



·前沿技术与交叉科学·

# 基于物理信息驱动傅里叶神经算子的 Vlasov 方程求解方法<sup>\*</sup>

付 伟, 王 川, 张天爵, 周洪吉

(中国原子能科学研究院 核技术综合研究所, 北京 100023)

**摘 要:** Vlasov 方程是研究等离子体物理的重要方程, 在高温、完全电离且忽略库伦碰撞的情况下, 其数值求解方法主要有欧拉类方法和拉格朗日类方法。考虑传统数值求解方法在高精度网格条件下计算资源快速增长、维度灾难等问题, 采用了基于物理信息驱动的傅里叶神经算子 (PFNO) 对 Vlasov 方程进行求解。该方法将傅里叶神经算子高维函数映射能力与 Vlasov 方程物理约束结合, 构建了数据-物理信息驱动的深度神经网络, 可提升模型在稀疏数据条件下的泛化性能, 同时具有网格无关性特征。数值实验表明, 该方法在保证求解精度的同时, 相比传统有限元法和谱方法计算效率提升 1~2 个数量级, 且能并行处理大批量数据。该研究为高维 Vlasov 动力学方程的求解提供了新思路, 在惯性约束聚变、空间等离子体模拟等领域具有一定应用潜力。

**关键词:** 神经算子; Vlasov 方程; 深度学习; 物理信息驱动; 等离子体

中图分类号: O534.2

文献标志码: A

doi: 10.11884/HPLPB202537.250071

## Method for solving Vlasov equation based on physics-informed Fourier neural operator

Fu Wei, Wang Chuan, Zhang Tianjue, Zhou Hongji

(Department of Nuclear Technology and Application, China Institute of Atomic Energy, Beijing 100023, China)

**Abstract:** [Background] The Vlasov equation is a cornerstone in plasma physics, governing the evolution of distribution functions in high-temperature, collisionless plasmas. Conventional numerical methods, including Eulerian and Lagrangian approaches, often encounter severe computational challenges due to the rapid increase in cost with fine grid resolutions and the curse of dimensionality. These limitations restrict their effectiveness in large-scale kinetic plasma simulations needed in fusion research and space plasma studies. [Purpose] This work aims to develop an efficient and scalable computational framework for solving the Vlasov equation that mitigates the drawbacks of traditional methods. The study particularly addresses the need for maintaining accuracy and physical consistency while significantly reducing computational costs in high-dimensional simulations. An approach based on the physics-informed Fourier neural operator (PFNO) is introduced. [Methods] The method integrates the high-dimensional function mapping ability of the Fourier neural operator with the physical constraints of the Vlasov equation. A physics-informed loss function is constructed to enforce mass, momentum, and energy conservation laws. The framework was evaluated through benchmark tests against finite element and spectral solvers, and its parallel performance was assessed on large-scale computing platforms. [Results] The PFNO approach demonstrates accuracy comparable to conventional solvers while achieving computational efficiency improvements of one to two orders of magnitude. The method shows strong generalization under sparse-data conditions, exhibits grid independence, and scales effectively in parallel computing environments, enabling efficient treatment of high-dimensional plasma dynamics. The study presents a novel paradigm for solving high-dimensional Vlasov equations by combining deep learning operators with physical principles. [Conclusions] The PFNO framework enhances efficiency without sacrificing physical accuracy, making it a promising tool for applications in inertial confinement fusion, astrophysical plasma modeling, and space

\* 收稿日期: 2025-04-10; 修订日期: 2025-08-21

基金项目: 中国原子能科学研究院英才基金项目 (YC232505000404)

联系方式: 付 伟, fuwei@ciae.ac.cn.

plasma simulations. Future research directions include extension to multi-species and relativistic plasma systems.

**Key words:** neural operator, Vlasov equation, deep learning, physics-informed, plasma

等离子体的分布密度函数  $f(x, v, t)$  所遵循的玻耳兹曼方程, 在等离子体温度足够高且忽略碰撞的情况下, 可以用 Vlasov 方程描述<sup>[1]</sup>。该方程广泛应用于惯性约束聚变<sup>[2]</sup>、空间等离子体<sup>[3]</sup>、激光-等离子体相互作用等领域<sup>[4]</sup>, 是描述等离子运动的重要方程。传统的数值求解方法主要有两类<sup>[5-7]</sup>: 一是拉格朗日类方法, 典型代表是 PIC (Particle in Cell) 方法, 通过模拟大规模粒子群在电磁场作用下的运动, 自洽地计算出物理空间网格上粒子的密度, 并利用统计学方法得出分布函数的具体形式。然而, 尽管该方法能用一定量的粒子获得较好的结果, 但在某些情况下, 粒子跟踪方法固有的数值噪声会显著影响粒子分布函数的精确度, 这种噪声随着粒子数量  $N$  的增加以  $1/N$  的比例增长, 尤其在描述带电粒子束的复杂运动时尤为重要。二是欧拉类方法, 其在相空间上利用有限差分、有限元等方法, 对 Vlasov 方程进行离散化处理, 进而得到网格点处的分布函数值, 尽管欧拉类方法能够考虑复杂的边界条件, 但它们存在守恒性差和求解大质量矩阵的困难, 因此在向多维空间推广时面临挑战。此外, 20 世纪 90 年代发展起来的半拉格朗日方法因其无数值耗散、高精度的特点<sup>[8]</sup>, 也广泛用于等离子体动力学模拟。

