Solving Inverse Problems Using Finite-Element Physics Informed Neural Networks in Presence of Noise

Anthony LoRe Starleaf & Dr. Siddharth Parida

Embry-Riddle Aeronautical University

Abstract

This study builds upon a previous investigation of Finite-Element Physics-Informed Neural Networks (FE-PINNs) by performing an analysis of their sensitivity to noise. FE-PINNs were previously shown to be capable of performing a two-dimensional linear elastic full waveform inversion on a soil column. As a further step towards applying this methodology to problems involving real data, FE-PINNs were used to inversely determine the elastic modulus of a single quad element, with varying degrees of noise (0-20%) present in the training data. It was found that, depending on the accuracy of the initial estimate of the element's elastic modulus, FE-PINN can successfully solve the inverse problem with up to 20% noise in the training data.

Inverse Problems

Consider a dynamical system

$$\mathbf{y} = f(t, \mathbf{x}, \boldsymbol{\Theta}) \tag{1}$$

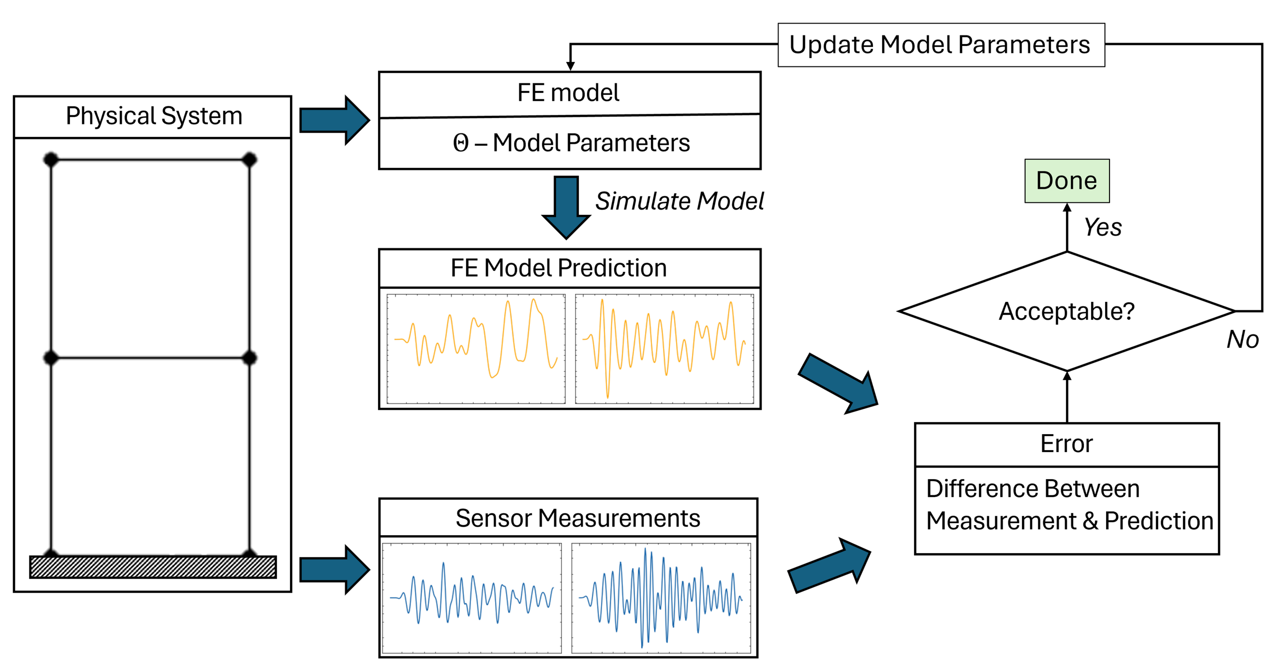
where $\mathbf{x}$ is the system state vector, t is time, $\boldsymbol{\Theta}$ is a vector of system parameters, and ν is a noise vector. Suppose sensor measurements ($\mathbf{y_m}$) of the system are available.

