调的是，与解决微分方程的任何经典数值方法（基于网格的方法，例如RK4、有限差分法FDM、有限元法FEM）不同，这种预测是在没有对时空域进行任何离散化的情况下获得的，利用了自动微分[]，可以打破维度灾难 [PMR+16, LMMK21]，能够以

表 1: 求解微分方程三类方法的对比

求解方法	数值离散化方法	常规神经网络	PINN
特点: ✓ 优点 × 缺点	✓ 理论基础完备、可理解, 具有现成的误差估计以及收敛性和稳定性保证 ✓ 高效率 ✓ 高精度 × 遭受维度灾难 × 网格生成仍很复杂 × 无法无缝整合噪声数据 × 解决逆问题的成本过高	✓ 无网格 ✓ 克服维度灾难, 能应对高维和非线性问题 ✓ 对新数据的泛化能力 × 需要大数据和强算力 × 过拟合 × 黑箱问题	✓ 常规神经网络的优点 + ✓ 小数据、不易过拟合, 泛化能力强 ✓ 可处理不完善数据和不完整模型 ✓ 处理病态问题和逆问题时有效且高效 ✓ 可理解性高 ✓ 网络参数少、速度快 ✓ 灵活性好、跨计算平台

机器精度精确评估配置点上的微分算子 [BB23]。PINN不需要处理令人望而却步的小步数, 因此可以轻松处理不规则和移动域问题 [GKLS24], 并能在更高维度上良好扩展。自动微分是PINN相对于其他经典数值方法求解DE的一个显著优势, 与传统数值方法相比, 自动微分利用链式规则通过网络反向传播并计算导数, 它并不依赖于离散该域的网格, 但对其而言抽样方法变得更加重要 [WZT⁺23]。

PINN将数据与数学物理模型无缝集成, 即使在部分理解、不确定的情况下也是如此。由于PINN表述本身的平滑性或规律性, 即使问题的假设并不完美, 也有可能找到有意义的解决方案 [YLRK20]。

PINN [RPK19]这种利用自动微分的无网格方法, 可以直接处理非线性问题(神经网络善于处理非线性问题的固有特性), 而无需承诺任何先验假设、线性化或局部时间步进。PINN在处理病态问题和逆问题时有效且高效, 例如, 没有指定初始条件或边界条件的正演问题和反演问题, 或PDEs中某些参数未知的问题, 在这些情况下, 经典数值方法可能会失效 [KKL⁺21]。

1.4 Related Work & Our Contributions

Related Work.

[Bat23b]还针对在求解刚性ODE的IVP时遇到的①已知解数据(即训练数据)非常有限(主要指初始或边界条件)、②积分区间如果过大将导致PINN难以准确预测解的问题, 通过嵌入更多物理信息、优化训练数据损失以确保其更全面地考虑所有初始条件以及渐进式学习策略即逐步增加积分区间并在每个区间内优化PINN的参数、采用移动网格来最小化微分方程的残差等改进策略使得改进后的PINN在求解刚性ODE的IVP具有更高的精度和更稳定的训练过程。此外, 这些改进策略也被证明在解决边界值问题(BVP)时同样有效, 例如高雷诺数稳态对流-扩散方程的求解。

[Bat23a]进一步比较了软约束和硬约束在解决高阶Lane-Emden-Fowler类型方程(如二阶Lane-Emden方程、三阶Emden-Fowler方程和四阶Lane-Emden-Fowler方程)时的性能。基准测试结果表明: 软约束灵活性较高, 能够处理多种不同的物理场景和边界条件, 但可能需要更多的训练数据和迭代次数来收敛到满意的解; 硬约束能够更直接地满足

微分方程的约束，因此可能在较少的迭代次数内达到较高的精度，但可能需要更复杂的试验解选择和调整过程。

Our Contributions.

