AE5093 — Scientific Applications of Deep Learning

Homework 3

Daniel J. Pearson

Aerospace Engineering Department
Worcester Polytechnic Institute

1 Introduction

In this assigment, a Feed Forward Neural Network (FFNN) is trained and informed using a physical model of a system. The network will be trained on two PDEs: the 1D viscous Burgers equation and the 2D wave equation. The goal is to demonstrate the ability of the FFNN to learn the underlying physics of the system and make predictions based on the model. For both cases, it will be fed a series of boundary conditions and interior points to learn.

2 1D Burgers Equation

The 1D viscous Burgers equation is a fundamental partial differential equation (PDE) that describes the motion of a viscous fluid. It is given by:

$$u_t + u * u_x = \nu u_{xx} \tag{1}$$

where u is the velocity field, t is time, x is the spatial coordinate, and ν is the kinematic viscosity.

2.1 Problem Setup

For this problem, the domain is defined as $x \in [-1, 1]$ and $t \in [0, 1]$. With the following boundary conditions:

$$u(x, 0) = -\sin(\pi x) \tag{2}$$

$$u(-1, t) = u(1, t) = 0 \tag{3}$$

where,

$$\nu \in \{\frac{0.01}{\pi}, \frac{0.0001}{\pi}, 0.0\}$$

.

For each viscous case, the following parameters are used:

- Epochs = 5000
- LearningRate = 1000
- Neurons = 50
- $N_i = 5000$ (interior points)
- $N_b = 256$ (boundary points)
- $N_{ic} = 256$ (initial condition points)

The NN will be trained on the interior points, boundary points and initial conditions. The total loss function is defined by:

$$L = L_{pde} + L_{ic} + L_{bc} \tag{4}$$

where,

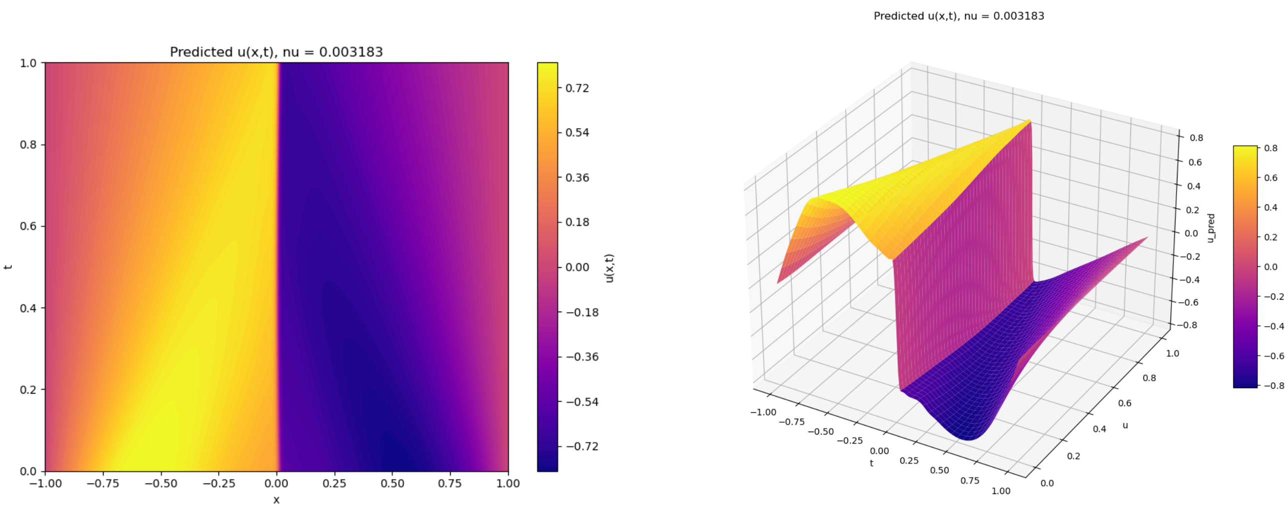
$$L_{pde} = u_t + uu_x - \nu u_{xx} \tag{5}$$

$$L_{ic} = \frac{1}{N}\sum_{i=1}^{N}(\theta(u|x_{ic}, t_{ic}) - u_{ic})^2 \tag{6}$$

$$L_{bc} = L_{ic} = \frac{1}{N}\sum_{i=1}^{N}(\theta(u|x_{bc}, t_{bc}) - u_{bc})^2 \tag{7}$$

