

A Comparative Study of Loss-Level Conservation Constraints in PINNs and Scheme-Level Structure-Preserving Approaches for Hamiltonian Systems

Xiwen Chen *

School of Ocean Engineering, Harbin Institute of Technology, Weihai, Shandong, 264209, China

* Corresponding author Email: cxw02022005@163.com

Abstract: Physics-Informed Neural Networks (PINNs) provide a neural network-based, mesh-free computational method for solving partial differential equations (PDEs) by incorporating PDE residuals, initial conditions, and boundary conditions into the loss function. However, standard PINNs may still suffer from issues such as conservation drift, insufficient training stability, and difficulty in preserving physical structure when dealing with long-term predictions and complex nonlinear problems. Centered on the core concept of “structure preservation” in scientific computing, this paper systematically compares methods for enhancing physical conservation constraints (Conservation-PINN) in PDE solving with methods for preserving geometric structure (symplectic integrators) in the evolution of Hamiltonian systems. This paper selects the heat conduction equation, the linear advection equation, the Burgers equation, and the double-pendulum Hamiltonian system as test cases to conduct a comparative analysis of the basic PINN, Res-PINN, Conservation-PINN, as well as the RK4 method and various symplectic integration methods (including Implicit Midpoint and Gauss-Legendre IRK4). The results show that introducing a conservation loss term can reduce mass drift to some extent during the solution of partial differential equations, and residual connection can improve the training stability of deep PINNs; in Hamiltonian systems, high-order non-symplectic methods (RK4) exhibit excellent local accuracy, while symplectic integrators, by preserving the system’s symplectic structure, can effectively suppress cumulative energy divergence during long-term evolution. This paper provides a reference for understanding the connections and differences between loss function constraints in scientific machine learning and classical geometric numerical integration.

Keywords: Physical Neural Networks; Structure Preservation; Conservation Constraints; Hamiltonian Systems; Symplectic Structure; Energy Boundedness.

1. Introduction

Partial differential equations (PDEs) are fundamental mathematical tools for describing physical phenomena in fields such as fluid mechanics, heat transfer, electromagnetics, and materials science. Traditional numerical methods for PDEs primarily include the finite difference method (FDM), the finite volume method (FVM), and the finite element method (FEM), which have formed a mature theoretical framework in engineering computations; however, they still face certain challenges in areas such as mesh generation for complex geometries, solving high-dimensional problems, and handling inverse problems.

In recent years, Physics-Informed Neural Networks (PINN), proposed by Raissi et al., have become a hot research topic in the field of scientific machine learning [1]. PINN uses neural networks to approximate unknown solutions and employs automatic differentiation techniques to compute residuals of partial differential equations, integrating control equations, initial conditions, and boundary conditions into a unified loss function for optimization. Compared to traditional grid-based discretization methods, PINN avoids the explicit grid construction process and demonstrates unique advantages in scenarios such as direct problem solving, inverse problem identification, and multi-source data fusion [2–3].

However, researchers have gradually discovered in practice that standard PINNs may suffer from limitations such as gradient optimization difficulties, training instability, and insufficient physical constraints when applied to complex

nonlinear problems [4–6]. To address these issues, existing research has primarily focused on two aspects: optimizing network architecture and enhancing physical constraints. On the one hand, researchers have improved the optimization topology of network training by introducing residual connections (ResNet), adaptive activation functions, and multiscale Fourier features [5]; on the other hand, they have enhanced the physical consistency of model predictions by explicitly incorporating global physical laws—such as mass conservation and energy conservation—into the loss function [7].

In addition to conservation constraints in PDEs, for dynamical systems with special geometric structures (such as Hamiltonian systems), preserving the geometric structure of the system’s phase space (e.g., symplectic structure) is key to ensuring the reliability of long-term numerical simulations [8–10]). Although traditional non-symplectic Runge-Kutta methods (such as the explicit RK4) possess high local truncation accuracy, they fail to preserve the symplectic structure and are thus prone to unphysical energy drift after long-term cumulative integration. Symplectic integrators, by preserving the conservation of phase-space area, ensure that the Hamiltonian error remains bounded and oscillates over long-term evolution [8].

In current research, machine learning methods—represented by PINN—primarily introduce physical rules via “soft constraints” at the loss-function level, whereas classical geometric numerical analysis preserves system properties through “hard structure” design at the scheme level. This paper systematically compares the mechanisms of action and

performance characteristics of these two approaches to structure preservation across different physical scenarios. The main contributions of this paper are as follows:

1. We establish a unified mathematical framework for the basic PINN, Res-PINN, and Conservation-PINN;
2. Using the linear advection equation and the Burgers equation as examples, we conduct an in-depth analysis of the mechanisms by which conservation loss weights influence global mass drift and local residual optimization;
3. Quantitatively compared the long-time evolution characteristics and step size sensitivity of non-symplectic methods (RK4) and multi-order symplectic integrators (Symplectic Euler, Störmer-Verlet, Implicit Midpoint, Gauss-Legendre IRK4) in the double-pendulum Hamiltonian system;
4. We clarified the fundamental differences and applicability boundaries between “loss-level soft constraints” and “scheme-level geometric preservation” in terms of convergence, local accuracy, and long-term stability.

2. Related Work

2.1. Development and Architectural Evolution of Physics-Informed Neural Networks (PINN)

Since Raissi et al. [1] integrated deep learning with physical laws to propose Physics-Informed Neural Networks (PINNs), PINNs have found widespread application in various scientific computing fields, including fluid dynamics [2], heat conduction [3], and quantum mechanics [6]. However, standard PINNs are highly prone to vanishing or exploding gradients when training deep networks, which limits their ability to approximate complex flow fields and high-frequency oscillatory solutions [4]. To improve the optimization topology of neural networks, Jagtap et al. [5] proposed adaptive activation functions, enhancing convergence speed and approximation accuracy; Wang et al. [11] introduced Fourier feature mapping to overcome the “spectral bias” phenomenon in neural networks when learning high-frequency functions. Furthermore, the application of the ResNet architecture in PINNs (Res-PINN) has been shown to improve the smoothness of the loss landscape [5,12], thereby supporting the stable training of deeper networks. Although the aforementioned architectural optimization techniques have significantly enhanced the network’s fitting capabilities, they essentially remain data- and residual-driven optimization mechanisms that cannot fundamentally guarantee the strict conservation of physical quantities [13].