- 本文通过相关文献调研在更广泛的解微分方程（包括ODE和PDE）领域综述和归纳了PINN相对于常规神经网络方法、标准数值积分的优势（见表 1）。
-

2 Method & Result

2.1 PINN Principle & Implementation

2.1.1 Governing Equations

我们考虑一个定义在空间或时空域 $\Omega \subseteq \mathbb{R}^d$ 上的物理系统，其中未知数 $u(\mathbf{x}) : \mathbb{R}^d \rightarrow \mathbb{R}^m$ 为系统状态变量，是空间或时空坐标 $\mathbf{x} \in \Omega$ 的函数，对于时间无系统 $\mathbf{x} = (x_1, \dots, x_d)$ ，对于时间依赖系统 $\mathbf{x} = (x_1, \dots, x_{d-1}, t)$ 。支配系统的物理规律常以ODE、PDE表征，这些方程称为控制方程 [HLZ⁺23]，如下

$$\text{Differential equation: } \mathcal{F}(u; \theta)(\mathbf{x}) \equiv \mathcal{F}(u, \mathbf{x}; \theta) = 0, \mathbf{x} \in \Omega. \quad (1)$$

$$\text{Initial Conditions: } \mathcal{I}(u; \theta)(x, t_0) = 0, x \in \Omega_0. \quad (2)$$

$$\text{Boundary Conditions: } \mathcal{B}(u; \theta)(x, t) = 0, x \in \partial\Omega. \quad (3)$$

对于时间依赖系统（即动态系统），我们需要为状态变量（有时还包括其导数）在初始时刻 t_0 设置初始条件 $\mathcal{I}(u; \theta)(x, t_0) = 0, x \in \Omega_0$ ，如式(2)。 $\theta \in \Theta$ 对系统进行参数化或控制，其中 θ 可以是一个矢量，也可以是一个包含在控制方程中的函数。对于以偏微分方程为特征的系统，我们还需要对空间域 $\partial\Omega$ 边界上的状态变量进行约束，以使系统适定。对于边界点 $x \in \partial\Omega$ ，我们有边界条件 $\mathcal{B}(u; \theta)(x, t) = 0, x \in \partial\Omega$ ，如式(3)。如果对初始条件和边界条件没有约束，则有： $\mathcal{I}(u; \theta) \triangleq 0$ 和 $\mathcal{B}(u; \theta) \triangleq 0$ 。

2.1.2 Framework of PINN via Soft Constrains

假设有一个服从方程 (1) 的系统和一个数据集 $\{u(\mathbf{x}_i)\}_{i=1, \dots, N}$ ，然后就可以构建神经网络 $u_w(\mathbf{x})$ 并用以下损失函数对其进行训练，即

$$\begin{aligned} \mathcal{L} = & \frac{\lambda_r}{|\Omega|} \int_{\Omega} \|\mathcal{F}(u_w; \theta)(\mathbf{x})\|^2 d\mathbf{x} + \frac{\lambda_i}{|\Omega_0|} \int_{\Omega_0} \|\mathcal{I}(u_w; \theta)(x)\|^2 dx \\ & + \frac{\lambda_b}{|\partial\Omega|} \int_{\partial\Omega} \|\mathcal{B}(u_w; \theta)(\mathbf{x})\|^2 d\mathbf{x} + \frac{\lambda_d}{N} \sum_{i=1}^N \|u_w(\mathbf{x}_i) - u(\mathbf{x}_i)\|^2, \end{aligned} \quad (4)$$

其中 $\int_{\Omega} \|\mathcal{F}(u_w; \theta)(\mathbf{x})\|^2 d\mathbf{x}$ 是使得PINN u_w 满足微分方程约束的残差损失； $\int_{\Omega_0} \|\mathcal{I}(u_w; \theta)(\mathbf{x})\|^2 d\mathbf{x}$ 和 $\int_{\partial\Omega} \|\mathcal{B}(u_w; \theta)(\mathbf{x})\|^2 d\mathbf{x}$ 分别是使得PINN满足初始条件和边界条件的相应损失； $\frac{1}{N} \sum_{i=1}^N \|u_w(\mathbf{x}_i) - u(\mathbf{x}_i)\|^2$ 是PINN在匹配监督数据集的常规数据损失。

为简化记号，我们用

$$l_r \triangleq \|\mathcal{F}(u_w; \theta)(\mathbf{x})\|^2, \quad (5)$$

$$l_i \triangleq \|\mathcal{I}(u_w; \theta)(\mathbf{x})\|^2, \quad (6)$$

$$l_b \triangleq \|\mathcal{B}(u_w; \theta)(\mathbf{x})\|^2, \quad (7)$$

$$l_d \triangleq \|u_w(\mathbf{x}_i) - u(\mathbf{x}_i)\|^2, \quad (8)$$

并记 $\Omega_r \triangleq \Omega, \Omega_i \triangleq \Omega_0, \Omega_b \triangleq \partial\Omega, \Omega_d \triangleq \mathcal{D}_d$ 因此，这些损失可以统一写为

$$\mathcal{L}_k = \frac{\lambda_k}{|\Omega_k|} \int_{\Omega_k} l_k d\mathbf{x}, k \in \{r, i, b, d\}. \quad (9)$$

