2D MHD Equilibrium Solver using Physics-Informed Neural Networks

Byoungchan Jang[1], Alan Kaptanoglu[1], Matt Landreman[1], William Dorland[1] [1]*Department of Physics, University of Maryland*

Motivation

- Magnetohydrodynamic equilibrium calculations are a crucial tool for magnetic confinement.
- A fast solver for MHD equilibria is needed for real-time reconstructions and incorporation in optimization loops such as those that occur in stellarator shape optimization.
- Here we explored physics-informed neural networks (PINNs) as a solver for producing 2D MHD equilibria.

Grad-Shafranov Equation

- Grad-Shafranov Equation is a 2D ideal MHD equilibrium equation
- We are looking at a **particular set of analytic solutions:** *Solov'ev Profile*
- *The reason we use an analytic solution is not to* ***train*** *but to* ***check***

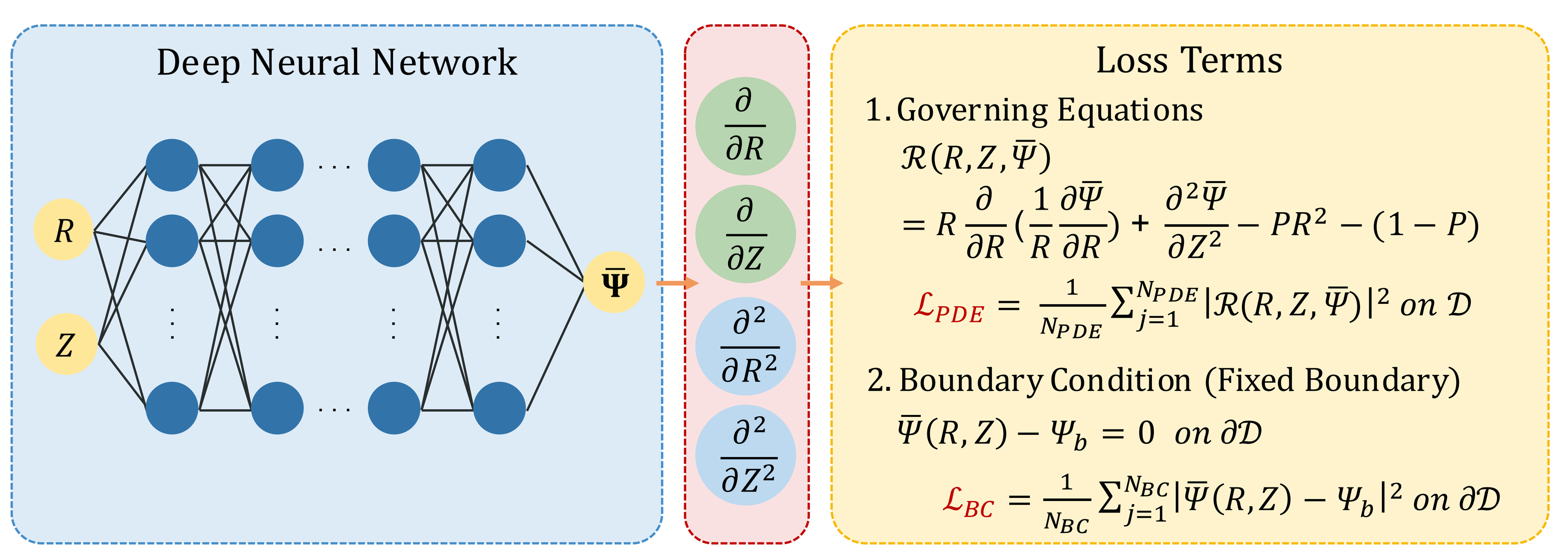
$$\psi_{rr} - \frac{1}{r}\psi_r + \psi_{zz} + \mu_0 r^2 \frac{dp(\psi)}{d\psi} + \frac{1}{2}\frac{dF^2(\psi)}{d\psi} = 0$$

$$\mu_0 r^2 \frac{dp(\psi)}{d\psi} = -C \qquad \frac{1}{2}\frac{dF^2(\psi)}{d\psi} = -A \qquad P = CR_0^2/(A + CR_0^2)$$

$$\Psi_{RR} - \frac{1}{R}\Psi_R + \Psi_{ZZ} - PR^2 - (1-P) = 0 \qquad P: Solov'ev\ Pressure\ Profile$$