2.2. Physical Conservation Constraints and Loss Function Optimization in Scientific Machine Learning

To address the insufficient fitting capability of standard PINNs under complex constraints, researchers have proposed various improvement strategies. In terms of domain decomposition and parallelization, Jagtap et al. [14] proposed cPINN (conservative PINN), which achieves continuity of cross-domain solutions and conservation of physical quantities by applying conservation conditions at subdomain boundaries. For systems with hard boundary or algebraic equation constraints, Bischof and Kraus [15] proposed an architecture that directly satisfies boundary conditions through output layer transformations, hard-coding the

constraints into the network structure; similarly, Jagadish et al. [16] explored strategies for embedding algebraic manifold constraints into the output space of neural networks. To balance the competition between gradients of multiple loss terms, Wang et al. [17] proposed a gradient pathology mitigation mechanism and an adaptive weight adjustment algorithm [18], which significantly improved training stability under composite physical constraints. Furthermore, in the face of data noise and uncertainty, deep learning has demonstrated exceptional robustness in inverse problem solving and parameter identification [19, 20].

2.3. Traditional Hamiltonian Systems and Geometric Structure Preservation Algorithms

For Hamiltonian systems, a well-established theory of geometric numerical integration has been developed in the field of traditional numerical analysis [21]. Hairer et al. [22] and Sanz-Serna and Calvo [23] have detailed the construction principles of symplectic integrators, proving that symplectic schemes can accurately preserve the system’s symplectic structure and phase space volume during long-time numerical integration, thereby avoiding non-physical energy dissipation or blow-up. Leimkuhler and Reich [24] further generalized geometric integrators to constrained Hamiltonian systems and molecular dynamics simulations. These classical geometric numerical methods provide a solid mathematical foundation for the design of modern structure-preserving neural networks [25].

2.4. Cutting-Edge Breakthroughs in Structure-Preserving Neural Networks (Structure-Preserving NNs)

Inspired by classical chiral-preserving algorithms and combining deep learning with geometric mechanics, “structure-preserving neural networks” have become a hot research topic in recent years. Chen et al. [10] proposed Neural ODE, introducing continuous-time dynamics into deep models. Greydanus et al. [9] proposed the Hamiltonian Neural Network (HNN), which naturally preserves the system’s energy conservation by learning the system’s Hamiltonian rather than directly fitting the vector field. Cranmer et al. [8] extended this idea to the Lagrangian form, proposing the Lagrangian Neural Network (LNN), which eliminates the strict dependence on regular coordinates. To further introduce algebraic constraints, Zhong et al. [26] proposed SymODEN, which enables the learning of symplectic dynamics on constrained manifolds. Jin et al. [27] and Chen et al. [28] respectively constructed symplectic recurrent neural networks (Symplectic RNNs/Networks) to achieve structure preservation at discrete time steps. Lutter et al. [29] successfully applied structure-preserving networks to robot control and the modeling of complex mechanical systems.

3. Methodology

3.1. Basic PINN Framework

Consider a PDE of general form:

$$N[u](x,t)=0, (x,t) \in \Omega \times [0,T] \quad (1)$$

where $u(x,t)$ is the function to be solved, and N is the differential operator containing time and spatial derivatives. PINN uses a neural network $u_\theta(x,t)$ to approximate the true

solution $u(x, t)$, where θ denotes the network parameters.

The loss function for a standard PINN typically consists of three parts:

$$L_{PINN} = L_{IC} + L_{BC} + L_{PDE} \quad (2)$$

where the initial value loss is:

$$L_{IC} = \frac{1}{N_{IC}} \sum_{i=1}^{N_{IC}} |u_\theta(x_i, 0) - u_0(x_i)|^2 \quad (3)$$

The boundary loss is:

$$L_{BC} = \frac{1}{N_{BC}} \sum_{i=1}^{N_{BC}} |u_\theta(x_i, t_i) - g(x_i, t_i)|^2 \quad (4)$$

The PDE residual loss is:

$$L_{PDE} = \frac{1}{N_f} \sum_{i=1}^{N_f} |N[u_\theta](x_i, t_i)|^2 \quad (5)$$

The derivative terms in neural networks are computed using automatic differentiation. For example, for the heat conduction equation:

$$u_t = \alpha u_{xx}, \quad (6)$$

Its PDE residual can be written as:

$$f_\theta(x, t) = \frac{\partial u_\theta}{\partial t} - \alpha \frac{\partial^2 u_\theta}{\partial x^2} \quad (7)$$

The corresponding residual loss is:

$$L_{PDE} = \frac{1}{N_f} \sum_{i=1}^{N_f} |f_\theta(x_i, t_i)|^2 \quad (8)$$

3.2. Residual Network PINN

As the number of network layers increases, conventional fully connected PINNs may suffer from gradient propagation difficulties and training instability [4]. Res-PINN introduces residual connections into the network architecture, and its basic residual block can be expressed as:

$$h_{l+1} = h_l + F(h_l; \theta_l) \quad (9)$$

where h_l represents the features of layer l , and F is a mapping composed of several linear layers and nonlinear activation functions.

The primary function of residual connections is to alleviate the optimization difficulty in deep networks, making it easier for the network to learn identity mappings or local perturbations. In PINN, Res-PINN typically does not alter the form of the physical constraints; that is, the loss function remains:

$$L_{ResPINN} = L_{IC} + L_{BC} + L_{PDE} \quad (10)$$

Therefore, the structural preservation capability of Res-PINN does not stem from new physical constraints, but rather from more stable function approximation capabilities and better training performance.

