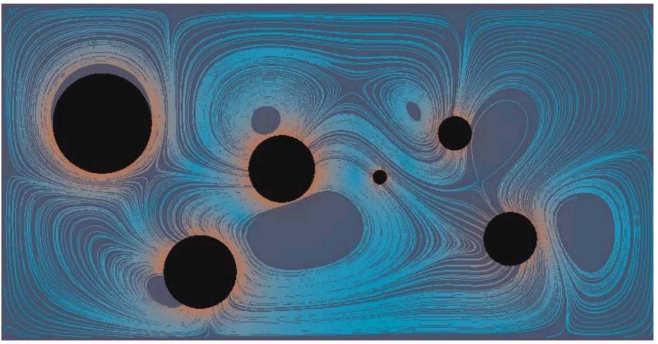
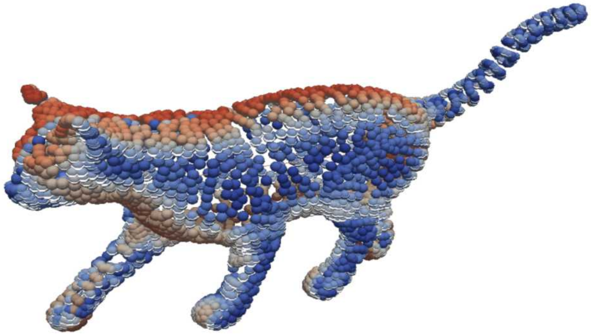
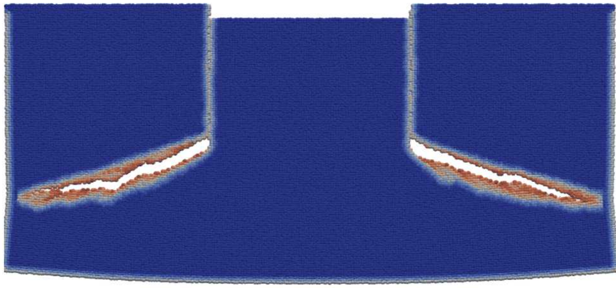
SAND2020-6936C







Physics-informed graph neural nets

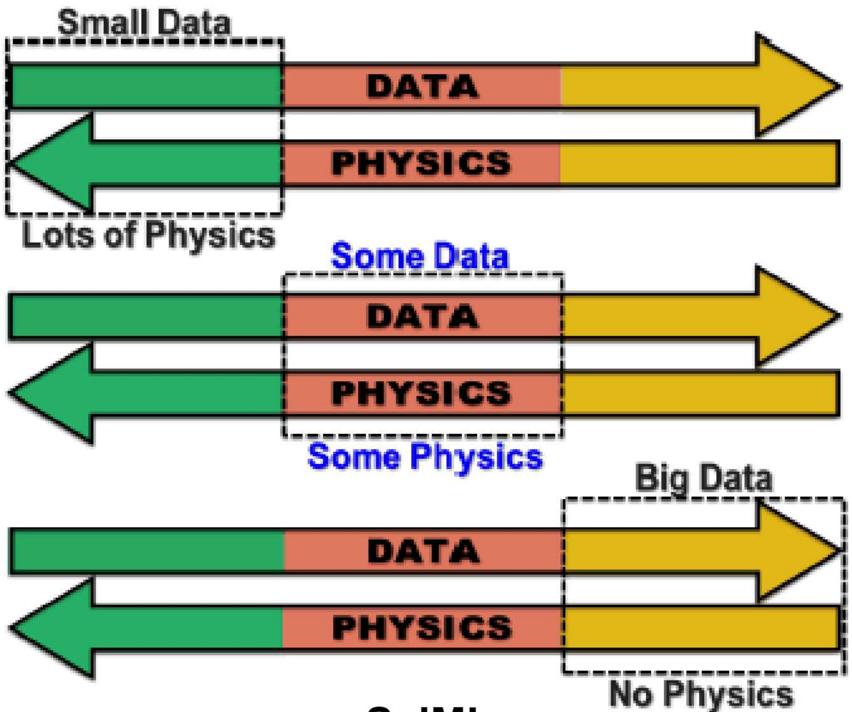
A unification of NN architectures with mimetic PDE discretization

Nat Trask
Center for Computing Research
Sandia National Laboratories

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.

What is scientific machine learning?

Traditional scientific computing:
Known model, known theory leading to good discretization with FEM, data primarily for V+V, parameter estimation



Traditional machine learning:
No physics, unknown input/output relationship, learn on huge amounts of data + universal approximation

SciML:
Known model form, unknown constitutive relationships or closures, small amount of high fidelity data

Objective: Develop ML tools to extract physics preserving data-driven models and learn inexpensive surrogates

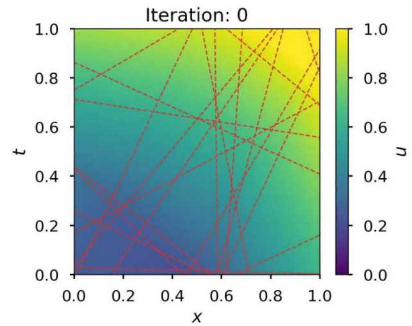
What's out there right now?

Most methods pursue some notion of physics regularization to weakly endow network with desirable properties

Make list of desired features and penalize them after the fact: PDE structure, BC, IC, conservation, etc

$$\mathbf{L} = \mathbf{L}_{data} + \epsilon \mathbf{L}_{physics}$$

$$\mathbf{L} = ||u_{data} - \mathcal{NN}||_{\ell_2}^2 + \epsilon||\mathcal{L}[u_{data}] - \mathcal{L}[\mathcal{NN}]||_{\ell_2}^2$$



The Good….

It actually works – many first of their kind results in surrogate models, SPDE, inverse problems, etc

The Bad….

Many penalty parameters lead to large numbers of hyperparameters, challenging to train, demonstrate convergence/stability, difficult to handle multiphysics

Can we use ideas from physics-compatible PDE discretization to do physics-informed machine learning *in a strong sense*?

Isaac E Lagaris, Aristidis Likas, and Dimitrios I Fotiadis. Articial neural networks for solving ordinary and partial differential equations. IEEE Transactions on Neural Networks, 1998

Dongkun Zhang, Lu Lu, Ling Guo, and George Em Karniadakis. Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems., 2019.

