Sandia National Laboratories

Exceptional service in the national interest

SCIENTIFIC MACHINE LEARNING AND TENSORFLOW TUTORIAL

Operator Learning

Ravi G Patel

Scientific Machine Learning Department

February 1 – 2, 2024

Numerical PDEs: Analysis, Algorithms, and Data Challenges

ICERM

Brown University

U.S. DEPARTMENT OF ENERGY NNSA National Nuclear Security Administration

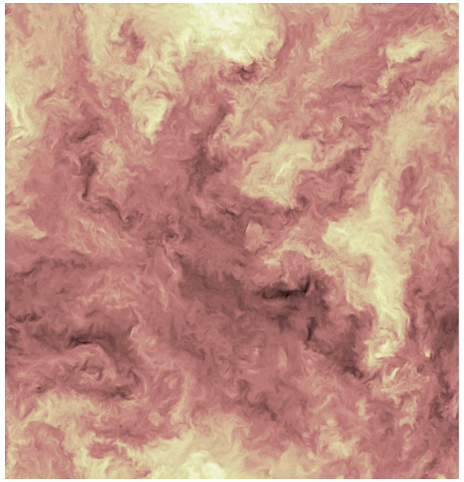
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

SAND2024-00936O

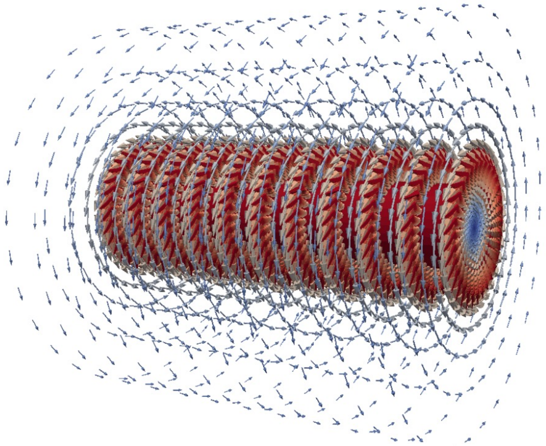
PHYSICAL SYSTEMS ARE DESCRIBED BY PDE'S

- We describe many physical systems with fields, i.e., functions over a spatio-temporal domain
- When we model field data, we're trying to find a unknown PDE
  - An equation with field valued solutions that match our data
  - Historical examples,

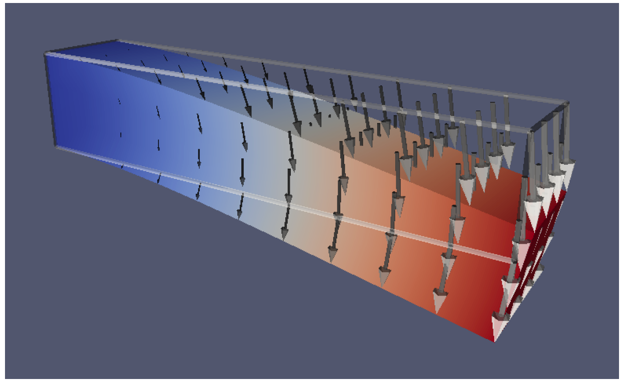
Physics	Incompressible fluid	Electromagnetism	Elasticity
Fields	Fluid velocity	Fluid velocity	Stress/strain
PDE	Navier-Stokes	Electric/magnetic	Hook's Law



Forced isotropic turbulence
JHTDB, turbulence.pha.jhu.edu



Magnetic field from oscillating current
Elmer FEM, elmerfem.org



Cantilever bean under gravity
FENICS, fenicsproject.org

PDE MODELING IS HARD

- Multiphase, multiphysics problems
  - Can’t easily model with intuition
  - Data driven methods provide an alternative

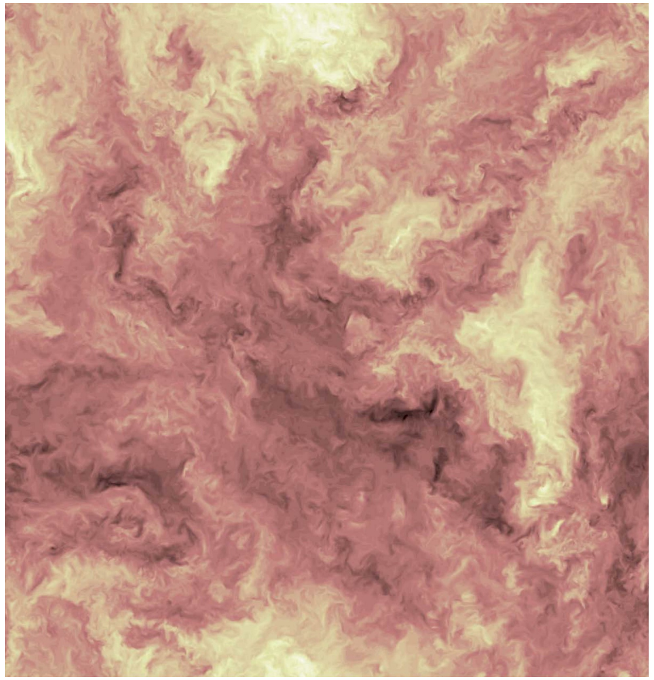
$$\frac{D\left\langle \boldsymbol{u}'\otimes\boldsymbol{u}'\right\rangle}{Dt}+\nabla\cdot\left(-\nu\nabla\left\langle \boldsymbol{u}'\otimes\boldsymbol{u}'\right\rangle+\boxed{\frac{2}{3\rho}I\otimes\left\langle \boldsymbol{u}'p'\right\rangle+\left\langle \boldsymbol{u}'\otimes\boldsymbol{u}'\otimes\boldsymbol{u}'\right\rangle}\right)$$

$$=-\left\langle \boldsymbol{u}'\otimes\boldsymbol{u}'\right\rangle\cdot\nabla\boldsymbol{u}-\left\langle \boldsymbol{u}'\otimes\boldsymbol{u}'\right\rangle\cdot\nabla\boldsymbol{u}$$

$$\boxed{\begin{aligned}&+\left\langle\frac{p'}{\rho}\left(\nabla\boldsymbol{u}+\nabla\boldsymbol{u}^T\right)\right\rangle\\&-2\nu\left\langle\left(\nabla\left\langle\boldsymbol{u}\right\rangle\right)^T\cdot\left(\nabla\left\langle\boldsymbol{u}\right\rangle\right)\right\rangle\end{aligned}}$$