近年随着 GPU 计算硬件的能力提升, 深度学习技术开始在机器视觉、自然语言处理、科学计算等领域凸显出强大的应用潜力。近年来, 利用深度学习方法求解 PDE 方程已取得一定的验证结果。深度学习方法在偏微分方程的应用主要可以分为两类, 一类为纯数据驱动的方法, 例如卷积神经网络<sup>[9]</sup>、神经算子方法<sup>[10]</sup>等; 另一类为物理信息驱动的神经网络方法<sup>[11]</sup> (PINN, physics informed neural networks), 其特征将 PDE 方程显式或隐式地耦合进神经网络结构或者损失函数中, 约束神经网络训练过程中的行为, 从而更好地符合物理规律。目前 PINN 方法已在麦克斯韦方程<sup>[12]</sup>、纳维-斯托克斯方程<sup>[13]</sup>和薛定谔方程<sup>[14]</sup>等诸多类型方程中取得了较好的训练效果。已有部分研究者开展了基于 PINN 方法求解 Vlasov 方程的实践<sup>[15]</sup>, 研究表明利用 PINN 方法求解 Vlasov 方程的缺点在于收敛困难, 往往只能拟合给定初始条件下的解, 对于不同初始条件的解泛化效果不佳, 同时可能存在梯度消失的风险。神经算子是近年来兴起的一种新型深度学习架构, 专为高效求解复杂物理系统的偏微分方程而设计。与传统神经网络不同, 神经算子 (Neural Operator) 基于函数空间和函数空间的映射关系, 能够直接在无限维函数空间中学习物理规律的隐式表达, 突破了传统数值方法对离散网格的依赖。以傅里叶神经算子 (Fourier Neural Operator, FNO) 为代表的先进模型, 通过频域卷积操作捕捉物理场演化的全局特征, 在处理湍流模拟、材料应力预测等复杂问题时展现出显著优势。在计算速度方面, 相较于传统有限元方法, 神经算子可将计算速度提升成百数千倍, 同时保持对不规则计算域和不同分辨率输入数据的强适应能力。

综上, 本文将从朗道阻尼和双束不稳定性这两个典型 Vlasov 方程算例出发, 基于傅里叶神经算子模型结构, 采用物理信息和训练数据加权的损失函数, 在函数空间逼近 Vlasov 方程初始条件与方程解的映射, 实现 Vlasov 方程的快速求解。

## 1 基础方程的提出

带电粒子运动过程中, 主要受到电磁力影响, 相空间的 Vlasov 方程通常与 Maxwell 方程组耦合成为 Vlasov-Maxwell 方程组, 该方程组可自洽地描述粒子在自场和外场下的运动规律, 是最为经典的动力学方程。Vlasov 方程在静电近似下, 可将简化为 Vlasov-Poisson 方程, 其形式可以表示为

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \frac{\partial f}{\partial \mathbf{x}} + \frac{q}{m} \mathbf{E} \cdot \frac{\partial f}{\partial \mathbf{v}} = 0 \quad (1)$$

式中:  $f$  为粒子密度分布函数;  $\mathbf{v}$  为粒子速度;  $q$  为粒子所带电荷量;  $m$  为粒子质量。电场  $E$  满足 Poisson 方程

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\epsilon_0} \quad (2)$$

方程的初始条件:  $f(x, v, 0) = f_0(x, v)$ , 边界条件通常为周期性条件。该方程为非线性偏微分方程, 仅在极少数情况下存在解析解。为保证数值稳定性, 我们采用如下的方式对该方程进行无量纲化。

$$\hat{m} = \frac{m}{m_e}, \quad \hat{q} = \frac{q}{-e}, \quad \hat{t} = \frac{t}{\omega_p^{-1}}, \quad \hat{x} = \frac{x}{c\omega_p^{-1}}, \quad \hat{v} = \frac{v}{c}, \quad \hat{E} = \frac{E}{-\omega_p c m_e / e} \quad (3)$$

式中:  $m_e$  为电子质量;  $\omega_p$  为等离子体频率;  $e$  为电子电荷。我们进一步考虑离子不动的情形, 记  $f_e$  为电子密度函数,