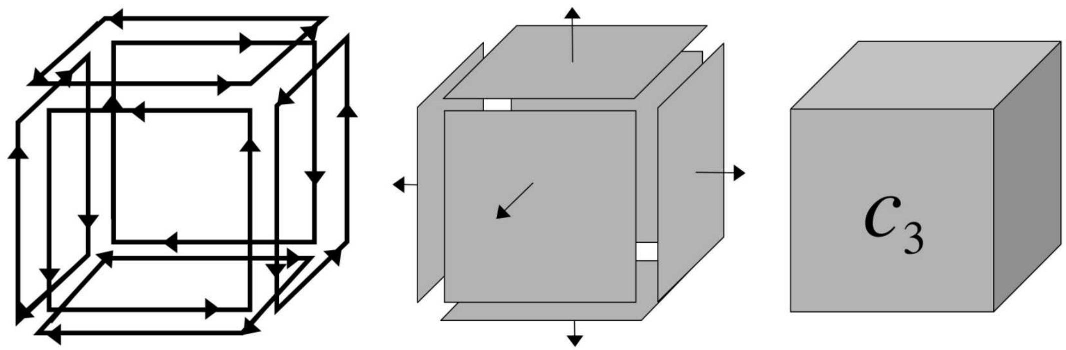
Xuhui Meng and George Em Karniadakis. A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse PDE problems., 2020.

Zhiping Mao, Ameya D Jagtap, and George Em Karniadakis. Physics-informed neural networks for high-speed flows., 2020.

Dongkun Zhang, Ling Guo, and George Em Karniadakis. Learning in modal space: Solving time-dependent stochastic pdes using physics-informed neural networks, 2019.

Zhiping Mao, Zhen Li, and George Em Karniadakis. Nonlocal flocking dynamics: Learning the fractional order of pdes from particle simulations. 2019

What are physics compatible discretizations for PDEs?



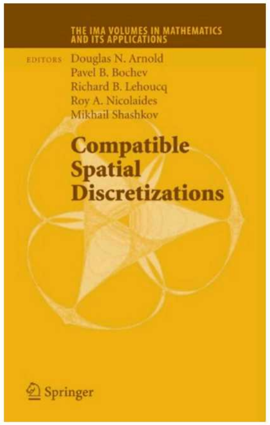
$$0 \leftarrow \partial\partial c_3 \xleftarrow{\;\partial\;} \partial c_3 \xleftarrow{\;\partial\;} c_3$$

Methods for solving PDEs which:

Use generalized Stokes theorems to approximate differential operators

Preserve topological structure in governing equations

Mimic properties of continuum operators (thus sometimes called **mimetic discretizations**)



Arnold, D. N., Bochev, P. B., Lehoucq, R. B., Nicolaides, R. A., & Shashkov, M. (Eds.). (2007). *Compatible spatial discretizations* (Vol. 142). Springer Science & Business Media.

Two key ingredients:

1: A topological structure

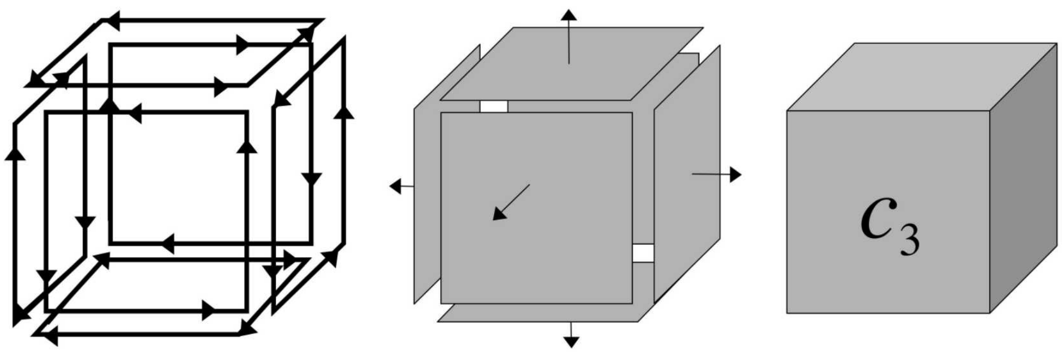
In PDE discretization this is a mesh, with boundary operators linking cells, faces, edges, and nodes

We will use a graph as an inexpensive low-dimensional mesh surrogate

2: Metric information

Measures associated with mesh entities, ensuring discrete exterior derivatives converge to div/grad/curl

Graphs are purely topological with no natural metric, we will use ML to extract metric information from data

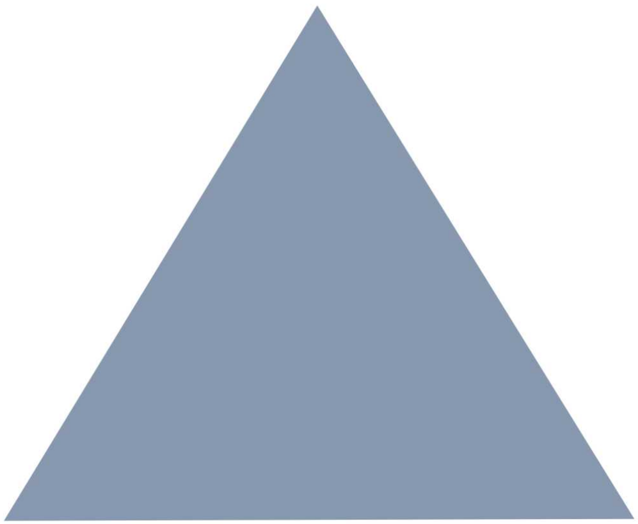


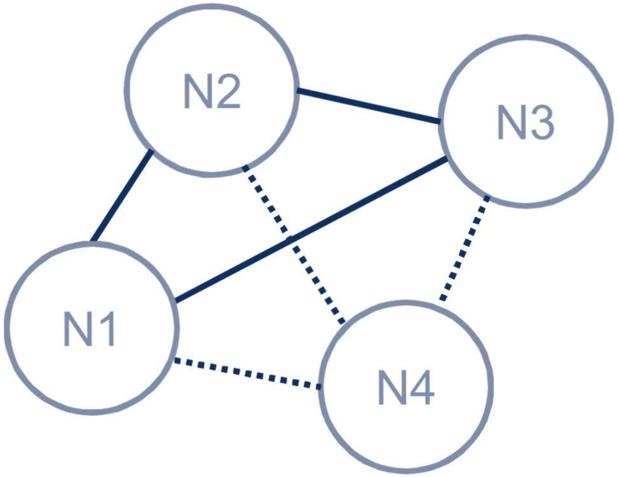
$$0 \leftarrow \partial\partial c_3 \xleftarrow{\partial} \partial c_3 \xleftarrow{\partial} c_3$$

$$\nabla \cdot \mathbf{u} = \frac{1}{\mu(C)} \sum_{f \in \partial C} \int_f \mathbf{u} \cdot d\mathbf{A}$$

Exterior calculus preliminaries: chain complex

$$C_0 \xleftarrow{\partial_0} C_1 \xleftarrow{\partial_1} C_2 \xleftarrow{\partial_2} C_3$$