Needs modeling

Reynolds stress modeling for single phase turbulence



Forced isotropic turbulence
JHTDB, turbulence.pha.jhu.edu

PDE MODELING IS HARD

- Multiphase, multiphysics problems
  - Can’t easily model with intuition
  - Data driven methods provide an alternative

$$\frac{\partial\langle\alpha_p\rangle}{\partial t}+\nabla\cdot\langle\alpha_p\rangle\langle\boldsymbol{u}_p\rangle_p=0,$$

$$\frac{\partial\langle\alpha_p'^2\rangle}{\partial t}+\nabla\cdot\langle\alpha_p'^2\rangle\langle\boldsymbol{u}_p\rangle_p+\nabla\cdot\langle\alpha_p'^2\boldsymbol{u}_p''\rangle$$
$$=-2\langle\alpha_p'\boldsymbol{u}_p''\rangle\cdot\nabla\langle\alpha_p\rangle-\langle\alpha_p'^2\rangle\nabla\cdot\langle\boldsymbol{u}_p\rangle_p-2\langle\alpha_p\rangle\langle\alpha_p'\nabla\cdot\boldsymbol{u}_p''\rangle-\langle\alpha_p'^2\nabla\cdot\boldsymbol{u}_p''\rangle,$$

$$\frac{\partial\langle\alpha_p\rangle\langle\boldsymbol{u}_p\rangle_p}{\partial t}+\nabla\cdot\langle\alpha_p\rangle(\langle\boldsymbol{u}_p\rangle_p\otimes\langle\boldsymbol{u}_p\rangle_p+\langle\mathscr{P}\rangle_p)=\langle\alpha_p\rangle(\langle\mathscr{A}\rangle_p+\boldsymbol{g}),$$

$$\frac{\partial\langle\alpha_p\rangle\langle\boldsymbol{P}\rangle_p}{\partial t}+\nabla\cdot\langle\alpha_p\rangle(\langle\boldsymbol{u}_p\rangle_p\otimes\langle\boldsymbol{P}\rangle_p+\langle\boldsymbol{u}_p''\otimes\boldsymbol{P}\rangle_p+\langle\boldsymbol{Q}\rangle_p)$$
$$=-\langle\alpha_p\rangle(\langle\boldsymbol{P}\rangle_p\cdot\nabla\langle\boldsymbol{u}_p\rangle_p+\langle\boldsymbol{P}\cdot\nabla\boldsymbol{u}_p''\rangle_p)^\dagger-\frac{2}{\tau_p}\langle\alpha_p\rangle\langle\boldsymbol{P}\rangle_p+\frac{12\langle\alpha_p\rangle}{\sqrt{\pi}d_p}\langle\alpha_p\Theta^{1/2}(\Delta^*-\boldsymbol{P})\rangle_p,$$

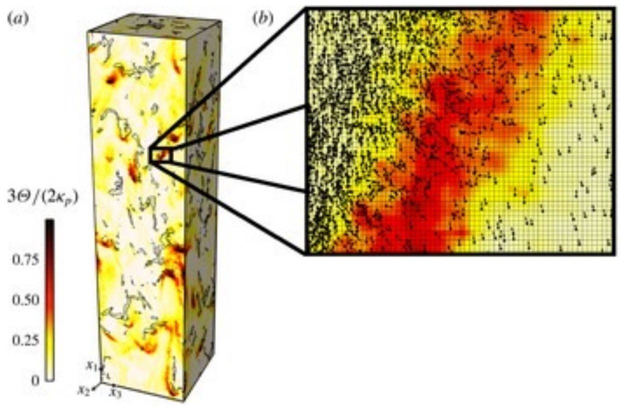
$$\frac{\partial\langle\alpha_p\rangle\langle\boldsymbol{u}_p''\otimes\boldsymbol{u}_p''\rangle_p}{\partial t}+\nabla\cdot\langle\alpha_p\rangle(\langle\boldsymbol{u}_p\rangle_p\otimes\langle\boldsymbol{u}_p''\otimes\boldsymbol{u}_p''\rangle_p+\langle\boldsymbol{u}_p''\otimes\boldsymbol{u}_p''\otimes\boldsymbol{u}_p''\rangle_p+\langle\boldsymbol{P}\otimes\boldsymbol{u}_p''\rangle_p^\dagger)$$
$$=-\langle\alpha_p\rangle(\langle\boldsymbol{u}_p''\otimes\boldsymbol{u}_p''\rangle_p\cdot\nabla\langle\boldsymbol{u}_p\rangle_p)^\dagger+\langle\alpha_p\rangle\langle\boldsymbol{P}\cdot\nabla\boldsymbol{u}_p''\rangle_p^\dagger+\frac{\langle\alpha_p\rangle}{\tau_p}(\langle\boldsymbol{u}_f'''\otimes\boldsymbol{u}_p''\rangle_p-\langle\boldsymbol{u}_p''\otimes\boldsymbol{u}_p''\rangle_p)^\dagger$$
$$+\frac{\langle\alpha_p\rangle}{\rho_p}(\langle\boldsymbol{u}_p''\otimes\nabla\cdot\boldsymbol{\sigma}_f'\rangle_p-\langle\boldsymbol{u}_p''\otimes\nabla p_f'\rangle_p)^\dagger,$$