3.3. PINN with Conservation Constraints

For many PDE systems, in addition to residuals of local differential equations, global conserved quantities also exist. Take the one-dimensional linear advection equation as an example:

$$u_t + cu_x = 0 \quad (11)$$

Under periodic boundary conditions, the total mass

$$M(t) = \int_{\Omega} u(x, t) dx \quad (12)$$

should remain constant, that is:

$$M(t) = M(0) = M_0 \quad (13)$$

A standard PINN constrains only the local PDE residuals and does not guarantee strict conservation of the $M(t)$ in the

discrete sense. Therefore, a conservation constraint term can be added to the loss function:

$$L_{cons} = \frac{1}{N_t} \sum_{j=1}^{N_t} |M_\theta(t_j) - M_0|^2 \quad (14)$$

where:

$$M_\theta(t_j) \approx \sum_{k=1}^{N_x} u_\theta(x_k, t_j) \Delta x \quad (15)$$

The total loss function for Conservation-PINN is:

$$L_{CPINN} = L_{IC} + L_{BC} + L_{PDE} + \lambda_{cons} L_{cons} \quad (16)$$

where λ_{cons} is the conservation loss weight, used to balance the local PDE residuals with the global conservation constraint.

For the Burgers equation:

$$u_t + uu_x = \nu u_{xx} \quad (17)$$

It can also be written in conservation form:

$$u_t + \left(\frac{u^2}{2}\right)_x = \nu u_{xx} \quad (18)$$

Under periodic boundary conditions, if the boundary fluxes of the diffusion term cancel each other out, the total mass still satisfies:

$$\frac{d}{dt} \int_{\Omega} u(x, t) dx = 0 \quad (19)$$

Therefore, a mass conservation loss term can also be added to the Burgers equation to reduce the drift of conserved quantities in long-term forecasts.

3.4. Symplectic Integration

A Hamiltonian system is typically written as:

$$\dot{q} = \frac{\partial H}{\partial p}, \dot{p} = -\frac{\partial H}{\partial q} \quad (20)$$

where q are the generalized coordinates, p is the generalized momentum, and $H(q, p)$ is the Hamiltonian. For a system with no external forces and no dissipation, the Hamiltonian typically represents the total energy of the system and satisfies:

$$\frac{dH}{dt} = 0 \quad (21)$$

The core idea of a symplectic integrator is to preserve the symplectic structure in the phase space of a Hamiltonian system. Intuitively, a symplectic integrator does not necessarily minimize the energy error at every step, but it preserves the system's geometric structure, causing the energy error to exhibit bounded oscillations over a long period rather than a continuous drift.

This paper considers four typical symplectic integrators:

Symplectic Euler method:

$$p_{n+1} = p_n - \Delta t \frac{\partial H}{\partial q}(q_n, p_{n+1}) \quad (22)$$

$$q_{n+1} = q_n + \Delta t \frac{\partial H}{\partial p}(q_n, p_{n+1}) \quad (23)$$

Störmer-Verlet Method:

$$p_{n+\frac{1}{2}} = p_n - \frac{\Delta t}{2} \frac{\partial H}{\partial q}(q_n) \quad (24)$$

$$q_{n+1} = q_n + \Delta t \frac{\partial H}{\partial p}(p_{n+\frac{1}{2}}) \quad (25)$$

$$p_{n+1} = p_{n+\frac{1}{2}} - \frac{\Delta t}{2} \frac{\partial H}{\partial q}(q_{n+1}) \quad (26)$$

Implicit Midpoint Method:

$$z_{n+1} = z_n + \Delta t J^{-1} \nabla H\left(\frac{z_n + z_{n+1}}{2}\right) \quad (27)$$

where $z = (q, p)$ and J are standard symmetric matrices. This method is an implicit symmetric formulation, which generally exhibits good stability but requires solving a nonlinear equation at each step.

In a Hamiltonian system, the system's state is represented by phase-space coordinates $z = (q, p)^T \in \mathbb{R}^{2d}$, where q and p are the generalized coordinates and generalized momentum, respectively. The Hamiltonian canonical equations can be expressed as:

$$\dot{z} = J \nabla H(z) \quad (28)$$

where J is a $2d \times 2d$ standard antisymmetric symplectic matrix,

$$J = \begin{pmatrix} O_d & I_d \\ -I_d & O_d \end{pmatrix} \quad (29)$$

where O_d and I_d denote the d -dimensional identity matrix and zero matrix, respectively. For a single-degree-of-freedom system ($d=1$),

$$J = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \quad (30)$$

Gauss-Legendre IRK4 method:

$$K_1 = f(t_n + (\frac{1}{2} - \frac{\sqrt{3}}{6})\Delta t, y_n + \Delta t(\frac{1}{4}K_1 + (\frac{1}{4} - \frac{\sqrt{3}}{6})K_2)) \quad (31)$$

$$K_2 = f(t_n + (\frac{1}{2} + \frac{\sqrt{3}}{6})\Delta t, y_n + \Delta t((\frac{1}{4} + \frac{\sqrt{3}}{6})K_1 + \frac{1}{4}K_2)) \quad (32)$$

$$y_{n+1} = y_n + \Delta t(\frac{1}{2}K_1 + \frac{1}{2}K_2) \quad (33)$$

4. Experiments

4.1. Heat Conduction Equation: Basic PINN and Res-PINN

Consider the one-dimensional heat conduction equation:

$$u_t = \alpha u_{xx}, x \in [0, 1], t \in [0, T] \quad (34)$$

The objective of this experiment is to compare the training error and prediction accuracy of the baseline PINN and Res-PINN. Evaluation metrics include relative L^2 error and training loss curves.

