A Machine Learning Approach to Solve Partial Differential Equations

Nathaniel Jones

Department of Mathematics

West Chester University

Research & Creative Activity Day

04/17/2024

Why Solve PDEs With AI?

- Traditional numerical methods are often great, but can be prone to instability
- Some problems extremely difficult to solve:
  - Inverse problems
  - Problems with incomplete initial conditions (e.g. weather modeling)

- Physics-informed Neural Networks (PINNs) are a promising new way to tackle these issues
- Main benefit is robustness obtained from continuous model

Burgers' Equation Background

Johannes Martines Burgers
(source: University of Maryland)

- Model of turbulence in compressible fluids (such as gases)
- Equation first appears in Bateman (1915)
- Named after J.M. Burgers for his treatise on the subject in 1948
- Jump discontinuity in the solution adds difficulty in developing numerical methods

Burgers' Equation Initial Boundary Value Problem

Governing Equation

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = v\frac{\partial^2 u}{\partial x^2}$$

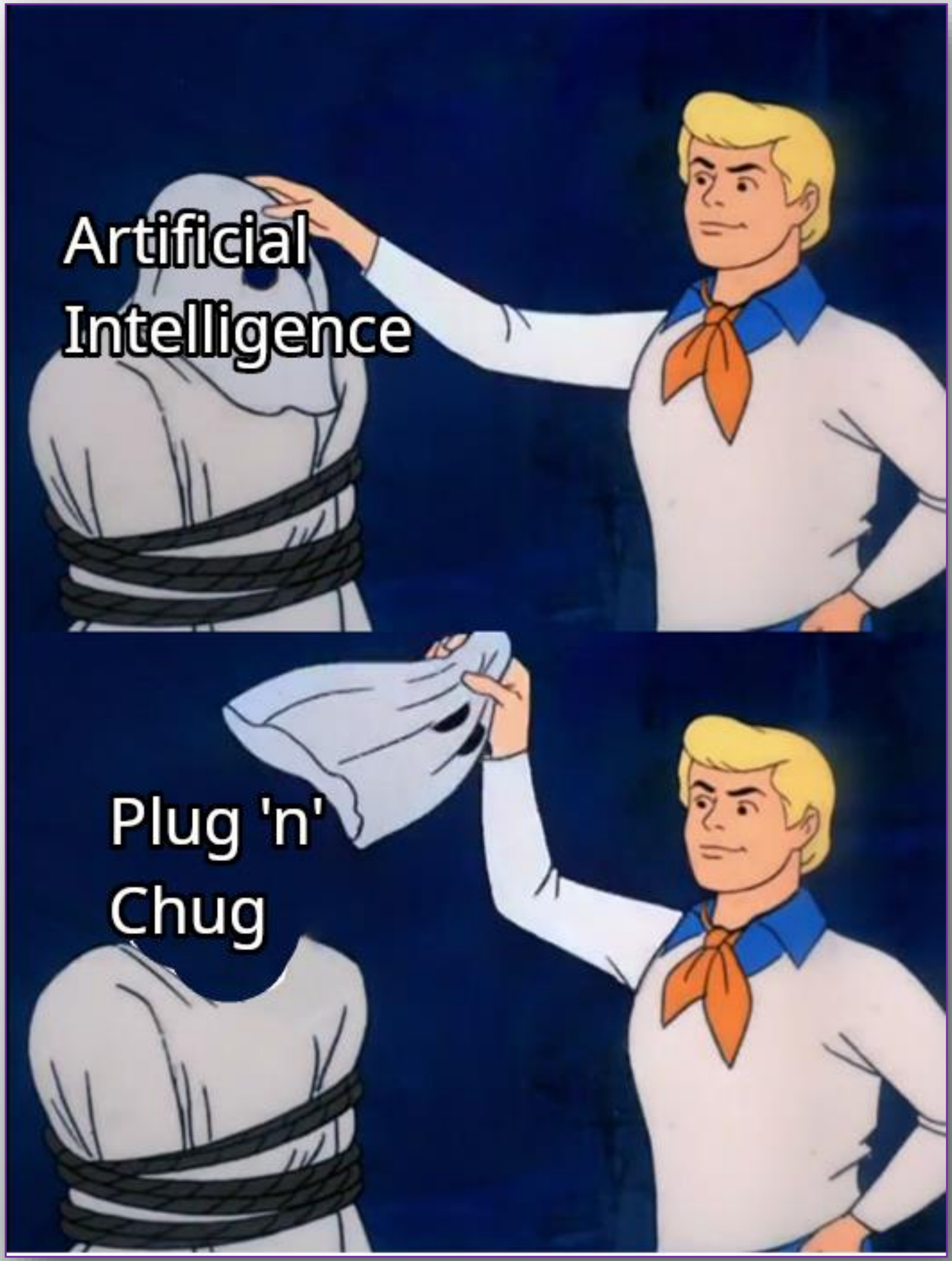
Initial Condition

$$u(x,0) = -sin(\pi x)$$

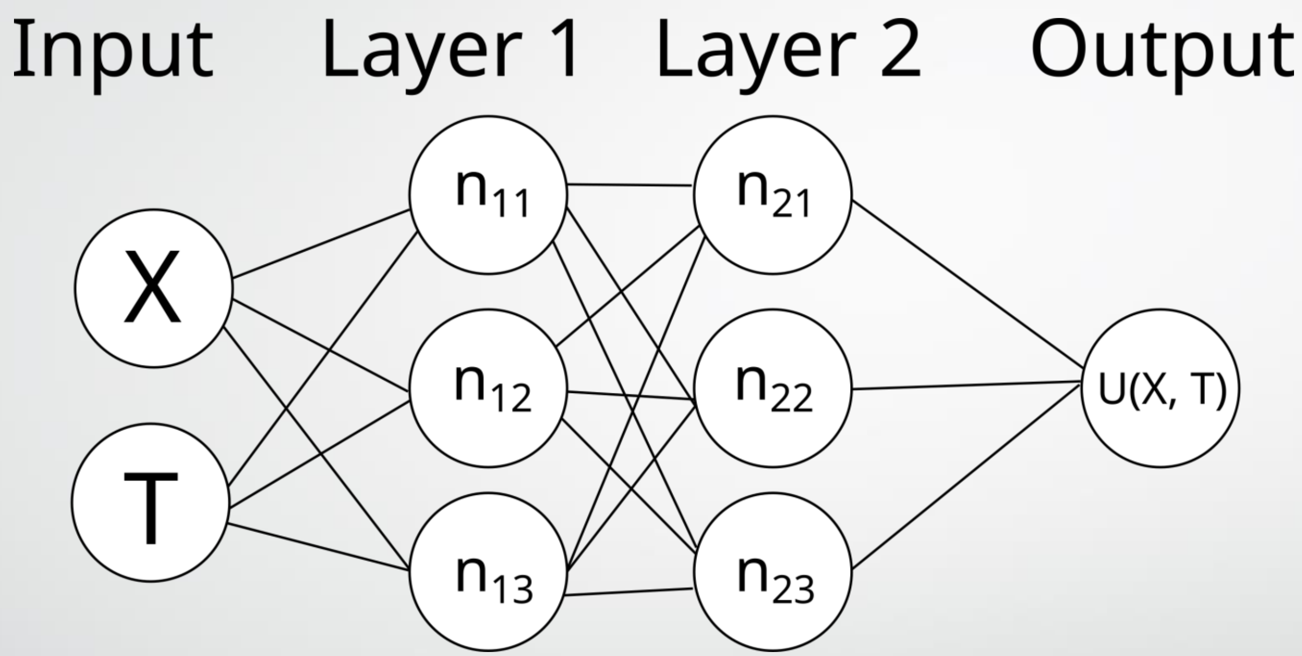
Boundary Condition

$$u(-1,t) = 0$$
$$u(1,t) = 0$$

"Artificial Intelligence"