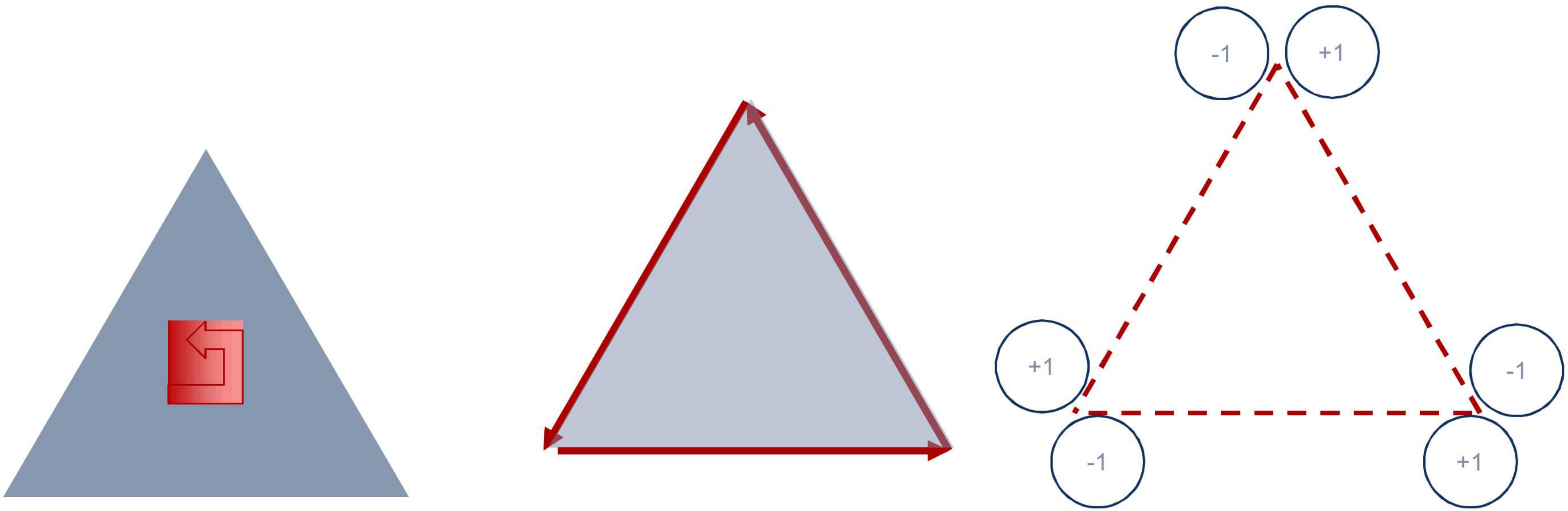




Compat. PDE	Comb. Hodge
Mesh entities	K-cliques

Exterior calculus preliminaries: chain complex

$$C_0 \xleftarrow{\partial_0} C_1 \xleftarrow{\partial_1} C_2 \xleftarrow{\partial_2} C_3$$



$f \in C_2$ $\quad$ $\partial_2 f \in C_1$ $\quad$ $\partial_1 \partial_2 f \in C_0$

Exact sequence property: $\forall k, \partial_k \partial_{k+1} = 0$

Exterior calculus preliminaries: cochain complex

$$C_0 \xleftarrow{\partial_0} C_1 \xleftarrow{\partial_1} C_2 \xleftarrow{\partial_2} C_3$$

$$C^0 \xrightarrow{d_0} C^1 \xrightarrow{d_1} C^2 \xrightarrow{d_2} C^3$$

Coboundary operators define maps $d_k : C^k \rightarrow C^{k+1}$ satisfying $d_{k+1} d_k = 0$

Boundary and coboundary operators satisfy the *generalized Stokes theorem*

$$\int_\omega du = \int_{\partial\omega} u$$

Comb. Hodge	Compat. PDE
$grad[s](i,j) = \int_{e_{ij}} \nabla s \cdot d\mathbf{l} = s_j - s_i$ $curl[X](F) = \int_F \nabla \times X \cdot d\mathbf{A} = \sum_{e \in \partial F} \int_e X \cdot d\mathbf{l}$	$grad[s](i,j) = s_j - s_i$ $curl[X](i,j,k) = X_{ij} + X_{jk} + X_{ki}$

Exterior calculus preliminaries: codifferentials

$$C_0 \xleftarrow{\partial_1} C_1 \xleftarrow{\partial_2} C_2 \xleftarrow{\partial_3} C_3$$

$$C^0 \underset{d_0^\star}{\overset{d_0}{\rightleftarrows}} C^1 \underset{d_1^\star}{\overset{d_1}{\rightleftarrows}} C^2 \underset{d_2^\star}{\overset{d_2}{\rightleftarrows}} C^3$$

Introducing inner products $(\cdot,\cdot)_k$, we define the *codifferential* operator $d_k^\star : C^{k+1} \to C^k$ as

$$(v, d_k^\star u)_k = (d_k v, u)_{k+1}$$

Again, $d_{k-1}^\star * d_k^\star = 0$

Exterior calculus preliminaries: codifferential examples

$$C^0 \underset{d_0^\star}{\overset{d_0}{\rightleftarrows}} C^1 \underset{d_1^\star}{\overset{d_1}{\rightleftarrows}} C^2 \underset{d_2^\star}{\overset{d_2}{\rightleftarrows}} C^3$$

$$(v, d_k^\star u)_k = (d_k v, u)_{k+1}$$

How to choose inner-products?

PDE context

Covolume methods - Hodge star

Mimetic finite difference – As needed to get accuracy

Mixed FEM – L2, but carefully design FEM spaces

Graph context

Purely topological, so no concern with consistency

Therefore, classically choose ℓ_2 inner product so that codifferential is simply the adjoint of the coboundary matrix

Why not use data to bridge the gap?

What does all this give you?

$$C^0 \underset{d_0^\star}{\overset{d_0}{\rightleftarrows}} C^1 \underset{d_1^\star}{\overset{d_1}{\rightleftarrows}} C^2 \underset{d_2^\star}{\overset{d_2}{\rightleftarrows}} C^3$$

$$(v, d_k^\star u)_k = (d_k v, u)_{k+1}$$

- Differential operators which locally and globally conserve fluxes, circulations, potentials
- Invertible Hodge Laplacians $\Delta_k = d_{k+1}^\star d_{k+1} + d_k d_k^\star$
- Exact sequence properties $d_{k+1} d_k = d_k^\star d_{k+1}^\star = 0$

Corollary: Treatment of non-trivial nullspaces

Linear PDE $\nabla \times \nabla \times u = f$ admit $\tilde{u} = u + \nabla\phi, \forall\phi$