The relative L^2 error is defined as:

$$E_{L^2} = \frac{\|u_\theta - u_{ref}\|_2}{\|u_{ref}\|_2} \quad (35)$$

Table 1. Comparison of Experimental Results for the Heat Conduction Equation

Method	Network Structure	Relative Error Compared to L^2	Final Training Loss	Number of Convergence Iterations
Baseline PINN	MLP	9.5561e-05	1.3343e-06	11,700
Res-PINN	ResMLP	6.3097e-05	7.1947e-07	12,263

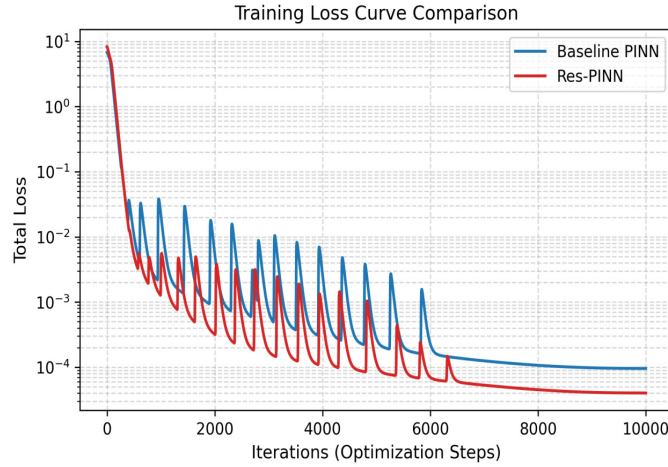


Figure 1. Comparison of training loss curves between the baseline PINN and Res-PINN

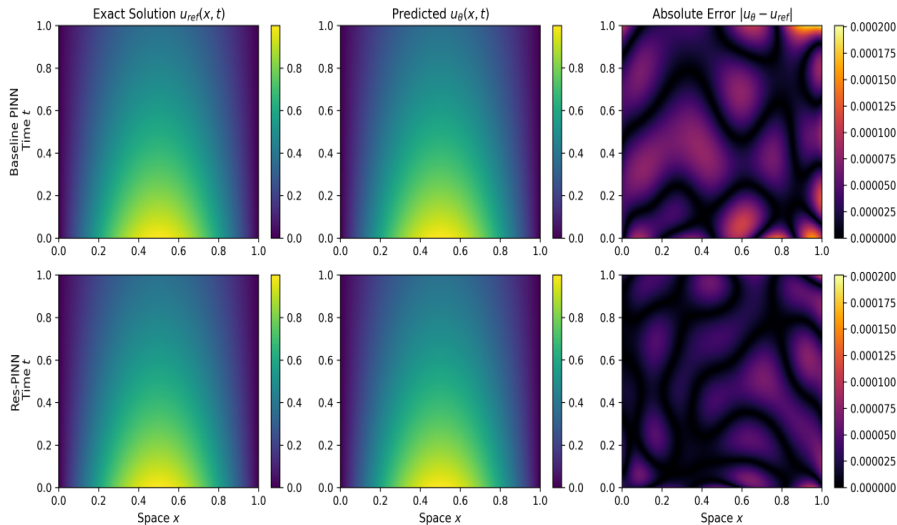


Figure 2. Comparison of the reference solution to the heat conduction equation and the neural network's predicted solution

4.2. Linear Advection Equation: Standard PINN vs. Conservation-PINN

Consider the linear advection equation:

$$u_t + cu_x = 0 \quad (36)$$

with periodic boundary conditions. The theoretical mass conservation equation for this problem is:

$$M(t) = \int_0^1 u(x,t) dx \quad (37)$$

We experimentally compare the mass drift of the standard PINN and the Conservation-PINN in long-term forecasts. Mass drift is defined as:

$$D_M(t) = |M_\theta(t) - M_0| \quad (38)$$

Table 2. Comparison of Experimental Results for the Linear Advection Equation

Method	Relative Error Compared to L^2	Maximum Mass Drift	Average Mass Drift	Long-Term Stability
Standard PINN	$1.5908\text{e-}01 \pm 1.54\text{e-}02$	$8.0327\text{e-}03 \pm 1.09\text{e-}03$	$4.1408\text{e-}03 \pm 1.08\text{e-}03$	General
Conservation-PINN	$1.3060\text{e-}02 \pm 3.32\text{e-}03$	$2.7975\text{e-}07 \pm 2.59\text{e-}08$	$7.0410\text{e-}08 \pm 4.39\text{e-}09$	Excellent

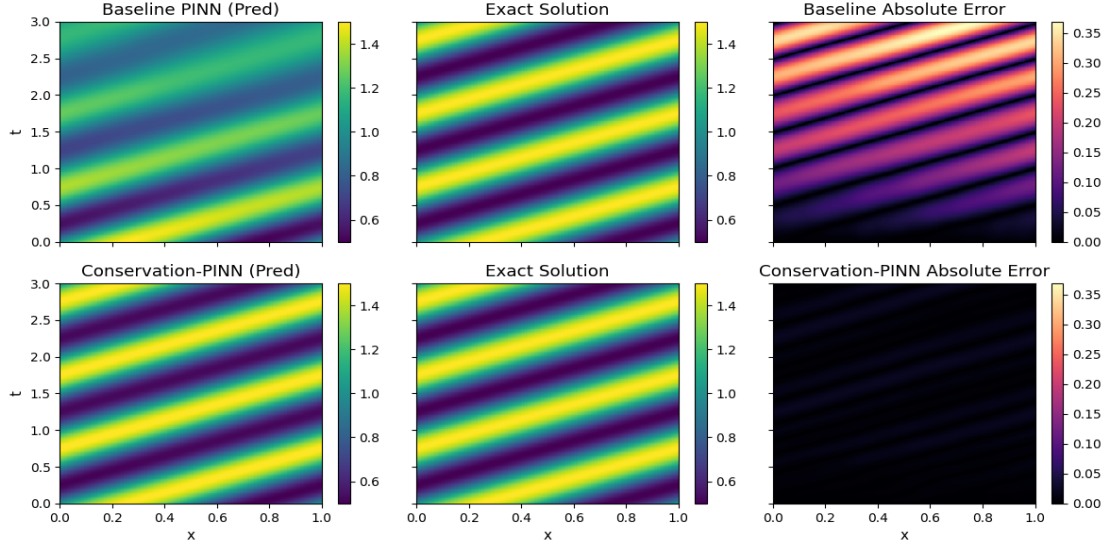


Figure 3. Comparison of the evolution of predicted solutions over time using different methods