2.2 Results - No LR Annealing

2.2.1 Case 1: High Viscosity



(a) Prediction Gradient for Case 1: High Viscosity (b) Prediction for Case 1: High Viscosity

Figure 1: Prediction for Case 1: High Viscosity

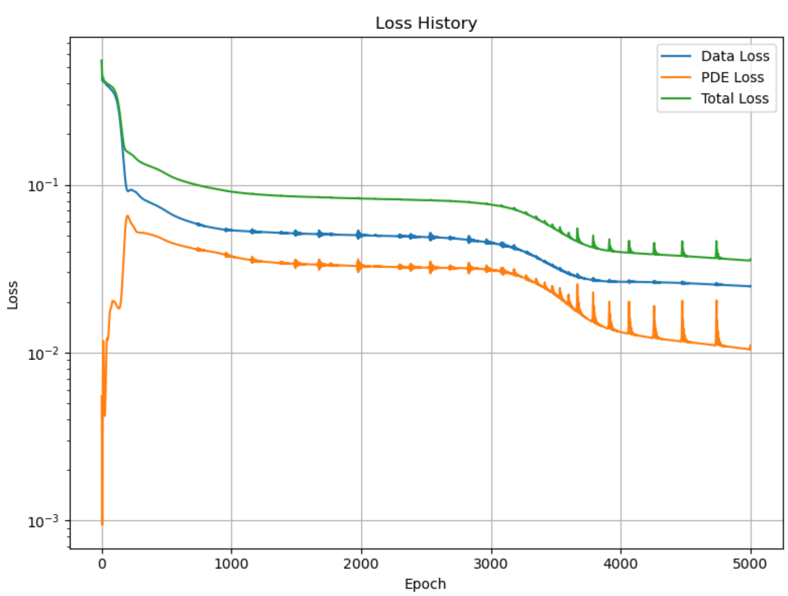
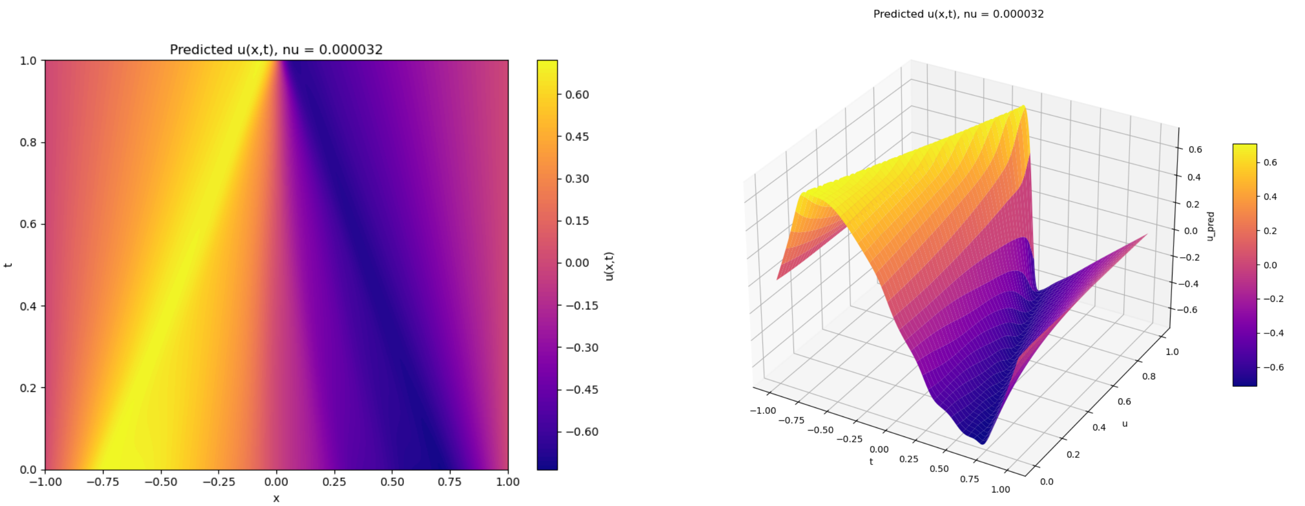


Figure 2: Loss for Case 1: High Viscosity

For the first case, the FFNN was trained on the highest viscocity case. The loss function converged to a value of $10^{-1}.4$ after approximately 4500 epochs. The prediction gradient and 3d plot of the prediction are shown in Figure 1. Since this function does not have a closed form solution, the prediction is compared to a numerical solution. Using the gradient, it is clear that the boundary and initial conditions are satisfied and is a smooth and continuous function expected for a viscous fluid. This case is the most difficult to learn, as the viscocity is high and there is a signfiicant jump discontinuity in the solution.