$$\frac{\partial\langle\alpha_f\rangle}{\partial t}+\nabla\cdot\langle\alpha_f\rangle\langle\boldsymbol{u}_f\rangle_f=0.$$

$$\frac{\partial\langle\alpha_f\rangle\langle\boldsymbol{u}_f\rangle_f}{\partial t}+\nabla\cdot\langle\alpha_f\rangle(\langle\boldsymbol{u}_f\rangle_f\otimes\langle\boldsymbol{u}_f\rangle_f+\langle\boldsymbol{u}_f'''\otimes\boldsymbol{u}_f'''\rangle_f)$$
$$=\frac{1}{\rho_f}(\nabla\cdot\langle\boldsymbol{\sigma}_f\rangle-\nabla\langle p_f\rangle)-\langle\alpha_f\rangle\varphi\langle\mathscr{A}\rangle_p+\langle\alpha_f\rangle\boldsymbol{g},$$

$$\frac{\partial\langle\alpha_f\rangle\langle\boldsymbol{u}_f'''\otimes\boldsymbol{u}_f'''\rangle_f}{\partial t}+\nabla\cdot\langle\alpha_f\rangle(\langle\boldsymbol{u}_f\rangle_f\otimes\langle\boldsymbol{u}_f'''\otimes\boldsymbol{u}_f'''\rangle_f+\langle\boldsymbol{u}_f'''\otimes\boldsymbol{u}_f'''\otimes\boldsymbol{u}_f'''\rangle_f)$$
$$=-\langle\alpha_f\rangle(\langle\boldsymbol{u}_f'''\otimes\boldsymbol{u}_f'''\rangle_f\cdot\nabla\langle\boldsymbol{u}_f\rangle_f)^\dagger+\frac{1}{\rho_f}(\nabla\cdot\langle\boldsymbol{\sigma}_f\otimes\boldsymbol{u}_f'''\rangle-\nabla\langle p_f\boldsymbol{u}_f'''\rangle)^\dagger$$
$$-\frac{1}{\rho_f}(\langle\boldsymbol{\sigma}_f\cdot\nabla\boldsymbol{u}_f'''\rangle-\langle p_f\nabla\boldsymbol{u}_f'''\rangle)^\dagger$$
$$+\frac{\langle\alpha_f\rangle\varphi}{\tau_p}[\langle\boldsymbol{u}_f'''\otimes\boldsymbol{u}_p''\rangle_p-\langle\boldsymbol{u}_f'''\otimes\boldsymbol{u}_f'''\rangle_p+\langle\boldsymbol{u}_f'''\rangle_p\otimes(\langle\boldsymbol{u}_p\rangle_p-\langle\boldsymbol{u}_f\rangle_f)]^\dagger$$
$$+\frac{\langle\alpha_f\rangle\varphi}{\rho_p}[\langle\boldsymbol{u}_f'''\rangle_p\otimes\nabla\langle p_f\rangle+\langle\boldsymbol{u}_f'''\otimes\nabla p_f'\rangle_p-\langle\boldsymbol{u}_f'''\rangle_p\otimes\nabla\cdot\langle\boldsymbol{\sigma}_f\rangle-\langle\boldsymbol{u}_f'''\otimes\nabla\cdot\boldsymbol{\sigma}_f'\rangle_p]^\dagger.$$

Reynolds averaged transport equations[1] for particle-laden turbulent flow



Granular temperature field in cluster-induced turbulence[1]

[1]Capecelatro et al., *J. Fluid Mech.* (2015)

PDE MODELING IS HARD

- Multiphase, multiphysics problems
  - Can’t easily model with intuition
  - Data driven methods provide an alternative

- Adding a single extra phase greatly complicates the model
- Need a scalable approach to PDE modeling

$$\frac{\partial \langle \alpha_p \rangle}{\partial t} + \nabla \cdot \langle \alpha_p \rangle \langle \boldsymbol{u}_p \rangle_p = 0,$$

$$\frac{\partial \langle \alpha_p'^2 \rangle}{\partial t} + \nabla \cdot \langle \alpha_p'^2 \rangle \langle \boldsymbol{u}_p \rangle_p + \nabla \cdot \langle \alpha_p'^2 \boldsymbol{u}_p'' \rangle = -2\langle \alpha_p' \boldsymbol{u}_p'' \rangle \cdot \nabla \langle \alpha_p \rangle - \langle \alpha_p'^2 \rangle \nabla \cdot \langle \boldsymbol{u}_p \rangle_p - 2\langle \alpha_p \rangle \langle \alpha_p' \nabla \cdot \boldsymbol{u}_p'' \rangle - \langle \alpha_p'^2 \nabla \cdot \boldsymbol{u}_p'' \rangle,$$

$$\frac{\partial \langle \alpha_p \rangle \langle \boldsymbol{u}_p \rangle_p}{\partial t} + \nabla \cdot \langle \alpha_p \rangle (\langle \boldsymbol{u}_p \rangle_p \otimes \langle \boldsymbol{u}_p \rangle_p + \langle \mathscr{P} \rangle_p) = \langle \alpha_p \rangle (\langle \mathscr{A} \rangle_p + \boldsymbol{g}),$$

