Koopman Operators, Fourier Analysis, and Neural Networks for Nonlinear PDEs

Arush Gupta

March 8, 2025

Chapter 1

Combining Koopman Theory and SPINNs for PDE Solutions

Q: Can this be used in conjunction with SPINNs?

Yes! The Koopman framework and SPINNs can be combined to enhance the representation and learning of PDE solutions.

1. Koopman + SPINNs: The Big Picture

- **SPINNs** (Sparse Physics-Informed Neural Networks) use a meshless, sparse neural representation to approximate PDE solutions.
- **Koopman theory** transforms nonlinear dynamics into a linear system by finding appropriate eigenfunctions.
- The combination allows SPINNs to approximate **Koopman eigenfunctions**, making solutions both efficient and interpretable.

2. How They Complement Each Other

Feature	Koopman Theory	SPINNs
Main Idea	Linearize nonlinear dynamics	Sparse, meshless neural PDE solver
Advantage	Makes nonlinear problems more tractable	Reduces computational cost & improves inter
Challenge	Approximate eigenfunctions introduce error	Needs a good basis for function approxin
How They Fit	Koopman gives a transformed linear PDE	SPINNs provide a structured neural basis for learning

Q: How does Koopman help SPINNs?

- **Finding Sparse Representations** – Koopman identifies dominant eigenfunctions, reducing the degrees of freedom required in SPINNs.
- **Better Initialization** – Koopman's transformation provides a structured initialization instead of training SPINNs from scratch.
- **Handling Nonlinear Residuals** – Koopman's Fourier decomposition of residuals aligns with SPINNs' kernel-based approximations.

Q: How do SPINNs help Koopman?

- **Approximating Koopman Eigenfunctions** – SPINNs can learn Koopman eigenfunctions using their sparse, meshless representation.

- **Correcting Koopman Approximation Errors** – SPINNs adaptively refine the residual correction for better accuracy.
- **Solving Transformed PDEs Efficiently** – Once the Koopman transform is applied, SPINNs can efficiently solve the resulting linear PDEs.

Q: How does SPINN refine Koopman approximations?

Koopman approximations introduce an error term:

$$\frac{d\hat{\phi}}{dt} = K\hat{\phi} + \epsilon(x)$$

where $\hat{\phi}$ is the approximated Koopman eigenfunction, and $\epsilon(x)$ is the unmodeled nonlinear component.

1. Fourier Analysis of Errors

- Taking the Fourier transform of $\epsilon(x)$ helps identify frequency components missed by Koopman.
- The error modifies the spectrum, meaning the Koopman operator alone does not fully capture system evolution.

2. SPINN-Based Residual Correction

- Instead of leaving $\epsilon(x)$ unknown, we train a **SPINN** to approximate it:

$$\epsilon(x) \approx \text{SPINN}(x).$$

- SPINNs refine the error term using sparse basis functions and kernel-based approximations.

Q: How is the SPINN correction term trained?

SPINNs are **not purely data-driven** but incorporate **physics-informed constraints** to model the residuals directly.

1. Koopman Projection and Residual Identification

- Start with the nonlinear system:

$$\frac{dx}{dt} = f(x)$$

- Koopman approximates evolution using a linear transformation:

$$\frac{d\phi(x)}{dt} = K\phi(x)$$

- The error term $\epsilon(x)$ captures the missed nonlinearities.

2. Embedding SPINN Correction in the Koopman Framework

- **Sparse Fourier Kernel Networks**: SPINNs use kernel-based transformations to approximate residual errors.
- **Physics-Guided Loss Function**: Instead of standard data loss, SPINNs minimize a physics-informed residual loss:

$$L_{\text{SPINN}} = \sum_i \left| \frac{d\hat{\phi}}{dt} - K\hat{\phi} - \text{SPINN}(x) \right|^2$$

- **Iterative Refinement**: Koopman eigenfunctions and SPINN corrections are iteratively adjusted until the residual $\epsilon(x)$ is minimized.

Q: Why is this approach valuable?

- **Koopman provides structure** → SPINNs provide efficiency.
- **Faster & more accurate PDE solvers** for physics, fluid dynamics, and complex systems.
- **Bridges physics-based and data-driven approaches**, avoiding black-box ML.

Chapter 2

Understanding Koopman Operators, Fourier Transforms, and Their Role in Nonlinear Dynamics

2.1 Why is predicting the future state of a system difficult?

Predicting the future state of a system using differential equations is challenging because:

- Many dynamical systems are **nonlinear**, meaning their solutions cannot be easily decomposed into simple functions.
- Numerical methods suffer from **the curse of dimensionality**, making high-dimensional problems intractable.
- Chaos and sensitivity to initial conditions cause errors to grow exponentially.

2.2 How does Koopman theory help with nonlinear dynamics?

Koopman theory provides a way to linearize nonlinear systems by lifting them into a higher-dimensional space.