2.2.2 Case 2: Low Viscosity



(a) Prediction Gradient for Case 2: Low Viscosity (b) Prediction for Case 2: Low Viscosity

Figure 3: Prediction for Case 2: Low Viscosity

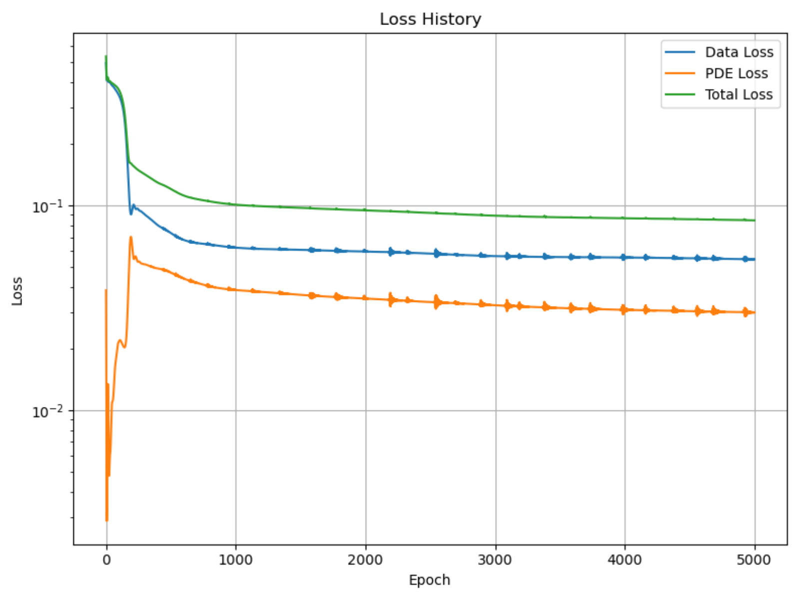
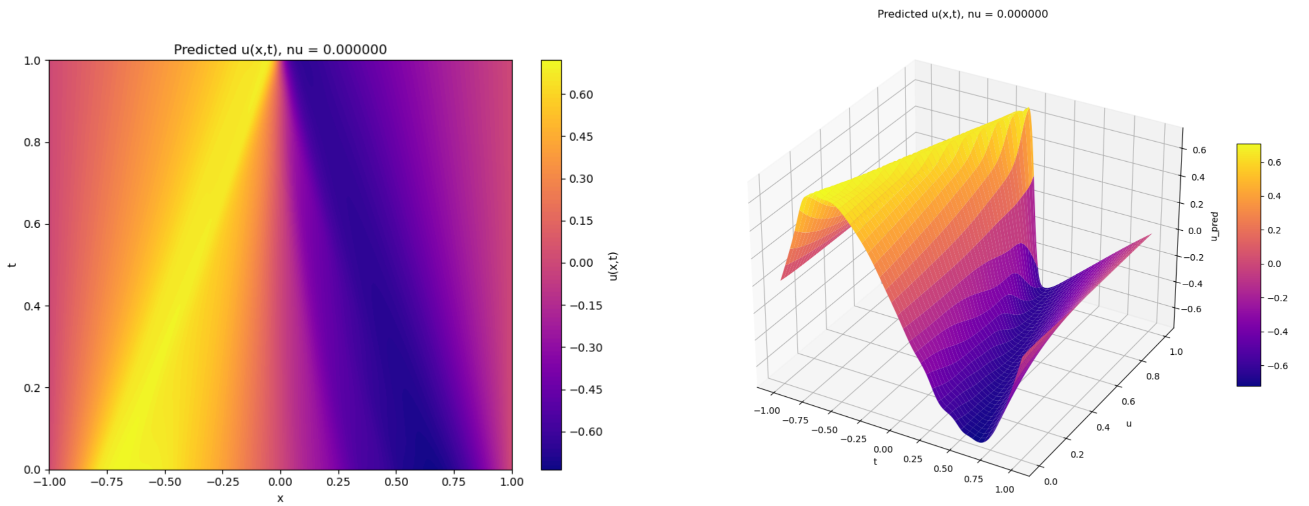


