Improving physics-informed training for forward and inverse problems in earthquake dynamics

Cody Rucker (crucker@uoregon.edu) and Brittany A. Erickson (bae@uoregon.edu)
Department of Computer Science, University of Oregon

We show that exact enforcement of boundary conditions improves approximation accuracy of PINNs at an increased cost to training time.

Plane strain domain

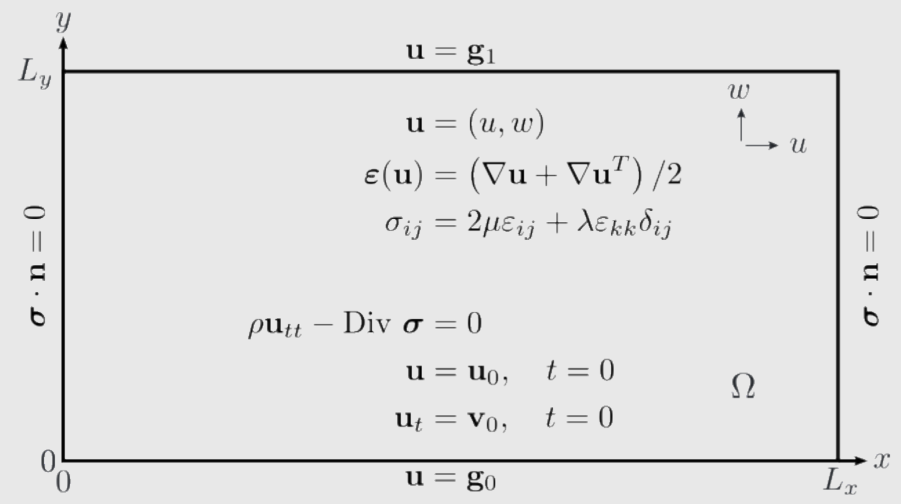

Fig. 1: 2D schematic of a dynamic plane strain problem. Strain (ε)-displacement($\mathbf{u}$) relation is shown and stress $\boldsymbol{\sigma}$ is given in terms of shear modulus μ and Lamé's first parameter λ.

Physics-informed neural networks

PINN architecture:
Given a generic initial-boundary-value problem(IBVP)

$$\mathcal{L}\left[\mathbf{u};\lambda\right](\mathbf{x}) = \mathbf{k}(\mathbf{x}), \quad \mathbf{x} \in \widehat{\Omega},$$
$$\mathcal{B}\left[\mathbf{u};\lambda\right](\mathbf{x}) = \mathbf{g}(\mathbf{x}), \quad \mathbf{x} \in \partial\widehat{\Omega},$$

we define a neural network $\mathcal{N}$ which aspires to be the IBVP solution u. To this end we define the loss components

$$MSE_{\widehat{\Omega}}(\theta) = \frac{1}{N_{\widehat{\Omega}}}\sum_{i=1}^{N_{\widehat{\Omega}}}\left|\mathcal{L}[\mathcal{N};\lambda](\mathbf{x}^i_{\widehat{\Omega}};\theta) - \mathbf{k}^i\right|^2,$$
$$MSE_{\partial}(\theta) = \frac{1}{N_{\partial}}\sum_{i=1}^{N_{\partial}}\left|\mathcal{B}[\mathcal{N};\lambda](\mathbf{x}^i_{\partial};\theta) - \mathbf{g}^i\right|^2,$$

over collocation points $\{\mathbf{x}^i_{\widehat{\Omega}}\}_{i=1}^{N_{\widehat{\Omega}}}$ and $\{\mathbf{x}^i_{\partial}\}_{i=1}^{N_{\partial}}$. Then, $\mathcal{N}(\mathbf{x};\theta^*) = u(\mathbf{x})$ when

$$\theta^* = \arg\min_{\theta} MSE_{\widehat{\Omega}}(\theta) + MSE_{\partial}(\theta).$$