$$\mathbf{y_m} = g(t, \boldsymbol{\Theta}) + \nu \tag{2}$$

Inverse Problem: Given $\mathbf{y_m}$, estimate $\boldsymbol{\Theta}$

Existing Methods

Finite Element Model Updating (FEMU)

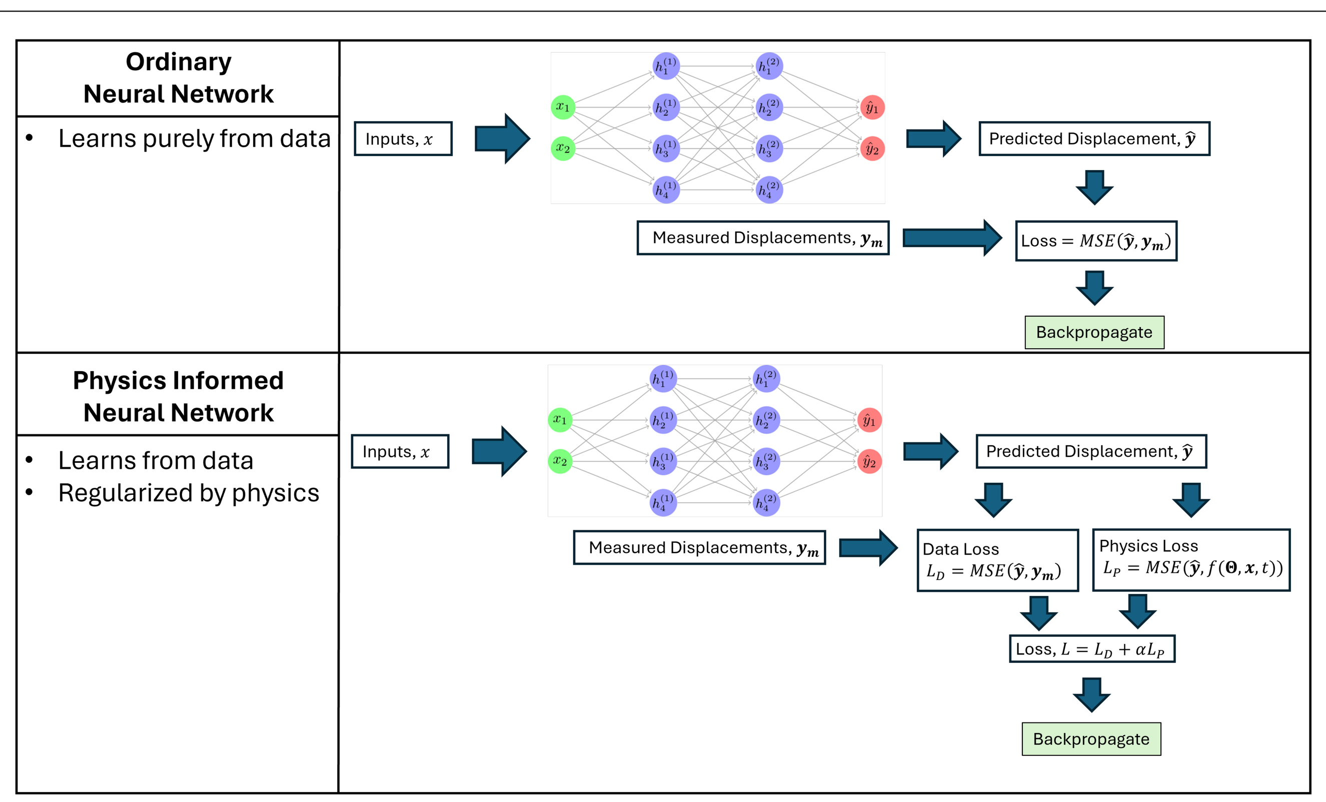


While powerful, this method has a few weaknesses

- Computationally expensive to simulate
- High fidelity models are rarely available
- Final result depends strongly on model resolution