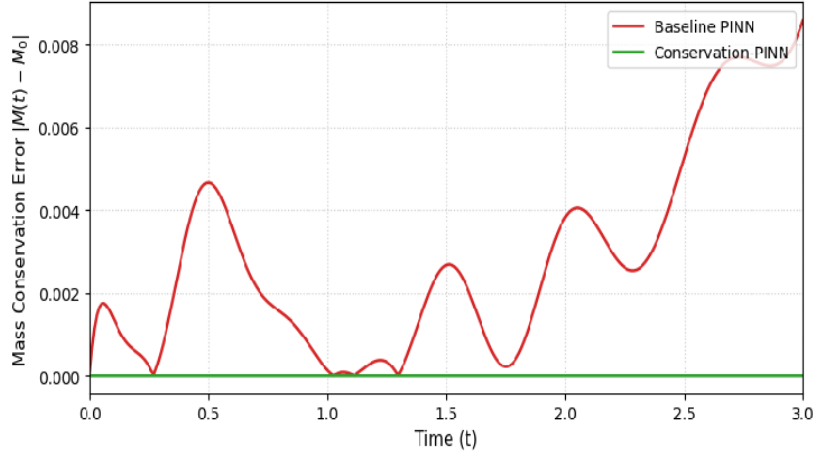


Figure 4. Mass drift curves for the standard PINN and Conservation-PINN

4.3. Burgers Equation: Conservation-PINN

Consider the viscous Burgers equation:

$$u_t + uu_x = \nu u_{xx} \quad (39)$$

Table 3. Comparison of Results for Different λ_{cons} s in the Burgers Equation

λ_{cons}	Relative L^2 Error	Maximum Mass Drift	PDE Residual Loss	Training Stability
0	6.38e-02	2.05e-03	3.85e-04	Minor drift
0.1	7.19e-02	3.27e-03	5.41e-04	Minor drift
1	4.54e-02	3.51e-04	1.45e-04	Optimal
10	5.96e-02	3.74e-04	3.16e-04	Training Oscillations

Under periodic boundary conditions, the system satisfies mass conservation. The experiment focuses on examining the impact of Conservation-PINN on solution field errors and conserved quantity drift for the Burgers equation, and analyzes the sensitivity of the λ_{cons} .

4.4. Double-Pendulum Hamiltonian System: RK4 and Symplectic Integrator

The double-pendulum system's Hamiltonian can be derived from the Lagrangian and expressed as the canonical Hamilton equations:

$$\dot{q} = \frac{\partial H}{\partial p}, \dot{p} = -\frac{\partial H}{\partial q} \quad (40)$$

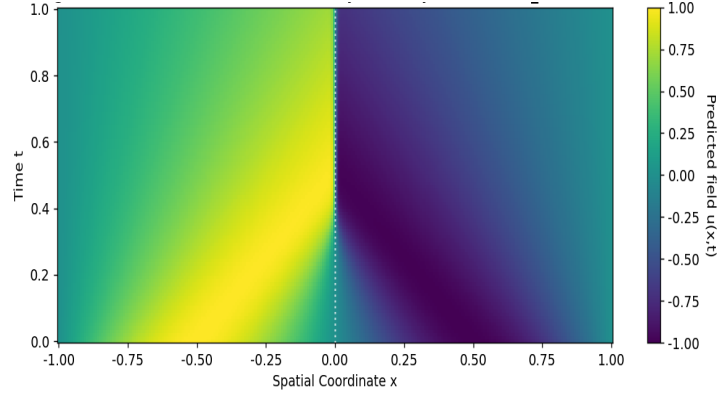


Figure 5. Reference solution for the Burgers equation and Conservation-PINN predicted solution

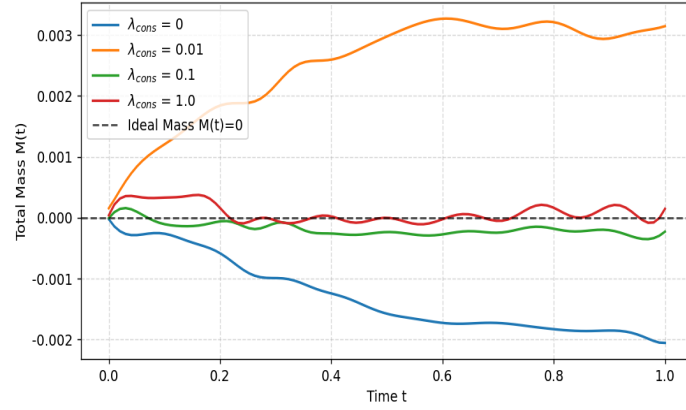


Figure 6. Mass drift curves under different λ_{cons} s

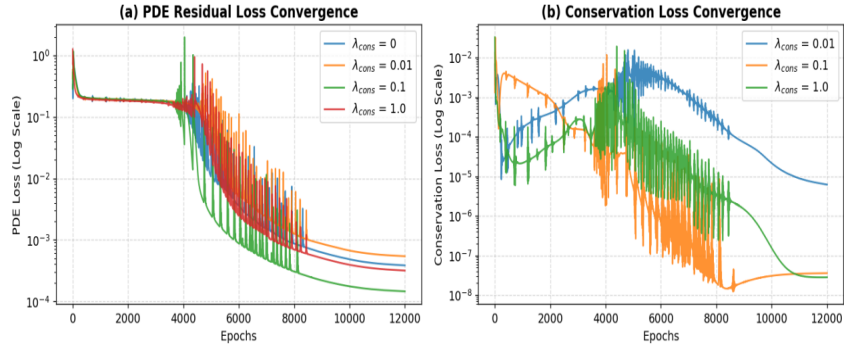


Figure 7. Variations in PDE residual loss and conservation loss under different λ_{cons}

In this paper, we use automatic differentiation to compute the partial derivatives of the Hamiltonian with respect to the q and the p , thereby obtaining the right-hand side of the system. We experimentally compare the Hamiltonian drift of RK4, Symplectic Euler, Störmer-Verlet, and Implicit Midpoint

methods during long-term simulations.