PINN 的损失具有灵活性和可扩展性，如果没有可用数据或初始/边界限制，我们可以直接省略相应的损失项。这些损失的学习率或权重可以通过设置超参数 $\lambda_r, \lambda_i, \lambda_b$ 和 λ_d 来调整。为了计算式 (4)，我们需要评估几个积分项，其中涉及 $u_w(\mathbf{x})$ 的高阶导数计算（PINNs 利用计算图的自动微分来计算这些导数项）。然后，使用 Monte-Carlo 抽样法利用一组配置点对积分进行近似。我们使用 $\mathcal{D}_r, \mathcal{D}_i, \mathcal{D}_b$ 和 \mathcal{D}_d 来表示配置点数据集，用 N_r, N_i, N_b 和 N_d 表示数据量。那么，损失函数可以近似为

$$\begin{aligned} \mathcal{L} = & \frac{\lambda_r}{N_r} \sum_{i=1}^{N_r} \|\mathcal{F}(u_w; \theta)(\mathbf{x}_i)\|^2 + \frac{\lambda_i}{N_i} \sum_{i=1}^{N_i} \|\mathcal{I}(u_w; \theta)(\mathbf{x}_i)\|^2 \\ & + \frac{\lambda_b}{N_b} \sum_{i=1}^{N_b} \|\mathcal{B}(u_w; \theta)(\mathbf{x}_i)\|^2 + \frac{\lambda_d}{N_d} \sum_{i=1}^N \|u_w(\mathbf{x}_i) - u(\mathbf{x}_i)\|^2. \end{aligned} \quad (10)$$

式 (10) 可以使用 SGD 等一阶方法和 L-BFGS 等二阶优化器进行高效训练。

2.1.3 DeepXDE Implementation

[BB23]是用Pytorch实现的，我们用DeepXDE [LMMK21]实现并与之进行了结果对比。用DeepXDE编写的代码更短、更接近于数学公式。在DeepXDE中求解微分方程如同使用内置模块“搭积木”，包括指定计算域（几何和时间）、微分方程、边界/初始条件、约束、训练数据、神经网络结构和训练超参数。这些模块或组件都是松散耦合的，因此DeepXDE具有良好的结构和高度可配置性。

2.2 Results for PINN Solving Oscillator ODEs

为了便于直接对比，我们选用与 [BB23]中相同的线性和非线性谐振ODE（即Tutorial Example和Van der Pol osillators）求解。

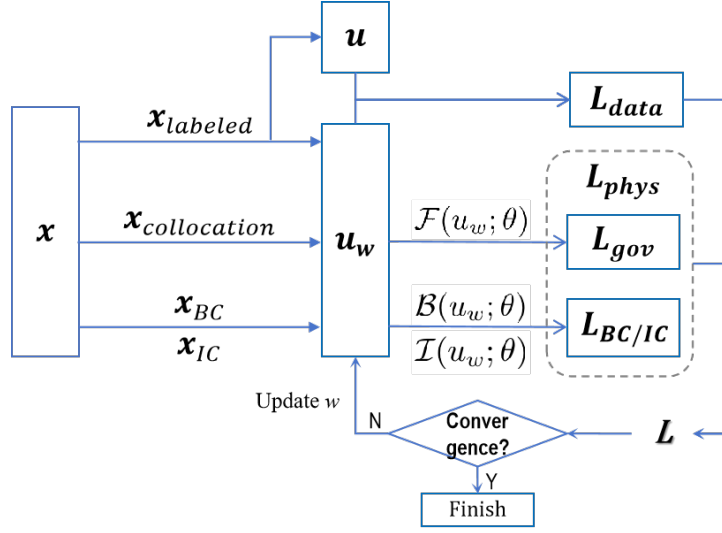


图 1: Schematic diagram of PINN via soft constrains.

