Detecting Surface-Breaking Cracks using PINNs

Network Architecture and Train Optimization

Itamar Fiorino

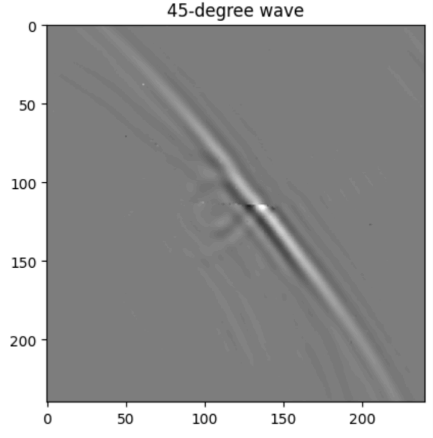
May 2024

Table of contents

Background

Motivation

- On aircraft, small cracks can lead to catastrophic failures, which means on-demand detection of cracks is critical
- The approach taken here is to send ultrasonic waves through the metal
- We identify cracks when waves slow down or there is back-scattering [2]



Wave Equation

The wave equation, a PDE on (in this case) displacement U, characterizes the movement of a wave through a medium with wave velocity V:

$$U_{tt} = V \cdot \Delta U$$

where V may depend on space, and Δ is the Laplacian of the spatial coordinates. It is a *second-order linear hyperbolic* PDE. In one spatial dimension, the wave equation is easily solved through a change of variables, in higher dimensions the solutions are more involved.

Inverse Problem

In our case, our objective is to both find a PDE for the wave, and to find the velocities:

- Velocity through the medium
- Disruptions in the velocity (due to the crack)

This is an *inverse problem*.

PINN Architecture

Loss Metrics

To control the boundary conditions, which in this case correspond to the wave coinciding with the measurements, we use a standard MSE loss,

$$MSE_u = \frac{1}{n}\sum |U(t_k, x_k, y_k) - U_k|^2$$

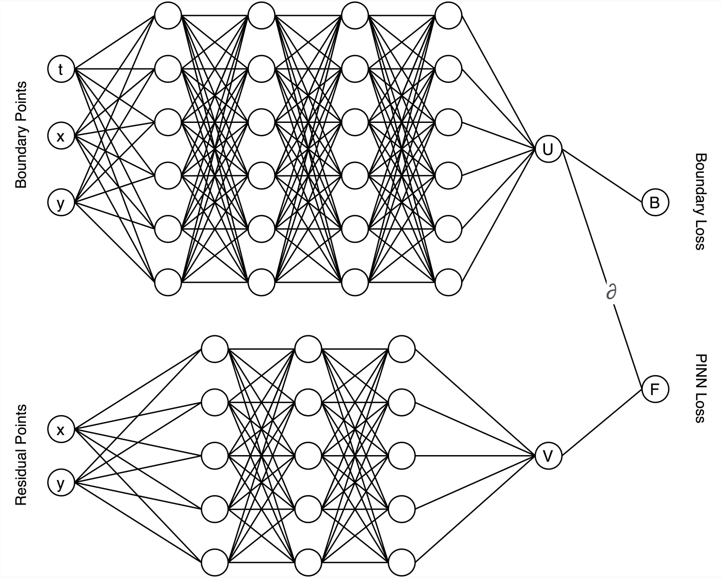
To control the PINN, we use the residual between the left hand side of the PDE, U_{tt} and the right hand side $V \cdot \Delta U$:

$$MSE_f = \frac{1}{2}\sum |f(t_k, x_k, y_k)|^2$$

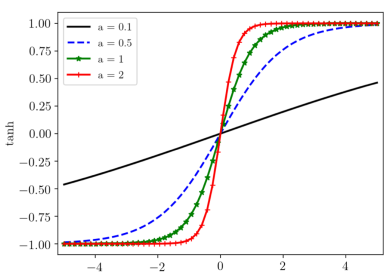
where

$$f(t_k, x_k, y_k) = U_{tt}(t_k k, x_k, y_k) - V(x_k, y_k) \cdot \Delta U(t_k, x_k, y_k)$$

Architecture



Adaptive Activation



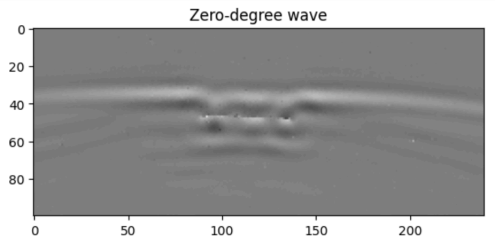
Like many PINNs, we use the (smooth) Tanh function for activation. In order to speed up training (as demonstrated in [1]), we use an adaptive (learnable) activation function.

$$x^k = \sigma(na\mathcal{L}(x^{k-1}))$$

Training

Preprocessing

- PCA reduces noise and dimensionality
- Only frames where the wave is present/contacting are included, and image size is reduced.
- Only a small percentage of the data is used, both to reduce batch size and to demonstrate the efficacy of the PINN.