Proposed alternative: Finite Element-based Physics Informed Neural Networks (FE-PINN)

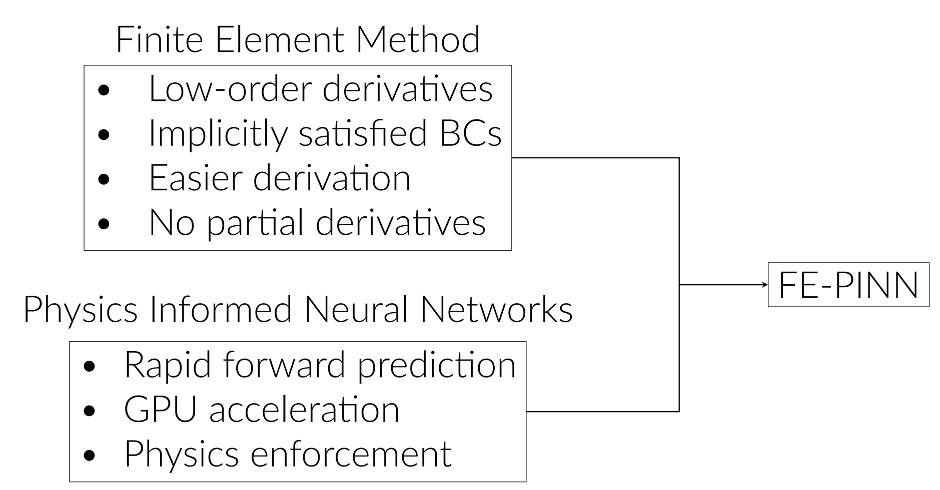
Traditional Physics-Informed Neural Networks



Strengths: Rapid forward prediction, potential for GPU acceleration, higher-fidelity surrogate model
Weakness: Many partial derivatives in physics loss term often lead to convergence issues

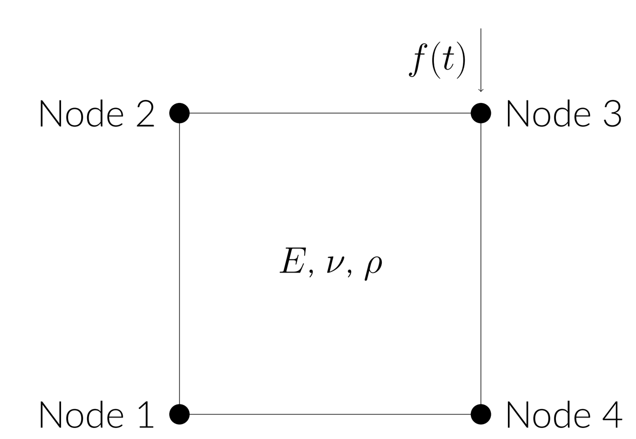
Finite Element Physics-Informed Neural Networks

1. Can be used to increase fidelity of existing FE model by incorporating data
2. Can be used to estimate parameters of FE model

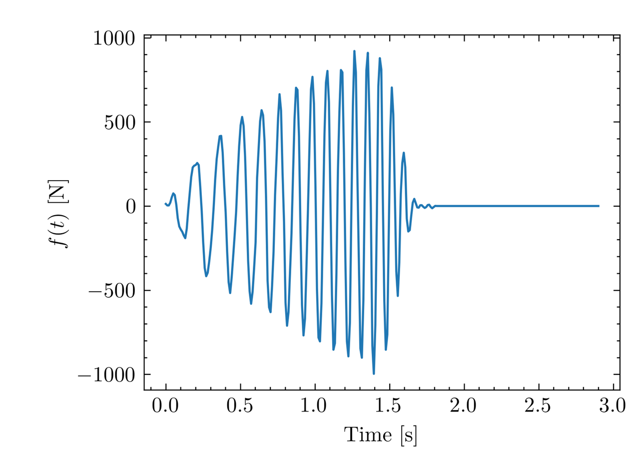


Computational Experiment

The FE-PINN algorithm was used to determine the Young's modulus (E) of the quad element shown below, subjected to varying amounts of noise. The model's initial estimate of E was also varied.