$$\frac{\partial \langle \alpha_p \rangle \langle \boldsymbol{P} \rangle_p}{\partial t} + \nabla \cdot \langle \alpha_p \rangle (\langle \boldsymbol{u}_p \rangle_p \otimes \langle \boldsymbol{P} \rangle_p + \langle \boldsymbol{u}_p'' \otimes \boldsymbol{P} \rangle_p + \langle \boldsymbol{Q} \rangle_p) = -\langle \alpha_p \rangle (\langle \boldsymbol{P} \rangle_p \cdot \nabla \langle \boldsymbol{u}_p \rangle_p + \langle \boldsymbol{P} \cdot \nabla \boldsymbol{u}_p'' \rangle_p)^{\dagger} - \frac{2}{\tau_p} \langle \alpha_p \rangle \langle \boldsymbol{P} \rangle_p + \frac{12 \langle \alpha_p \rangle}{\sqrt{\pi} d_p} \langle \alpha_p \Theta^{1/2} (\Delta^* - \boldsymbol{P}) \rangle_p,$$

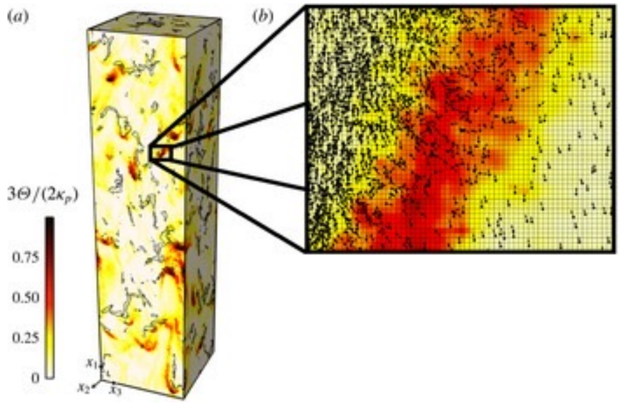
$$\frac{\partial \langle \alpha_p \rangle \langle \boldsymbol{u}_p'' \otimes \boldsymbol{u}_p'' \rangle_p}{\partial t} + \nabla \cdot \langle \alpha_p \rangle (\langle \boldsymbol{u}_p \rangle_p \otimes \langle \boldsymbol{u}_p'' \otimes \boldsymbol{u}_p'' \rangle_p + \langle \boldsymbol{u}_p'' \otimes \boldsymbol{u}_p'' \otimes \boldsymbol{u}_p'' \rangle_p + \langle \boldsymbol{P} \otimes \boldsymbol{u}_p'' \rangle_p^{\dagger}) = -\langle \alpha_p \rangle (\langle \boldsymbol{u}_p'' \otimes \boldsymbol{u}_p'' \rangle_p \cdot \nabla \langle \boldsymbol{u}_p \rangle_p)^{\dagger} + \langle \alpha_p \rangle \langle \boldsymbol{P} \cdot \nabla \boldsymbol{u}_p'' \rangle_p^{\dagger} + \frac{\langle \alpha_p \rangle}{\tau_p} (\langle \boldsymbol{u}_f''' \otimes \boldsymbol{u}_p'' \rangle_p - \langle \boldsymbol{u}_p'' \otimes \boldsymbol{u}_p'' \rangle_p)^{\dagger} + \frac{\langle \alpha_p \rangle}{\rho_p} (\langle \boldsymbol{u}_p'' \otimes \nabla \cdot \boldsymbol{\sigma}_f' \rangle_p - \langle \boldsymbol{u}_p'' \otimes \nabla p_f' \rangle_p)^{\dagger},$$

$$\frac{\partial \langle \alpha_f \rangle}{\partial t} + \nabla \cdot \langle \alpha_f \rangle \langle \boldsymbol{u}_f \rangle_f = 0.$$

$$\frac{\partial \langle \alpha_f \rangle \langle \boldsymbol{u}_f \rangle_f}{\partial t} + \nabla \cdot \langle \alpha_f \rangle (\langle \boldsymbol{u}_f \rangle_f \otimes \langle \boldsymbol{u}_f \rangle_f + \langle \boldsymbol{u}_f''' \otimes \boldsymbol{u}_f''' \rangle_f) = \frac{1}{\rho_f} (\nabla \cdot \langle \boldsymbol{\sigma}_f \rangle - \nabla \langle p_f \rangle) - \langle \alpha_f \rangle \varphi \langle \mathscr{A} \rangle_p + \langle \alpha_f \rangle \boldsymbol{g},$$

$$\frac{\partial \langle \alpha_f \rangle \langle \boldsymbol{u}_f''' \otimes \boldsymbol{u}_f''' \rangle_f}{\partial t} + \nabla \cdot \langle \alpha_f \rangle (\langle \boldsymbol{u}_f \rangle_f \otimes \langle \boldsymbol{u}_f''' \otimes \boldsymbol{u}_f''' \rangle_f + \langle \boldsymbol{u}_f''' \otimes \boldsymbol{u}_f''' \otimes \boldsymbol{u}_f''' \rangle_f) = -\langle \alpha_f \rangle (\langle \boldsymbol{u}_f''' \otimes \boldsymbol{u}_f''' \rangle_f \cdot \nabla \langle \boldsymbol{u}_f \rangle_f)^{\dagger} + \frac{1}{\rho_f} (\nabla \cdot \langle \boldsymbol{\sigma}_f \otimes \boldsymbol{u}_f''' \rangle - \nabla \langle p_f \boldsymbol{u}_f''' \rangle)^{\dagger} - \frac{1}{\rho_f} (\langle \boldsymbol{\sigma}_f \cdot \nabla \boldsymbol{u}_f''' \rangle - \langle p_f \nabla \boldsymbol{u}_f''' \rangle)^{\dagger} + \frac{\langle \alpha_f \rangle \varphi}{\tau_p} [\langle \boldsymbol{u}_f''' \otimes \boldsymbol{u}_p'' \rangle_p - \langle \boldsymbol{u}_f''' \otimes \boldsymbol{u}_f''' \rangle_p + \langle \boldsymbol{u}_f''' \rangle_p \otimes (\langle \boldsymbol{u}_p \rangle_p - \langle \boldsymbol{u}_f \rangle_f)]^{\dagger} + \frac{\langle \alpha_f \rangle \varphi}{\rho_p} [\langle \boldsymbol{u}_f''' \rangle_p \otimes \nabla \langle p_f \rangle + \langle \boldsymbol{u}_f''' \otimes \nabla p_f' \rangle_p - \langle \boldsymbol{u}_f''' \rangle_p \otimes \nabla \cdot \langle \boldsymbol{\sigma}_f \rangle - \langle \boldsymbol{u}_f''' \otimes \nabla \cdot \boldsymbol{\sigma}_f' \rangle_p]^{\dagger}.$$