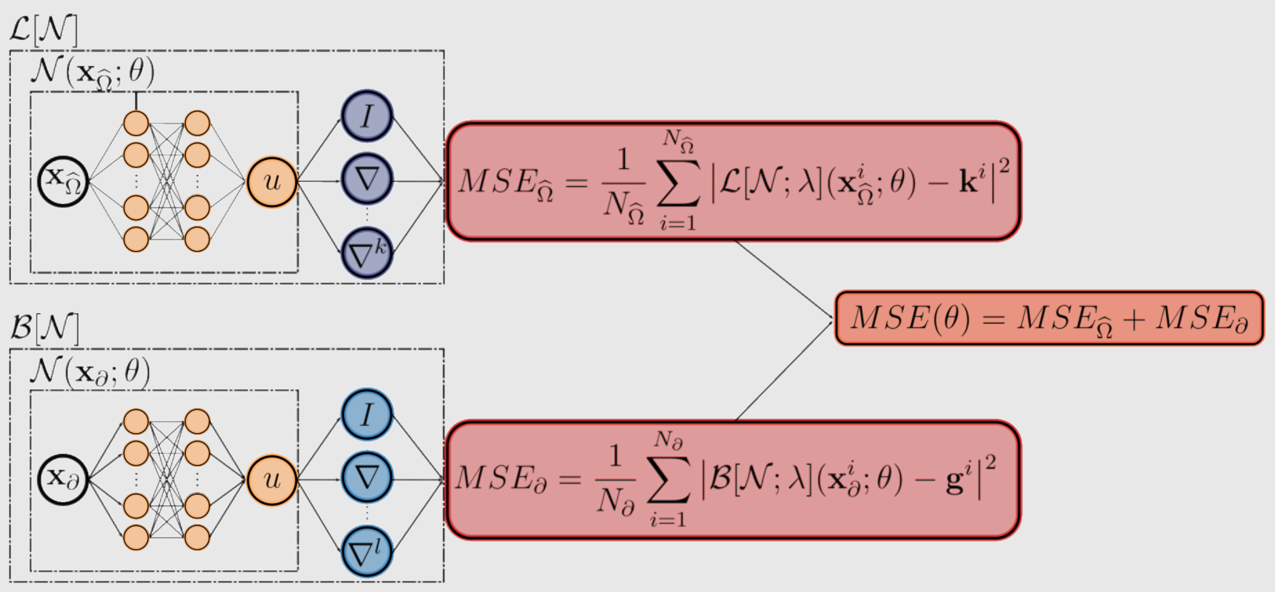

Fig. 2: A schematic of the PINN architecture for solving a generalized boundary value problem. Displacement approximation network $\mathcal{N}$ is trained on interior and boundary subdomains which are governed by operators $\mathcal{L}$ and $\mathcal{B}$, respectively.

Implicit Boundary Representation

The general solution form of an initial boundary value problem can be given by

$$u = \sum_{i=1}^{n} u_i W_i + \mathcal{N}\prod_{i=1}^{n}\phi_i^{\mu_i},$$

where ϕ_i approximates the distance to boundary i for which conditions u_i of order μ_{i-1} are interpolated by W_i.

Approximate Distance Functions (ADF): Given an implicit representation f of a curve, we construct a convex trimming region that contains a desired segment of f. Using R-function theory we can approximate the distance ϕ to the segment of f contained in t. Figure 3 shows the relevant functions for a simple Bezier curve.

