SAND2021-2176C

nPINNS: Nonlocal Physics-Informed Neural Networks

SIAM Conference on Computational Science and Engineering (CSE21)
March 4, 2021

PRESENTED BY
Michael Parks
Center for Computing Research
Sandia National Laboratories

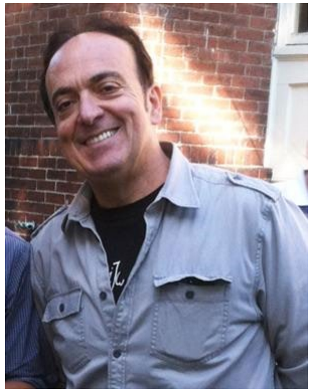
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

SAND2021-XXXX C

Collaborators and Funding

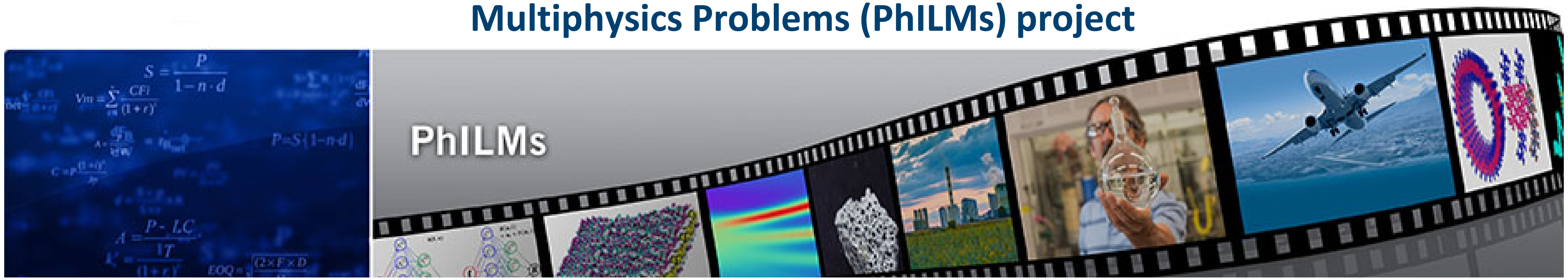
Dr. Goufei Pang
(Brown)

Dr. Marta D'Elia
(Sandia)



Prof. George Karniadakis
(Brown)

This work was supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research under the Physics-Informed Learning Machines for Multiscale and Multiphysics Problems (PhILMs) project



https://www.pnnl.gov/computing/philms/

Pacific Northwest NATIONAL LABORATORY

Sandia National Laboratories

BROWN

Outline

- ❑ **Nonlocal Models**
- ❑ **Using computational models in practice**
- ❑ **nPINNS: nonlocal Physics-Informed Neural Networks**
  - ❑ **Data-driven solutions**
  - ❑ **Data-driven discovery**
- ❑ **A nonlocal model for turbulent Couette flow**
- ❑ **Conclusions**

Discovery of Nonlocal Model Parameters from Data

- I can calibrate any isotropic elastic solid for given ω, δ.
- With detailed knowledge of microstructure, in some cases can derive δ.

- This raises several questions:
  - Is the choice of influence function or horizon important?
  - Does choice of these parameters make a difference in getting a physically correct answer or a physically incorrect answer?
  - Is any specific choice just as good as any other? Is there a single best choice for a specific application? Are there multiple good choices?
  - How do you tell?

- Data-driven methods present the opportunity to discover these parameters from data.
- Let's talk about Physics-Informed Neutal Networks (PINNs), both for (1) data-driven solution to PDEs, and (2) data-driven discovery of model parameters.

- Example: Nonlocal Isotropic Elastic Material

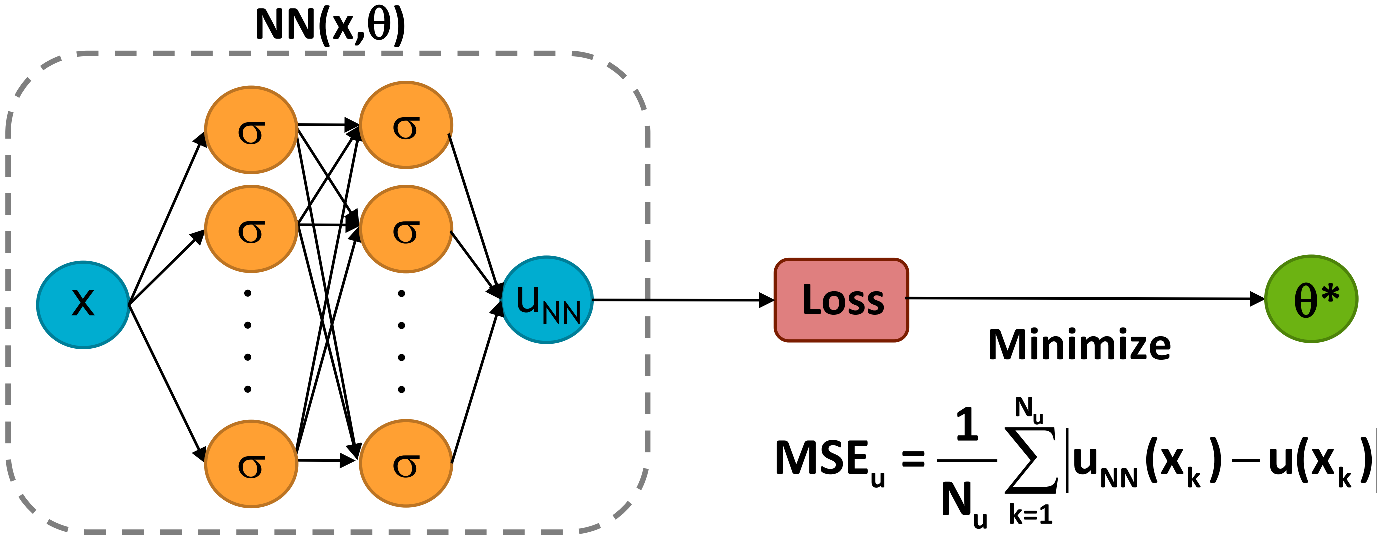
- Governing equations and parameters
  - $\rho\ddot{u}(x,t) = \int_{H_\delta} \left(T[x,t]\langle x' - x\rangle - T[x',t]\langle x - x'\rangle \right) dV_{x'}$
  - $T[x,t]\langle x' - x\rangle = \left(\frac{3k\theta}{m} \underline{\omega}\underline{x} + \frac{15\mu}{m} \underline{\omega}\underline{e}^d \right) \frac{x' - x}{\|x' - x\|}$
  - k is bulk modulus, μ is shear modulus
  - ω is peridynamic influence function
  - δ is peridynamic horizon

Train a Neural Network to Solve a PDE (Naïve Approach)

- Train deep neural network (DNN) to solve this PDE:

$$f\left(x;\frac{\partial u}{\partial x_1},\cdots,\frac{\partial u}{\partial x_d};\frac{\partial^2 u}{\partial x_1 \partial x_1},\cdots,\frac{\partial^2 u}{\partial x_1 \partial x_d};\cdots;\lambda\right)=0$$

- Naïve approach: Train network minimizing loss based on provided training data



$$MSE_u = \frac{1}{N_u}\sum_{k=1}^{N_u}\left|u_{NN}(x_k) - u(x_k)\right|^2$$

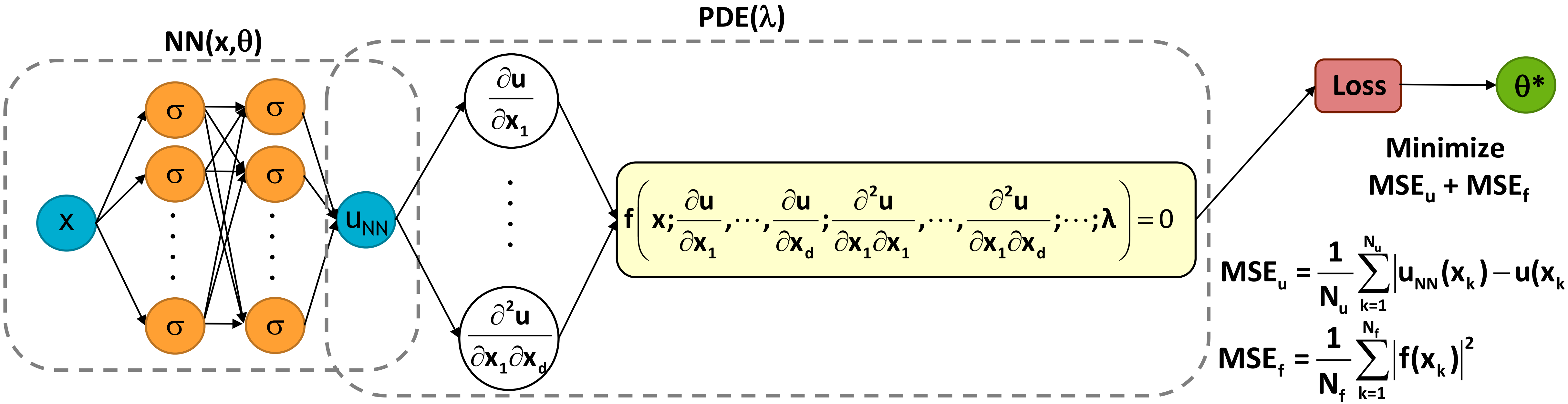
- In practice, this requires lots of data.
- There is no explicit notion of governing physics anywhere in this system.