Reynolds averaged transport equations[1] for particle-laden turbulent flow



Granular temperature field in cluster-induced turbulence[1]

[1]Capecelatro et al., *J. Fluid Mech.* (2015)

OPERATOR LEARNING ENABLES PDE DISCOVERY

- PDE's are composed of operators, i.e., mappings between functions,

$$\sum_i \mathcal{N}_i\left(u_1, u_2, \ldots\right) = 0$$

$$u_j \in \mathcal{U}_j : \mathbb{R}^+ \times \Omega \to \mathbb{R}^{m_j}$$

$$\mathcal{N}_i : \mathcal{U}_1 \times \mathcal{U}_2 \times \ldots \to \mathcal{V}$$

- E.g., the viscous Burger's equation has three operators,

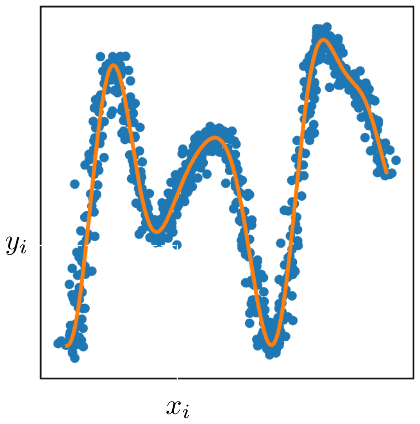
$$\partial_t u + \partial_x u^2 - \partial_x^2 u = 0$$

- If we're given the field data, we can find a PDE by finding it's component operators

OPERATOR LEARNING IS A GENERALIZATION OF REGRESSION

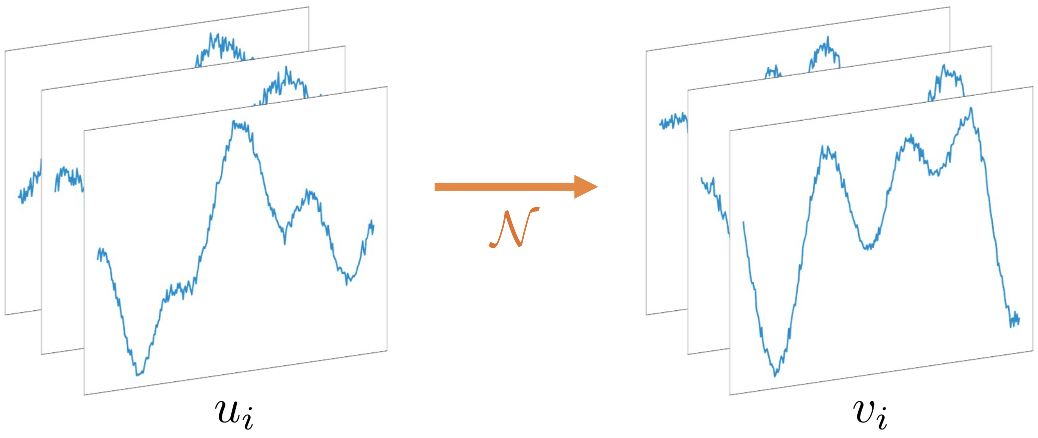
- Given pairs of functions, $\{u_i, v_i\}^{i=1,2,\dots}$, we can use regression to find the operator maps between the two, $\mathcal{N}(u_i) \approx v_i$

Fitting functions



$$\hat{f} = \underset{f}{\operatorname{argmin}} \sum_i ||y_i - f(x_i)||_2^2$$

Fitting operators



$$\hat{\mathcal{N}} = \underset{\mathcal{N}}{\operatorname{argmin}} \sum_i ||v_i - \mathcal{N}[u_i]||_{\mathcal{V}}$$

THREE COMPONENTS OF AN OPERATOR LEARNING METHOD

1. A discretization for functional data
   - To run on computers, we need to work with some discretization
   - Usually implicit in the dataset we receive
   - Often taken to be point evaluations of the functions
2. An interpolation scheme
   - Even though we only work with discretizations, we should be able to evaluate the functions anywhere they're defined
   - Can be learned or fixed
3. A parameterized mapping between discretizations
   - We'll use gradient based methods to optimize these parameters against the error norm

OPERATOR LEARNING PARAMETERIZATIONS

Modal operator regression for physics (MOR-Physics[1,2])