2.2.1 PINN Solving Linear Oscillator ODE vs NN

线性谐振方程（Tutorial Example）

$$\frac{dy}{dt} + 0.1y - \sin(\pi t/2) = 0. \quad (11)$$

式中, $t \in [0, 30]$, 初始条件 $y_0 = 1$ 。我们使用常规神经网络和PINN两种方法分别求解。

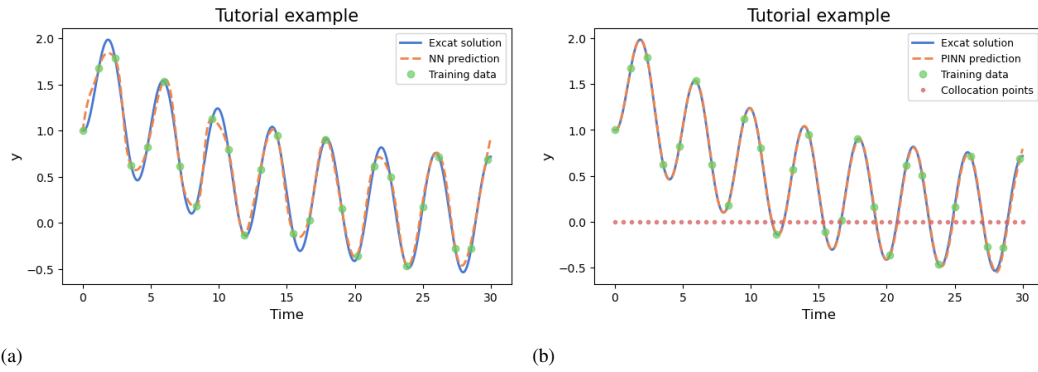


图 2: PINN vs NN Results之一.

极简数据结果。

2.2.2 PINN Solving Non-linear Oscillator ODE

VDP

表 2: A table of mathematical notations.

Notations	Description
u	state variables of the physical system
\mathbf{x}	spatial or spatial-temporal coordinates
x	spatial coordinates
t	temporal coordinates
θ	parameters for a physical system
w	weights of neural networks
$\frac{\partial}{\partial x_i}$	partial derivatives operator
\mathcal{D}_i^k	$\frac{\partial^k}{\partial x_i^k}$, k -order derivatives for variable x_i
∇	nabla operator (gradient)
Δ	Laplace operator
\int	integral operator
\mathcal{F}	differential operator representing the PDEs/ODEs/etc.
\mathcal{I}	initial conditions (operator)
\mathcal{B}	boundary conditions (operator)
Ω	spatial or spatial-temporal domain of the system
Θ	space of the parameters θ
W	space of weights of neural networks
\mathcal{L}	loss functions
\mathcal{L}_r	residual loss
\mathcal{L}_b	boundary condition loss
\mathcal{L}_i	initial condition loss
l_k	residual (error) terms
$\ \cdot\ $	norm of a vector or a function

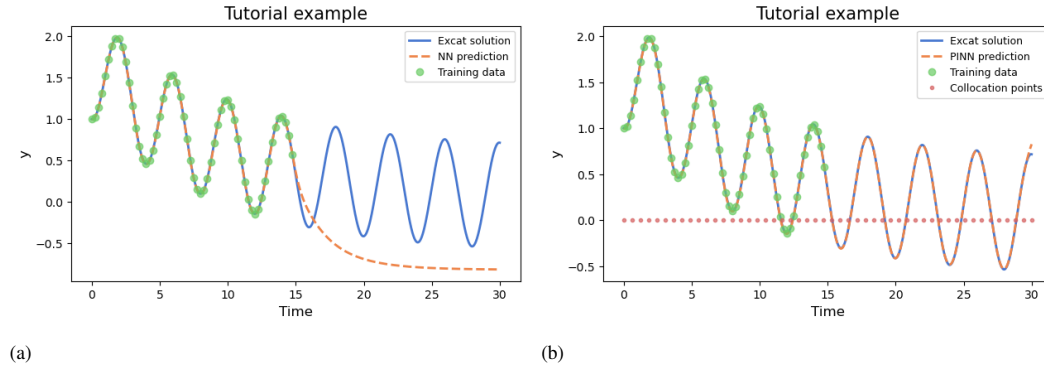


图 3: PINN vs NN Results之二.