PINNs*: Train a Neural Network to Solve a PDE

- **Train deep neural network (DNN) to solve this PDE:**

$$f\left(x;\frac{\partial u}{\partial x_1},\cdots,\frac{\partial u}{\partial x_d};\frac{\partial^2 u}{\partial x_1 \partial x_1},\cdots,\frac{\partial^2 u}{\partial x_1 \partial x_d};\cdots;\lambda\right)=0$$

- **Physics-Informed Neural Network (PINN) explicitly incorporates physics by constraining network output**



$$MSE_u = \frac{1}{N_u}\sum_{k=1}^{N_u}\left|u_{NN}(x_k)-u(x_k)\right|^2$$

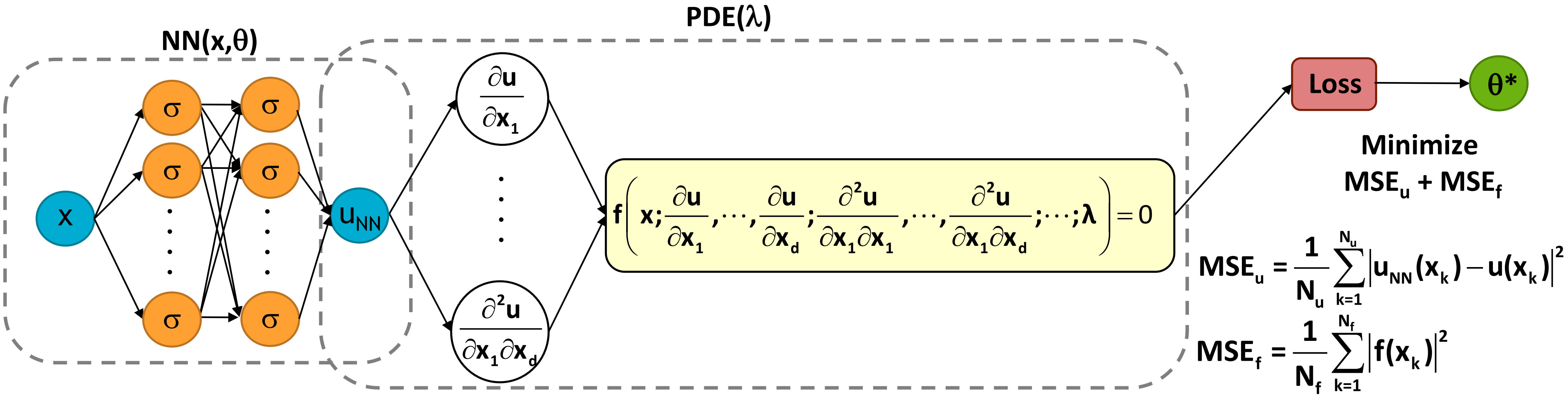
$$MSE_f = \frac{1}{N_f}\sum_{k=1}^{N_f}\left|f(x_k)\right|^2$$

- **In general, this requires much less data and can produce highly accurate solutions.**

* M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686-707.

PINNs*: Train a Neural Network to Solve a PDE

- PINNs can be used in two ways:
  - **Data-driven solutions to PDEs** (i.e., λ is known and we seek u(x)).
  - **Data-driven discovery of PDEs** (i.e., λ is unknown and we seek u(x) and λ). In this case, λ becomes a parameter of our PINN.



$$f\left(x;\frac{\partial u}{\partial x_1},\ldots,\frac{\partial u}{\partial x_d};\frac{\partial^2 u}{\partial x_1 \partial x_1},\ldots,\frac{\partial^2 u}{\partial x_1 \partial x_d};\ldots;\lambda\right)=0$$

$$MSE_u = \frac{1}{N_u}\sum_{k=1}^{N_u}\left|u_{NN}(x_k)-u(x_k)\right|^2$$

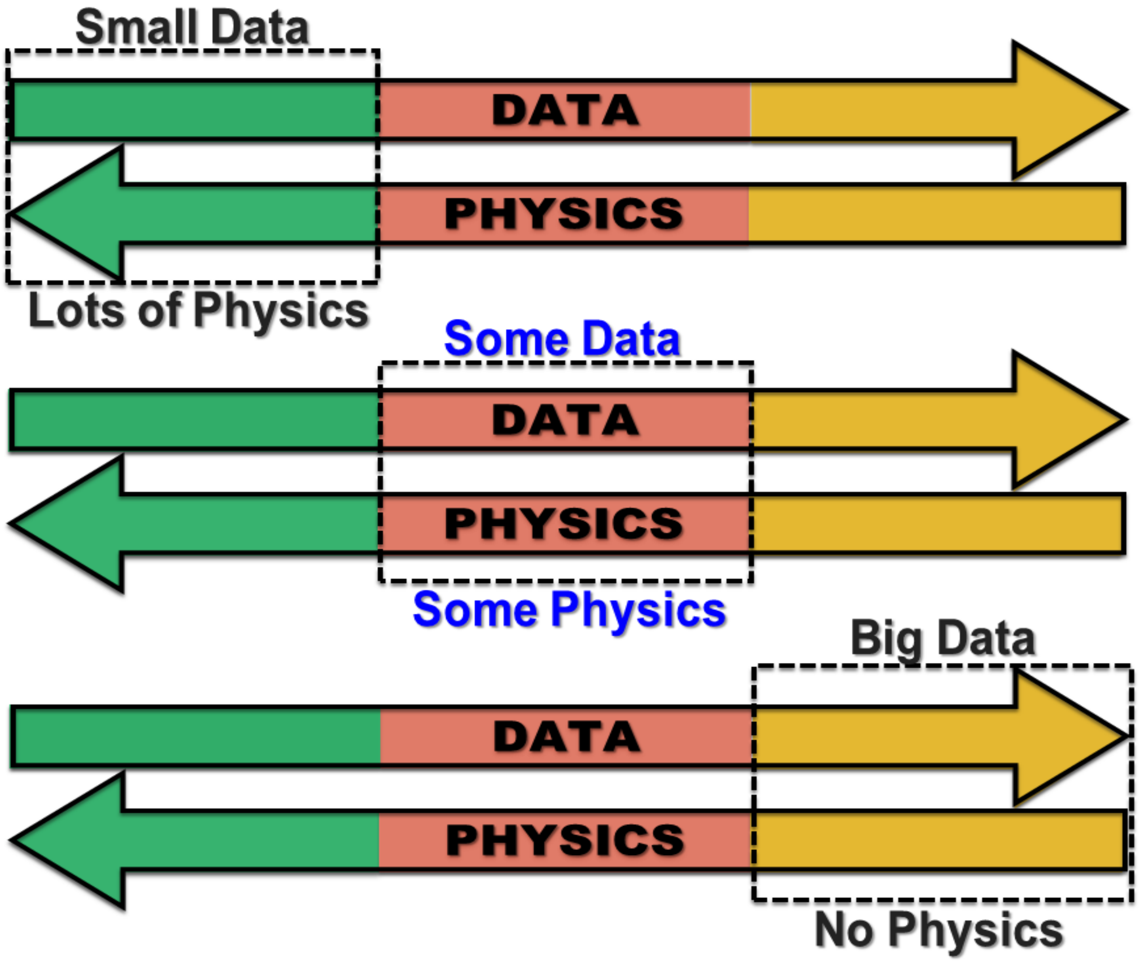
$$MSE_f = \frac{1}{N_f}\sum_{k=1}^{N_f}\left|f(x_k)\right|^2$$

* M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707.