Can restrict solutions perpendicular to null space by imposing gauge condition

$$d_1^\star d_1 u + d_0 \lambda = f$$

$$d_0^\star u = 0$$

Data-driven codifferential

$$C^0 \underset{d_0^\star}{\overset{d_0}{\rightleftarrows}} C^1 \underset{d_1^\star}{\overset{d_1}{\rightleftarrows}} C^2 \underset{d_2^\star}{\overset{d_2}{\rightleftarrows}} C^3$$

$$(v, d_k^\star u)_k = (d_k v, u)_{k+1}$$

For finite dimensional space, defining inner-product amounts to finding SPD matrix M_k, such that $(x, y)_k = x^\intercal M_k y$

Options for parameterizing inner-product:

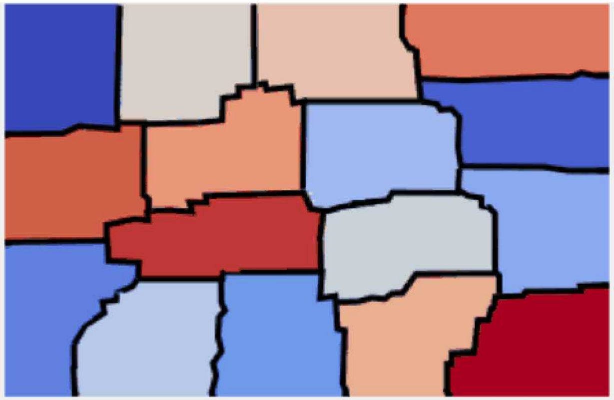
- Any SPD matrix can be expressed via Cholesky decomp $M_k = QQ^\intercal$, upper-triangular Q_{ij} with trainable weights
- Simpler example, $M_k = diag(\xi)$, with $\xi_k > 0$ trainable weights

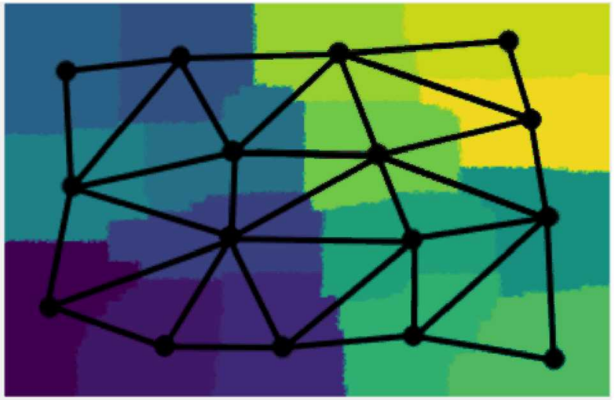
Second choice corresponds to higher-order generalizations of *resistor networks*

Physics-informed graph neural networks (pigNNs)

$$\nabla \cdot \mathbf{F} = f \qquad d_0^\star \mathbf{F} = f$$

$$\mathbf{F} + \kappa \nabla \phi = 0 \qquad \mathbf{F} + \xi d_0 \phi + \mathcal{N}_\eta(\phi) = 0$$





High-fidelity PDE solution

Apply graph-cut to coarse-grain chain complex

Average over partitions to obtain training data

Ideal optimization problem

$$d_0^\star \mathbf{F} = f$$

$$\mathbf{F} + \xi d_0 \phi + \mathcal{N}_\eta(\phi) = 0$$

$$\Rightarrow \quad a_\xi(\phi, v) + N_v^\eta(\phi) = b(v)$$

Invertible bilinear form

Nonlinear perturbation

Rewriting as variational problem, we have a nonlinear perturbation of a nice elliptic problem. Conservation is encoded **strongly** via codifferential

$$\underset{\xi,\eta}{argmin} ||\phi - \phi_{data}||^2 + ||\xi||^2$$

$$\text{s.t. } d_0^\star F = 0$$

If we can fit the model to data while imposing equality constraint, then during training we restrict to manifold of solvable models preserving physics

Optimization problem (PINNs version)

$$d_0^\star \mathbf{F} = f$$
$$\mathbf{F} + \xi d_0 \phi + \mathcal{N}_\eta(\phi) = 0$$

$$\Rightarrow \quad a_\xi(\phi, v) + N_v^\eta(\phi) = b(v)$$

Invertible bilinear form

Nonlinear perturbation

$$\underset{\xi,\eta}{argmin} ||\phi - \phi_{data}||^2 + ||\xi||^2 + \lambda||d_0^\star F||^2$$

Penalty parameter

Physics residual

An obvious choice to find our desired model is to regularize PINNs style and play with penalty parameter

Wang, Sifan, Yujun Teng, and Paris Perdikaris. "Understanding and mitigating gradient pathologies in physics-informed neural networks."*arXiv preprint arXiv:2001.04536*(2020).

Optimization problem ("PDE"-constrained)

$$\mathcal{L}_{\xi,\eta,\lambda} = ||\phi - \phi_{data}||^2 + ||\xi||^2 + \lambda^\intercal d_0^\star F$$

$$\text{s.t. } d_0^\star F = 0$$

Vector of Lagrange multipliers

Physics residual at each node

An iterative algorithm guaranteeing exact enforcement of physics at each iteration:

- *Solve forward problem with current parameters*

$$\phi \leftarrow \nabla_\lambda \mathcal{L}_{\xi,\eta,\lambda}(\phi) = 0$$

- *Solve adjoint problem to get Lagrange multipliers*

$$\lambda \leftarrow \nabla_\phi \mathcal{L}_{\xi,\eta,\lambda}(\phi) = 0$$

- *Update model coefficients*

$$\xi, \eta \leftarrow \nabla_{\xi,\eta} \mathcal{L}_{\xi,\eta,\lambda}(\phi) = 0$$