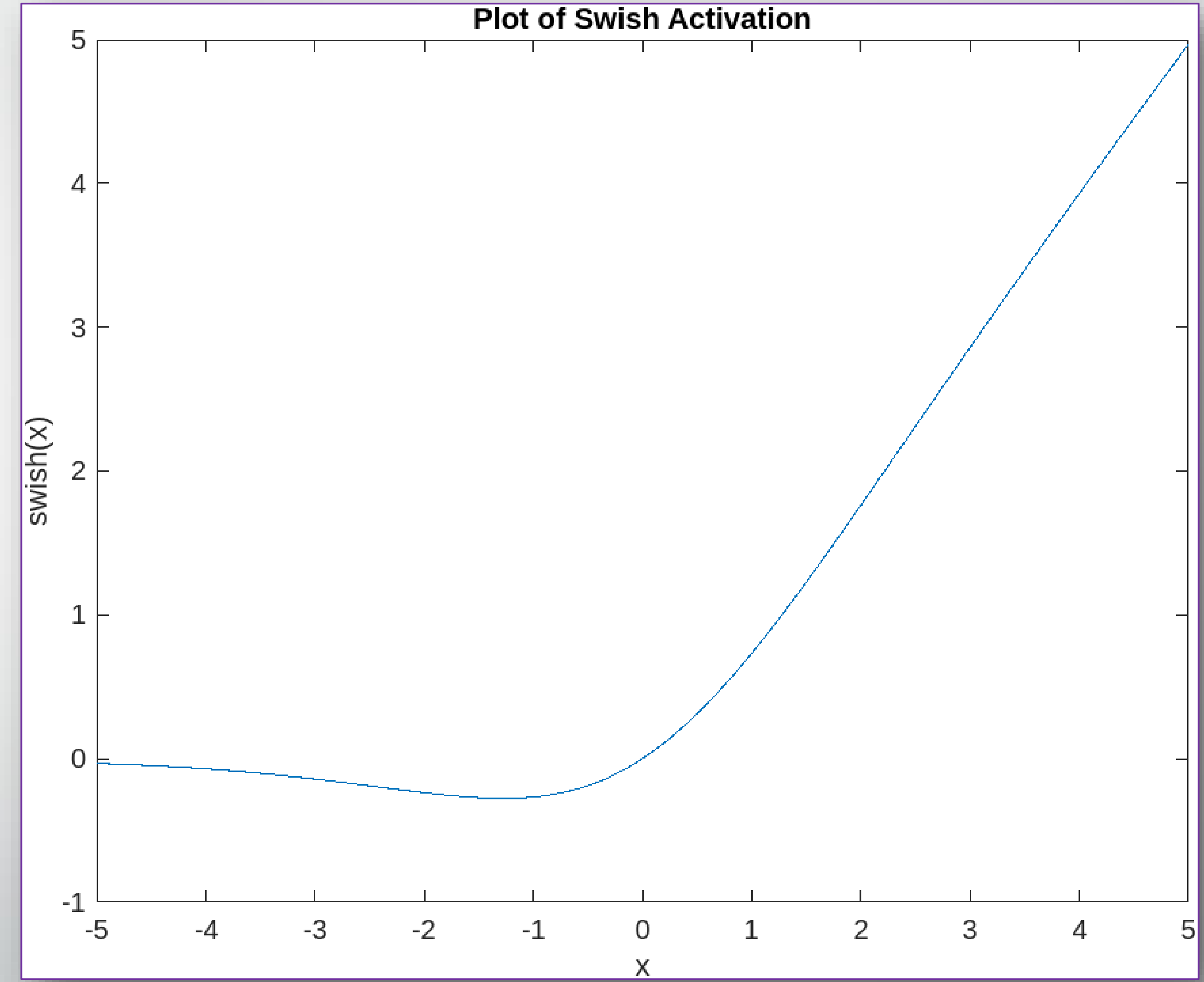


Inside a Neural Network (simplified)



$$\text{Neuron Output} = \sigma(w_1 \cdot (\text{Input 1}) + w_2 \cdot (\text{Input 2}) + \ldots)$$

Swish Activation Function



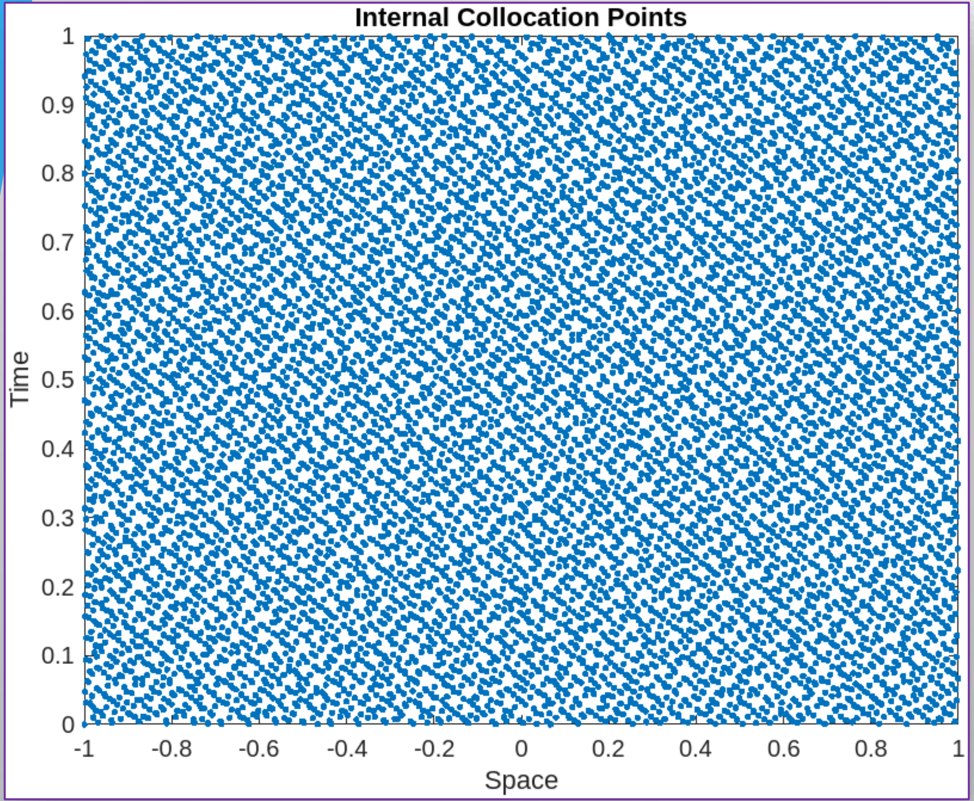