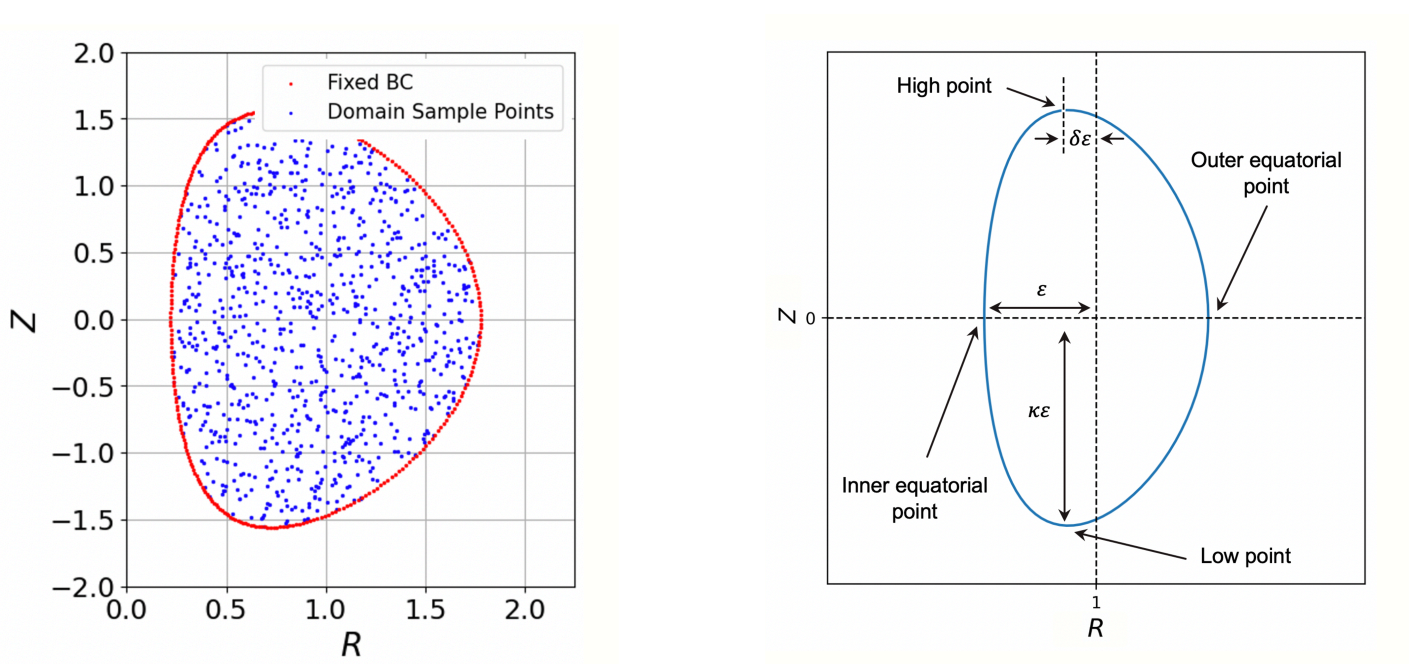
Physics-informed Neural Network

- PINNs calculate residual of **PDE** and Boundary condition
- **Optionally**, we can add **data**(i.e. experimental or numerical solution)
- PINNs shown here are implemented with **DeepXDE** by Lu, et al.



$$L_{total} = \lambda_{PDE}\mathcal{L}_{PDE} + \lambda_{BC}\mathcal{L}_{BC} + (\lambda_{Data}\mathcal{L}_{Data})$$