则方程(2)可以写为

$$\nabla \cdot \mathbf{E} = \int f_c dv - 1 \quad (4)$$

记电子电势为  $\varphi$ , 则方程(4)写为

$$-\Delta\varphi = \int f_c dv - 1 \quad (5)$$

至此, 通过方程(1)与方程(5)的耦合, 可在一定初始条件及边界条件下, 求解函数  $f_c(x, v, t)$ 。

## 2 模型构建

### 2.1 问题的提出

为描述在神经算子的理论框架下求解 PDE 方程的问题, 我们定义算子  $\mathcal{L}$  为

$$(\mathcal{L}_a u)x = 0, \quad a \in \mathcal{A}, \quad u \in \mathcal{U} \quad (6)$$

式中:  $a$  为 PDE 方程参数(可以为物理参数、边界条件或初始条件);  $u$  为待求解的分布函数;  $\mathcal{A}$  是参数所在的函数空间, 算子  $\mathcal{L}$  是函数空间中的非线性映射。根据所要求解的 PDE 方程, 神经算子可以定义映射  $g: \mathcal{A} \rightarrow \mathcal{U}$ , 其中  $\mathcal{A}$  初始条件或 PDE 方程参数所在的函数空间, 而  $\mathcal{U}$  为 PDE 方程解的函数空间。

基于神经算子的 PDE 方程的离散求解问题可以按如此定义: 给定数据对  $(a_i, u_i)$ , 找到算子  $g$  的一个近似  $g^+$ , 使得对于任何新数据  $a' \in \mathcal{A}$ , 我们有  $g^+(a') \approx g(a')$ , 即

$$\min_{\mathbb{E}_{a \sim u}} \|g^+(a) - g(a)\|^2 \quad (7)$$

由此可知, 上述的最小化问题和深度学习的训练目标是一致的, 因此我们可以建立 Vlasov 方程的初始函数空间, 形成训练数据集, 通过一定的网络结构寻找算子的近似值, 进而逼近 PDE 方程的解, 实现对于某类初始条件的快速求解。不同于物理信息驱动神经网络方法, 神经算子的目的是学习函数空间到函数空间的映射关系, 而非针对单一 PDE 实例进行优化。神经算子框架可在任意离散化网格上训练与推理, 突破传统数值方法对固定网格的依赖, 且通过从参数分布中学习, 模型可泛化至未见过的初始条件或物理参数, 显著提升模型计算效率。

### 2.2 模型结构

典型的神经算子框架包括: 编码层 (Lifting layer)、迭代核积分层 (iteration kernel integration layer) 及映射层 (projection layer)。整体结构可以描述为

$$g_\theta := Q \circ \sigma_T(W_{T-1} + K_{T-1} + b_{T-1}) \circ \dots \circ P \quad (8)$$

式中:  $Q$ 、 $P$  分别表示编码层和投影层;  $W$  为线性变换算子;  $K$  为积分核算子;  $b$  为偏置项;  $\sigma_T$  为非线性激活函数。模型架构中, 主要通过积分核算子的卷积运算捕捉待求解 PDE 方程的全局信息, 其表达形式如下

$$K = \int_D \kappa(x, y) v_i(y) dy \quad (9)$$

式中:  $v_i(y)$  是输入函数;  $\kappa(x, y)$  是积分核函数;  $D$  为积分域。神经算子方法的核心在于如何处理积分核算子, 不同的处理方法可以衍生出诸如神经网络、卷积神经网络、低秩神经算子等。为了高效地计算积分核算子, 根据傅里叶积分的卷积定理, 空间域的卷积对应于频域的乘积

$$\mathcal{F}[(Kv)](k) = \mathcal{F}[\kappa](k) \cdot \mathcal{F}[v](k) \quad (10)$$

进一步将积分核函数转化为傅里叶空间中的可学习的权重, 同时假设核函数具有平移不变性, 即  $(\kappa(x, y) = \kappa(x - y))$ , 对公式(10)两边进行傅里叶逆变换

$$K = \mathcal{F}^{-1}(\mathcal{F}[\kappa](k) \cdot \mathcal{F}[v](k))(x) = \mathcal{F}^{-1}(R(k) \cdot \mathcal{F}[v](k))(x) \quad (11)$$

根据公式(8)及公式(11)本文所采用的傅里叶神经算子网络结构如图 1 所示。图 1 中 Vlasov 方程的初始分布作为网络输入, 输入数据  $f(x, v, t = 0)$  经过全连接层升维处理 (lifting) 后, 进行傅里叶变换, 将数据信息由实空间变换至频域空间, 通过一个线性层过滤掉高频模式后, 与多层感知机 (MLP) 的输出相加求和, 使用 Relu 等激活函数输出至下一层, 进一步处理, 最终输出求解区域的分布函数。