- Given pairs of functions, $(u_i, v_i)^{i=1,2,\dots}$, let's find a Poisson-like PDE,

  $$\mathcal{N}(u) = v$$

  - Let's assume they're given as point-wise evaluations on a regular, periodic grid
- We parameterize the unknown operator as,

  $$\mathcal{N}u = \mathcal{F}^{-1} g(\kappa; \xi_g) \mathcal{F} h(u; \xi_h)$$

  Where g and h are neural networks. $\mathcal{F}$ is the Fourier transform
- Optimization problem becomes,

  $$\xi_g, \xi_h = \underset{\hat{\xi}_g, \hat{\xi}_h}{\operatorname{argmin}} \sum_i \left|\left| v_i - \mathcal{N}(u_i; \hat{\xi}_g, \hat{\xi}_h) \right|\right|_2^2$$

[1]R.G. Patel and Desjardins, *arxiv: 1810.08552* (2018)
[2]R.G. Patel et al., *CMAME* (2021)

MOR-PHYSICS: MOTIVATION

For smooth functions in a periodic domain,

Physical space

$$f(x) = \sum_{\kappa} = \tilde{f}_{\kappa} e^{j\kappa x}$$

$$\partial_x^{\gamma} f(x)$$

$\xrightarrow{\mathcal{F}}$

$\xleftarrow{\mathcal{F}^{-1}}$

Fourier space

$$f_{\kappa} = \int f(x) e^{-j\kappa x} dx$$

$$(j\kappa)^{\gamma} \tilde{f}_{\kappa}$$

Parameterization contains,

- Laplacian

$$\partial_x^2 u \longrightarrow \mathcal{F}^{-1}[(-\kappa^2)\mathcal{F}[u]]$$

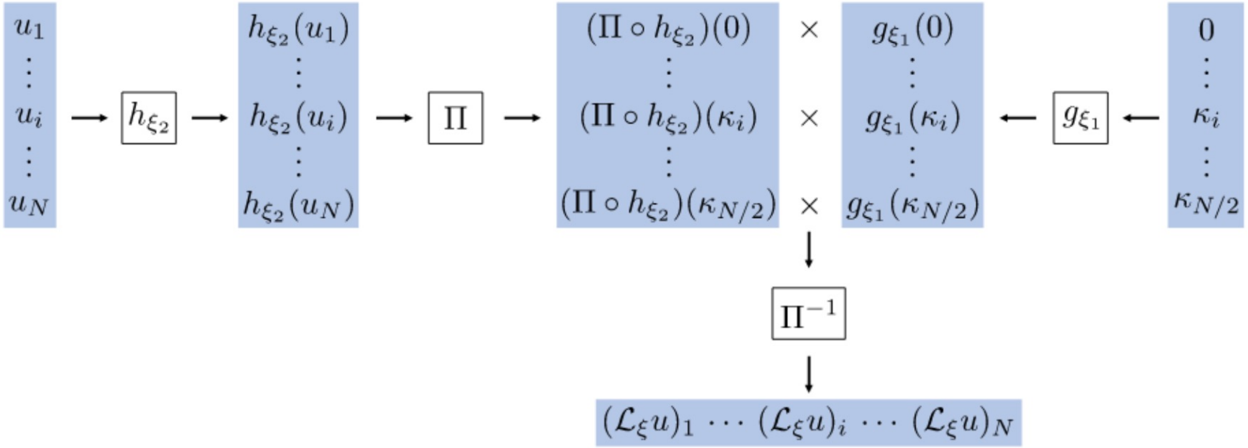
g h

- Burgers operator

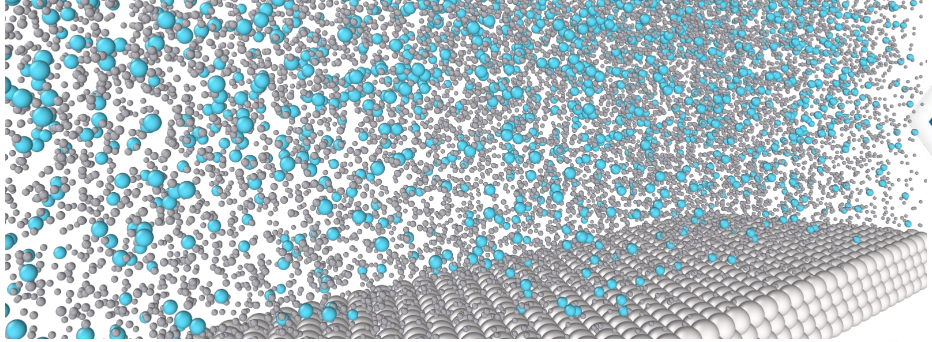
$$\partial_x u^2 \longrightarrow \mathcal{F}^{-1}[(j\kappa)\mathcal{F}[u^2]]$$

