Solving simple damped oscillation ODE with PINN

Mohammad Mahdi Abedi

March 24, 2024

1 Introduction

In scientific computing, solving ordinary differential equations (ODEs) is a fundamental task in different applications. ODEs describe various physical phenomena, including oscillatory motion, which is a common occurrence in mechanical systems. One such problem is the damped harmonic oscillator, governed by a second-order ODE. Here, we explore the utilization of Physics Informed Neural Networks (PINN; Raissi et al. 2019) in TensorFlow to solve the damped harmonic oscillator problem.

2 Problem Statement

Consider a damped harmonic oscillator described by the following second-order ODE:

$$u_{tt} + 2\xi u_t + u = 0$$

Where:

- u_{tt} is the second-order derivative of displacement u with respect to time t (acceleration).
- u_t is the first-order derivative of u with respect to t (velocity).
- u represents the displacement (position).
- ξ (or xi) is the damping coefficient.

3 Physics Informed Neural Networks (PINN)

PINN is a recent approach that combines deep learning with physical principles to solve differential equations. In our implementation, we construct a PINN model with two input layers: one for time points (t) and the other for the damping coefficient (ξ). The architecture consists of several fully connected layers orgnized into two branches that work with each input separtely and a combination part, aimed at learning the dynamics of the system.

4 Code Explanation

4.1 Utilities

The code includes several utility functions for building, training, and evaluating the Physics Informed Neural Network (PINN).

- **build_model**: This function constructs the architecture of the PINN model. It defines two input layers, one for time points (t) and the other for the damping coefficient (ξ). The model consists of fully connected layers with *tanh* activation functions, facilitating the learning of system dynamics.

- **compute_gradients**: This function computes the gradients required for the PINN loss function. It uses TensorFlow's automatic differentiation capability to compute gradients of model with respect to the input variable (t.
- **loss_pinn**: The loss function for the PINN is defined in this function. It comprises terms representing the differential equation (ODE) and initial conditions. The ODE term ensures that the model adheres to the physical laws governing the system (equation 1), while the initial condition terms enforce consistency with the given initial conditions (fixed during training and inference). It calculates u_t and u_{tt} using the **compute_gradients** function.
- **train_step**: This function defines a single training step for the PINN model. It computes gradients, applies gradient descent optimization to update the model parameters, and returns the loss value along with additional metrics that comprise the values of different terms of the loss function.

4.2 Training

The training section of the code encompasses the entire process of training the PINN model.

- Initializes the PINN model and sets the number of epochs. The model architecture is defined using the **build_model** function, and hyperparameters such as the number of neurons per layer and the learning rate are specified.
- Defines the initial conditions and parameters required for training. These include the initial position (x_0), initial time (t_0), initial velocity (v_0), number of time points for training (npt), and the range of time values (min_time and max_time).
- Executes the training loop, which iterates over the specified number of epochs. Within each epoch, random time points and a damping coefficient (ξ) are sampled. The PINN model is trained to minimize the loss function, which comprises terms representing the ODE and initial conditions. The learning rate is decreased dynamically using an adaptive scheduling strategy.

4.3 Exact Solution

The damped harmonic oscillator problem has an explicit exact solution, allowing for validation of the PINN model's performance. The exact solution is given by the formula:

$$u(t) = Ce^{-\xi t}\sin(\omega_d t + \phi)$$

where C, ω_d, and ϕ are derived from the initial conditions (x_0 and v_0) and the damping coefficient (ξ).

4.4 Results and Discussion

After training the PINN model, its performance is evaluated by comparing the predicted displacement against the exact solution for various time points. Figure 1 illustrates this comparison, demonstrating the close agreement between the PINN prediction and the true solution. This validation confirms the accuracy and effectiveness of the PINN approach in solving the damped harmonic oscillator problem. Note that we trained the model so that we can change the damping coefficient (ξ) after training.

5 Conclusion

In conclusion, we have demonstrated the application of Physics Informed Neural Networks (PINN) to solve the damped harmonic oscillator problem efficiently. By combining deep learning with physical principles, PINN offers a promising approach for solving differential equations in various scientific and engineering domains. Our model with two branches allows for solving the PINN parametrcally so that be able to have the solution for different values of time (t) and damping coefficient (ξ) after training

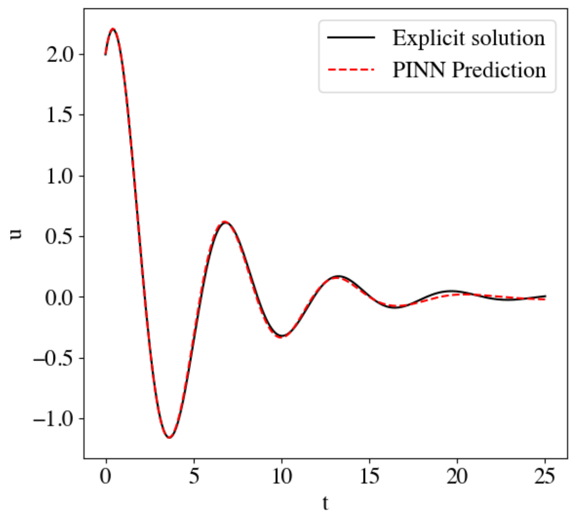

Figure 1: Comparison between PINN Prediction and Exact Solution for $\xi = 0.2$
Initial condition: $x_0 = 2.0$, $t_0 = 0.0$, $v_0 = 1.0$
In the inference stage, the PINN model can be evaluated for different values of time and ξ within predefined ranges.

References

Raissi M, Perdikaris P, Karniadakis GE. 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.