(a) Single quad element subjected to a dynamic load. Nodes 1 & 4 fixed.



(b) Forcing function.

Governing Equations

Strong form of equilibrium

$$\rho\frac{\partial^2 u}{\partial t^2} = E\frac{\partial^2 u}{\partial x^2} + G\left(\frac{\partial^2 v}{\partial x \partial y} + \frac{\partial^2 u}{\partial y^2}\right) + c\frac{\partial u}{\partial t} + b_x$$

$$\rho\frac{\partial^2 v}{\partial t^2} = E\frac{\partial^2 v}{\partial y^2} + G\left(\frac{\partial^2 u}{\partial x \partial y} + \frac{\partial^2 v}{\partial x^2}\right) + c\frac{\partial v}{\partial t} + b_y$$

With Boundary/Initial Conditions

$$u(x, y=0, t) = 0 \qquad b_x = 0$$
$$v(x, y=0, t) = 0 \qquad b_y = -f(t)\delta(x-5, y-5)$$
$$u(x, y, t=0) = 0 \qquad v(x, y, t=0) = 0$$

After application of the FE method,

$$\mathbf{M}(\rho)\ddot{\mathbf{u}} + \mathbf{C}(\nu, E)\dot{\mathbf{u}} + \mathbf{K}(\nu, E)\mathbf{u} = \mathbf{f}(t) \tag{3}$$

With Initial Conditions

$$\mathbf{u}(t=0) = 0 \qquad \dot{\mathbf{u}}(t=0) = 0 \tag{4}$$

With boundary conditions implicitly satisfied. This form is noticeably simpler due to the

- Reduced number of derivatives, and
- Absence of independent boundary conditions

Training Data

The model is trained on the x- and y- displacement histories of *only Node 3*. It is given no data on Node 2.

Model Architecture

Input Features	Output Features	Hidden Features	Hidden Layers	Activation
1	4	32	3	Sinusoid

Table 1. Hyperparameters of the neural network used to solve the inverse problem.

Results

A neural network with the parameters described in Table 1 was trained for 2000 epochs *or* until E converged to within 2% of the ground-truth value. The optimization was performed once for various noise levels and initialization errors.