How to Train Your Neural Network

- How do we get neural network to output the solution to the IBVP?
- Find the weights that minimize the **loss function:**

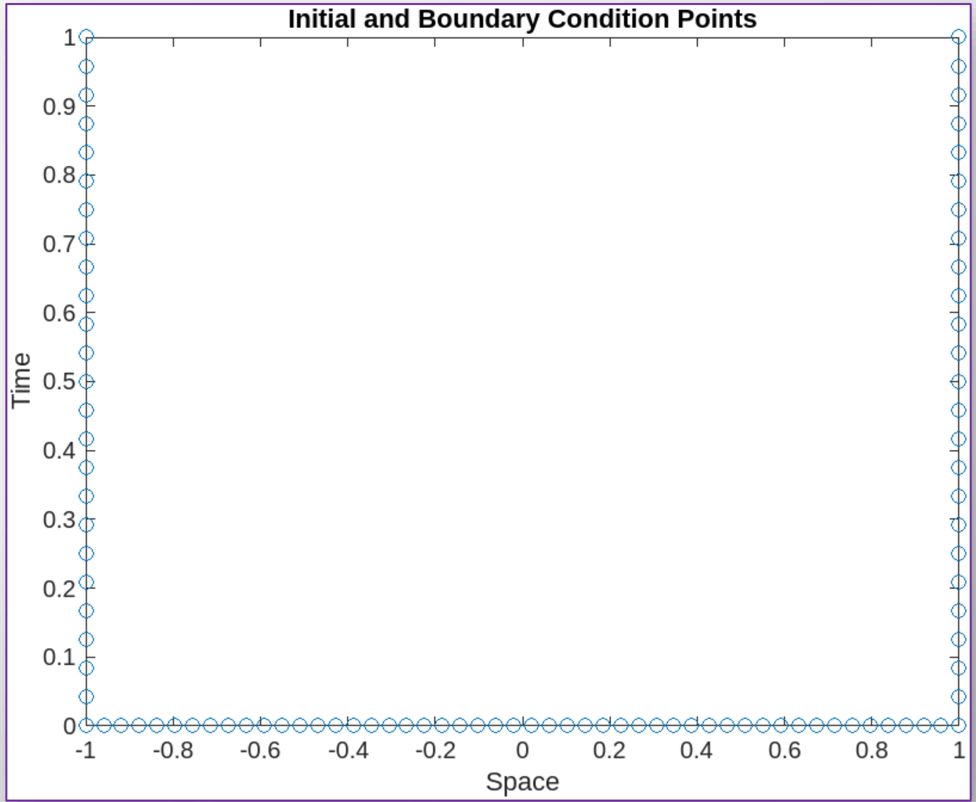
$$\boxed{L(\hat{u}) = L_{\text{Burgers}}(\hat{u}) + L_{\text{Data}}(\hat{u})}$$