### 2.3 物理信息嵌入

Vlasov 方程作为碰撞项可忽略的守恒型动力学方程, 其解需严格满足质量、动量及能量守恒律。神经算子纯

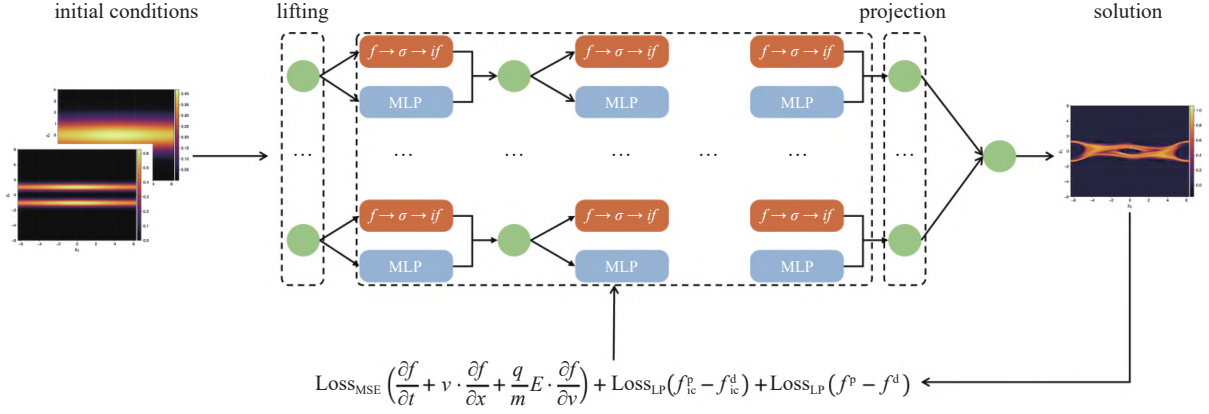


Fig. 1 Physics-informed Fourier neural operator network model

图 1 物理信息驱动的傅里叶神经算子网络模型

数据驱动的训练方式可能导致长时间预测中物理偏差累积,造成精度损失,尤其在训练数据有限的区域(如相空间低密度区)。为了有效约束神经网络的拟合行为,提高泛化能力,模型采用物理信息嵌入的混合损失函数,将物理定律显式编码为软约束项,以增强模型的物理现象表达能力。模型的总损失函数可以写为

$$\mathcal{L} = \mathcal{L}_{\text{data}} + \mathcal{L}_{\text{P}} = \mathcal{L}_{\text{data}} + \mathcal{L}_{\text{PDE}} + \mathcal{L}_{\text{IC}} + \mathcal{L}_{\text{P}} + \mathcal{L}_{\text{E}} + \mathcal{L}_{\text{q}} \quad (12)$$

式中:  $\mathcal{L}_{\text{data}}$  为训练数据的相对误差损失;  $\mathcal{L}_{\text{P}}$  为物理信息损失;  $\mathcal{L}_{\text{PDE}}$  为 Vlasov-Poisson 方程的损失;  $\mathcal{L}_{\text{IC}}$  为边界条件损失;  $\mathcal{L}_{\text{P}}$  为动量守恒损失;  $\mathcal{L}_{\text{E}}$  为总能量损失;  $\mathcal{L}_{\text{q}}$  为电荷量守恒损失。其中训练数据的损失依据相对误差均方根公式计算,即

$$\mathcal{L}_{\text{data}} = \frac{1}{N} \sum_{i,j,k=0}^n \frac{\|f'(x_i, v_j, t_k) - f(x_i, v_j, t_k)\|_2}{\|f(x_i, v_j, t_k)\|_2} \quad (13)$$

PDE 方程损失涉及到分布函数的微分及 Poisson 方程的耦合项处理,其中分布函数的微分可以借助神经网络的自动微分、数值差分方法及傅里叶变换三种方式实现。自动微分方法由于 FNO 网络结构的限制,需要建立点到点的映射,对计算资源的需求相比传统的神经网络更大;数值差分方法的精度依赖于网格点的选取,粗网格情况下截断误差明显;傅里叶变换方法可以直接在频域进行操作,且精度相对数值差分法更高,但需要周期性条件作为限制。综合考虑计算资源及精度要求,在本模型中,我们使用快速傅里叶变换方法直接计算输出函数的导数,计算公式如下

$$f'(x) = \mathcal{F}^{-1} [ik \hat{f}(k)](x) \quad (14)$$

对于一维相空间(1D1V)的 Vlasov-Poisson 方程,我们需要求解分布函数关于速度和空间位置的偏导,使用 PyTorch 提供的二维快速傅里叶变换方法,我们可以在复杂度  $O(n \log n)$  的条件下对该偏导数进行计算。

Vlasov 方程中耦合的电场项需要求解 Poisson 方程,对方程(5)的两边进行傅里叶变换,利用傅里叶变换的微分性质,可以推导出方程的解为

$$\varphi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} \frac{\hat{f}(k, t)}{k^2} e^{ikx} dk \quad (15)$$