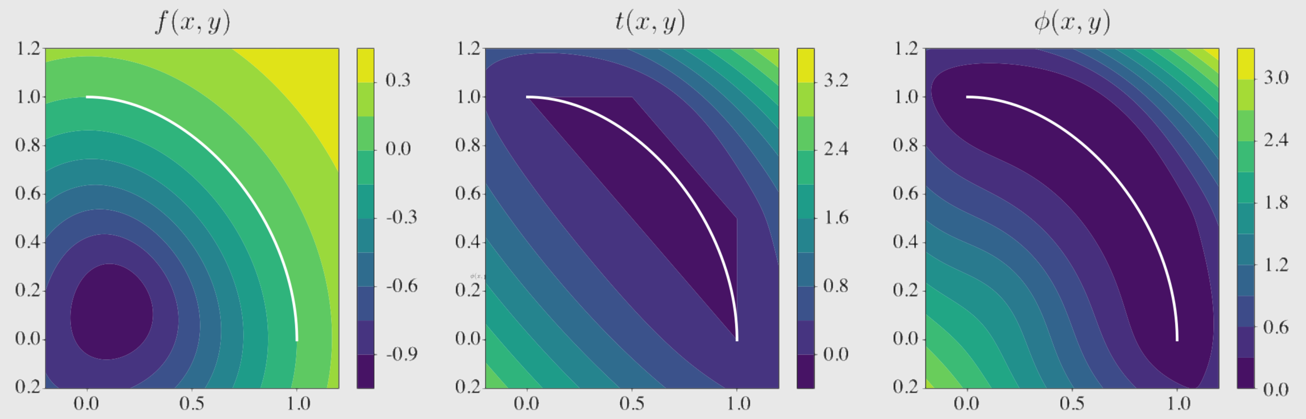

Fig. 3: Elements of an approximate distance function (ADF) to a Bezier curve defined by control points $P = [(0.0, 1.0)\ (0.5, 1.0)\ (1.0, 0.5)\ (1.0, 0.0)]$. From left top right, f shows the implicit Bezier curve, t the trimming function defined by the convex hull of P and ϕ the approximate distance function.

Interpolation Bases: From a set of boundary adf $\{\phi_i\}_{i=1}^{K}$ we define the interpolation basis $\{W_i\}_{i=1}^{K}$ by inverse distance weighting

$$W_i = \frac{\prod_{j=1;j\neq i}^{K}\phi_j^{\mu_j}}{\sum_{k=1}^{K}\prod_{j=1;j\neq k}^{K}\phi_j^{\mu_j}}.$$