- Minimization is done via steepest descent with respect to the network weights

Physics Informed Neural Net Training Process

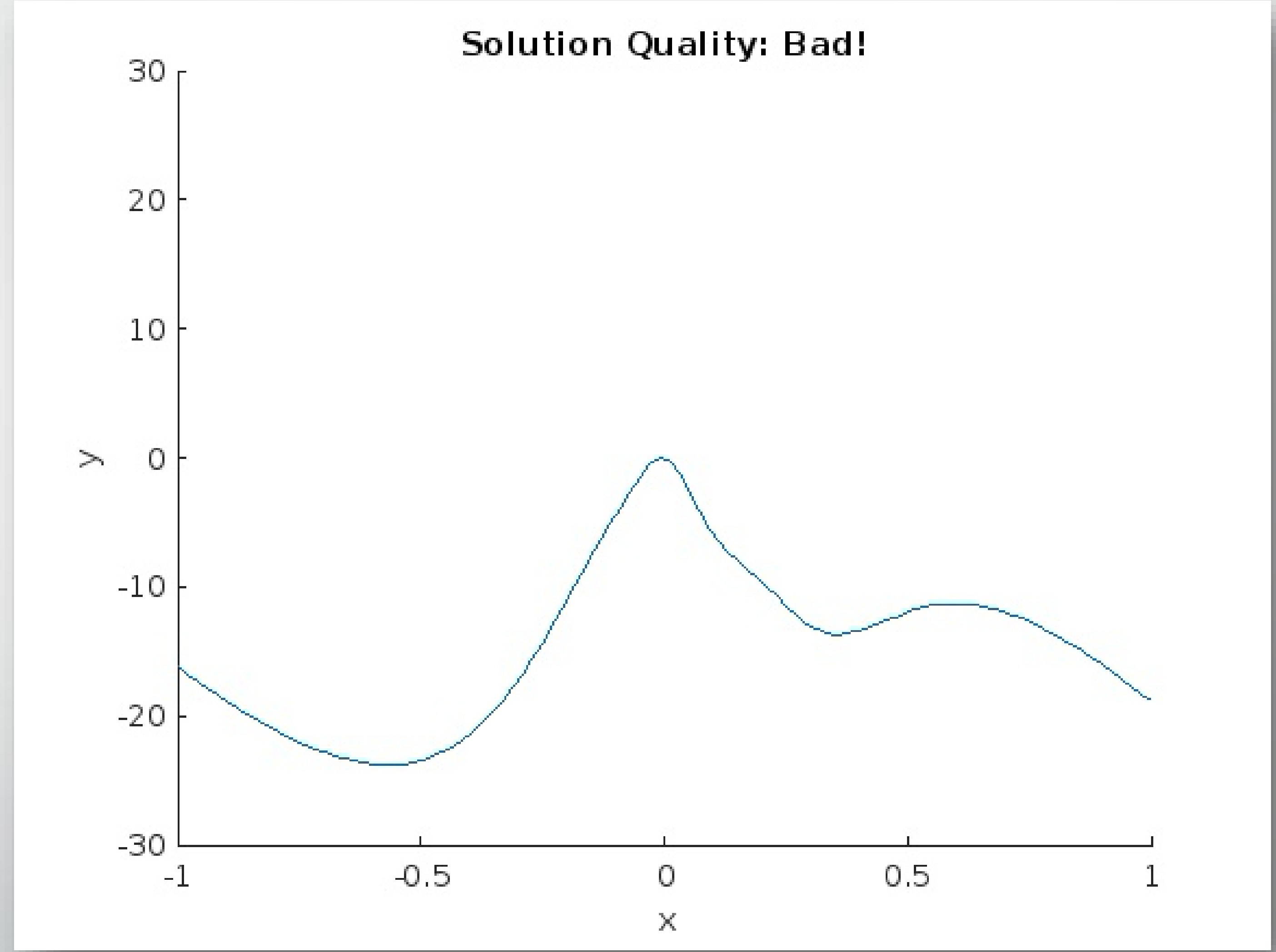


Does the network's solution evolve correctly at these points?