Figure 4: Loss for Case 2: Low Viscosity

For the second case, the FFNN was trained on the low viscocity case. The loss function converged to a value of $10^{-}1.2$ after approximately 3000 epochs. The prediction gradient and 3d plot of the prediction are shown in Figure 3. The prediction is much smoother than the first case and does not have a jump discontinuity. This case is expected to be easier to learn, despite a higher total loss.

2.2.3 Case 3: No Viscosity



(a) Prediction Gradient for Case 3: No Viscosity (b) Prediction for Case 3: No Viscosity

Figure 5: Prediction for Case 3: No Viscosity

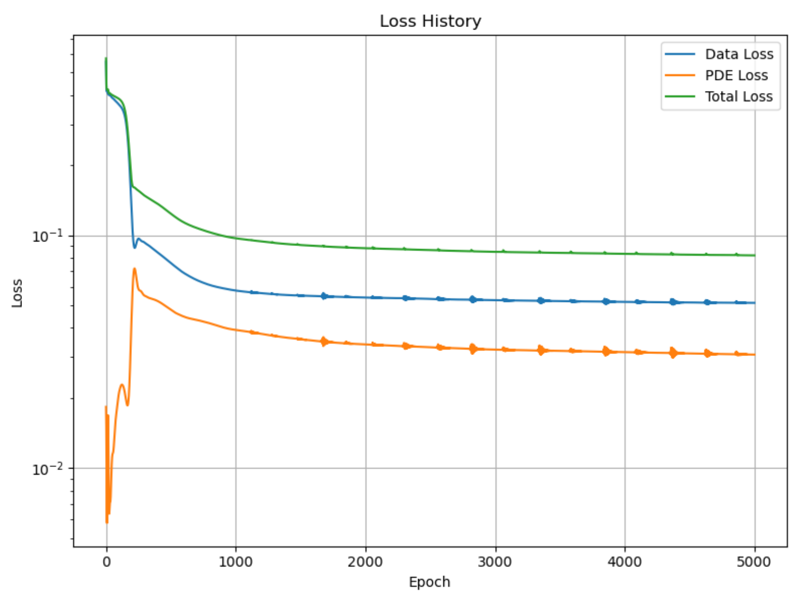


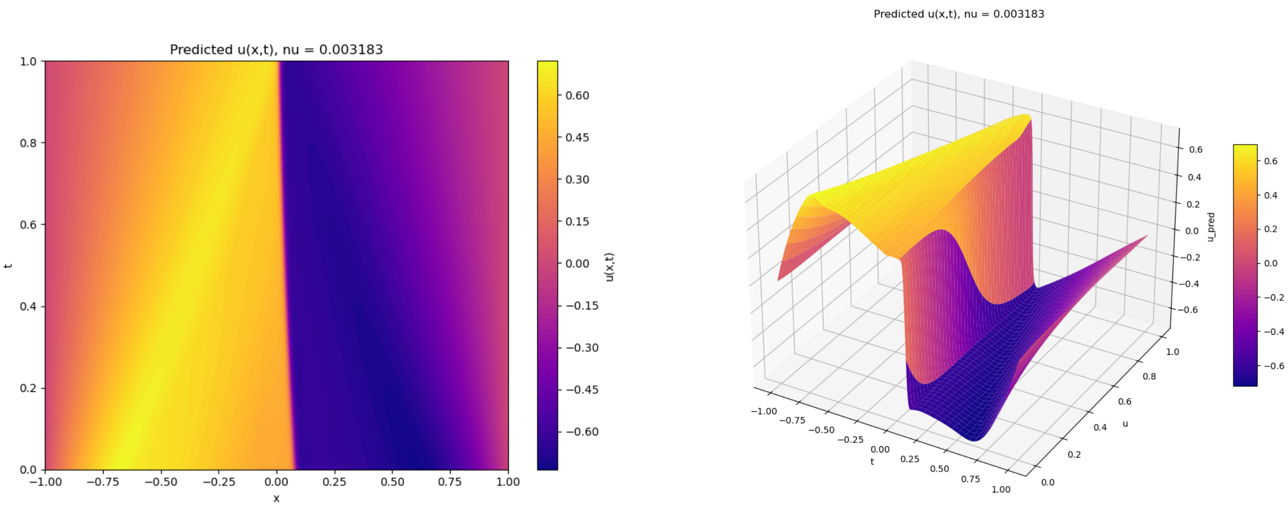
Figure 6: Loss for Case 3: No Viscosity