Knowledge of Physics vs. Data

- ❑ **PINNs = Neural Networks + Data + Physical Laws**

- ❑ **How much do we know about governing physics?**
- ❑ **How much data do we have?**



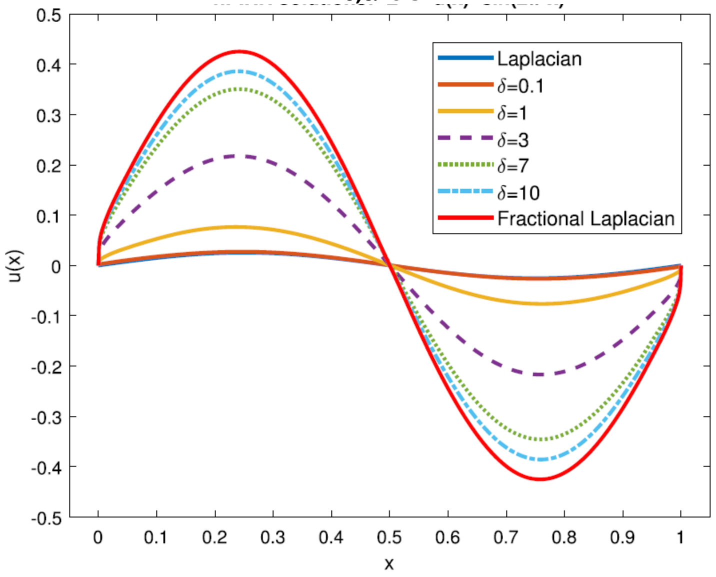
- ❑ **An alphabet of PINNS has been developed:**
  - ❑ cPINNs: conservative PINNs
  - ❑ vPINNs: variations PINNs
  - ❑ pPINNs: parareal PINNs
  - ❑ sPINNs: stochastic PINNs
  - ❑ fPINNs: fractional PINNs
  - ❑ LesPINNs: LES PINNs
  - ❑ **nPINNs: Nonlocal PINNs**
  - ❑ xPINNs: eXtended PINNs

- ❑ Next: **Universal Nonlocal Laplace Operator**

Universal Nonlocal Laplace Operator

- ❑ **Given broad spectrum of experimental data, we desire flexible operator.**
  - ❑ **i.e., operator discovery using parameterized classical Laplacian with data governed by a nonlocal Laplacian will not work well. But we don't know in advance the functional form data obeys.**

Solutions to $-\mathcal{L}_{\delta,\alpha=0.5}u(x) = -\sin(2\pi x)$



- ❑ **Use this operator*:**

$$-\mathcal{L}_{\delta,\alpha}u(x) = C_{\delta,\alpha}\int_{B_\delta(x)} \frac{u(y)-u(x)}{\|y-x\|_2^{d+\alpha}} \quad \forall x \in \Omega$$

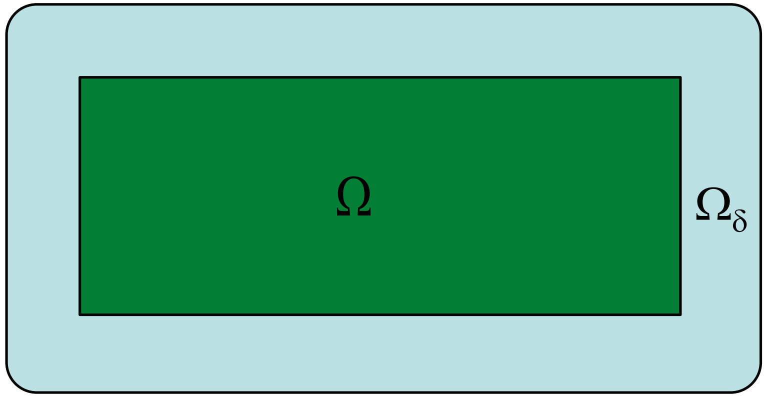
- ❑ **So that these properties are satisfied:**

$$\lim_{\delta\to 0}\left(-\mathcal{L}_{\delta,\alpha}\right)u(x) = -\Delta u(x) \qquad \forall \alpha \in (0,2) \qquad \text{(Classical Laplacian)}$$

$$\lim_{\delta\to \infty}\left(-\mathcal{L}_{\delta,\alpha}\right)u(x) = (-\Delta)^{\alpha/2}u(x) \qquad \forall \alpha \in (0,2) \qquad \text{(Fractional Laplacian)}$$

- ❑ **Apply nPINNs to this problem:**

$$-\mathcal{L}_{\delta,\alpha}u(x) = f(x) \quad x \in \Omega$$
$$u(x) = g(x) \quad x \in \Omega_\delta$$



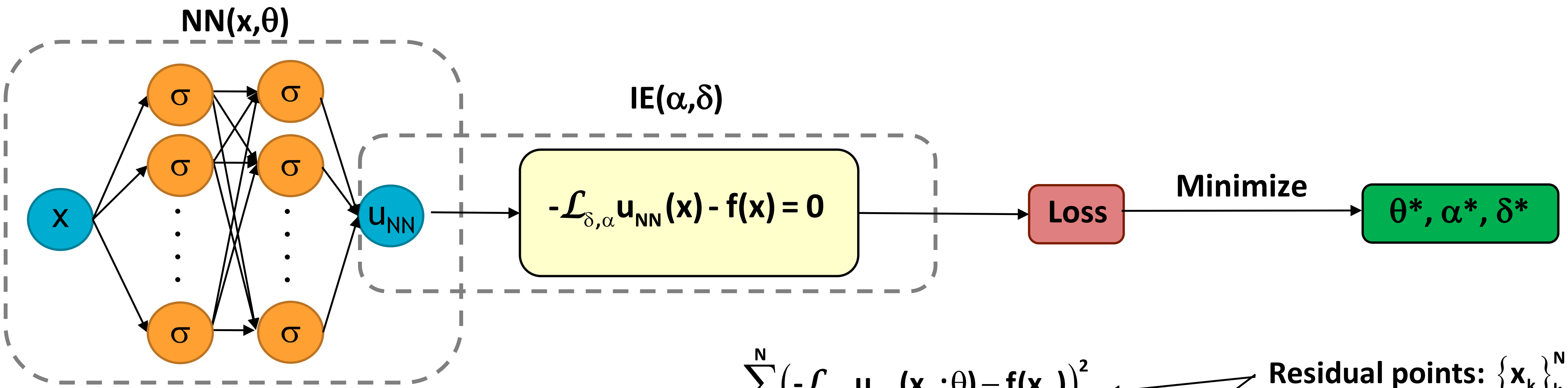
* This operator bridges fractional and local operators. For more on related unification results, see
M. D'Elia, M. Gulian, H. Olson, G. E. Karniadakis. A Unified Theory of Fractional, Nonlocal, and Weighted Nonlocal Vector Calculus, 2020 arXiv:2005.07686.