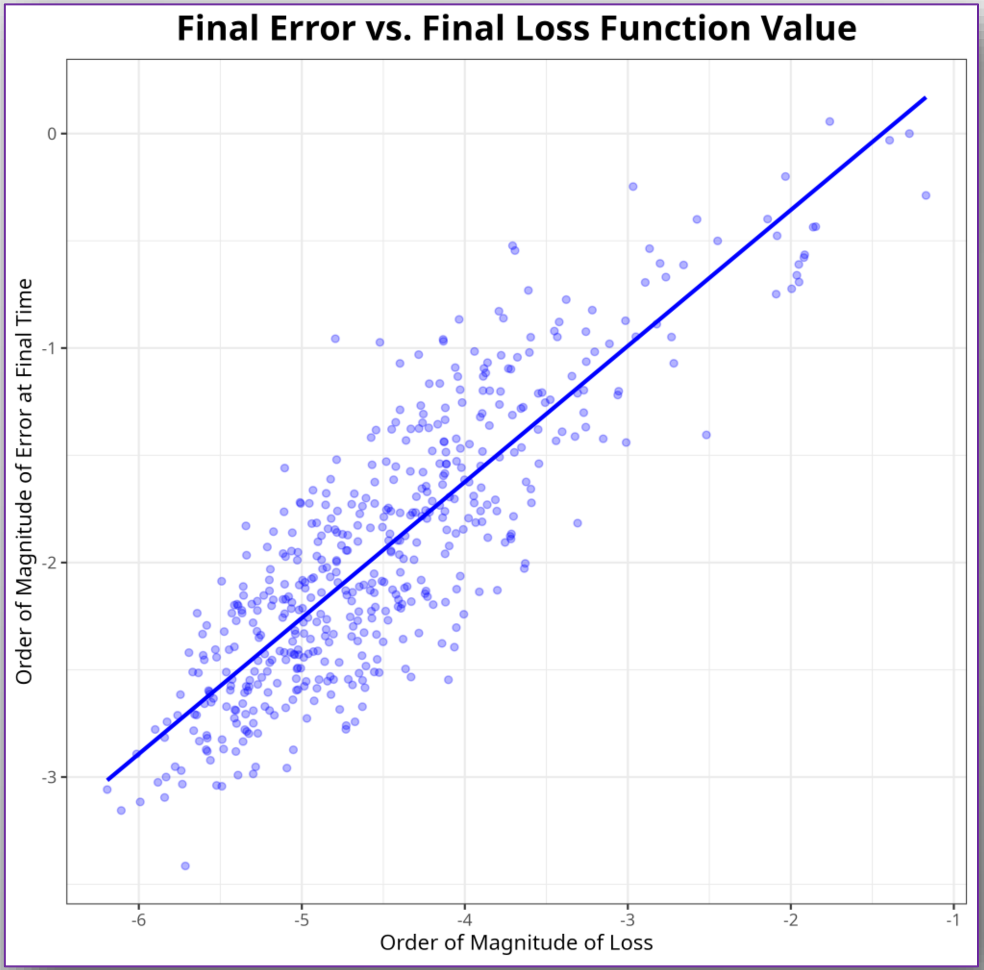


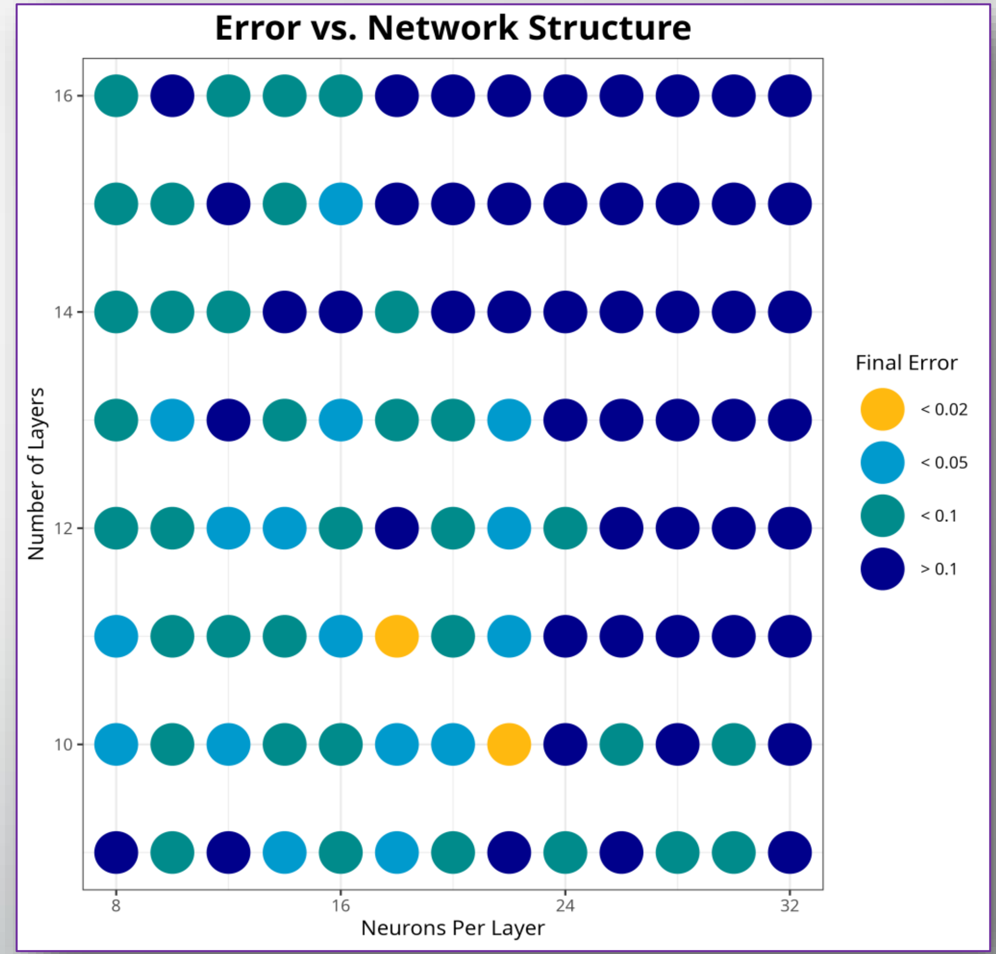
Does the network have the correct output at these points?