Consider a nonlinear system:

$$\frac{dx}{dt} = f(x). \tag{2.1}$$

Instead of studying x directly, we study an observable function $g(x)$, which evolves according to the Koopman operator K:

$$\frac{d}{dt}g(x) = Kg(x). \tag{2.2}$$

2.3 What are Koopman eigenfunctions?

Koopman eigenfunctions $\phi(x)$ satisfy:

$$\frac{d\phi}{dt} = K\phi. \tag{2.3}$$

They help us express nonlinear dynamics in terms of linear evolution, making the system easier to analyze.

2.4 Why does Koopman approximation introduce error?

In practical applications, we use an approximate eigenfunction $\hat{\phi}(x)$ with an error term $\epsilon(t)$:

$$\frac{d\hat{\phi}}{dt} = K\hat{\phi} + \epsilon(x). \tag{2.4}$$

This error term captures the deviation from an exact Koopman representation.

2.5 How do Fourier transforms help analyze this error?

The error $\epsilon(t)$ can be decomposed using the Fourier transform:

$$\epsilon(t) = \int_{-\infty}^{\infty} \hat{\epsilon}(\omega) e^{i\omega t} d\omega. \tag{2.5}$$

Applying this in the Koopman framework:

$$\int_0^t e^{K(t-\tau)} \epsilon(\tau) d\tau = \int_{-\infty}^{\infty} \hat{\epsilon}(\omega) \left[\int_0^t e^{K(t-\tau)} e^{i\omega\tau} d\tau \right] d\omega. \tag{2.6}$$

Evaluating this integral gives:

$$\tilde{\phi}(\omega) = \sum_j \frac{\hat{\phi}(0)}{i\omega - \lambda_j} + \sum_j \frac{\hat{\epsilon}(\omega)}{i\omega - \lambda_j}. \tag{2.7}$$

This shows that:

- Koopman eigenvalues determine the system's dominant frequencies.
- Nonlinear residuals introduce corrections to these frequencies.

2.6 Does this method require data?

If the dynamical equations are known, we do not need experimental data. However, we may need numerical simulations to:

- Compute Koopman eigenfunctions using methods like Extended Dynamic Mode Decomposition (EDMD).
- Train neural networks (e.g., Fourier Neural Operators) to approximate solution operators.

2.7 Can Koopman theory be applied to Burgers' equation?

Yes. Consider the Burgers' equation:

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2}. \tag{2.8}$$

Koopman methods can approximate its solution by lifting the dynamics into an infinite-dimensional space, where the nonlinear advection term is handled in a linearized fashion.

2.8 Are neural networks better at handling high dimensions or nonlinearity?

Neural networks excel at handling high-dimensional problems but struggle with nonlinearity unless:

- Proper architectures (e.g., deep networks) are used.
- Regularization methods are applied to avoid overfitting.
- Physics-informed techniques (e.g., PINNs, Koopman-based deep learning) are incorporated.

Chapter 3

3.1 Introduction

The study of nonlinear dynamical systems is crucial in physics, engineering, and computational sciences. Traditional numerical methods such as finite difference and finite element methods (FDM, FEM) suffer from computational inefficiency when dealing with high-dimensional and chaotic systems. In contrast, modern machine learning methods, including Koopman operators, Fourier neural operators, and physics-informed neural networks, offer alternative approaches for solving these problems efficiently.

This document provides an in-depth exploration of Koopman theory, Fourier transforms, and their applications in deep learning-based solvers for partial differential equations (PDEs).

3.2 Koopman Representation of Nonlinear Dynamics

Consider a nonlinear dynamical system:

$$\frac{d}{dt}x = f(x), \tag{3.1}$$

where $x \in \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}^n$ is a nonlinear function.

The Koopman operator K acts on observables $g(x)$ such that:

$$\frac{d}{dt}g(x) = Kg(x). \tag{3.2}$$

We seek Koopman eigenfunctions $\phi(x)$ that evolve linearly:

$$\frac{d}{dt}\phi = K\phi. \tag{3.3}$$

In practical scenarios, an approximate transformation $\hat{\phi}(x)$ is learned, introducing an error term $\epsilon(t)$:

$$\frac{d}{dt}\hat{\phi} = K\hat{\phi} + \epsilon(x). \tag{3.4}$$

3.3 Fourier Decomposition of Nonlinear Residuals

The error term $\epsilon(t)$ can be analyzed using the Fourier transform:

$$\epsilon(t) = \int_{-\infty}^{\infty} \hat{\epsilon}(\omega) e^{i\omega t} d\omega. \tag{3.5}$$

Substituting this into the Koopman framework:

$$\int_0^t e^{K(t-\tau)}\epsilon(\tau)d\tau = \int_{-\infty}^{\infty} \hat{\epsilon}(\omega)\left[\int_0^t e^{K(t-\tau)}e^{i\omega\tau}d\tau\right]d\omega. \tag{3.6}$$

Evaluating the integral gives:

$$\tilde{\phi}(\omega) = \sum_j \frac{\hat{\phi}(0)}{i\omega - \lambda_j} + \sum_j \frac{\tilde{\epsilon}(\omega)}{i\omega - \lambda_j}. \tag{3.7}$$

This result shows that:

- Koopman eigenvalues determine the system's dominant frequencies.
- The nonlinear residual modifies the spectrum via frequency-dependent corrections.