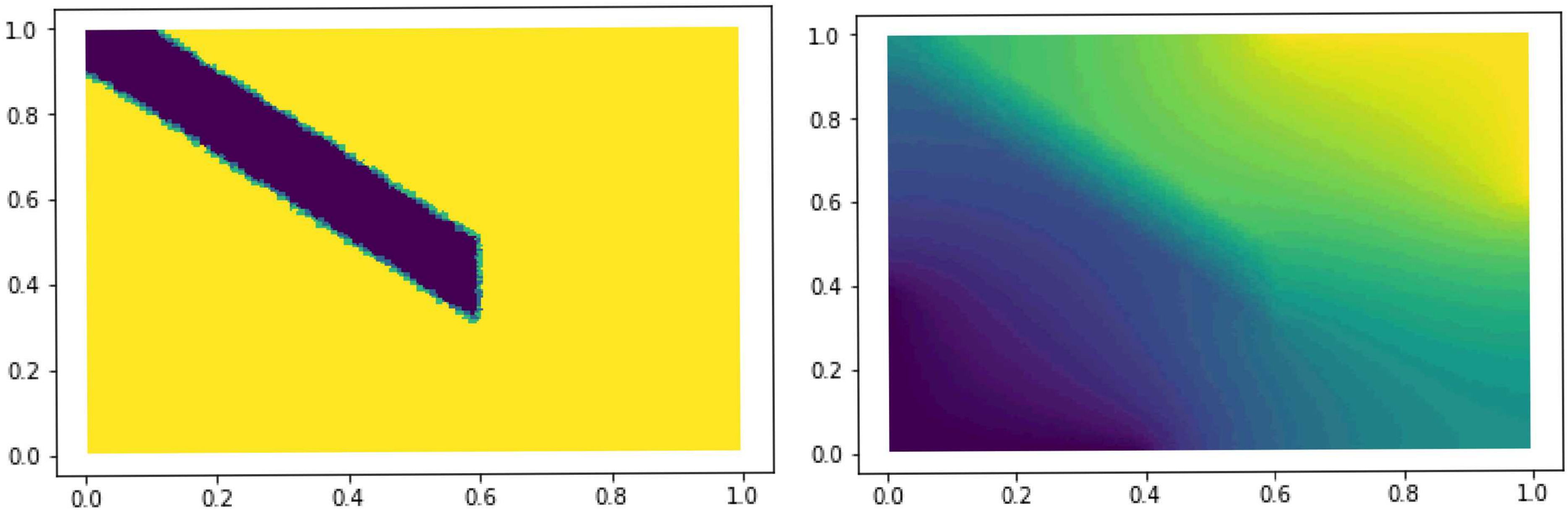
Back to Darcy…

$$\nabla \cdot \mathbf{F} = f$$

$$\mathbf{F} + \kappa \nabla \phi = 0$$

$$d_0^\star \mathbf{F} = f$$

$$\mathbf{F} + \xi d_0 \phi + \mathcal{N}_\eta(\phi) = 0$$



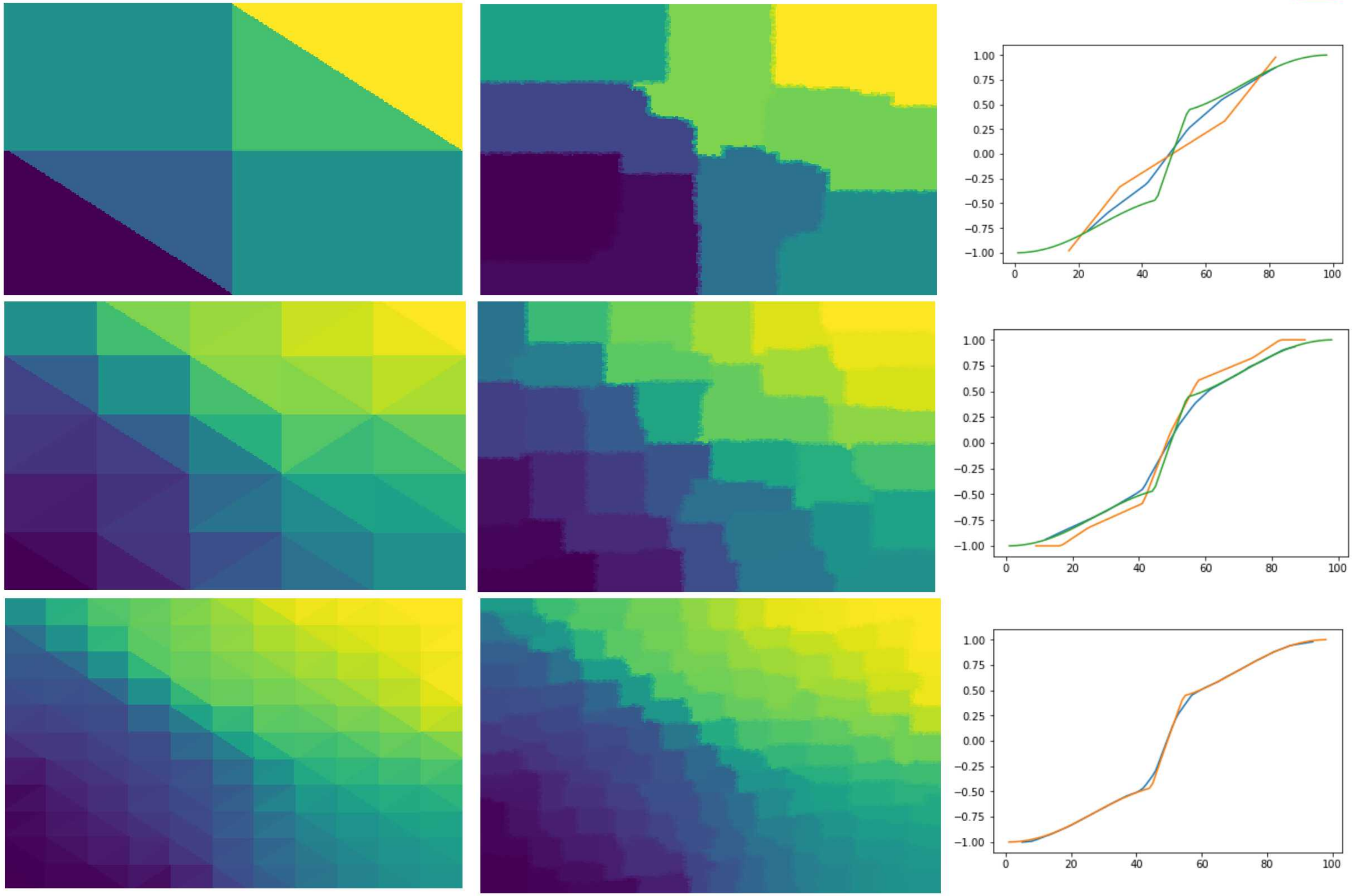
Comparison to traditional covolume: improved accuracy at low resolution