将网格数据代入公式(15)即可计算出网格点上的电场。综上,物理模型部分的损失函数可以写为

$$\mathcal{L}_{\text{P}} = \frac{1}{N} \sum \frac{\partial f}{\partial t} + v \cdot \frac{\partial f}{\partial x} - \frac{\partial \varphi}{\partial x} \frac{\partial f}{\partial v} + \frac{1}{N} \sum \frac{f'_{\text{IC}} - f_{\text{IC}2}}{f_{\text{IC}2}} + \frac{1}{N} \sum q'_i - q_{i2} + \frac{1}{N} \sum E'_i - E_{i2} \quad (16)$$

等式右边的四项分别代表了 Vlasov 方程本身的残差损失、边界条件残差损失、电荷与能量守恒条件的残差损失,其中  $f'_{\text{IC}}$  和  $f_{\text{IC}2}$  分别代表边界上分布函数的预测值和仿真值,  $q$  为电荷数,  $E$  为总能量。

### 3 数值实验

数值实验部分将以朗道阻尼及双流不稳定性两个基本问题作为求解结果的基准,重点介绍神经算子训练数据集的建立及训练、评估结果。数据集的产生采用了 Gkey II 软件生成 Vlasov 方程的数值解,该软件是一个模块化



框架, 包含多种等离子体物理和流体动力学求解器。在 Gkey II 中, Vlasov-Maxwell 系统在空间上采用间断 Galerkin 有限元方法离散, 在时间上使用强稳定性格式进行 Runge-Kutta 方法离散, 每个时间步进行完全显式的更新<sup>[16]</sup>。此外, Gkey II 软件提供了高效并行化能力, 数据集的获取可在并行环境下进行。

### 3.1 朗道阻尼

1946 年, 朗道通过解析延拓和围道积分, 首次从 Vlasov-Poisson 方程中导出这一现象。可以说朗道阻尼是 Vlasov 方程的直接推论, 并在实验中得到证实。朗道阻尼表明即使没有碰撞, 等离子体仍可通过动力学过程耗散波动能, 该现象对等离子体稳定性(如约束聚变、空间物理)至关重要。朗道阻尼对于 Vlasov 方程的数值求解算法精度要求较高, 因为其本质上依赖于粒子分布函数中共振粒子的贡献, 为了捕捉这一效应, 必须在速度空间上采用非常细的网格划分。这不仅会显著增加计算量, 还可能引入数值噪声和数值耗散, 影响最终的阻尼率计算精度。另一方面, 随着时间演化, 粒子分布函数会出现丝化现象(Filamentation), 这种现象使得分布函数在相空间中变得极其复杂。如果数值网格分辨率不足, 难以准确捕捉这些细微结构, 从而导致朗道阻尼计算的误差增大, 因此, 对于朗道阻尼的计算往往作为评价 Vlasov 数值仿真软件精度的基本算例。朗道阻尼的初始分布函数为麦克斯韦分布

$$f_0 = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{v^2}{2v_i}\right) \cdot (1 + \alpha \cos kx) \quad (17)$$

式中:  $1 + \alpha \cos kx$  是扰动项;  $v_i$  是电子热速度;  $k$  为波数。

采用上述的初始分布函数, 其中  $k = 0.5$ , 扰动振幅在  $[0, 0.5]$  的区间内采样 200 个点, 空间求解域为  $[-\pi/k, \pi/k]$ , 速度求解域为  $[-6, 6]$ , 时间范围为  $[0, 32]$ , 网格分辨率为  $128 \times 128 \times 65$ , 采用 Gkey II 生成训练数据, 使用的模型数据如表 1 所示。

训练过程中的损失函数变化情况如图 2 所示。从图 2 中

表 1 模型与训练参数

Table 1 Model and training parameters

fully connected layer	hidden layers	mode number in x direction	mode number in v direction	learning rate
128	5×32	12	12	0.001

可以看出加权后的训练损失接近  $6.65 \times 10^{-3}$ , 而训练后的模型在验证集上的相对误差为 0.31%。为进一步验证傅里叶神经算子的计算准确性, 考虑  $\alpha = 0.01$  的线性朗道阻尼情况, 由分布函数可以计算得到电场能量密度, 数值计算的电场能量密度衰减与神经网络预测的电场能量密度衰减对比如下图 3 所示, 其中电场强度均使用  $E_0 = -\omega_p c m_e / e$

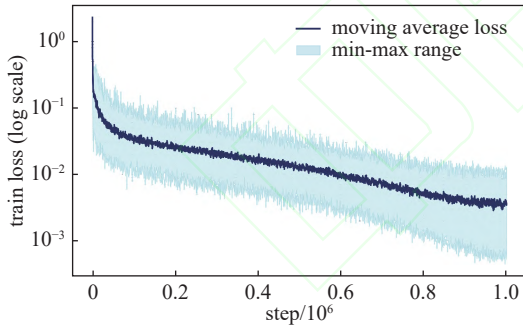


Fig. 2 The training loss of the Landau damping dataset as a function of iteration steps