For the third case, the FFNN was trained on the no viscocity case. The loss function converged to a value of $10^{-}1.2$ after approximately 2000 epochs. The prediction gradient and 3d plot of the prediction are shown in Figure 5. The non-viscous case also matched well with numerical solutions. The prediction is smooth and continous, as expected.

2.3 Results - LR Annealing

For the second set of cases, the FFNN was trained with a learning rate annealing approach. Using, $\lambda = 0.9$ the following results are obtained.

2.3.1 Case 1: High Viscosity



(a) Prediction Gradient for Case 1: High Viscosity (b) Prediction for Case 1: High Viscosity

Figure 7: Prediction for Case 1: High Viscosity

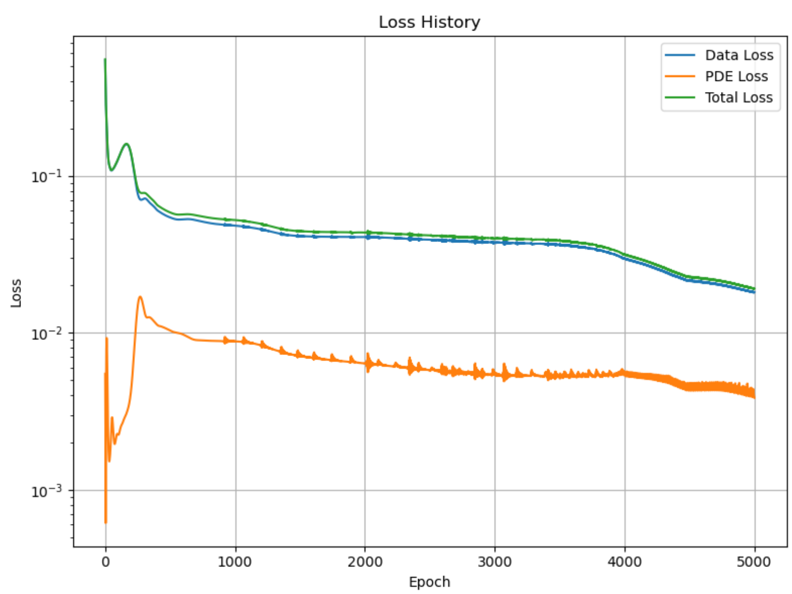
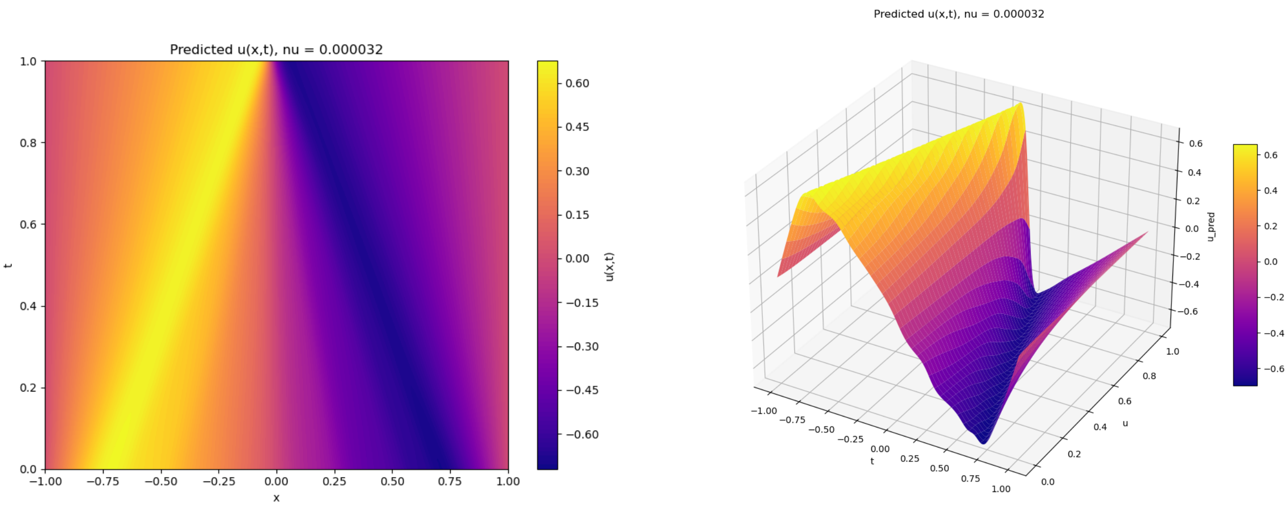