Nonlocal Physics-Informed Neural Networks (nPINNs)

- nPINNS can be summarized in 3 steps:

1. Collect observations or high fidelity simulations of the solution, u_{obs}
2. Approximate the solution with a fully connected NN: $u(x) \approx u_{NN}(x;\theta)$
3. Minimize the loss function with respect to the unknown parameters

$$-\mathcal{L}_{\delta,\alpha} u(x) = f(x) \quad x \in \Omega$$
$$u(x) = g(x) \quad x \in \Omega_\delta$$



Residual points: $\{x_k\}_{k=1}^{N}$

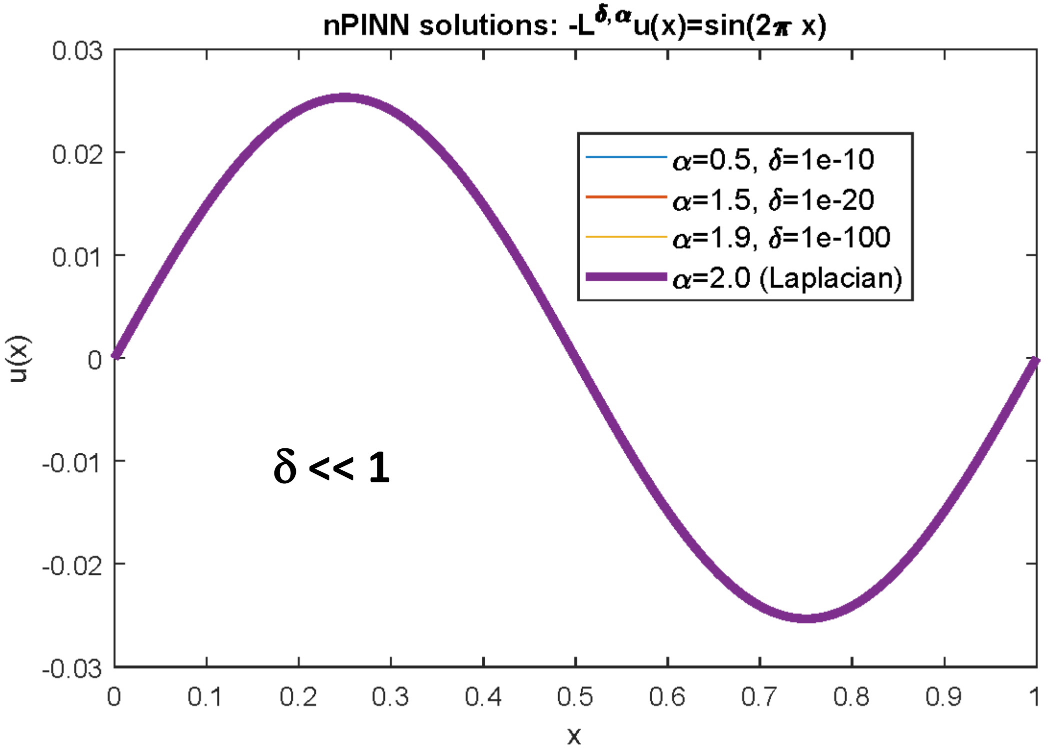
Observation points: $\{\hat{x}_k\}_{k=1}^{N_{obs}}$

- Forward mode (data-driven solution): $$Loss(\theta) = \frac{\sum_{k=1}^{N}\left(-\mathcal{L}_{\delta,\alpha} u_{NN}(x_k;\theta) - f(x_k)\right)^2}{\sum_{k=1}^{N} f(x_k)^2}$$

- Inverse mode (data-driven discovery): $$Loss(\theta,\delta,\alpha) = \frac{\sum_{k=1}^{N}\left(-\mathcal{L}_{\delta,\alpha} u_{NN}(x_k;\theta) - f(x_k)\right)^2}{\sum_{k=1}^{N} f(x_k)^2} + \frac{\sum_{k=1}^{N_{obs}}\left(u_{NN}(\hat{x}_k;\theta) - u_{obs}(\hat{x}_k)\right)^2}{\sum_{k=1}^{N_{obs}} u_{obs}(\hat{x}_k)^2}$$

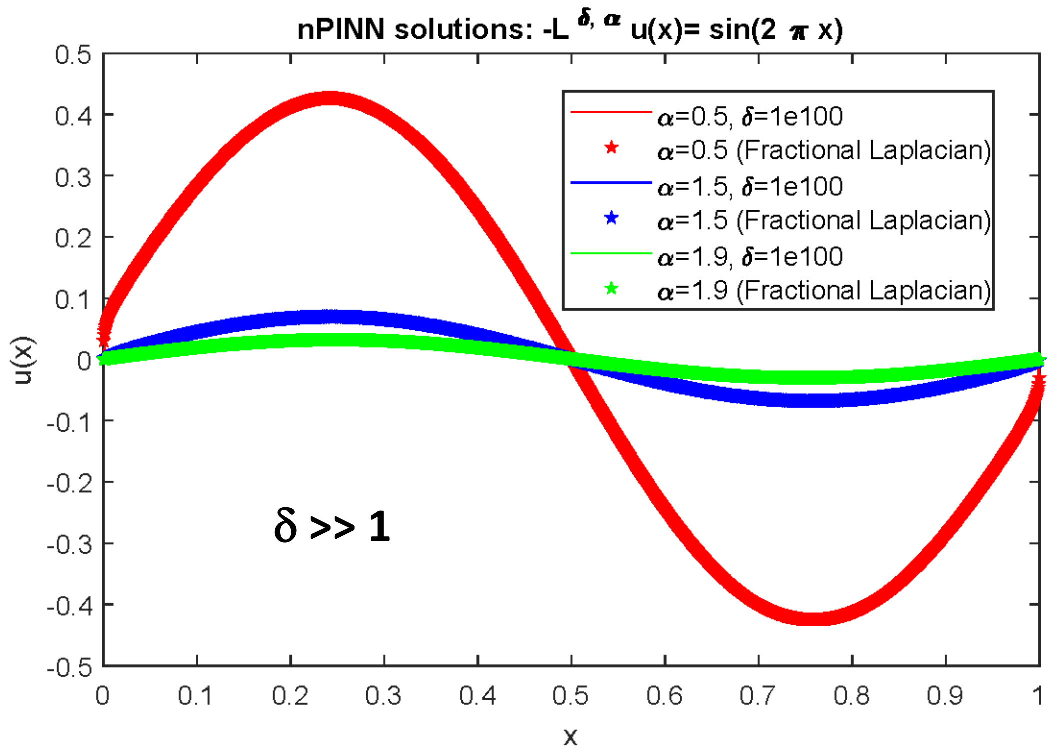
Computational Results: Data Driven Solutions

- nPINNs solutions show universal Laplace operator reproduces classical and fractional Laplacians



$$\lim_{\delta\to 0}\left(-\mathcal{L}_{\delta,\alpha}\right)u(x) = -\Delta u(x) \qquad \forall\alpha\in(0,2)$$

(Classical Laplacian)



$$\lim_{\delta\to\infty}\left(-\mathcal{L}_{\delta,\alpha}\right)u(x) = \left(-\Delta\right)^{\alpha/2}u(x) \quad \forall\alpha\in(0,2)$$

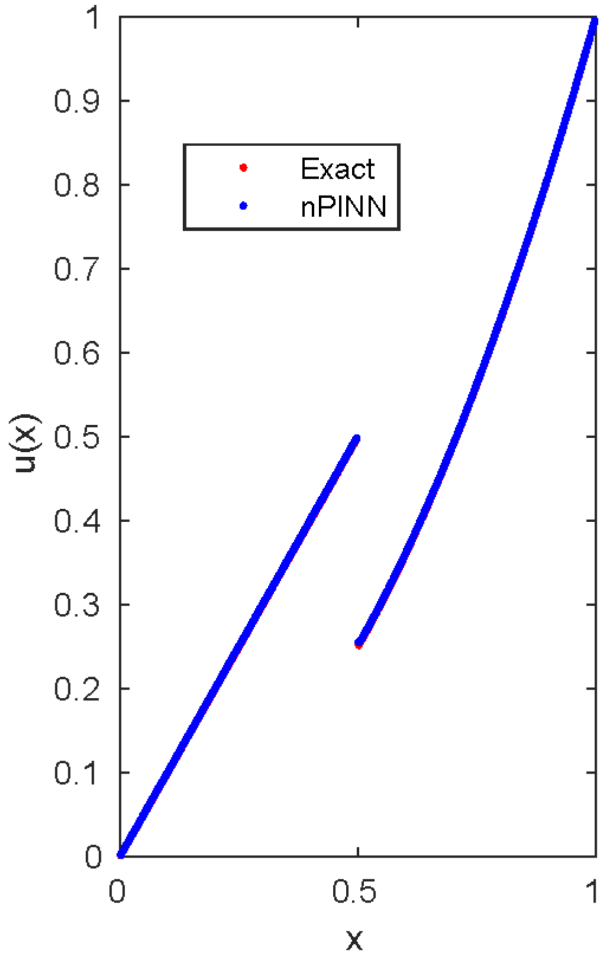
(Fractional Laplacian)

Computational Results: Data Driven Solutions

- nPINNs can reproduce discontinuous solutions (α=0, δ = 0.3)*

$$u(x)=\begin{cases} x & x\in[-\delta,0.5) \\ x^2 & x\in(0.5,1+\delta] \end{cases}$$