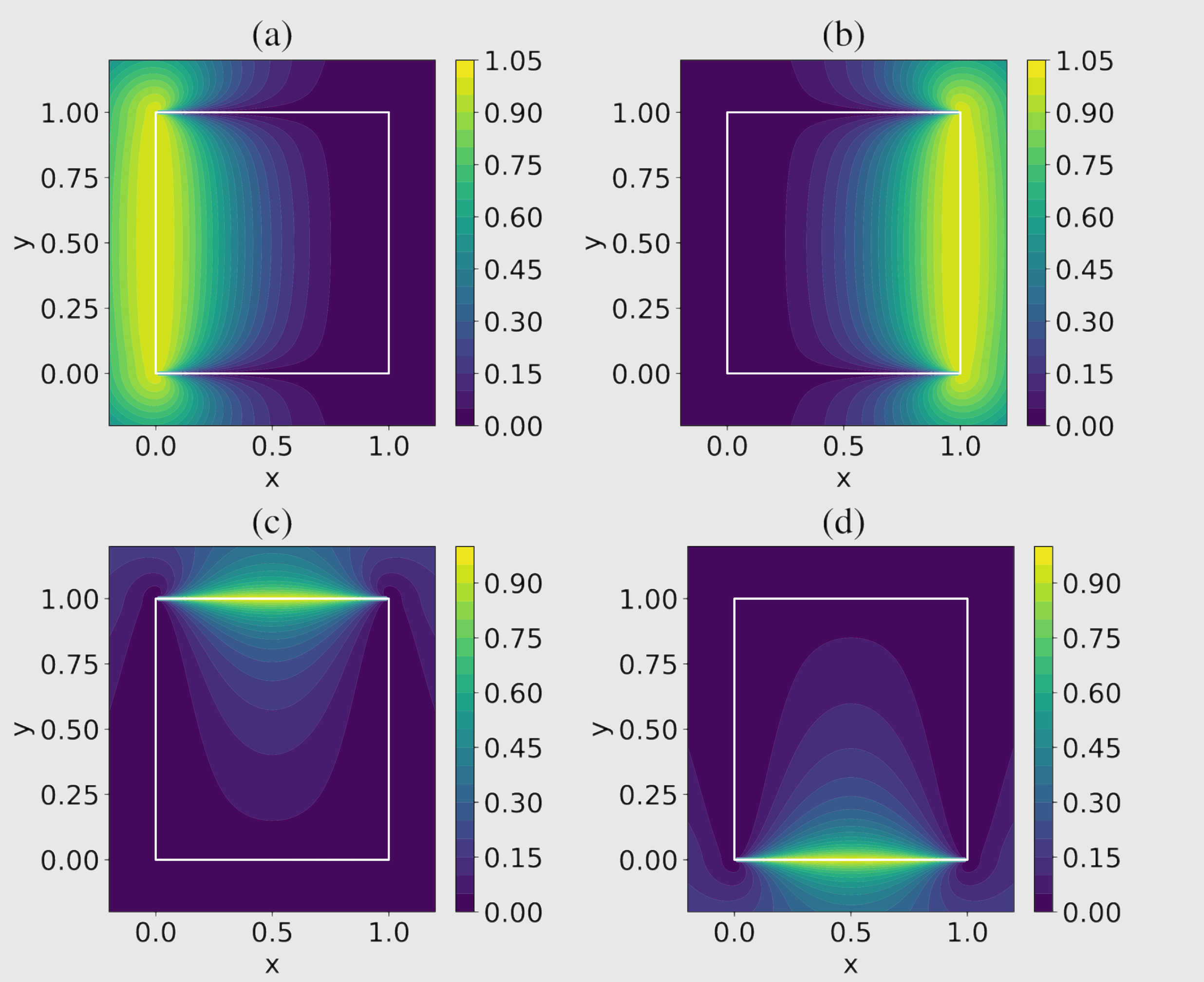

Fig. 4: Interpolation bases for each spatial boundary on Ω. (a) and (b) show left and right bases where $\mu = 2$ is used to interpolate Neumann conditions. Similarly, (c) and (d) show top and bottom with $\mu = 1$ for interpolating Dirichlet conditions.

Results

We first construct a solution to the plane strain problem where each boundary in space is constrained by a Dirichlet condition.