Figure 8: Loss for Case 1: High Viscosity

Compared to the case without annealing, the FFNN was able to converge to a smoother loss function with the data loss function converging significantly faster and closer to the total loss of $10^{-}1.7$. The prediction gradient and 3d plot of the prediction are shown in Figure 7. The prediction is much smoother than the first case and still captures the jump discontinuity.

2.3.2 Case 2: Low Viscosity



(a) Prediction Gradient for Case 2: Low Viscosity (b) Prediction for Case 2: Low Viscosity

Figure 9: Prediction for Case 2: Low Viscosity

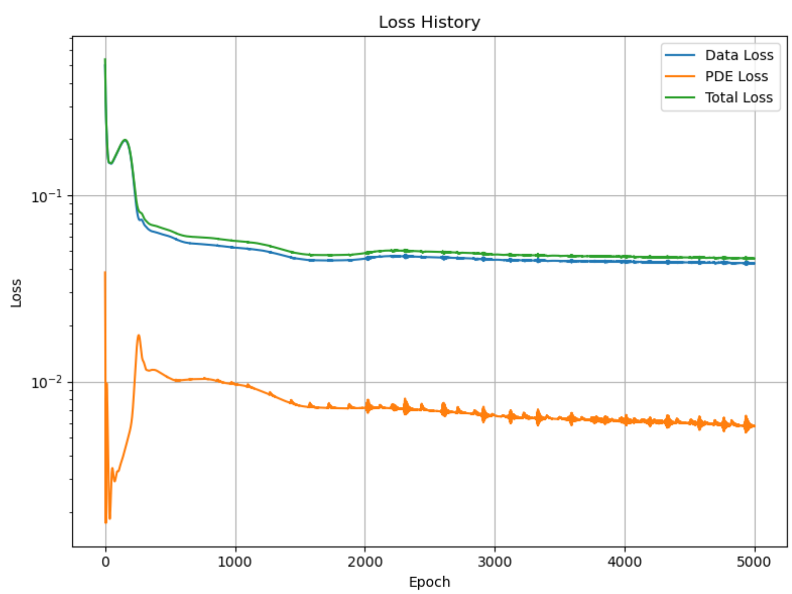
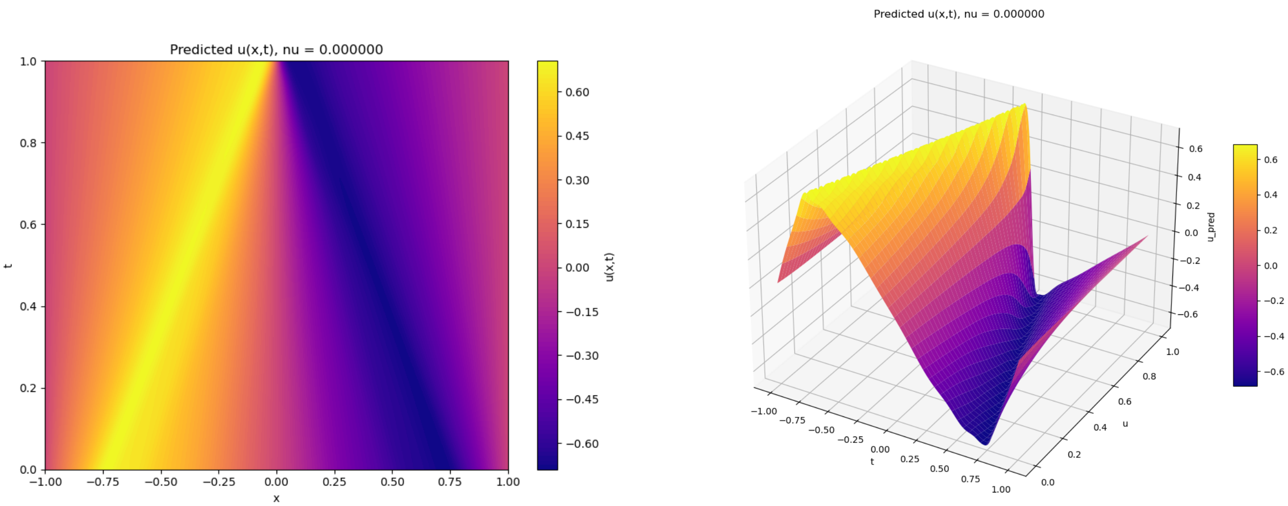