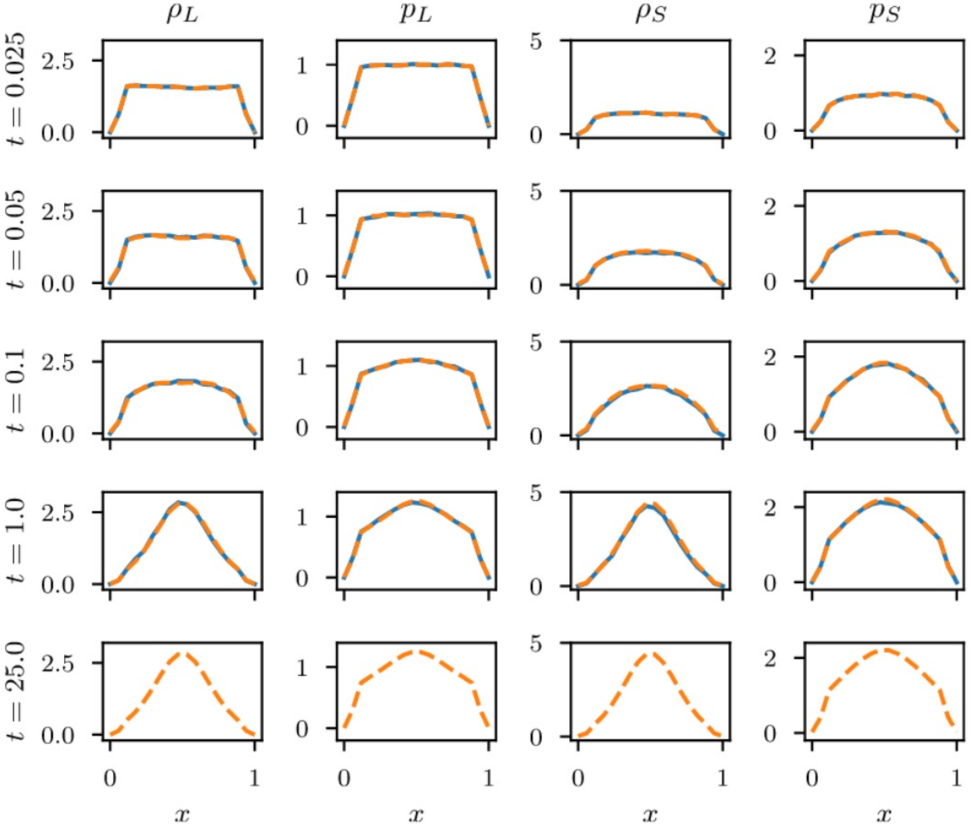
g h

MOR-PHYSICS: EXAMPLE



MOR-Physics parameterization[1]





MOR-physics learns dynamics of colloidal system from molecular dynamics simulations. Generalizes to unseen concentration and colloid diameter[2]

[1]R.G. Patel and Desjardins, *arxiv: 1810.08552* (2018)
[2]R.G. Patel et al., *CMAME* (2021)

OPERATOR LEARNING PARAMETERIZATIONS

Deep Operator Network (DeepONet)[1]

- Given pairs of functions, $(u_i, v_i)^{i=1,2,\dots}$, let's find a Poisson-like PDE,

  $$\mathcal{N}(u) = v$$

  - Let's assume they're given as point-wise evaluations on two (possibly different) sets of collocation points