Collocation Points



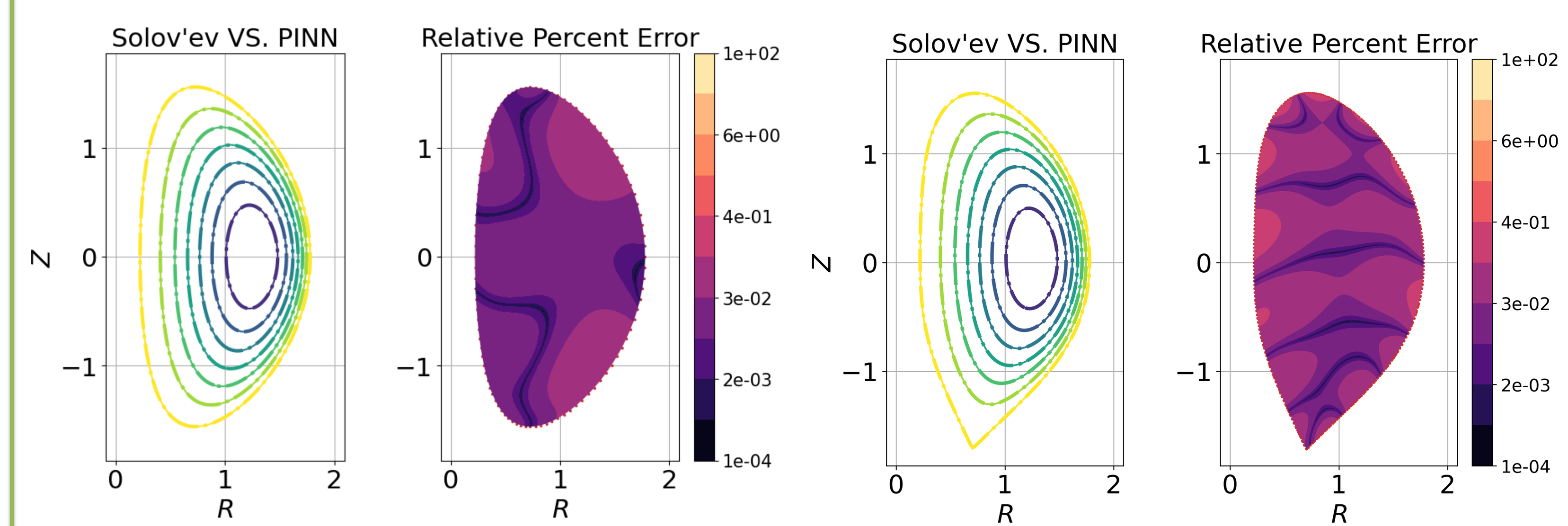
- Collocation points are where the PDE residuals are evaluated
- Possible to add data from numerical simulation or experiment

$$R = 1 + \varepsilon\cos(\tau + \sin^{-1}(\delta)\sin(\tau))$$

$$Z = \varepsilon\kappa\sin(\tau)$$

Baseline Equilibria

$$error = |\psi_{PINN} - \psi_{analytic}|/|\psi_a| \text{ where } \psi_a = \psi_{PINN} \text{ in magnetic axis}$$



- The plot on the left shows the solid and dot-dashed curves that indicate the PINN and analytic solutions, respectively.

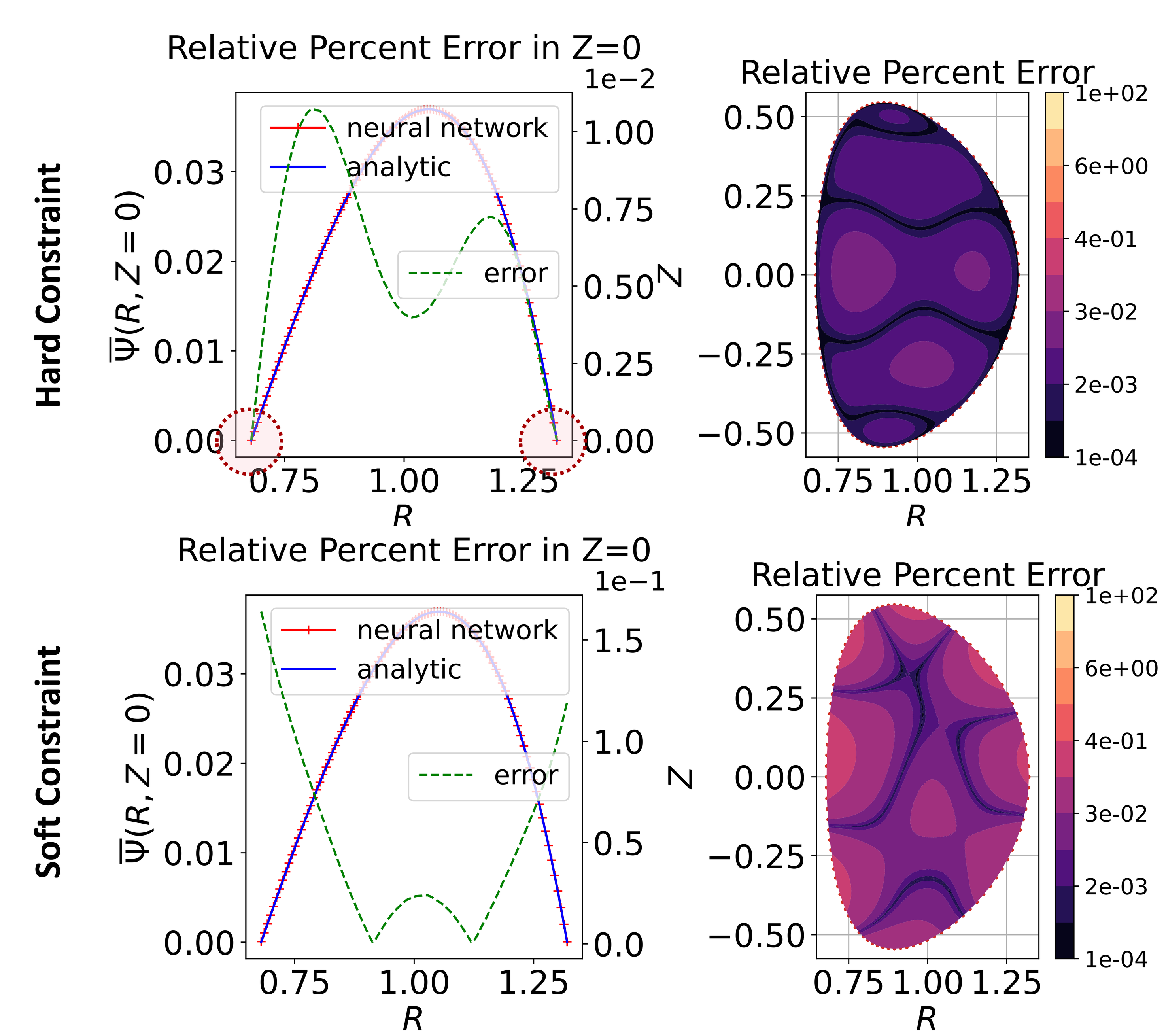
Configuration					
Devices	P	average error(%)	max error(%)	training time	inference time
ITER	0	0.015	0.162	20s	10ms
NSTX	0	0.032	0.142	22s	15ms
Spheromak	0	0.027	0.204	36s	12ms
ITER	1	0.021	0.136	17s	11ms
NSTX	1	0.026	0.186	32s	13ms
Spheromak	1	0.075	0.325	37s	10ms
FRC	1	0.057	0.214	27s	17ms