Figure 10: Loss for Case 2: Low Viscosity

Comparing the second case with and without learning rate annealing, the FFNN was yet again able to converge to a smoother loss function with less epochs. The data loss function converged to a value of $10^{-}1.5$ after approximately 2000 epochs.

2.3.3 Case 3: No Viscosity



(a) Prediction Gradient for Case 3: No Viscosity (b) Prediction for Case 3: No Viscosity

Figure 11: Prediction for Case 3: No Viscosity

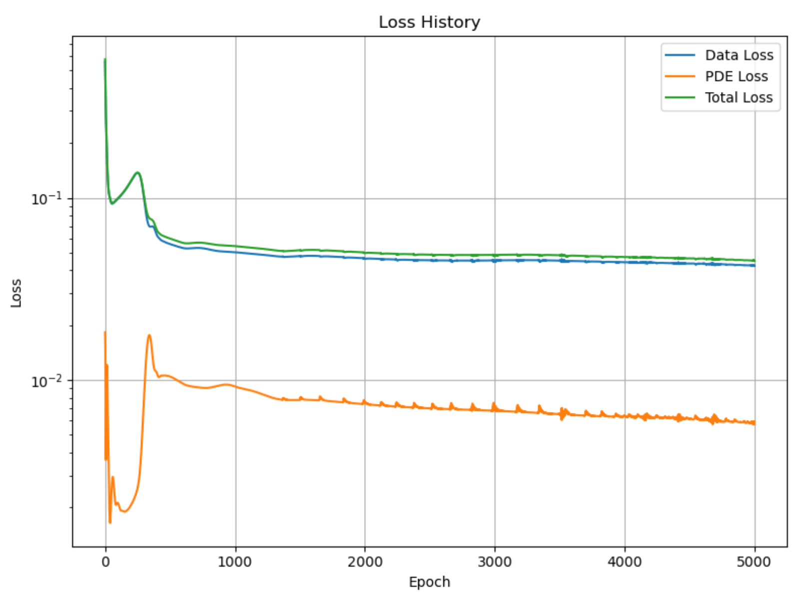


Figure 12: Loss for Case 3: No Viscosity

Comparing the third case with and without learning rate annealing, yet again the FFNN was able to converge to a smoother loss function with less epochs. The data function converged to a value of $10^{-}1.5$ after approximately 2500 epochs.

2.4 Conclusion

Overall, the FFNN was able to learn the underlying physics of the system and make accurate predictions based on the PDE and boundary conditions. The use of learning rate annealing significantly improved the speed of convergence and the smoothness of the loss function. Comparing to a numerical solution, the FFNN performed well and was able to capture the jump discontinuity in the viscous case.

3 2D Wave Equation

For this problem, the same domain as the previous case is used, but the wave equation is given by:

$$u_{tt} = c^2(u_{xx} + u_{yy}) \tag{8}$$

where c is the wave speed. The closed form solution is given by:

$$u(x, y, t) = \sin(k\pi x)\sin(k\pi y)cos(\omega t) \tag{9}$$

where k is the wave number and ω is the angular frequency given by:

$$\omega = \sqrt{2}c\pi k \tag{10}$$

The following parameters are used for the 2D wave equation:

- k $= \{2, 10, 25\}$
- Epochs $=$ 10000
- LearningRate $= 5 \cdot 10^{-4}$
- Neurons $=$ 50
- $N_i = 5000$ (interior points)
- $N_b = 256$ (boundary points)
- $N_{ic} = 256$ (initial condition points)
- $N_{bc} = 256$ (boundary condition points)