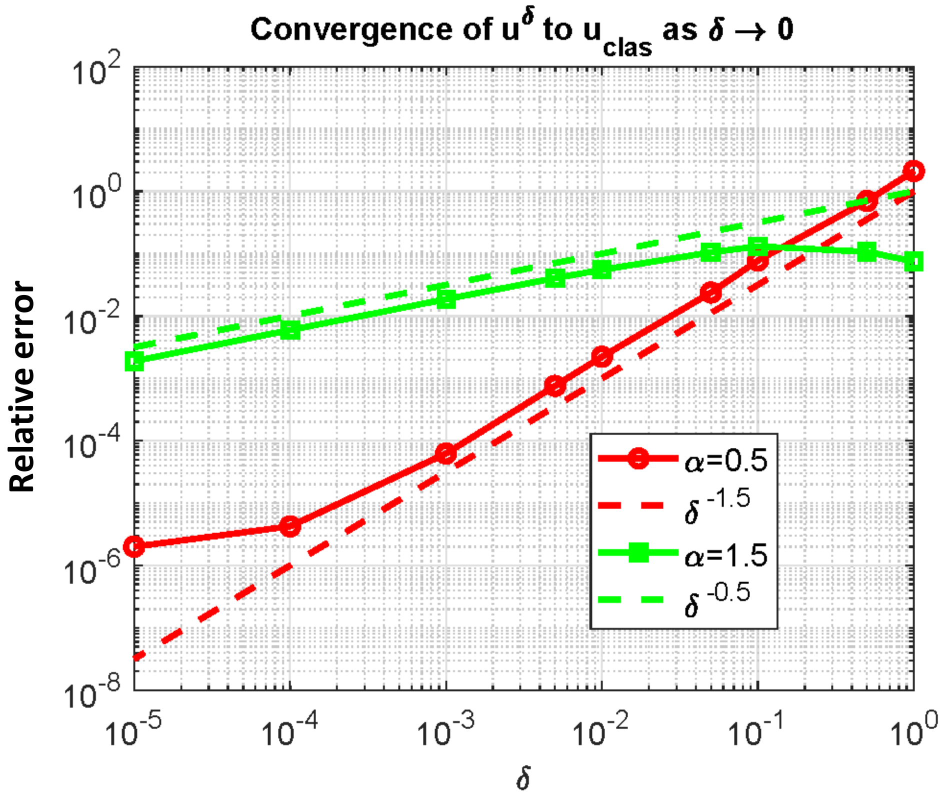
$$f(x)=\begin{cases} 0 & x\in[0,0.5-\delta) \\ -\frac{2}{\delta^2}\left[\frac{1}{2}\delta^2-\delta+\frac{3}{8}+(2\delta-\frac{3}{2}-\ln\delta)x + (\frac{3}{2}+x^2\ln\delta-(x^2-x)\ln\frac{1}{2}-x\right] & x\in[0.5-\delta,0.5) \\ -\frac{2}{\delta^2}\left[\frac{1}{2}\delta^2-\delta-\frac{3}{8}+(2\delta+\frac{3}{2}+x\ln\delta) - (\frac{3}{2}+x^2\ln\delta+(x^2-x)\ln x-\frac{1}{2}\right] & x\in(0.5,0.5+\delta) \\ -2 & x\in[0.5+\delta,1.0], \end{cases}$$

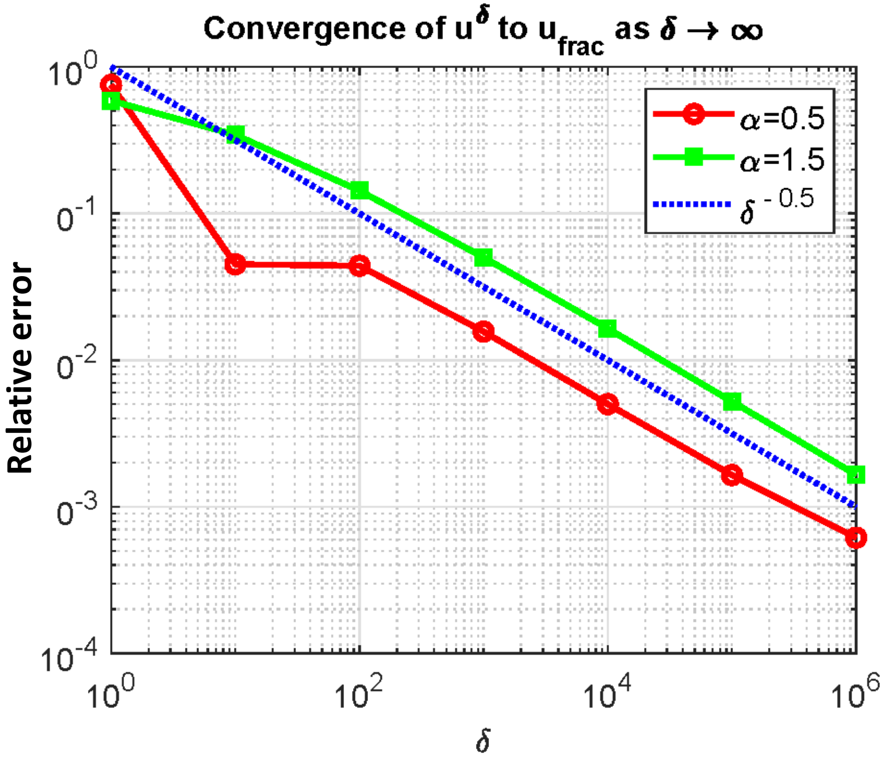


*X. Chen, M. Gunzburger, Continuous and discontinuous finite element methods for a peridynamics model of mechanics, Comput. Methods Appl. Mech. Eng. 200(9-12) (2011) 1237-1250.

Computational Results: Data Driven Solutions

- Convergence of nPINN solution to solution of classical Laplacian (δ→0) and fractional Laplacian (δ→∞)





$$\left|\left(-\mathcal{L}_{\delta,\alpha}\right)\mathbf{u}(\mathbf{x}) - (-\Delta)\mathbf{u}(\mathbf{x})\right| \sim \delta^{2-\alpha} \quad \text{as } \delta \to 0$$

$$\left|\left(-\mathcal{L}_{\delta,\alpha}\right)\mathbf{u}(\mathbf{x}) - (-\Delta)^{\alpha/2}\mathbf{u}(\mathbf{x})\right| \sim \delta^{\max\{\alpha-2,-\alpha\}} \quad \text{as } \delta \to \infty$$

Computational Results: Data Driven Discovery

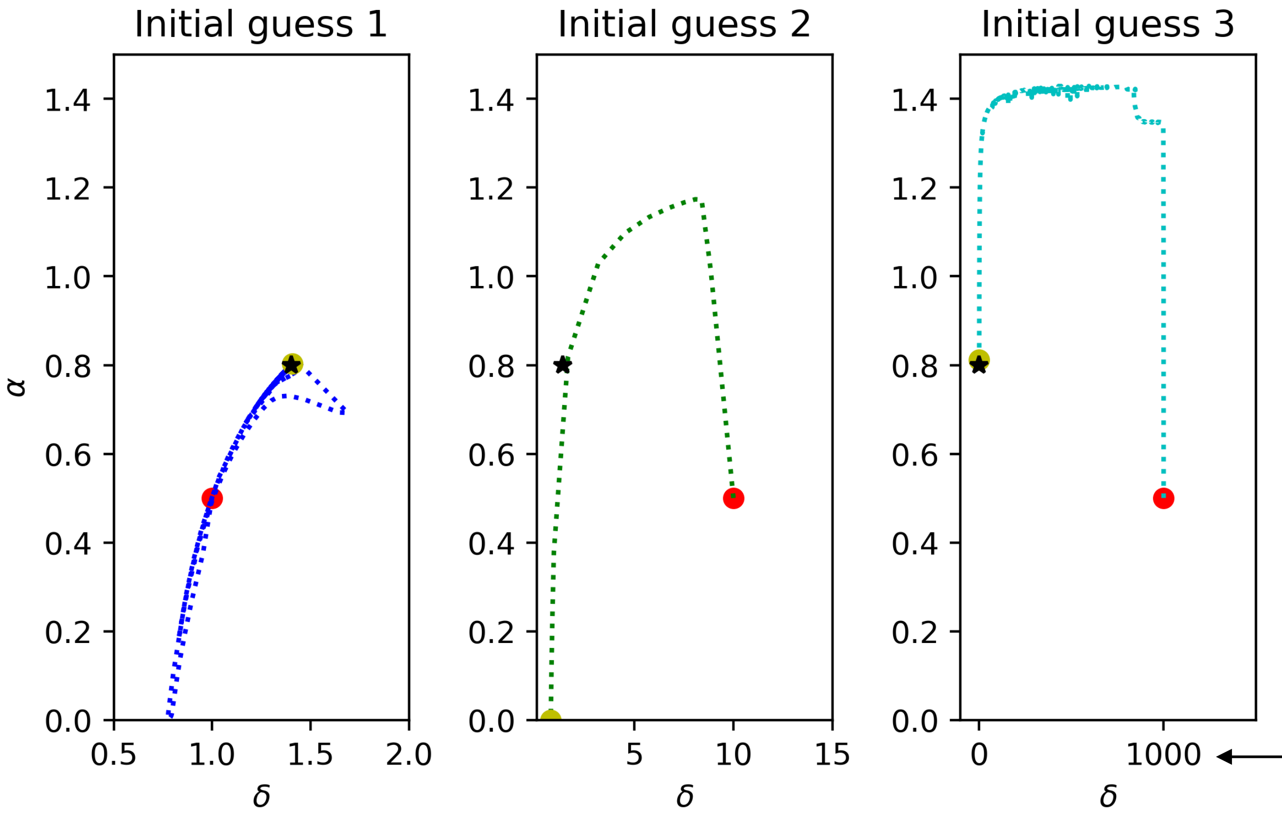
- nPINNs can discover parameterized operator from data: Seek $(\delta,\alpha) \in (0,\infty) \times (0,2)$.
- $\Omega = (0,1)$, $g(x) = 0$, $f(x) = \sin(2\pi x)$
- Training data: 100 uniformly spaced points in $\Omega \cup \Omega_\delta$
- Optimal $(\delta^*,\alpha^*) = (1.4, 0.8)$