图 2 朗道阻尼训练集的损失函数随迭代步的变化

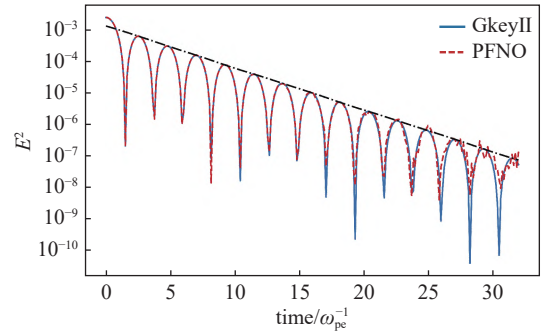


Fig. 3 Decay of the electric field energy density in linear Landau damping

图 3 线性朗道阻尼中的电场能量密度衰减

进行无量纲化。

对于  $\alpha$  非常小的情况(线性朗道阻尼), 可利用围道积分推导理论的衰减率为 0.1533。进一步对电场能量密度进行拟合, 我们可以计算出数值仿真的衰减率为 0.1555, 而通过傅里叶算子模型输出的分布函数计算的衰减率为 0.1528, 均十分接近理论值, 证明 PFNO 模型的计算精度能够保证基本的物理现象。对于非线性朗道阻尼情况, 同样计算了等离子体电场能量密度随时间演化的物理图像, 如图 4 所示。可以看到, 电场能量密度随指数衰减, 然后进入增长阶段, PFNO 模型预测的电场能量密度基本与数值结果保持一致, 通过拟合得到非线性朗道阻尼前一段的衰减率为  $\gamma_1 = 0.2837$ , 后一阶段的增长率为  $\gamma_2 = 0.703$ 。

### 3.2 双束不稳定性

双束不稳定性(Two-stream Instability)是等离子体物理和带电粒子束动力学中的一种重要现象, 即两个带电粒

子束在相互作用时,由于电磁场的耦合或共振效应导致系统能量重新分配,从而引发扰动增长、束流发散或能量损失。其中描述对称双束不稳定性的初始分布函数为

$$f(x, v, 0) = \frac{1}{2\sqrt{2\pi} \cdot v_{th}} \left[ \exp\left(-\frac{(v-u)^2}{2v_{th}^2}\right) + \exp\left(-\frac{(v+u)^2}{2v_{th}^2}\right) \right] \cdot [1 + \alpha \cos(kx)] \quad (18)$$

式中:  $v_{th}$  为热运动速度;  $u$  为漂移速度。设空间求解域为  $[-\pi/k, \pi/k]$ , 速度求解域为  $[-6, 6]$ , 时间范围为  $[0, 32]$ , 波数  $k = 0.5$ , 电子热运动速度、热速度与漂移速度之比、扰动大小分别在  $[0.1, 1]$ ,  $[0.5, 5]$ ,  $[0, 0.5]$  的区间范围随机采样, 获取 1405 个训练样本, 模型参数及训练参数如表 2 所示。

表 2 模型与训练参数

Table 2 Model and training parameters

fully connected layer	hidden layers	mode number in $x$ direction	mode number in $v$ direction	learning rate
128	5×64	12	12	0.0005

网格分辨率为  $128 \times 128 \times 65$ , 训练过程中的损失函数及验证集相对误差损失变化如图 5 所示。

从图 5 中可以看到, 训练集上的加权损失函数下降至 0.087, 验证集上数据的相对误差为 5.90%。为测试模型的泛化能力, 分别选取  $v_{th} = 0.2$ ,  $u = 1.0$ ,  $\alpha = 0.05$  (案例 1), 以及  $v_{th} = 0.15$ ,  $u = 0.6$ ,  $\alpha = 0.15$  (案例 2) 的情况下分布函数的演化情况如图 6 所示。

对于分布函数, 案例 1 和案例 2 的相对误差分别是 6.12% 和 5.09%, 从图 6 中可以看到, PFNO 模型很好的捕捉了分布函数的全局演化形式, 而误差则主要发生在分布函数内部的

涡旋部位。为阐明双束不稳定性的发生情况, 对案例 1 绘出电场能量随时间的变化关系如图 7 所示。

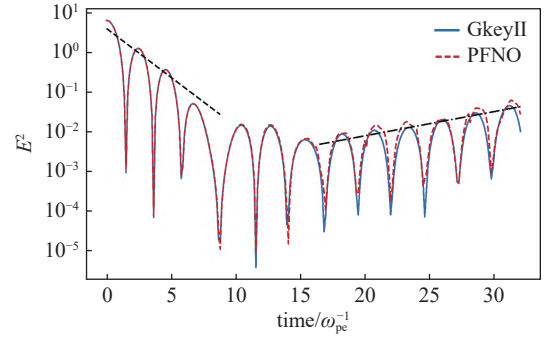


Fig. 4 The temporal evolution of plasma electric field energy density in nonlinear Landau damping