Hard Constraints

- For some problems, it is shown that the hard constraints produce faster and more accurate solutions to a given PDE.

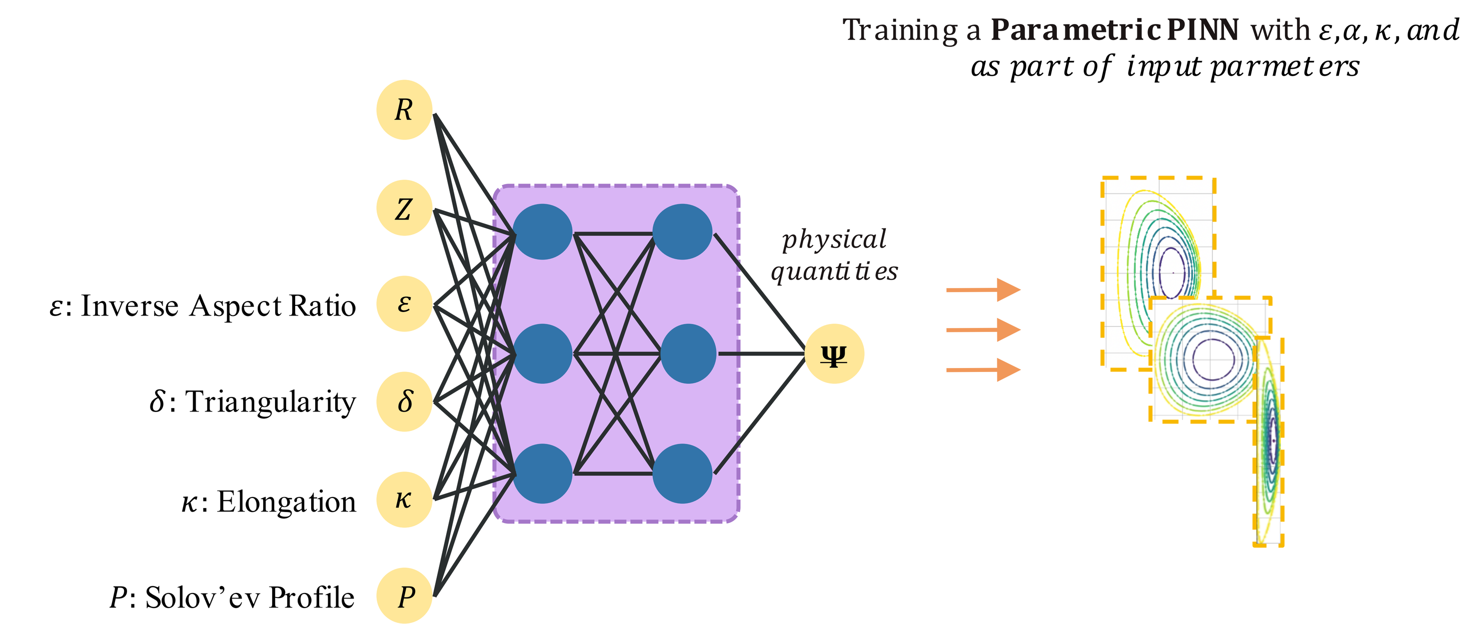
$$\bar{\Psi}(R,Z)_{hc} = G(R,Z)\bar{\Psi}(R,Z) \qquad G(R,Z) = 0 \text{ on boundary}$$