$$-\mathcal{L}_{\delta,\alpha} u(x) = f(x) \quad x \in \Omega$$
$$u(x) = g(x) \quad x \in \Omega_\delta$$

● Initial guess

● Converged value

★ True solution



Scale change

Initial guesses: $(\delta_1,\alpha_1) = (1,0.5)$ $(\delta_2,\alpha_2) = (10,0.5)$ $(\delta_3,\alpha_3) = (1000,0.5)$

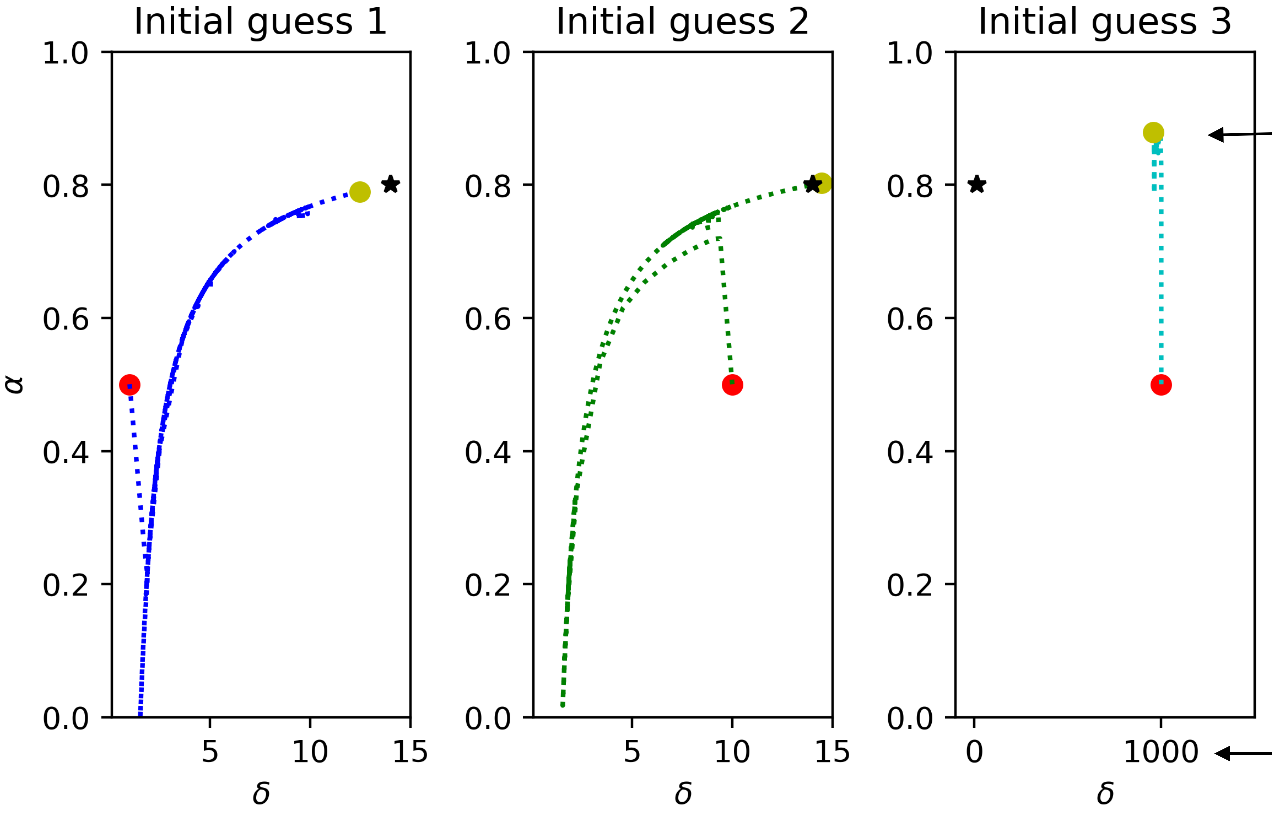
Relative errors: $e_1 = 2.64\times10^{-4}$ $e_2 = 1.43\times10^{-2}$ $e_3 = 2.69\times10^{-4}$

Computational Results: Data Driven Discovery

- nPINNs can discover parameterized operator from data: Seek $(\delta,\alpha) \in (0,\infty)\,(0,2)$.
- $\Omega = (0,1)$, $g(x) = 0$, $f(x) = \sin(2\pi x)$
- Training data: 100 uniformly spaced points in $\Omega \cup \Omega_\delta$
- Optimal $(\delta^*,\alpha^*) = (14.0, 0.8)$ ← Increase δ^* by 10× from previous example

$$-\mathcal{L}_{\delta,\alpha} u(x) = f(x) \quad x \in \Omega$$
$$u(x) = g(x) \quad x \in \Omega_\delta$$

- ● Initial guess
- ● Converged value
- ★ True solution



$\mathcal{L}_{\delta=958,\alpha=0.879}$ ***mimics*** $\mathcal{L}_{\delta=14,\alpha=0.8}$ for f(x), g(x), etc.

Scale change

Initial guesses:	$(\delta_1,\alpha_1) = (1,0.5)$	$(\delta_2,\alpha_2) = (10,0.5)$	$(\delta_3,\alpha_3) = (1000,0.5)$
Relative errors:	$e_1 = 5.46\times10^{-4}$	$e_2 = 1.06\times10^{-4}$	$e_3 = 8.17\times10^{-4}$

Wrong answer, but comparable error!