Challenges

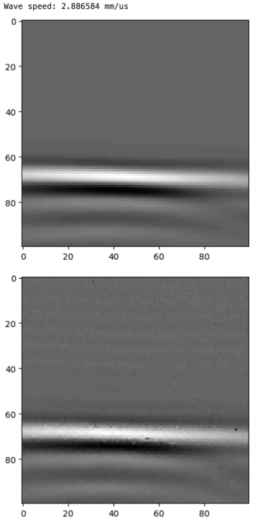
The PINN was very slow to converge, if it did at all. Several things were necessary to improve convergence:

- Pretraining (particularly for constant speed, which was simpler)
- Boundary condition weighting
    - Removing one component of the loss altogether (can increase speed if derivatives are not calculated)
    - Exponential weight scheduling
    - Cyclic weight scheduling
- Isolating learning to only U or V
- Other hyperparameters

Results

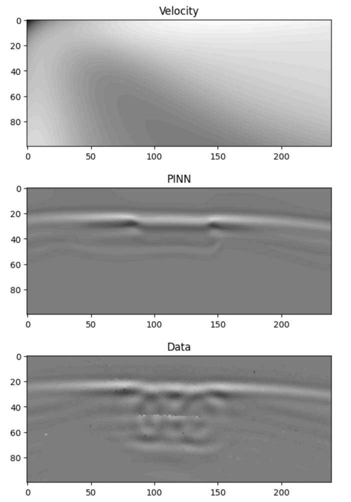
Wave Speed

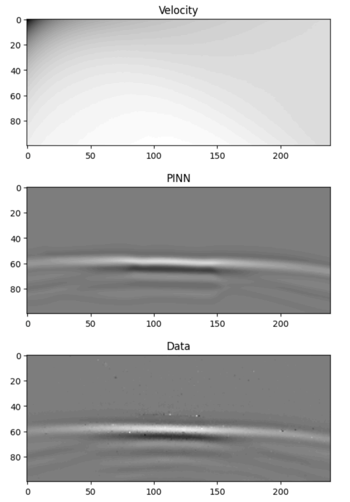
For the network that learned wave speed independently of position, a (4×64) network was used *without* adaptive activation. Pretraining and exponential loss weighting $(10^{-5} \rightarrow 10^{-1})$ was very effective. Determined Wave speed coincides with the laboratory consensus of $2.9\frac{mm}{\mu s}$



Local Minima

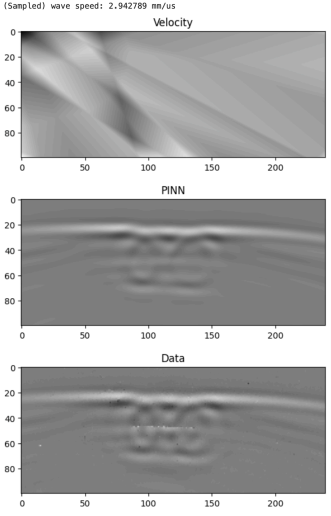
Using pretraining and exponential loss weighting (two frames):

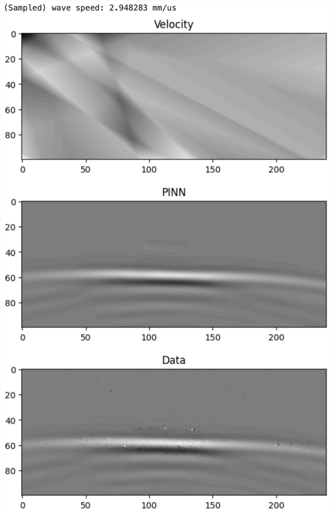




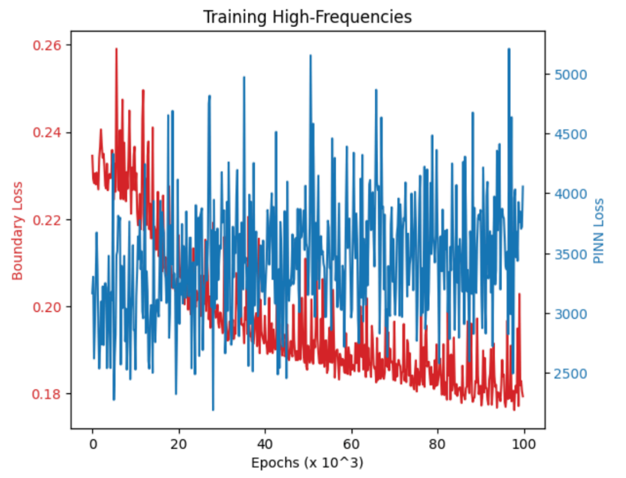
Simultaneous Convergence

Training both losses simultaneously ($\lambda = 10^{-3}$), $\sigma = \text{ReLU}$):





Simultaneous Convergence



George Em Karniadakis Ameya D Jagtap, Kenji Kawaguchi.
Adaptive activation functions accelerate convergence in deep and physics-informed neural networks.
Journal of Computational Physics, 2020.

James Blackshire Daniel Sparkman Khemraj Shukla, Patricio Clark Di Leoni and George Em Karniadakis.
Physics-informed neural network for ultrasound nondestructive quantification of surface breaking cracks.
2020.