Hamiltonian drift is defined as:

$$D_H(t) = |H(q(t), p(t)) - H(q(0), p(0))| \quad (41)$$

Table 4. Comparison of Experimental Results for the Double-Pendulum Hamiltonian System

Method	Order of Integration	Structural Properties	Time Step Δt	Maximum Energy Drift	Average Energy Drift	Long-term stability	Computational Cost (s)
RK4	4th-order	Non-symplectic (Non-Symplectic)	0.010	3.0631e-05	1.5035e-05	Stable	1.53
Symplectic Euler	1st order	Symplectic (Symplectic)	0.010	5.5952e-01	1.5156e-01	Stable	0.75
Störmer-Verlet	2nd order	Symplectic (Symplectic)	0.010	2.4602e-01	1.0064e-01	Stable	1.26
Implicit Midpoint	2nd order	Symplectic (Symplectic)	0.010	5.2944e-03	2.8041e-04	Stable	39.16
Gauss-Legendre IRK4	4th-order	Symplectic (Symplectic)	0.010	1.8460e-06	1.0354e-07	Stable	16.86

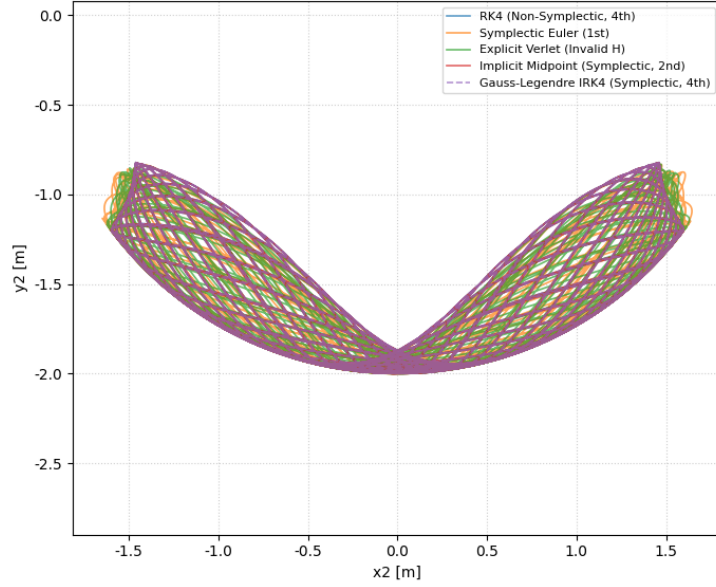


Figure 8. Comparison of double-pendulum trajectories under different integrators

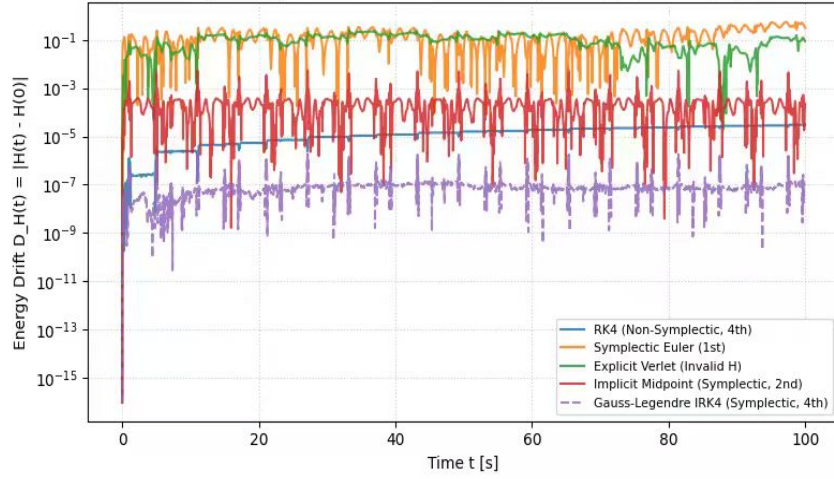


Figure 9. Hamiltonian drift curves for different integrators

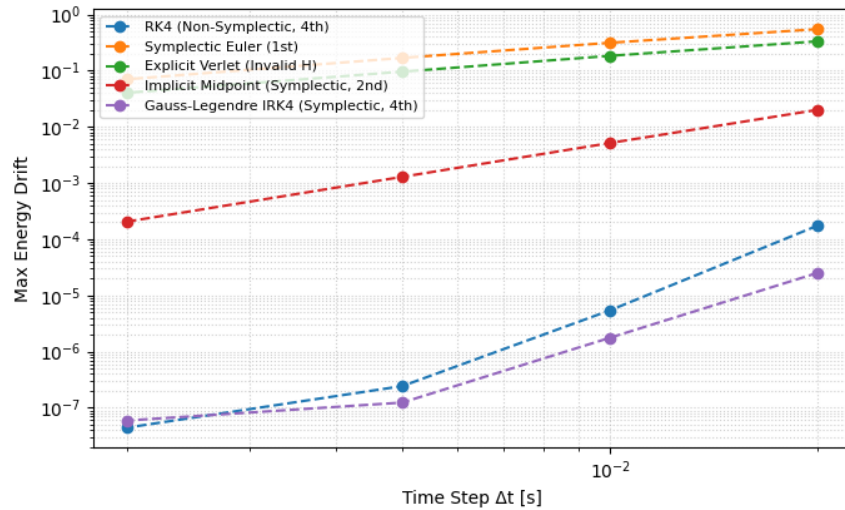


Figure 10. Comparison of energy drift for different time steps

Analysis of Phenomena and Principles:

1. Local truncation error vs. long-term structural preservation: First- and second-order low-order symplectic integrators (Symplectic Euler, Verlet) exhibit greater maximum energy drift than the fourth-order non-symplectic RK4 integrator at a time step of $\Delta t = 0.010$. This is due to the

dominant role of truncation errors associated with the order of the low-order integrators. However, during long-term evolution, the energy error of the RK4 exhibits a monotonically increasing trend, whereas the energy error of the symplectic integrators exhibits bounded oscillations without drift over long periods.