Sandia National Laboratories

N = 2^2

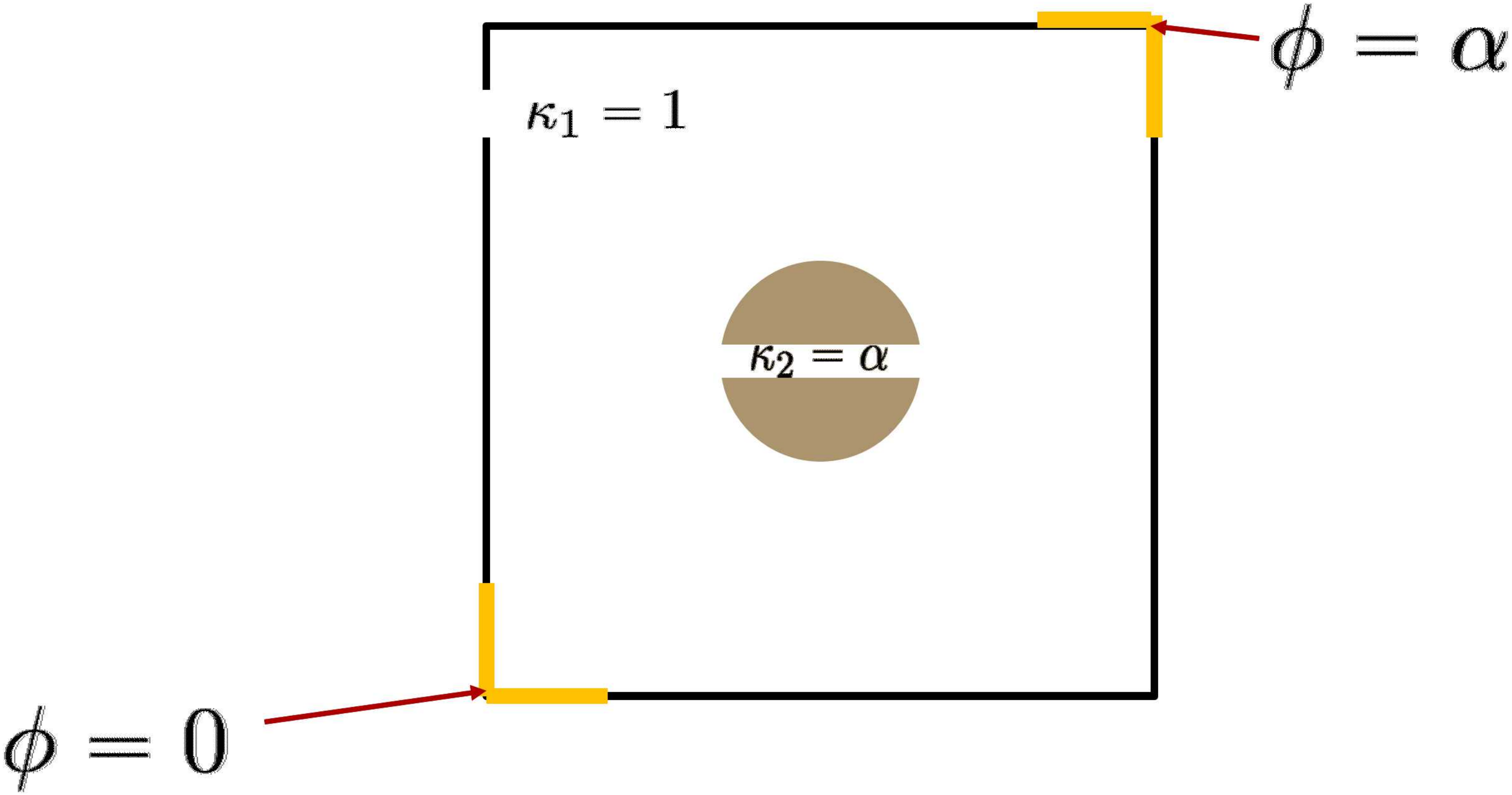
N = 5^2

N = 10^2



Comparison of pressure for same # DOF for FVM (left) and pigNN (center)
Right: profile along diagonal shows better fit to solution (green) by pigNN (blue) vs FVM (orange)