图 4 非线性朗道阻尼中等离子体电场能量密度随时间的演化

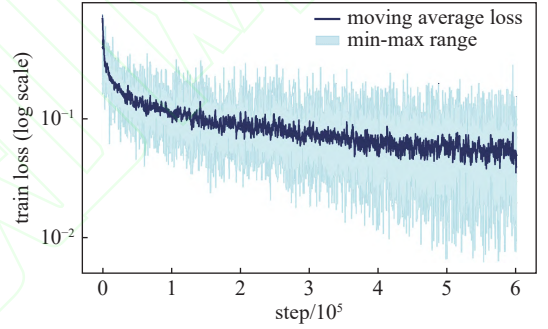


Fig. 5 The training loss of the two-stream instability dataset as a function of iteration steps

图 5 双束不稳定性训练集的训练损失随迭代步的变化

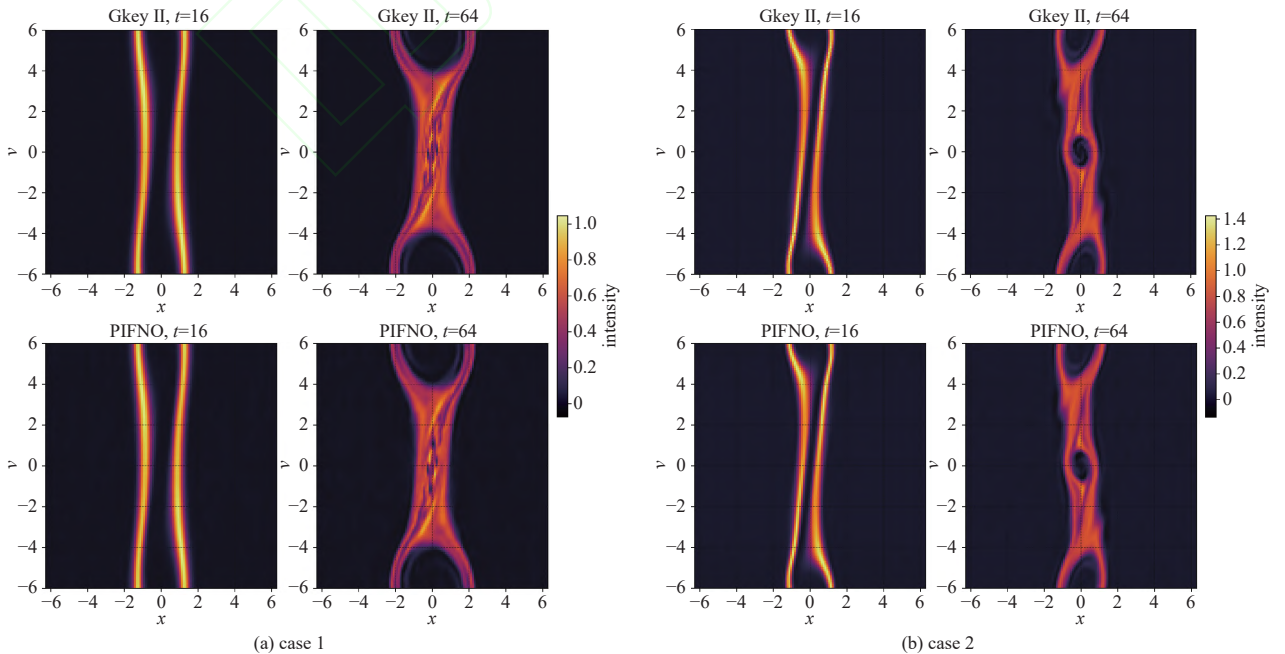


Fig. 6 Prediction results of two-stream instability

图 6 双束不稳定性分布函数的预测结果

从图 7 中可以看到, 双束不稳定系统中电场能量密度在第一阶段以指数形式增长, 随后进入较为平稳的状态, PFNO 模型的求解结果与数值计算软件保持一致。

### 3.3 讨论与分析

从以上朗道阻尼及双束不稳定性两例中可以看到, 对于某一类初始函数分布, PFNO 模型可以很好地预测分布函数的形式, 其精度与传统数值方法可比拟。在计算速度方面, 由于仅需一次前向传播, PFNO 模型的计算速度一般在 ms 量级, 而传统的数值方法需要在时间和空间上进行多次迭代求解, 耗时分钟到数小时不等。在参数空间选取 25 个不同的初始分布, 使用不同模型参数量计算, PFNO 的相对