Vary Weights to Minimize the Loss Function



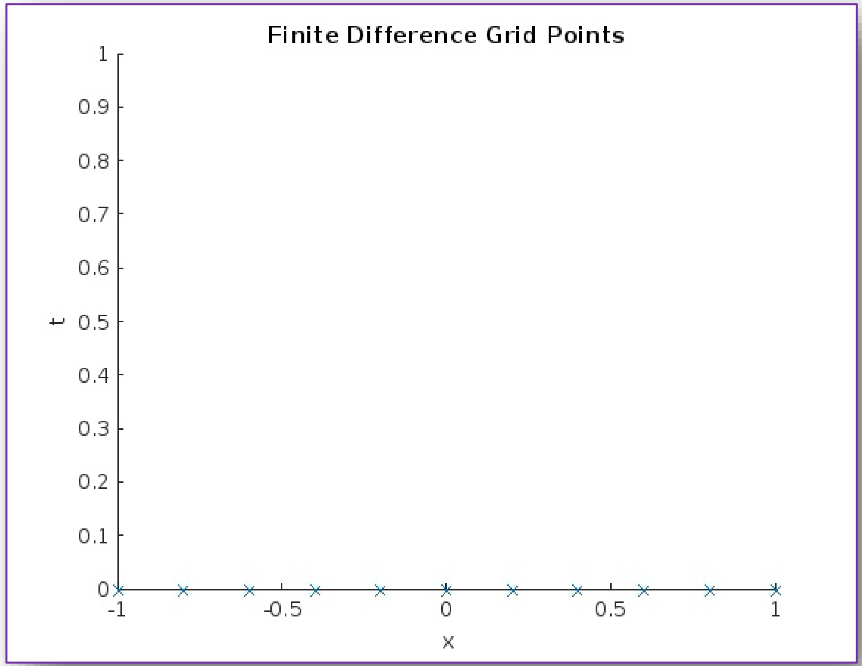
PINNs Experiment Results

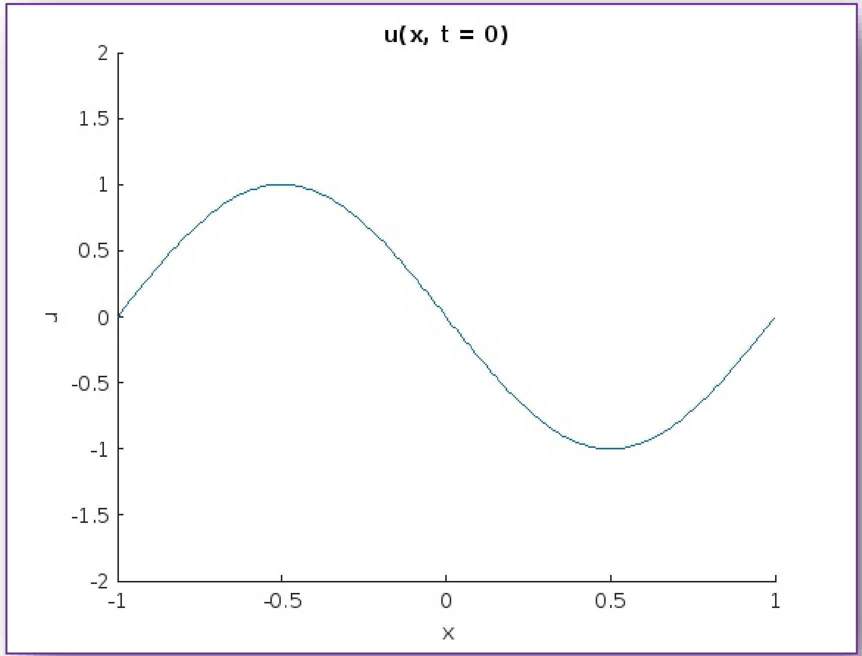




Traditional Finite Difference Method

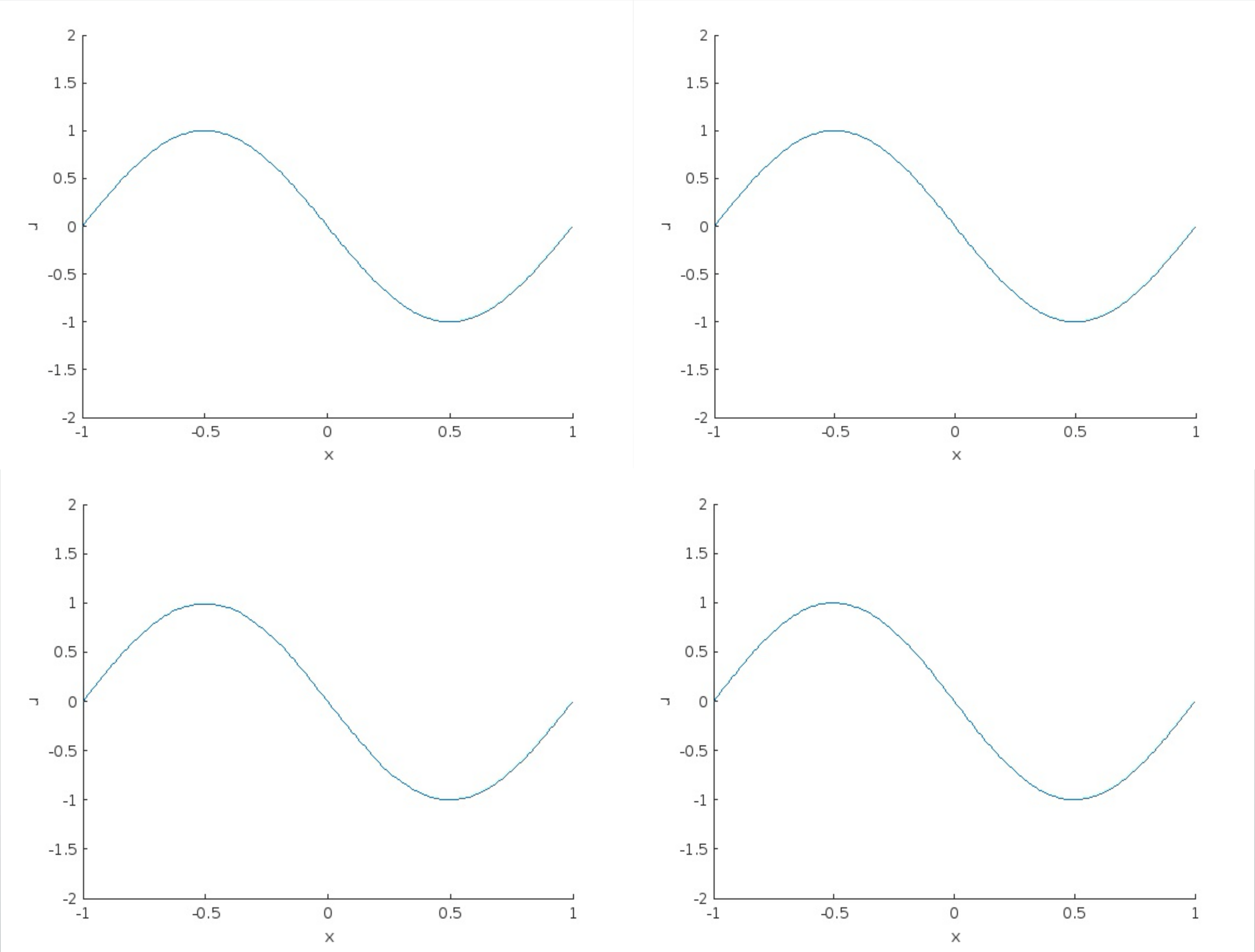
- Solution to IVBP is approximated at grid points
- First order in time, second order in space
- Point-to-point update method