- We parameterize the unknown operator as,

  $$\mathcal{N}(u, \xi_b, \xi_t)(y) = \sum_i b_i(u, \xi_b) t_i(y, \xi_t)$$

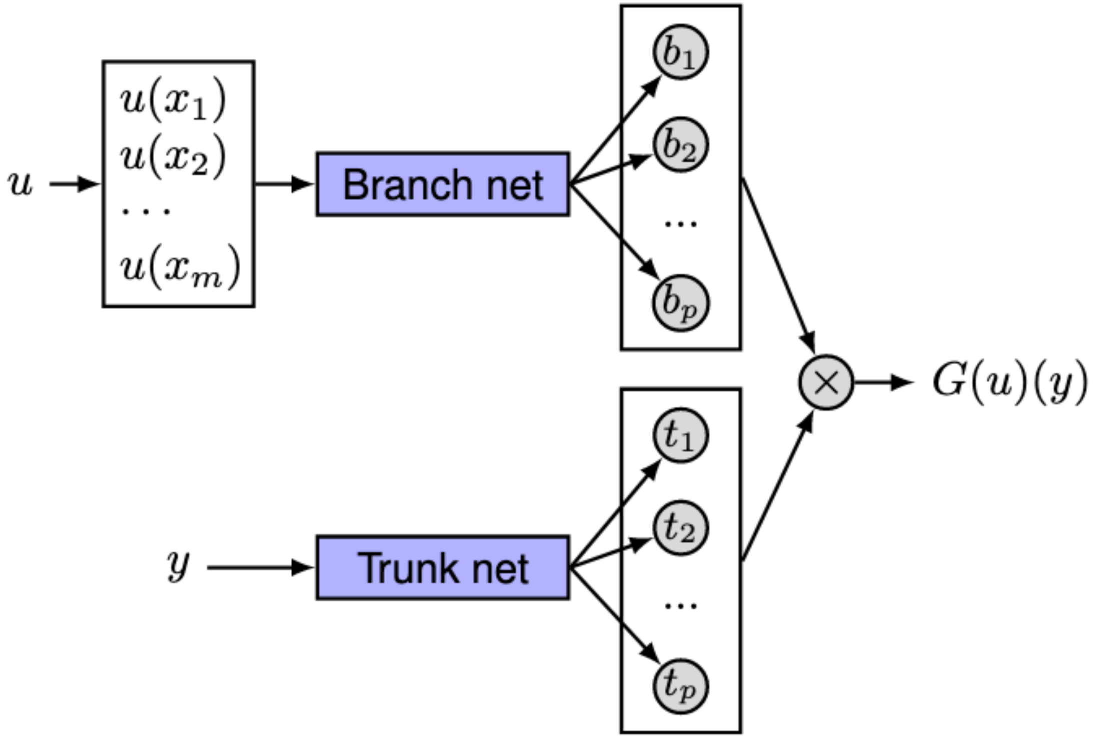
  Where b and t are neural networks

- Optimization problem becomes,

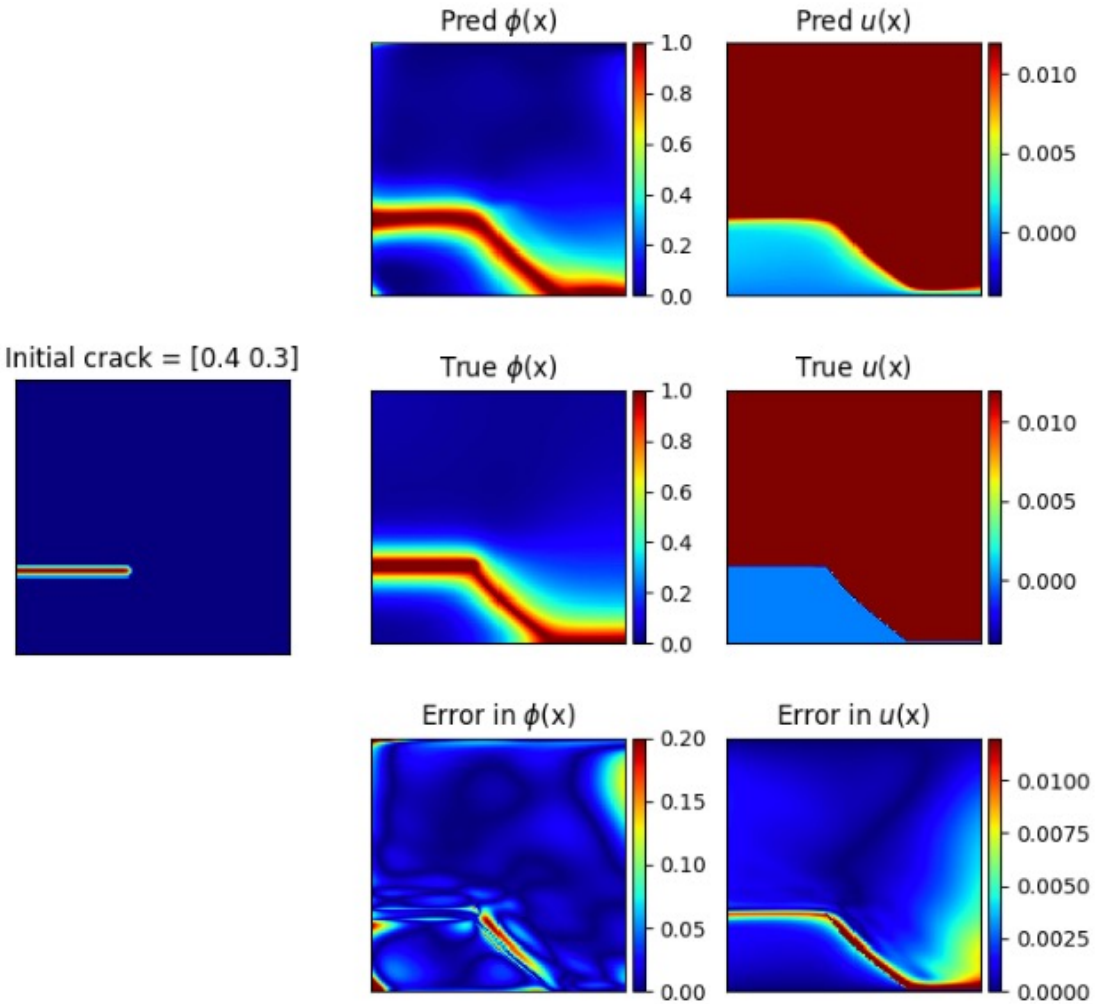
  $$\xi_b, \xi_t = \underset{\hat{\xi}_b, \hat{\xi}_t}{\operatorname{argmin}} \sum_i \left|\left| v_i - \mathcal{N}(u_i; \hat{\xi}_b, \hat{\xi}_t) \right|\right|_2^2$$

[1]Lu, Jin and Karniadakis, *Nature*, 2021

DEEPONET EXAMPLE



Unstacked DeepONet parameterization



Variational DeepONet learns crack path under shear loading. Generalizes to unseen crack tip locations.[2]

[1]Lu, Jin and Karniadakis, *Nature,* 2021
[2]Goswami et al., *CMAME,* 2022

INCLUDING A PRIORI KNOWLEDGE

- Operator learning is flexible
  - Can learn several operators simultaneously
  - Can combine a partially known PDE with unknown operators
- Often, we'll know more about a system than just that the data
  - Theoretical results
  - Symmetries
  - Conservation principles

TIME EVOLVING SYSTEMS

- Assume system is described by 1st order in time, autonomous PDE,

$$\partial_t u = \mathcal{N} u$$

- Discretize in time,

$$u^{n+1} = u^n + \Delta t \mathcal{N} u^n = (I + \Delta t \mathcal{N}) u^n$$

- Given observations, $\{v^n\}$, find,

$$\mathcal{N} = \underset{\hat{\mathcal{N}}}{\operatorname{argmin}} \sum_n \left|\left| v^{n+1} - (I + \Delta t \hat{\mathcal{N}}) v^n \right|\right|$$

- More generally,

$$\mathcal{N} = \underset{\hat{\mathcal{N}}}{\operatorname{argmin}} \sum_n \left|\left| v^{n+p} - (I + \Delta t \hat{\mathcal{N}})^p v^n \right|\right|$$

TIME EVOLVING SYSTEMS

- Alternatively, parameterize your operator with time

  $$u(\cdot, t) = \mathcal{N}(u^0; t)$$

  Lu, Jin and Karniadakis, *Nature,* 2021

  Li et al., *ICLR* (2021)

OTHER OPERATOR LEARNING FRAMEWORKS AND RELATED METHODS

- Fourier Neural Operator
  - Li et al., *ICLR* (2021)
- Nonlocal operator regression
  - H. You et al., *CMAME* (2020)
- Operator Inference
  - B. Peherstorfer and K. Willcox, *CMAME* (2016)
- Function-to-function regression
  - J.S. Morris, *Annu. Rev. Stat. Appl.* (2015)