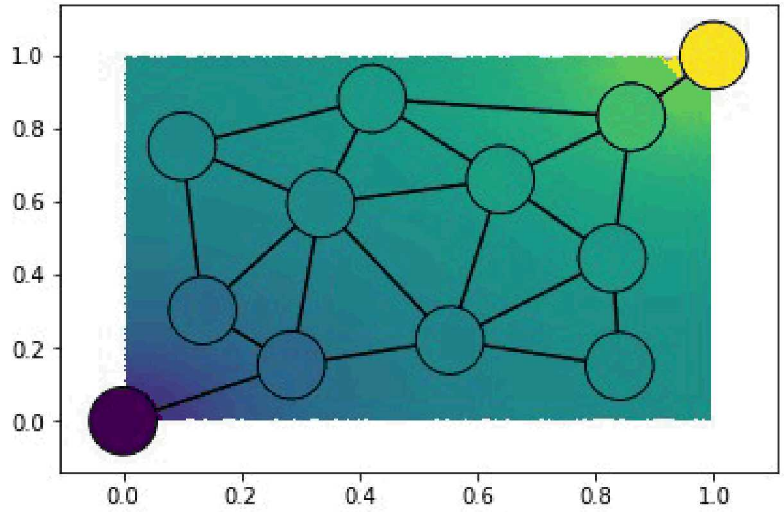
Nonlinear Darcy

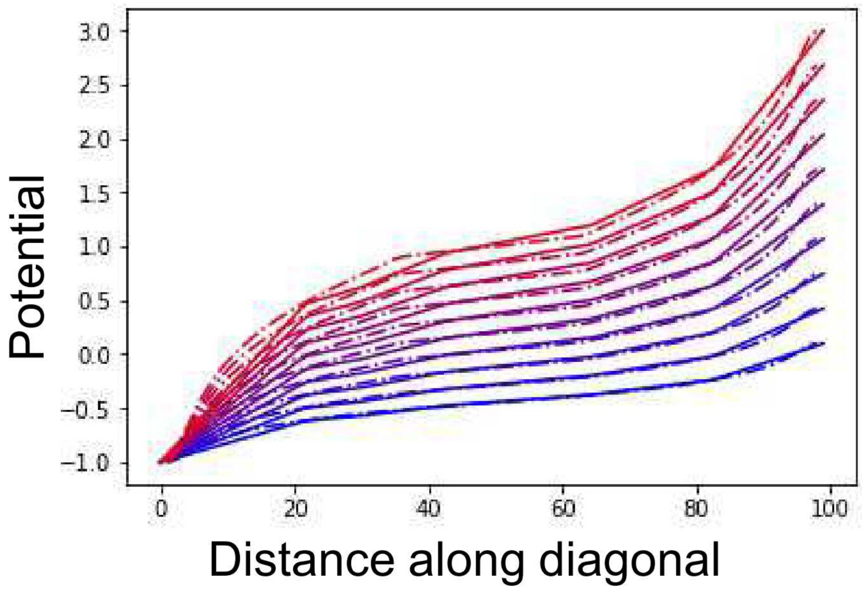


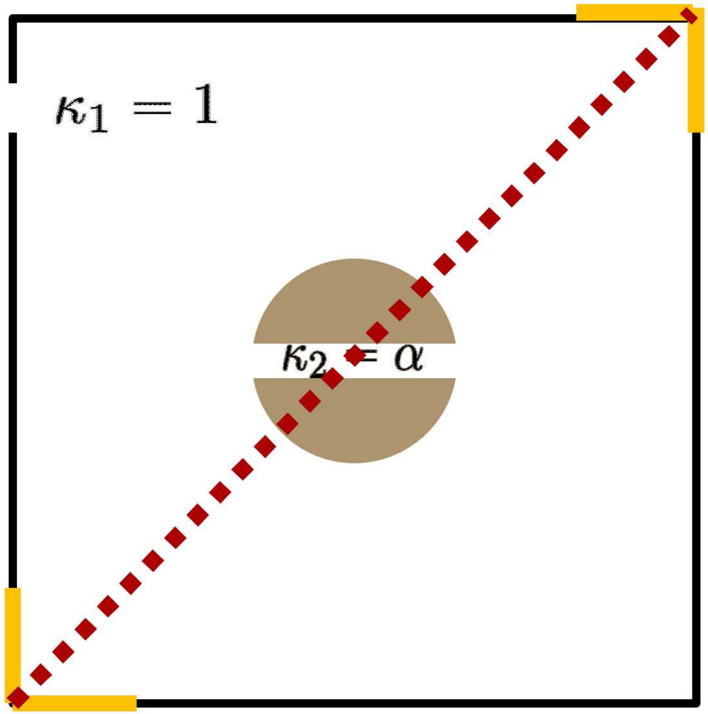
$$d_0^\star \mathbf{F} = f$$

$$\mathbf{F} + \xi d_0 \phi + \mathcal{N}_\eta(\phi) = 0$$

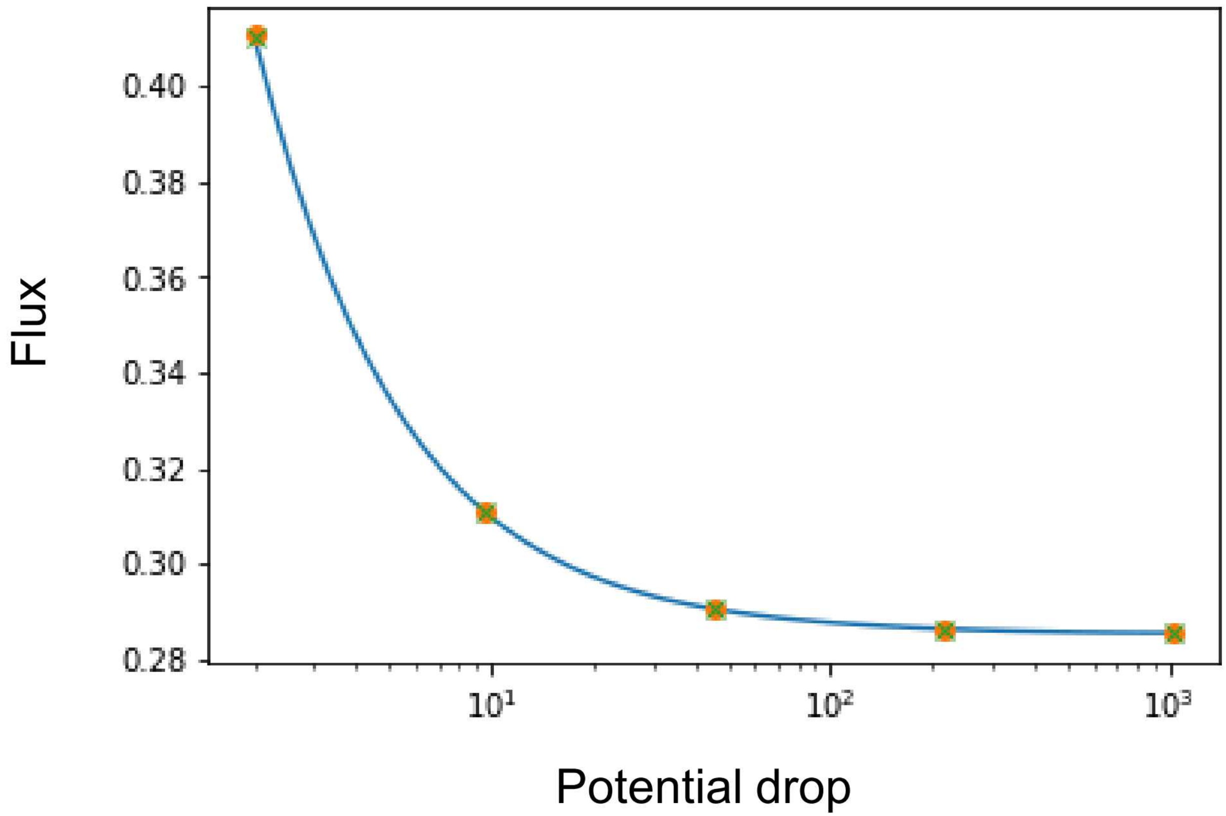
Nonlinear Darcy: potential profile across diagonal







Nonlinear Darcy – Dirichlet2Neumann map

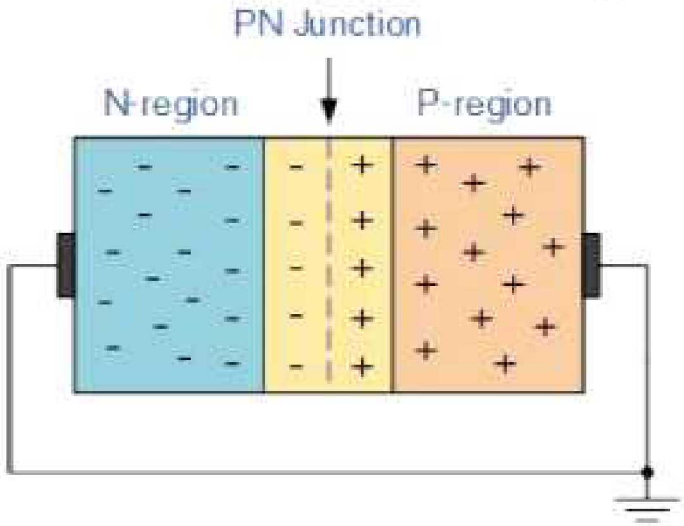


Training on **five** PDE solutions across **three decades** of data

An effective parameterization of D2N map, which may be embedded in other schemes

Compact models for semiconductors: PN-diode

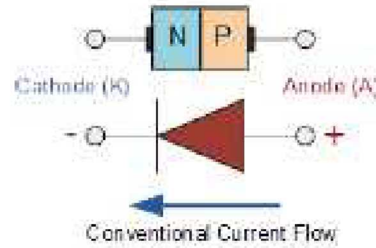
https://www.electronics-tutorials.ws/diode/diode_3.html



$$\nabla \cdot \epsilon \nabla \phi = -(p - n + N_D^+ - N_A^-)$$

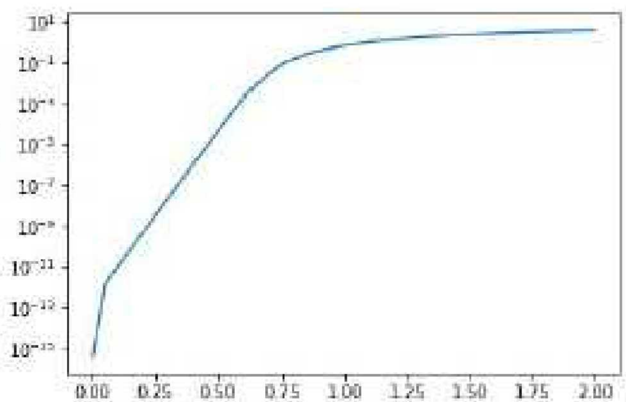
$$\frac{\partial n}{\partial t} = \frac{1}{q}\nabla \cdot (-\mu_n n E - D_n \nabla n) - R_n(n,p)$$