Finite Difference

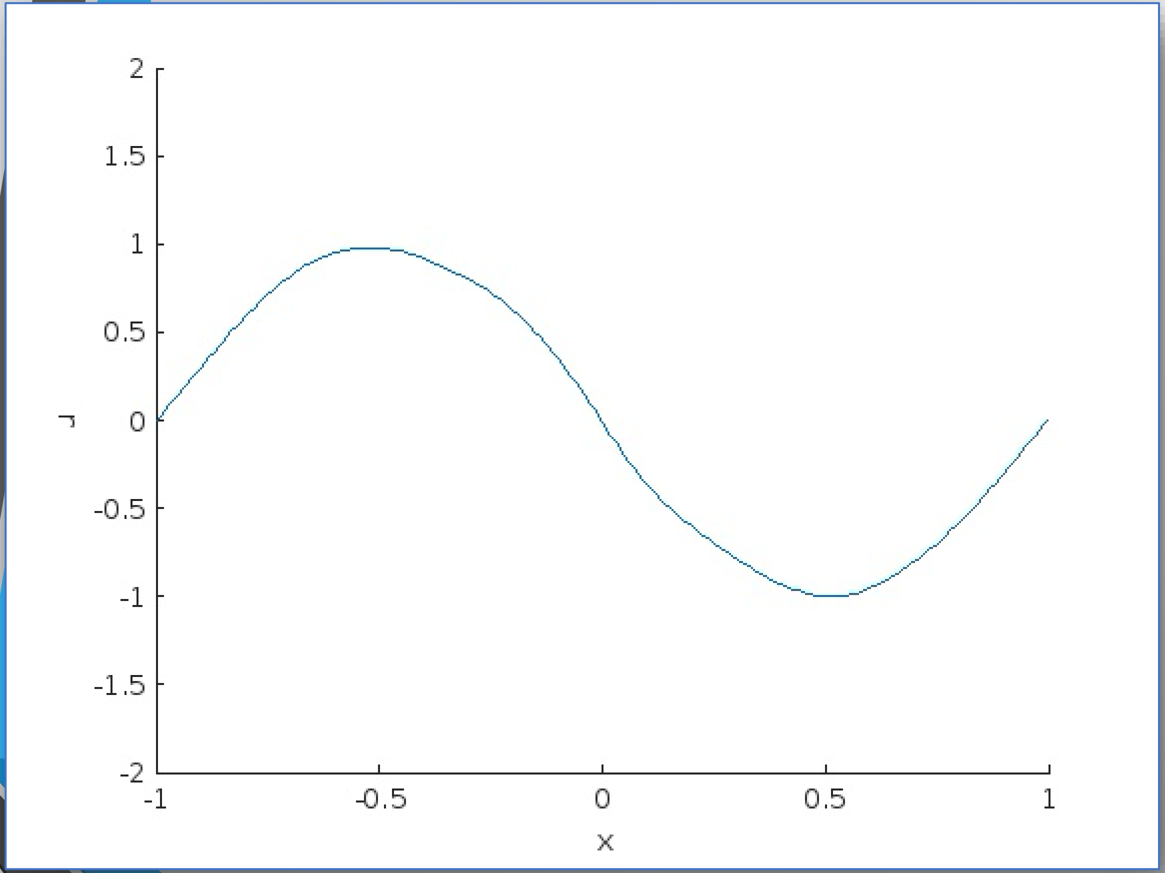
PINNs

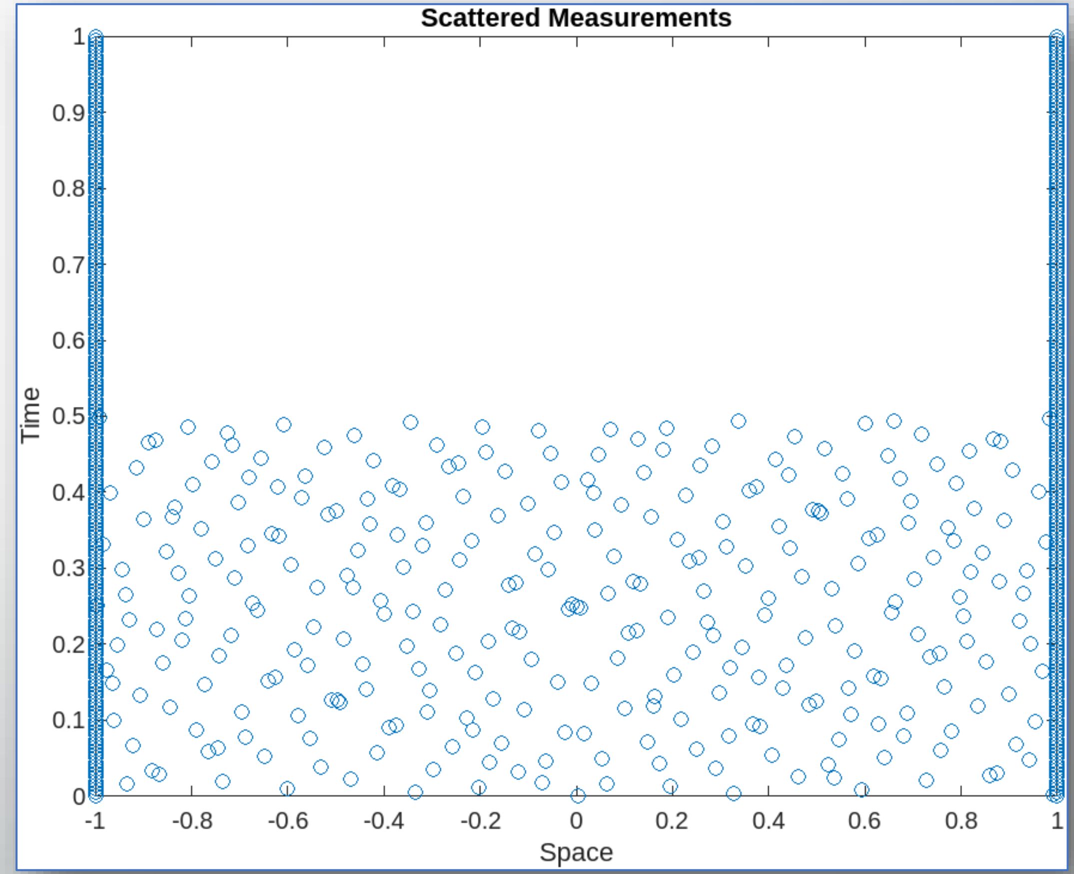


Viscosity = 0.0031

Viscosity = 0.0062

PINNs with Incomplete Initial Condition





The Tradeoff

- What we gain in robustness, we lose a lot in speed

V = 0.0031	Error at Final Time	Runtime (seconds)
PINNs	0.0618	**490.732**
Finite Difference	0.0589	0.002

Conclusion

- PINNs offer a lot in terms of robustness and ease-of-use
- But they won't replace traditional methods entirely
- One more tool in the toolbox to solve difficult problems

References

Bateman, H. (1915). Some recent researches on the motion of fluids. *Monthly Weather Review*, *43*(4), 163-170. https://doi.org/10.1175%2F1520-0493%281915%2943%3C163%3ASRROTM%3E2.0.CO%3B2

Burgers, J. M. (1948). A mathematical model illustrating the theory of turbulence. *Advances in applied mechanics*, *1*, 171-199. https://doi.org/10.1016/S0065-2156(08)70100-5

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.

MathWorks. (2023). *Solve Partial Differential Equations Using Deep Learning.* MATLAB and Simulink Help Center. https://www.mathworks.com/help/deeplearning/ug/solve-partial-differential-equations-using-deep-learning.html

Ramachandran, P., Zoph, B., & Le, Q. V. (2017). Searching for activation functions. *arXiv preprint arXiv:1710.05941*.