- Comparing side by side hard constraints show an order of magnitude better accuracy with lower training time.

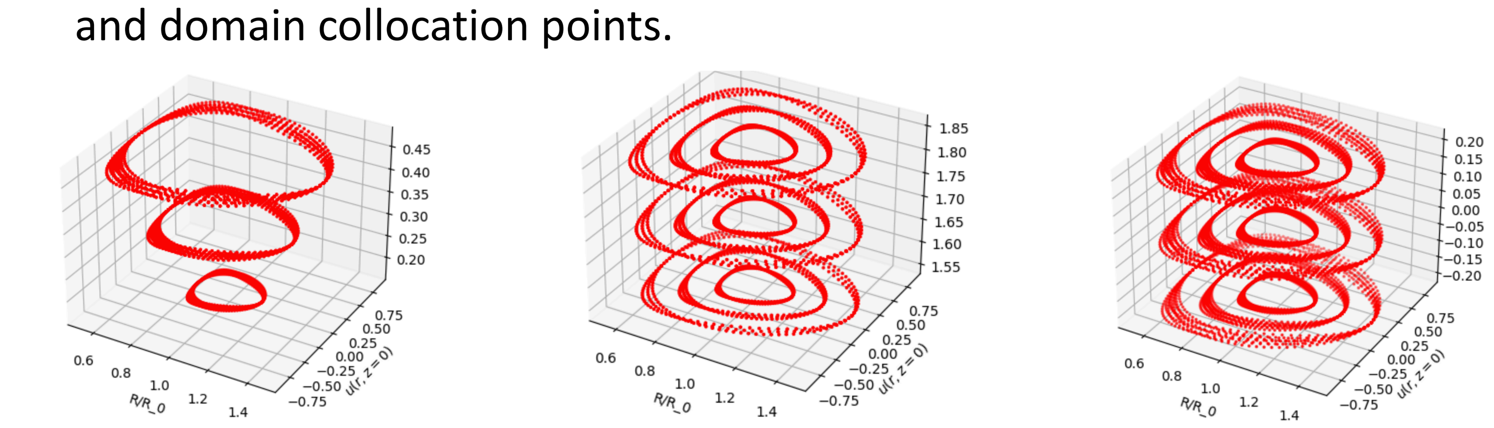


Parametric PINN

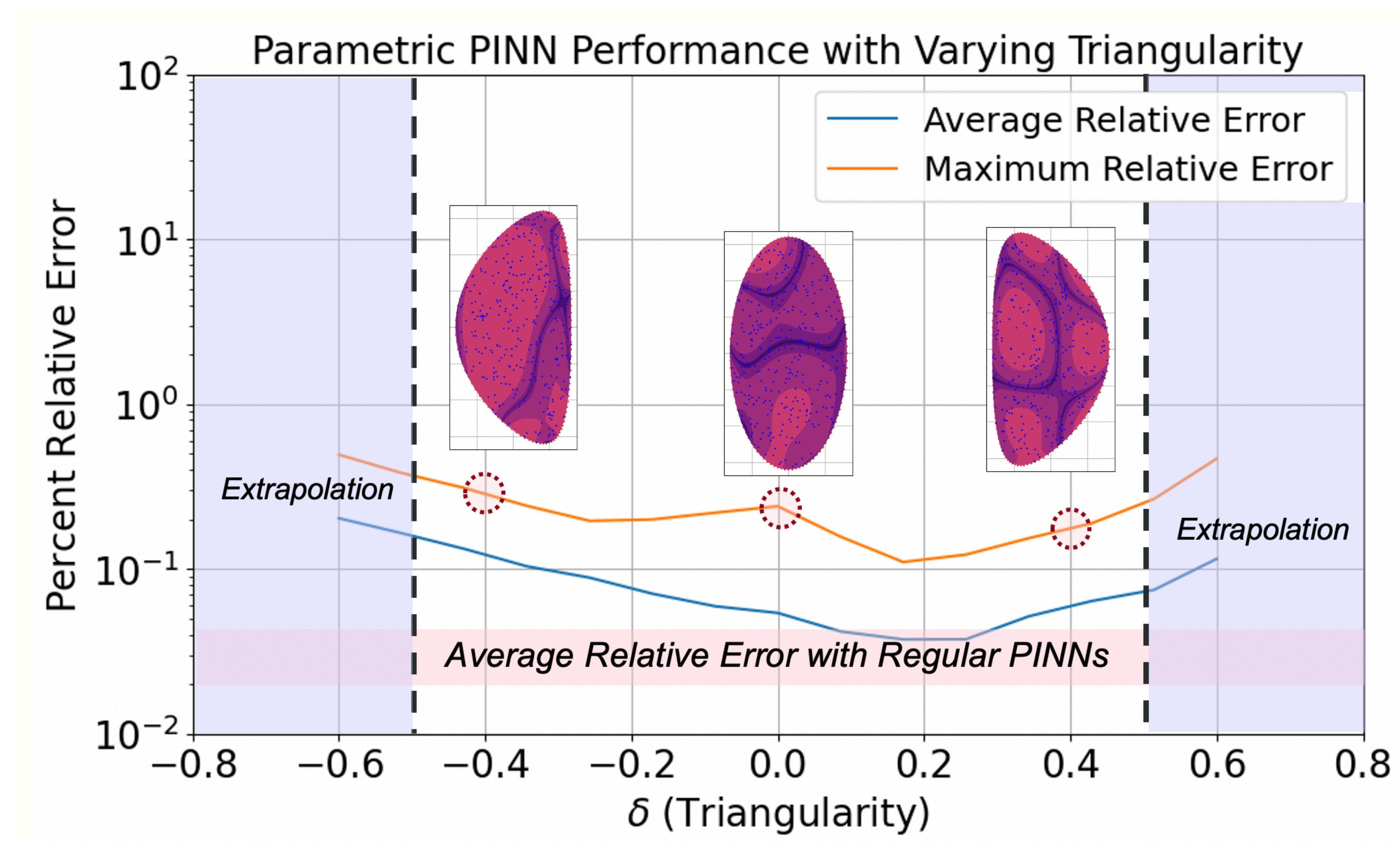
- Parametric PINNs expand on regular PINNs to include more input parameters.
- Allowing shape parameters as inputs can enable shape optimization with faster equilibrium reconstruction for various geometries.



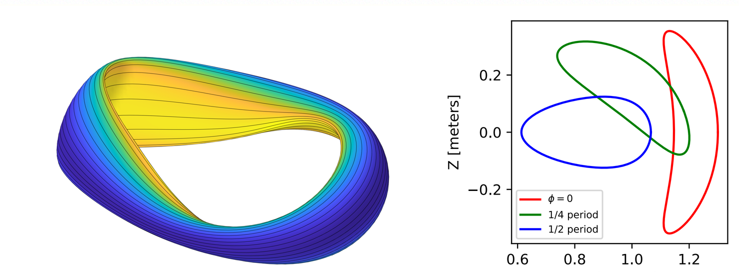
- For the result below, we used 409600 (i.e. $(100)(8)^4$) for both boundary and domain collocation points.



- Training took about 4 hours, and the inference is like regular PINNs.



Future Works



- Expand to 3D MHD to reconstruct equilibrium for stellarator optimization
- Finetune parametric PINNs for better performance: adaptive sampling and hard constraints may be helpful as well.

References

[1] M. Raissi et al. *Journal of Computational Physics* **378:**686-707 (2019).

[2] A. Cerfon et al. *Physics of Plasmas* **17.3**:032502 (2010).

[3] Lu, Lu, et al. *SIAM Review* **63.1:**208-228 (2021).

[4] D. A. Kaltsas et al. *Physics of Plasmas*, **29**(2), 022506 (2022)

[5] S. Markidis. *Frontiers in Big Data* **92**(2021).