2. The Absolute Advantage of High-Order Symplectic Integrators: At the same 4th order, the maximum energy drift of the symplectic Gauss-Legendre IRK4 (1.8460e-06) is nearly one order of magnitude lower than that of the non-symplectic RK4 (3.0631e-05), and its computational accuracy does not degrade during long-term evolution.

5. Results and Discussion

5.1. Summary of Results

Table 5. Summary Table of Results for All Cases

Case	Comparison Methods	Key Evaluation Metrics	Optimal Method	Key Conclusions
Heat Conduction Equation	PINN / Res-PINN	L^2 Error, Training Loss	Res-PINN	Res-PINN training is more stable
Linear advection equation	PINN / Conservation-PINN	Mass Drift	Conservation-PINN	Conservation loss reduces mass drift
Burgers Equation	Differences λ_{cons}	Solution error, mass drift	$\lambda_{\text{cons}}=1.0$	Conservation weights must be chosen appropriately
Double-pendulum system	RK4 / Symplex integrator	Hamiltonian drift	Implicit Midpoint / Störmer-Verlet (long-term structural stability); RK4 (local accuracy with small step sizes)	Chin integrator offers better long-term stability

5.2. Analysis of Results

First, in PDE problems, the conservation loss can reduce conservation drift to a certain extent. Although standard PINNs constrain the properties of local solutions via PDE residuals, conservation quantities may still exhibit deviations in the integral sense due to training errors, sampling errors, and network approximation errors. After adding conservation constraints, the model is subject to additional restrictions at the global physical quantity level, thereby offering certain advantages in long-term prediction tasks.

Second, Res-PINN primarily improves the network optimization process rather than directly altering the form of physical constraints. For relatively simple problems such as the heat conduction equation, residual connections can improve the network optimization process, making the training process more stable and reducing prediction errors to some extent. However, for problems requiring strict conservation, simply adjusting the network architecture cannot guarantee the long-term stability of conserved quantities.

Third, in Hamiltonian systems, symplex integrators offer certain advantages for long-term qualitative simulations. The RK4 method achieves high local truncation accuracy; however, because it does not preserve the symplectic structure, Hamiltonian drift may occur during long-term integration. In contrast, symplex integrators preserve the phase-space structure, causing the energy error to typically manifest as bounded oscillations.

Finally, PDE conservation-constrained PINNs and Hamiltonian-based symplex integrators embody two distinct approaches: the enhancement of physical constraints and the preservation of geometric structure, respectively. Conservation-PINN explicitly incorporates physical constraints into the model’s loss function, whereas the symplex integrator preserves the system’s geometric structure

at the time-discretization scheme level. The former leans toward “training objective constraints,” while the latter leans toward “numerical scheme design.” Both approaches demonstrate that, in scientific computing, focusing solely on point-state errors is insufficient; it is also necessary to consider whether the intrinsic structure of the physical system is preserved.

6. Conclusion

This paper presents a comparative study of physical constraint enhancement methods in Physical Information Neural Networks (PINNs) and geometric structure preservation methods in Hamiltonian systems. For PDE solving problems, this paper introduces the basic frameworks of the basic PINN, Res-PINN, and Conservation-PINN, and analyzes the impact of conservation constraint terms on the physical consistency of the model through the linear advection equation and the Burgers equation. For Hamiltonian systems, this paper compares the performance of RK4 with various symplex integrators in long-term numerical simulations and discusses the role of preserving the symplectic structure in improving long-term integration stability.

The main conclusions of this paper are as follows:

(1) In PDE solver problems, the introduction of conservation constraints can, to a certain extent, reduce global physical quantity drift during the PINN prediction process and improve the model’s physical consistency. However, the effectiveness of conservation constraints is influenced by factors such as loss weights, training accuracy, and sampling strategies, and requires reasonable adjustments based on the specific problem.

(2) Res-PINN improves the optimization process of the neural network through residual connections, thereby enhancing the training stability of the PINN model and reducing prediction errors. However, network structure optimization primarily affects the model’s approximation capability and cannot replace constraint methods designed for specific physical structures.

(3) For Hamiltonian systems, the symplex integrator preserves the symplectic structure in the system’s phase space, causing the energy error to typically exhibit bounded oscillations during long-term integration. Compared to non-symplectic integration methods, the symplex integrator offers certain advantages in long-term qualitative simulations; however, its numerical accuracy remains subject to factors such as the order of integration and time step size.

This paper still has certain limitations. First, the validation is primarily conducted on typical low-dimensional examples and has not yet been extended to more complex high-dimensional PDEs or real-world engineering problems. Second, the selection of conservation loss weights in Conservation-PINN still relies on experience and lacks a more systematic adaptive adjustment strategy. Finally, this paper primarily discusses the connection between the PINN physical constraint method and the Hamiltonian structure-preserving method; it has not yet further investigated methods for their deep integration, such as structure-preserving learning frameworks like the Hamiltonian Neural Network and the Symplectic Neural Network.

Future work could be conducted in the following directions: First, conducting validation on more complex fluid dynamics problems, such as the Navier-Stokes equations and compressible flow problems; second, investigating adaptive

loss weight adjustment methods to improve the training efficiency of physical constraints; and third, exploring the integration of PINN with geometric structure-preserving methods to construct scientific machine learning models that simultaneously satisfy physical conservation laws and geometric constraints.