Turbulence Modeling of Couette Flow

- 1D equation for Couette flow

$$\frac{d}{dy^+}\left(\frac{dU^+}{dy^+} - \left(\overline{uv}\right)^+\right) = 0, \quad y^+ \in [0, 2Re_\tau]$$

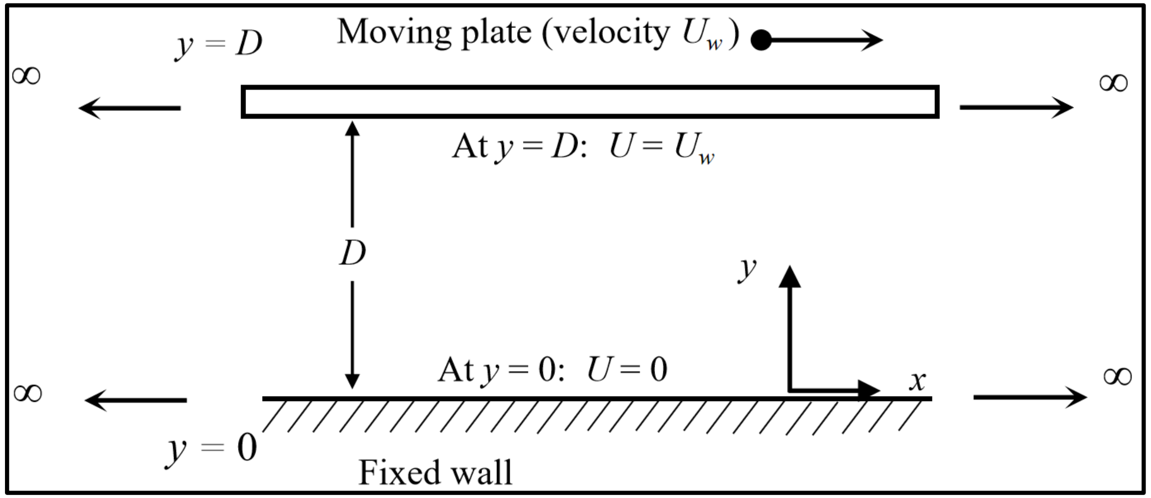
dimensionless Reynolds stress

dimensionless total shear stress

U+, y+ are dimensionless variables based on wall units

- Total shear stress equation

$$\frac{dU^+}{dy^+} - \left(\overline{uv}\right)^+ = 1, \quad y^+ \in [0, 2Re_\tau]$$



- **Propose new nonlocal model for Couette flow:**

$$\tilde{\mathcal{L}}_{\delta,\alpha} U^+ = 1, \quad \delta > 0, \ \alpha(y^+) \in (0,1)$$

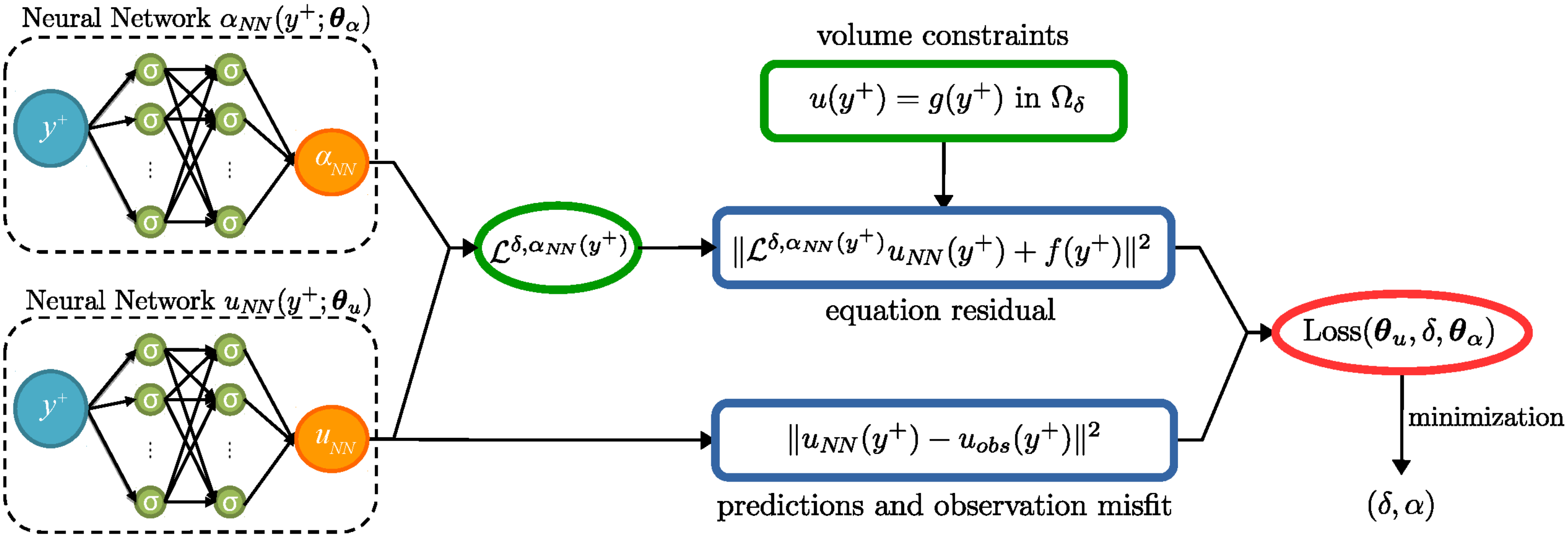
- New operator is not the operator $\mathcal{L}_{\delta,\alpha}$ we explored previously!

$$\lim_{\delta\to\infty}\left(\tilde{\mathcal{L}}_{\delta,\alpha}\right)U^+ \to$$ Combinaton of Caputo fractional derivatives

$$\lim_{\alpha(y^+)\to 1, \delta\to\infty}\left(-\mathcal{L}_{\delta,\alpha}\right)U^+ = \frac{dU^+}{dy^+}$$ ← Reduces to local model only in viscous sublayer where Reynolds stress << 1.

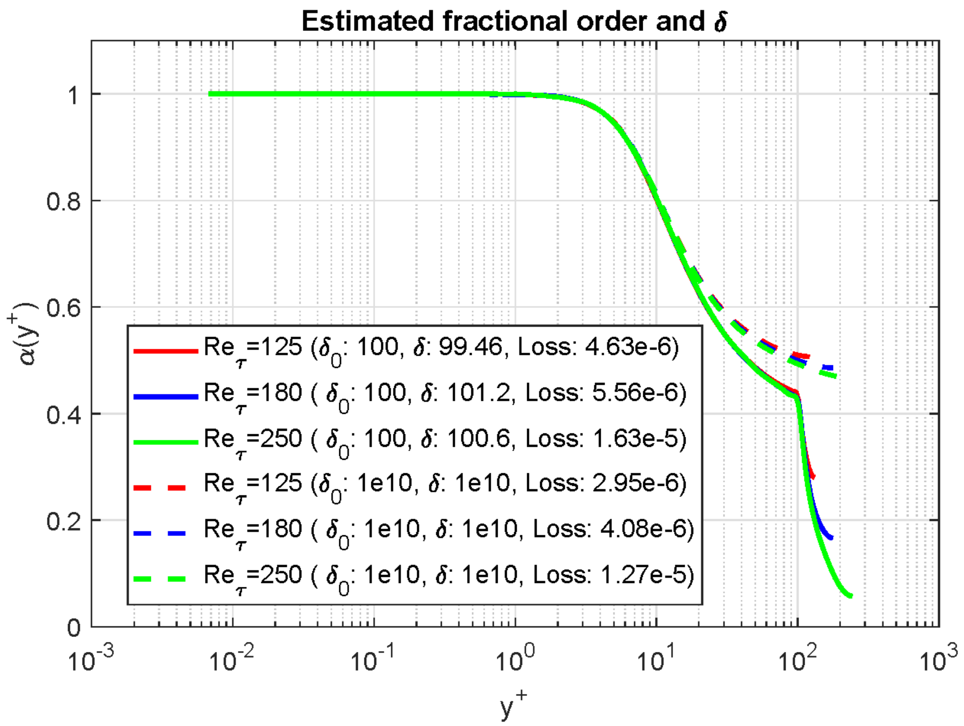
Turbulence Modeling of Couette Flow

- **Use nPINN to jointly estimate δ, α(y+). Use separate neural networks for U, α.**
- **Train using DNS data* for three different Reynolds numbers, Re_τ = 125, 180, 250.**
- **Use δ_0 = 100, 1e10.**