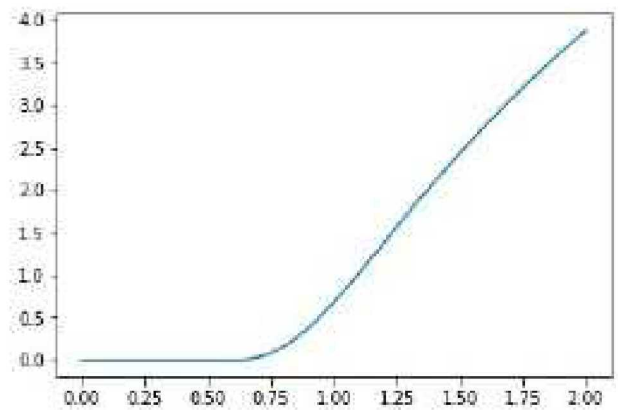
$$\frac{\partial p}{\partial t} = -\frac{1}{q}\nabla \cdot (\mu_n p E - D_p \nabla p) - R_p(n,p)$$



Traditional compact models fit ideal diode + resistor, and can be tuned to match either small or large voltage regimes

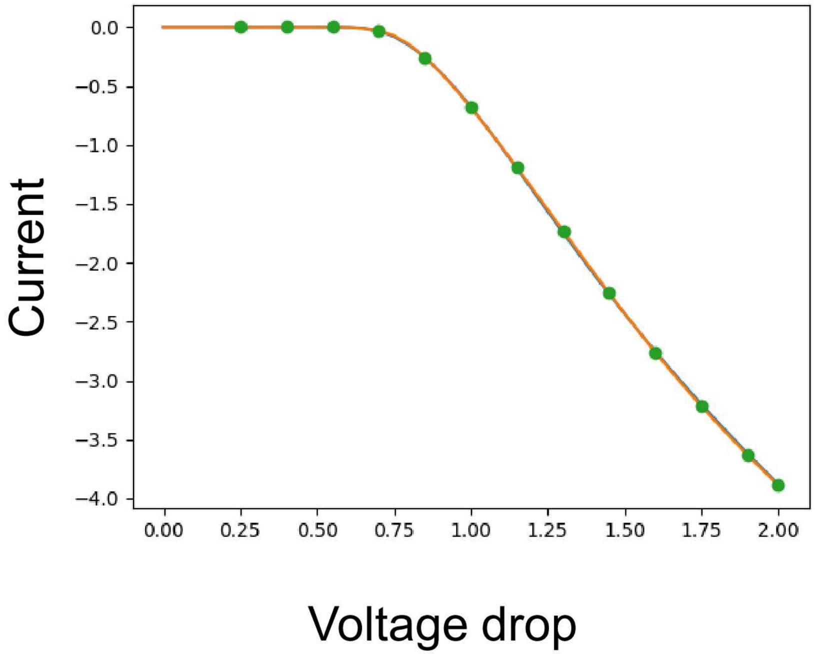
Locally exponential: I.D. model

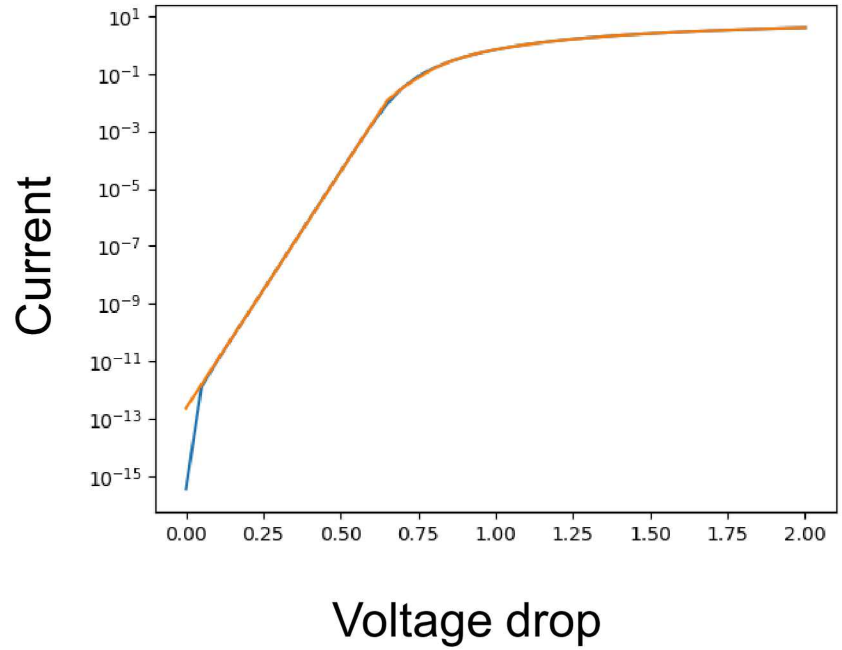




Locally linear: resistor model

Matching IV-curve – linear scale





Extract a conservative surrogate accurate over
fifteen orders of magnitude

May be embedded in a circuit simulator (e.g. Xyce) to couple coarse-grained high-fidelity PDE model in multiscale model w/ millions of components

Acknowledgements

- **PHILMs** – **Ph**ysics **I**nformed **L**earning **M**achine**s** for multiscale/multiphysics problems
  - ASCR MMICCs center at the intersection of machine learning and scientific computing
  - PI: George Karniadakis
  - SNL team: Mike Parks (PI), Pavel Bochev, Marta D'Elia, Mamikon Gulian, Ravi Patel, Mauro Perego, Nathaniel Trask
- **PIRAMID** – **P**hysics **I**nformed **R**apid and **A**utomated **ML** for compact model development
  - SNL LDRD to extract efficient compact circuit models from high-fidelity PDE simulation
  - Team: Andy Huang (PI), Xujiao Gao, Shahed Reza, Nathaniel Trask
- **DOE Early Career –** Physics informed graph neural networks for multiscale physics

Applications

Non-equilibrium closures for autoignition in turbulent combustion

Pulse shaping for pulsed power fusion applications on Z-machine

Development of surrogate models for radiation modeling of circuits

Fracture mechanics closures for ice sheet models

Multiscale modeling of lithium-ion batteries during failure

Multiscale closure for subsurface flow through fracture networks

Multiscale data-driven closures for kinetic effects and turbulence in plasmas

Several new projects – please reach out (natrask@sandia.gov) if you're on the job market!