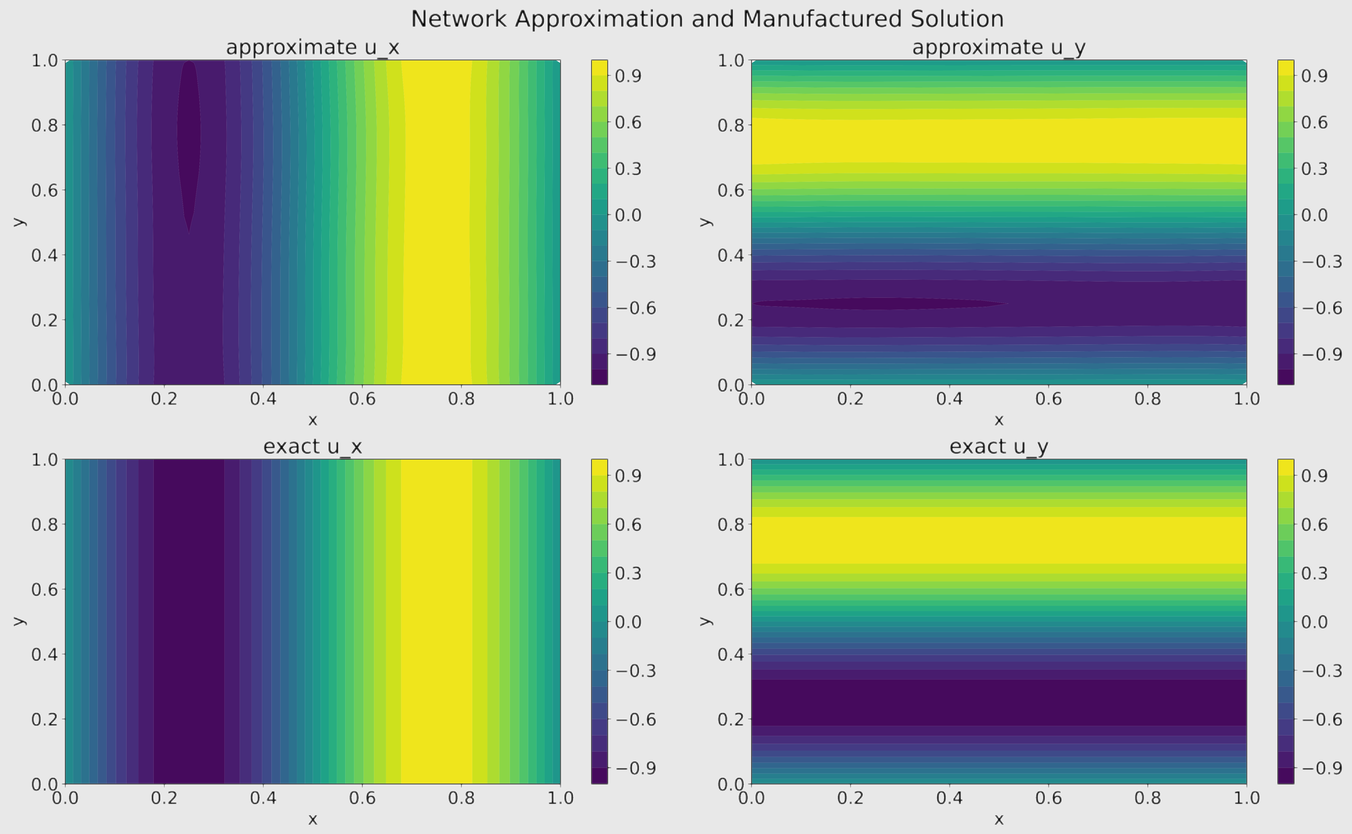

Fig. 5: Comparison of results for a neural network solution to the plane strain equations where each space-time boundary condition is hard-enforced. Network uses 3 hidden layers with 128 neurons each and was trained on 1000 random interior points for 1000 iterations of L-BFGS.

Hard-enforcement achieves lower $L2-$error but with an increase to total training time.