2.3 DeepXDE vs Pytorch 代码对比

3 Discussion

3.1 有无能量守恒约束的结果比较

{能量守恒约束对Duffing有用，对VDP无用}

3.2 计算效率对比: @GPU vs @CPU

4 Concluding Remarks

4.1 Summary

{① collocation点是无监督, 真实采集training点是有监督}

4.2 Future Work

此外, PINN与领域分解相结合可扩展至大型问题 [KKL⁺21], 并已进一步扩展到求解积分微分方程(IDE)、分数阶微分方程(FDE)和随机微分方程(SDE) [LMMK21]。

5 Code Availability

论文中已有部分代码示例, 详细代码将分享在 <https://github.com/mikelu-shanghai/PINNtoODEwithSmallData>。²

参考文献

- [Bat23a] Hubert Baty. Solving higher-order lane-emden-fowler type equations using physics-informed neural networks: benchmark tests comparing soft and hard constraints. *arXiv:2307.07302*, 2023.
- [Bat23b] Hubert Baty. Solving stiff ordinary differential equations using physics informed neural networks (pinns): simple recipes to improve training of vanilla-pinns. *arXiv:2304.08289*, 2023.
- [BB23] Hubert Baty and Leo Baty. Solving differential equations using physics informed deep learning: a hand-on tutorial with benchmark tests. *arXiv:2302.12260*, 2023.
- [BG21] Mohammad Mahdi Bejani and Mehdi Ghatee. A systematic review on overfitting control in shallow and deep neural networks. *Artif. Intell. Rev.*, 54(8):6391–6438, 2021.
- [CLS21] Zhao Chen, Yang Liu, and Hao Sun. Physics-informed learning of governing equations from scarce data. *Nature Communications*, 12:6136–48, 10 2021.
- [CYY⁺22] Kwan Ho Ryan Chan, Yaodong Yu, Chong You, Haozhi Qi, John Wright, and Yi Ma. Redunet: A white-box deep network from the principle of maximizing rate reduction. *Journal of Machine Learning Research*, 23(114):1–103, 2022.

²待论文接收后, 需要的读者可与我们联系。

- [GKLS24] Tamara G Grossmann, Urszula Julia Komorowska, Jonas Latz, and Carola-Bibiane Schönlieb. Can physics-informed neural networks beat the finite element method? *IMA Journal of Applied Mathematics*, 89(1):143–174, 05 2024.
- [HLZ⁺23] Zhongkai Hao, Songming Liu, Yichi Zhang, Chengyang Ying, Yao Feng, Hang Su, and Jun Zhu. Physics-informed machine learning: A survey on problems, methods and applications, 2023.
- [KKL⁺21] George Em Karniadakis, Ioannis G. Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3:422 – 440, 2021.
- [LBH15] Yann LeCun, Y. Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436–44, 05 2015.
- [LMMK21] Lu Lu, Xuhui Meng, Zhiping Mao, and George Em Karniadakis. Deepxde: A deep learning library for solving differential equations. *SIAM Review*, 63(1):208–228, 2021.
- [LPY⁺21] 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*, 43(6):B1105–B1132, 2021.
- [PMR⁺16] Tomaso A. Poggio, Hrushikesh Narhar Mhaskar, Lorenzo Rosasco, Brando Miranda, and Qianli Liao. Why and when can deep-but not shallow-networks avoid the curse of dimensionality: A review. *International Journal of Automation and Computing*, 14:503 – 519, 2016.
- [RPK19] 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*, 378:686–707, 2019.
- [SHS⁺18] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, L. Sifre, Dhharshan Kumaran, Thore Graepel, Timothy P. Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362:1140 – 1144, 2018.
- [WJX⁺22] Jared Willard, Xiaowei Jia, Shaoming Xu, Michael Steinbach, and Vipin Kumar. Integrating scientific knowledge with machine learning for engineering and environmental systems. *ACM Computing Surveys*, 55:1–37, 03 2022.
- [WZT⁺23] Chenxi Wu, Min Zhu, Qinyang Tan, Yadhu Kartha, and Lu Lu. A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed

neural networks. *Computer Methods in Applied Mechanics and Engineering*, 403:115671, 2023.

- [YLRK20] Alireza Yazdani, Lu Lu, Maziar Raissi, and George Em Karniadakis. Systems biology informed deep learning for inferring parameters and hidden dynamics. *PLOS Computational Biology*, 16(11):1–19, 2020.