3.4 Fourier Inversion Theorem

The assumption that $\epsilon(t)$ can be decomposed into a Fourier integral relies on the Fourier inversion theorem:

$$\epsilon^{\wedge}(\omega) = \int_{-\infty}^{\infty} \epsilon(t) e^{-i\omega t} dt. \tag{3.8}$$

The inverse transform recovers the original function:

$$\epsilon(t) = \int_{-\infty}^{\infty} \epsilon^{\wedge}(\omega) e^{i\omega t} d\omega. \tag{3.9}$$

If $\epsilon(t)$ has periodic components, the Fourier spectrum reveals dominant frequencies. If it contains broadband noise, the spectrum reflects that as well.

3.5 Neural Networks for High-Dimensional and Nonlinear PDEs

Traditional numerical solvers suffer in high-dimensional spaces and for strongly nonlinear systems. Neural networks, particularly neural operators, offer efficient alternatives:

3.5.1 Neural Operators: Fourier Neural Operator (FNO)

Neural operators learn mappings between function spaces, bypassing the need for step-by-step integration. The Fourier Neural Operator (FNO) transforms functions into frequency space:

$$u(x) \xrightarrow{\mathcal{F}} \hat{u}(k), \tag{3.10}$$

where $\mathcal{F}$ is the Fourier transform. The network learns transformations directly in spectral space, allowing for efficient solution of PDEs like the Navier-Stokes equations.

3.5.2 Physics-Informed Neural Networks (PINNs)

PINNs incorporate physical laws directly into the loss function:

$$\mathcal{L} = \|\mathcal{N}[u] - f\|^2 + \sum_i \|u(x_i) - u^*(x_i)\|^2. \tag{3.11}$$

This ensures solutions satisfy governing equations without requiring explicit discretization.

3.5.3 Koopman-Based Deep Learning

Koopman-based neural networks learn eigenfunctions $\phi(x)$ such that:

$$\mathcal{K}\phi = \Lambda\phi. \tag{3.12}$$

This linearizes nonlinear evolution in a higher-dimensional space, improving stability and generalization in PDE prediction.

3.6 Comparison of Methods

Feature	Strength	Weakness
High-Dimensional Handling	Excellent	Computationally expensive
Nonlinearity Handling	Good with constraints	Poor extrapolation outside training data
Chaos / PDE Solving	Works with modifications	Needs large datasets

3.7 Conclusion

Koopman theory, Fourier analysis, and modern neural networks provide powerful tools for analyzing and solving nonlinear dynamical systems. While neural networks handle high dimensions well, they require additional modifications to capture strong nonlinearities effectively. Future research should focus on hybrid methods combining Koopman representations with deep learning to improve interpretability and generalization.

Chapter 4

Fourier Representation of the Residual Error in Koopman Analysis

4.1 Conditions for Fourier Representation of Residual Error

To express the residual error $\epsilon(t)$ in terms of its Fourier transform, the following conditions must be satisfied:

4.1.1 1. Square-Integrability Condition

For the Fourier transform of $\epsilon(t)$ to exist, it must be square-integrable:

$$\int_{-\infty}^{\infty} |\epsilon(t)|^2 dt < \infty. \tag{4.1}$$

This ensures that $\epsilon(t)$ does not diverge as $t \to \pm\infty$.

4.1.2 2. Time-Dependent Residual Error

The residual error arises from approximations in the Koopman embedding and evolves over time:

$$\epsilon(t) = x(t) - \sum_j c_j \phi_j(x_0) e^{\lambda_j t}. \tag{4.2}$$

Since $\epsilon(t)$ is a function of time, its frequency components can be analyzed via Fourier methods.

4.1.3 3. Existence of a Meaningful Frequency Spectrum

The Fourier transform assumes that the function can be decomposed into a sum of sinusoids:

$$\epsilon(t) = \int_{-\infty}^{\infty} \hat{\epsilon}(\omega) e^{i\omega t} d\omega, \tag{4.3}$$

where $\hat{\epsilon}(\omega)$ is the Fourier transform of $\epsilon(t)$. This is valid if the error term exhibits periodic or quasi-periodic behavior.

4.1.4 4. Bounded Growth of the Error Term

For the Fourier transform to converge, $\epsilon(t)$ should not exhibit exponential divergence:

$$|\epsilon(t)| < Ce^{\alpha t}, \quad \text{where } \alpha \leq 0. \tag{4.4}$$

If $\alpha > 0$, the error term grows too fast, and the Fourier representation may not exist in the usual sense.

4.1.5 5. Stability of the System

The Koopman framework is most effective when the system evolves in a stable, quasi-periodic manner. If the system is chaotic, $\epsilon(t)$ may have broadband, non-periodic frequency content, making its Fourier transform less meaningful.