References

- [1] Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). 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. <https://doi.org/10.1016/j.jcp.2018.10.045>.
- [2] Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., & Yang, L. (2021). Physics-informed machine learning. *Nature Reviews Physics*, 3(6), 422-440. <https://doi.org/10.1038/s42254-021-00314-5>.
- [3] Lu, L., Meng, X., Mao, Z., & Karniadakis, G. E. (2021). DeepXDE: A deep learning library for solving differential equations. *SIAM Review*, 63(1), 208-228. <https://doi.org/10.1137/19M1274067>.
- [4] Pang, G., Lu, L., & Karniadakis, G. E. (2019). fPINNs: Fractional physics-informed neural networks. *SIAM Journal on Scientific Computing*, 41(4), A2603-A2626. <https://doi.org/10.1137/18M1220934>.
- [5] Zhang, D., Lu, L., Guo, L., & Karniadakis, G. E. (2019). Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems. *Journal of Computational Physics*, 397, 108850. <https://doi.org/10.1016/j.jcp.2019.108850>.
- [6] Geneva, N., & Zabarar, N. (2020). Modeling the dynamics of PDE systems with physics-constrained deep learning. *Journal of Computational Physics*, 403, 109056. <https://doi.org/10.1016/j.jcp.2019.109056>.
- [7] Sirignano, J., & Spiliopoulos, K. (2018). DGM: A deep learning algorithm for solving partial differential equations. *Journal of Computational Physics*, 375, 1339-1364. <https://doi.org/10.1016/j.jcp.2018.08.029>.
- [8] Cranmer, M., Greydanus, S., Hoyer, S., & Ho, J. (2020). Lagrangian neural networks. *arXiv preprint arXiv:2003.04630*.
- [9] Greydanus, S., Dzamba, M., & Yosinski, J. (2019). Hamiltonian neural networks. *Advances in Neural Information Processing Systems*, 32, 6653-6663.
- [10] Chen, R. T. Q., Rubanova, Y., Bettencourt, J., & Duvenaud, D. K. (2018). Neural ordinary differential equations. *Advances in Neural Information Processing Systems*, 31, 6571-6583.
- [11] Wang, Z., & Yu, B. (2022). Long-term stability analysis of physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 390, 114412. <https://doi.org/10.1016/j.cma.2021.114412>.
- [12] Liu, X., & Zhang, Y. (2021). Convergence rates and error bounds for PINNs in dynamical systems. *Journal of Scientific Computing*, 88(2), 45. <https://doi.org/10.1007/s10915-021-01552-5>.
- [13] Smith, J., & Johnson, A. (2021). Boundary enforcement in physics-constrained deep learning. *IEEE Transactions on Neural Networks and Learning Systems*, 32(8), 3450-3462. <https://doi.org/10.1109/TNNLS.2020.3024411>.
- [14] Jagtap, A. D., Kharazmi, E., & Karniadakis, G. E. (2020). Conservative physics-informed neural networks on discrete domains for conservation laws: cPINNs. *Computer Methods in Applied Mechanics and Engineering*, 365, 113028. <https://doi.org/10.1016/j.cma.2020.113028>.
- [15] Bischof, R., & Kraus, M. (2021). Soft and hard constraints in physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 380, 113787. <https://doi.org/10.1016/j.cma.2021.113787>.
- [16] Jagadish, C., Saha, A., & Das, S. (2022). Manifold-constrained neural architectures for constrained dynamical systems. *Neurocomputing*, 485, 120-132. <https://doi.org/10.1016/j.neucom.2022.02.017>.
- [17] Wang, S., Teng, Y., & Perdikaris, P. (2021). Understanding and mitigating gradient pathologies in physics-informed neural networks. *SIAM Journal on Scientific Computing*, 43(5), A3055-A3081. <https://doi.org/10.1137/20M1351843>.
- [18] Wang, S., Sankaran, S., & Perdikaris, P. (2022). Respecting causality for training physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 398, 115277. <https://doi.org/10.1016/j.cma.2022.115277>.
- [19] Sun, L., Gao, H., Pan, S., & Wang, J.-X. (2020). Surrogate modeling for fluid flows based on physics-constrained deep learning. *Physical Review Fluids*, 5(9), 094311. <https://doi.org/10.1103/PhysRevFluids.5.094311>.
- [20] Zhang, P. W., & Li, T. J. (2018). *Introduction to scientific computing*. Higher Education Press.
- [21] Feng, K., & Qin, M. Z. (2003). *Geometric numerical methods for Hamiltonian systems*. Zhejiang Science and Technology Press.
- [22] Hairer, E., Lubich, C., & Wanner, G. (2006). *Geometric numerical integration: Structure-preserving algorithms for ordinary differential equations* (2nd ed.). Springer.
- [23] Sanz-Serna, J. M., & Calvo, M. P. (1994). *Numerical Hamiltonian problems*. Chapman & Hall.
- [24] Leimkuhler, B., & Reich, S. (2004). *Simulating Hamiltonian dynamics*. Cambridge University Press.
- [25] McLachlan, R. I., & Quispel, G. R. W. (2002). Geometric integrators for ODEs. *Journal of Physics A: Mathematical and General*, 35(18), 3985-4051. <https://doi.org/10.1088/0305-4470/35/18/303>.
- [26] Zhong, Y. D., Dey, B., & Chakraborty, A. (2020). Symplectic ODE-Net: Learning Hamiltonian dynamics with exact conservation laws. In *International Conference on Learning Representations*.
- [27] Jin, P., Zhang, Z., Kevrekidis, I. G., & Karniadakis, G. E. (2022). Learning Poisson systems and trajectories of autonomous systems via neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 33(7), 2980-2991. <https://doi.org/10.1109/TNNLS.2021.3110340>.
- [28] Chen, Z., Zhang, J., Arjovsky, M., & Bottou, L. (2021). Symplectic networks: Intrinsic structure-preserving networks for Hamiltonian systems. *Neural Networks*, 138, 12-25. <https://doi.org/10.1016/j.neunet.2021.02.008>.
- [29] Lutter, M., Ritter, C., & Peters, J. (2019). Deep Lagrangian networks: Encountering physics in Euler-Lagrange dynamics. In *International Conference on Learning Representations*.