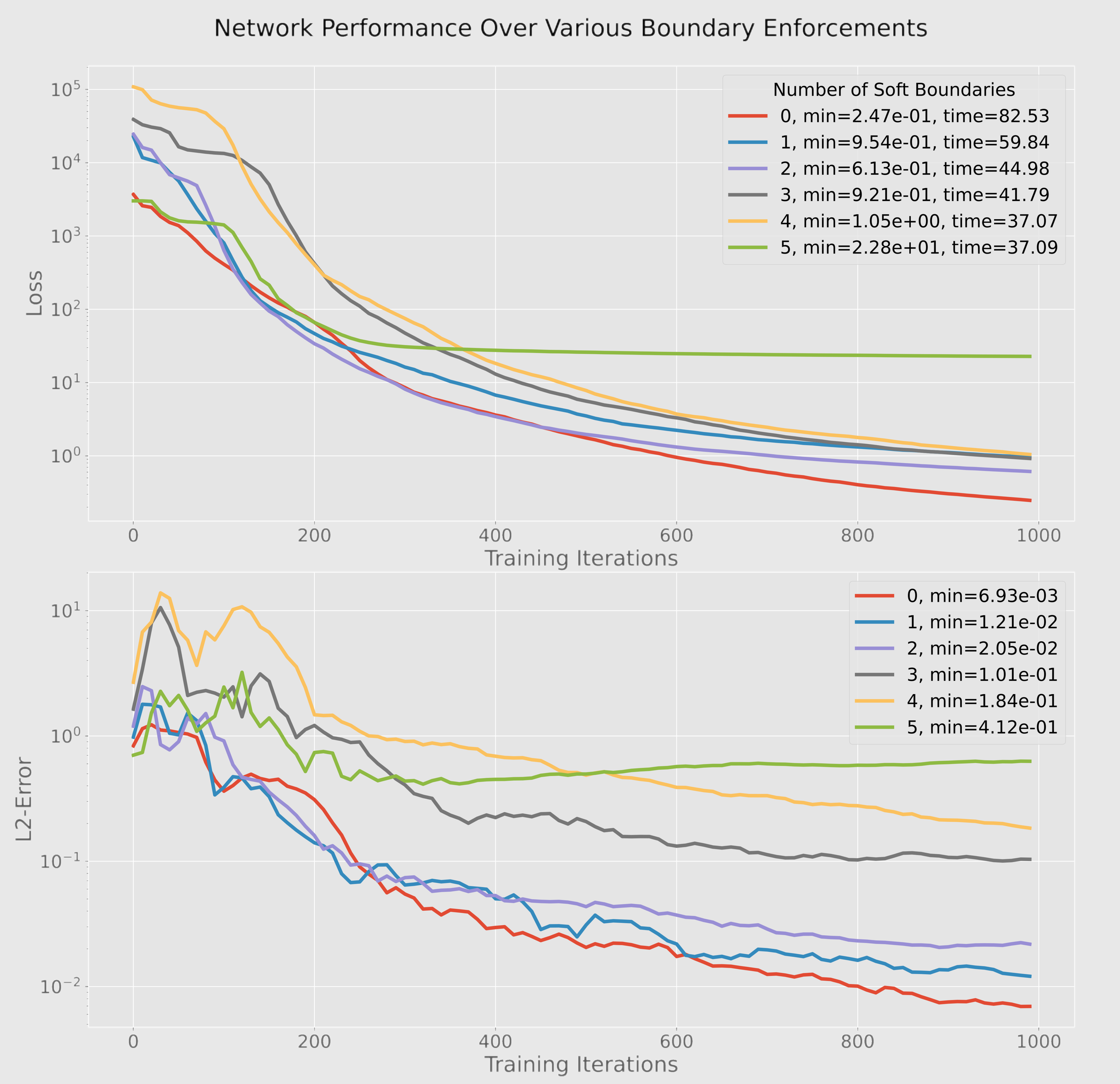

Fig. 6: Comparison of network performance over various configurations of hard and soft boundary enforcement. Each scenario uses a network with 3 hidden layers, 128 neurons each, trained on 1000 random points over 1000 iterations of L-BFGS. Number of training points is kept constant by projecting the randomly sampled interior points onto relevant boundaries in each scenario. Exact order for progressively adding soft boundaries is: top, bottom, right, left, initial, with the initial boundary representing two conditions: displacements and velocities.

Soft and Hard Enforcement of Dirichlet and Neumann Conditions

General Solution Form:

$$\widetilde{\mathcal{N}} = \sum_{i=1}^{n} u_i W_i + \mathcal{N}\prod_{i=1}^{n}\phi_i^{\mu_i},$$

Dirichlet Solution Structure:

$$u_i = \mathbf{g}_i$$

Neumann Solution Structure:

$$u_i = \mathcal{N} + \phi_i\left(D_1^{\phi_i}[\mathcal{N}] - \mathbf{h}_i\right)$$
$$D_1^{\phi}(\cdot)\left[-\nabla\phi\cdot\nabla(\cdot)\right]_{\partial\Omega}$$

PDE Residual:

$$MSE_{\Omega} = \frac{1}{N_{\Omega}}\sum_{i=1}^{N_{\Omega}}\left|\rho\widetilde{\mathcal{N}}_{tt} - \text{Div}\ \widetilde{\boldsymbol{\sigma}}\right|^2,$$

Dirichlet Loss:

$$MSE_i = \frac{1}{N_i}\sum_{k=1}^{N_i}\left|\widetilde{\mathcal{N}} - \mathbf{g}_i\right|^2,$$

Neumann Loss:

$$MSE_i = \frac{1}{N_i}\sum_{k=1}^{N_i}\left|\nabla\widetilde{\mathcal{N}}\cdot\mathbf{n} - \mathbf{h}_i\right|^2,$$

Objective function to be minimized

$$MSE = \sum_{\xi} MSI_{\xi},$$

Summary

- Exact enforcement of boundary conditions over implicitly defined sets.
- Mixed boundary enforcement for physics-informed neural networks.
- Hard and soft boundary enforcement are tested on a plane strain problem with Dirichlet conditions constraining spatial boundaries.
- Hard enforcement offers greater accuracy over soft enforcement but comes with an increase to the total training time.