% Noise	% Initial Error	Initial E [Pa]	Predicted E [Pa]	Actual E [Pa]	% Difference
0%	-15%	58846156.0	68276464.0	69230768	1.38
0%	-20%	55384620.0	68341040.0	69230768	1.29
0%	-25%	51923080.0	32707604.0	69230768	52.76
0%	-30%	48461536.0	20347384.0	69230768	70.61
0%	15%	79615384.0	68385776.0	69230768	1.22
0%	20%	83076928.0	68285928.0	69230768	1.36
0%	25%	86538464.0	68240496.0	69230768	1.43
0%	30%	89999992.0	68349984.0	69230768	1.27
5%	-15%	58846156.0	68328320.0	69230768	1.30
5%	-20%	55384620.0	68248992.0	69230768	1.42
5%	-25%	51923080.0	68424024.0	69230768	1.17
5%	-30%	48461536.0	-235790000	69230768	440.59
5%	15%	79615384.0	68281624.0	69230768	1.37
5%	20%	83076928.0	68383632.0	69230768	1.22
5%	25%	86538464.0	68385992.0	69230768	1.22
5%	30%	89999992.0	68376728.0	69230768	1.23
10%	-15%	58846156.0	68356320.0	69230768	1.26
10%	-20%	55384620.0	68311912.0	69230768	1.33
10%	-25%	51923080.0	32900322.0	69230768	52.48
10%	-30%	48461536.0	32813898.0	69230768	52.60
10%	15%	79615384.0	68277408.0	69230768	1.38
10%	20%	83076928.0	68375256.0	69230768	1.24
10%	25%	86538464.0	68352112.0	69230768	1.27
10%	30%	89999992.0	68256616.0	69230768	1.41
15%	-15%	58846156.0	68319232.0	69230768	1.32
15%	-20%	55384620.0	68303536.0	69230768	1.34
15%	-25%	51923080.0	68278648.0	69230768	1.38
15%	-30%	48461536.0	45739968.0	69230768	33.93
15%	15%	79615384.0	68121960.0	69230768	1.60
15%	20%	83076928.0	68330504.0	69230768	1.30
15%	25%	86538464.0	68445960.0	69230768	1.13
15%	30%	89999992.0	68296368.0	69230768	1.35
20%	-15%	58846156.0	68447648.0	69230768	1.13
20%	-20%	55384620.0	68414216.0	69230768	1.18
20%	-25%	51923080.0	-30599772.0	69230768	144.20
20%	-30%	48461536.0	32875180.0	69230768	52.51
20%	15%	79615384.0	68099240.0	69230768	1.63
20%	20%	83076928.0	68197576.0	69230768	1.49
20%	25%	86538464.0	68267680.0	69230768	1.39
20%	30%	89999992.0	68413616.0	69230768	1.18

Surrogate Modelling with Trained FE-PINN

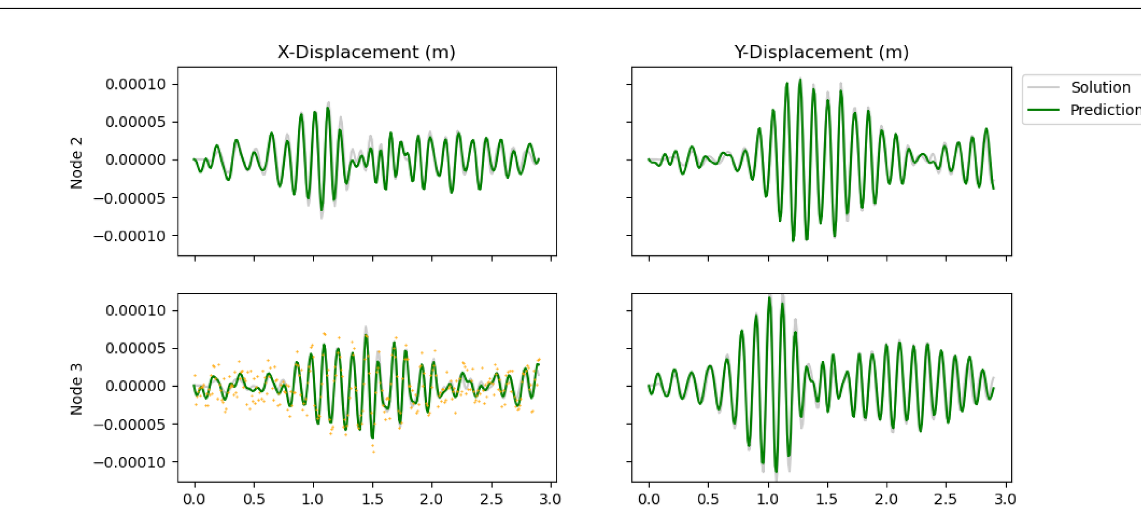


Figure 2. Displacement histories at both free nodes predicted by FE-PINN after training on data with 20% noise. Training data is shown in orange.

Conclusion

- Sensitive to the initial estimate of E
- Robust to the inclusion of *at least* 20% noise
- One step closer to applying FE-PINN to experimental data

lorestaa@my.erau.edu | paridas@erau.edu