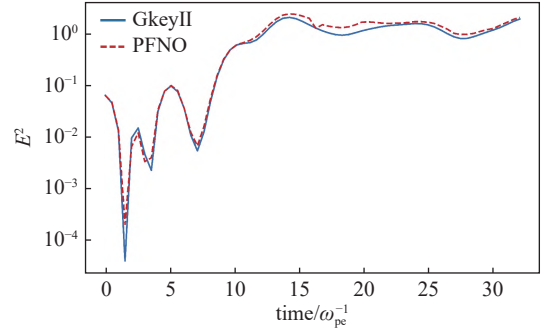


Fig. 7 The evolution of electric field energy density in Two-Stream instability

图 7 双束不稳定性中的电场能量密度变化过程

表 3 PFNO 计算精度与耗时对比

Table 3 Comparison of PFNO calculation accuracy and time consumption

resolution	number of parameters	PFNO $L^2$ error/%	time consumption by Gkey II/s	time consumption by PFNO/s	speed up
128×128	67 M	9.09	450	3.27	137.61
256×256	67 M	8.74	1 700	11.24	151.25
128×128	113 M	6.16	450	4.14	108.70
256×256	113 M	6.02	1 700	14.63	116.20

误差与计算耗时如表 3 所示。可以看到, 提高分辨率并未对 PFNO 模型的预测精度产生明显影响, 计算速度基本保持线性关系。

## 4 结 论

本文我们采用 PFNO 模型对 Vlasov 方程进行了求解, 讨论了一维情况下朗道阻尼及双束不稳定性两个基本算例的基准测试, 分别达到了 0.31% 和 5.90% 的相对  $L^2$  误差。数值结果表明 PFNO 模型在一定的精度范围内, 可对分布函数的演化形式进行快速求解, 计算时间相比传统数值方法缩短百倍, 尤其适用于需要多次参数扫描的场景。虽然神经算子可学习某一族函数的映射, 但该模型仍然一定程度上依赖高保真模拟数据集, 可进一步采用主动学习策略以减少数据需求。

### 参考文献:

- [1] Chen, Francis F. Introduction to plasma physics and controlled fusion [M]. Vol. 1. New York: Plenum Press, 1984.
- [2] Bardos, C, Anne N. A Vlasov equation with Dirac potential used in fusion plasmas [J]. Journal of Mathematical Physics 53, 2012, no. 11.
- [3] Palmroth, M., Ganse, U., et al. Vlasov methods in space physics and astrophysics [J]. Living Reviews in Computational Astrophysics, 2018, 4(1), p. 1.
- [4] Bertrand P, Albrecht-Marc M, et al. Vlasov models for laser-plasma interaction [J]. Transport Theory and Statistical Physics. 2005 Jan 1;34(1-2): 103-26.
- [5] Fijalkow, Eric. A numerical solution to the Vlasov equation [J]. Computer Physics Communications, 1999, 116. 2-3: 319-328.
- [6] Birdsall, C. K., et al. Plasma physics via computer simulation [M]. 2018.
- [7] Murawski K, Numerical solutions of magnetohydrodynamic equations [J]. Bulletin of the Polish Academy of Sciences-Technical Sciences, 2011, 59(2): 219-226.
- [8] Sonnendrücker, Eric, et al. The semi-Lagrangian method for the numerical resolution of the Vlasov equation [J]. Journal of Computational Physics, 1999, 149(2): 201-220.
- [9] Roska, Tamás, Leon O. Chua, et al. Simulating nonlinear waves and partial differential equations via CNN. I. Basic techniques [J]. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 2002(10): 807-815.
- [10] Kovachki, Nikola, Zongyi Li, et al. Neural operator: learning maps between function spaces with applications to PDES [J]. Journal of Machine Learning Research, 2023, 24(89): 1-97.
- [11] Raissi, M., Paris P., et al. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations [J]. Journal of Computational Physics, 2019, 378 686-707.
- [12] Zhang, Pan, et al. A Maxwell's equations based deep learning method for time domain electromagnetic simulations [J]. IEEE Journal on Multiscale and Multiphysics Computational Techniques, 2021, 6: 35-40.
- [13] Eivazi, Hamidreza, Mojtaba Tahani, et al. Physics-informed neural networks for solving Reynolds-averaged Navier-Stokes equations [J]. Physics of Fluids, 2022(7): 34.

- [14] Raissi, Maziar. Deep hidden physics models: deep learning of nonlinear partial differential equations[J]. Journal of Machine Learning Research, 2018(25): 1-24.
- [15] Qin, Yilan, Jiayu Ma, et al. Data-driven modeling of Landau damping by physics-informed neural networks. Physical Review Research, 2023, no. 3: 033079.
- [16] Juno, James, Ammar Hakim, et al. Discontinuous Galerkin algorithms for fully kinetic plasmas. Journal of Computational Physics 2018, 353: 110-147.