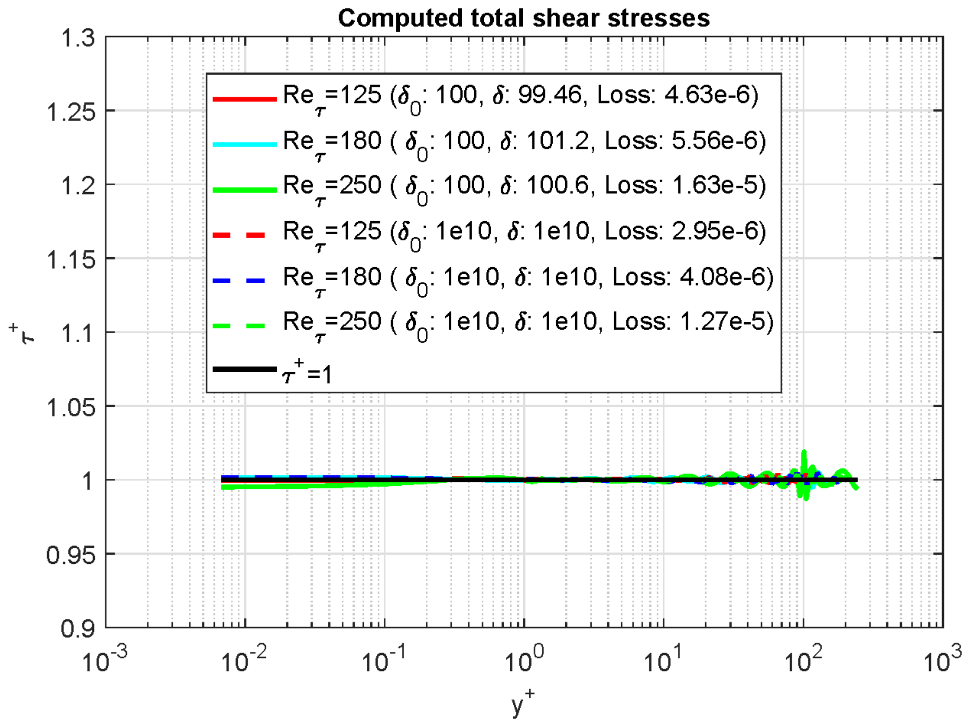


*V. Avsarkisov, S. Hoyas, M. Oberlack, Jose Pedro Garcia-Galache, Turbulent plane Couette flow at moderately high Reynolds number, J. Fluid Mech. 751 (2014).

Turbulence Modeling of Couette Flow



Profile of fractional order in wall units



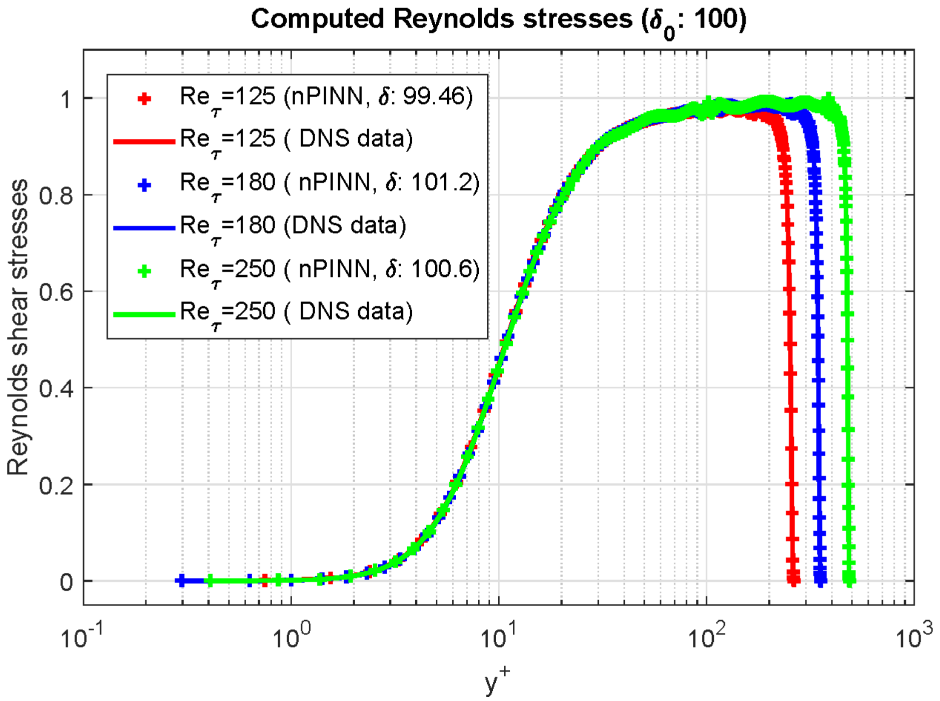
Computed total shear stress (true stress $\tau^+=1$)

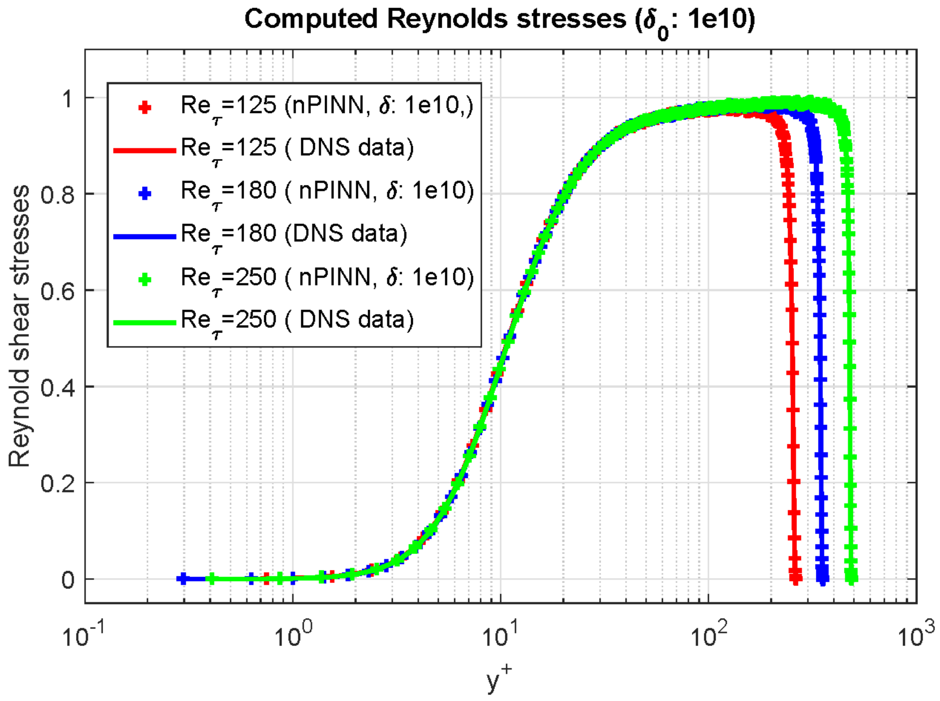
Observations:

- ❑ **Fractional order ≈ 1 near walls. Agrees with limit behavior for small Reynolds stress.**
- ❑ **Loss function not sensitive to changes in δ.**
- ❑ **Estimated fractional order profiles $\alpha(y^+)$ on top of each other independent of δ, Re_τ. Suggests existence of universal fractional order $\alpha(y^+)$ that reproduces DNS data independent of these Reynolds numbers.***
- ❑ **Fractional orders different for $y^+ > 20$, but with similar losses. Operators are distinct, but action on velocity is essentially the same (Mimic operator).**

*For more, see P.P. Mehta, G. Pang, F. Song, G.E. Karniadakis, Discovering a universal variable-order fractional model for turbulent Couette flow using a physics-informed neural network, Fract. Calc. Appl. Anal., 22 (6) (2019), pp. 1675-1688.

Turbulence Modeling of Couette Flow





Observations:

- ❑ **Computed Reynolds stresses on top of those reported from DNS dataset.**
- ❑ **Very different values of δ produce same stresses. These and other results (not shown) imply larger values of δ are more physically meaningful, and there is a threshold above which the nPINN reaches the same accuracy.**

Summary

- ❑ **Nonlocal Models**
- ❑ **Using computational models in practice**
- ❑ **nPINNS: nonlocal Physics-Informed Neural Networks**
  - ❑ **Data-driven solutions**
  - ❑ **Data-driven discovery**
- ❑ **A nonlocal model for turbulent Couette flow**
- ❑ **Conclusions**

Journal of Computational Physics 422 (2020) 109760

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

ELSEVIER

nPINNs: Nonlocal physics-informed neural networks for a parametrized nonlocal universal Laplacian operator. Algorithms and applications

G. Pang [a], M. D'Elia [b,*], M. Parks [c], G.E. Karniadakis [a]

[a] Division of Applied Mathematics, Brown University, RI, United States of America
[b] Computational Science and Analysis, Sandia National Laboratories, CA, United States of America
[c] Center for Computing Research, Sandia National Laboratories, NM, United States of America

Guofei Pang, Marta D'Elia, Michael L. Parks, George E. Karniadakis, **nPINNs: nonlocal Physics-Informed Neural Networks for a parametrized nonlocal universal Laplacian operator. Algorithms and Applications**, Journal of Computational Physics, 422, 109760, 2020.