Scientific Machine Learning Workshop

Lecture 1: Introduction and Overview

Instructor: Ulisses M. Braga-Neto

Department of Electrical and Computer Engineering

Scientific Machine Learning Lab (SciML Lab)

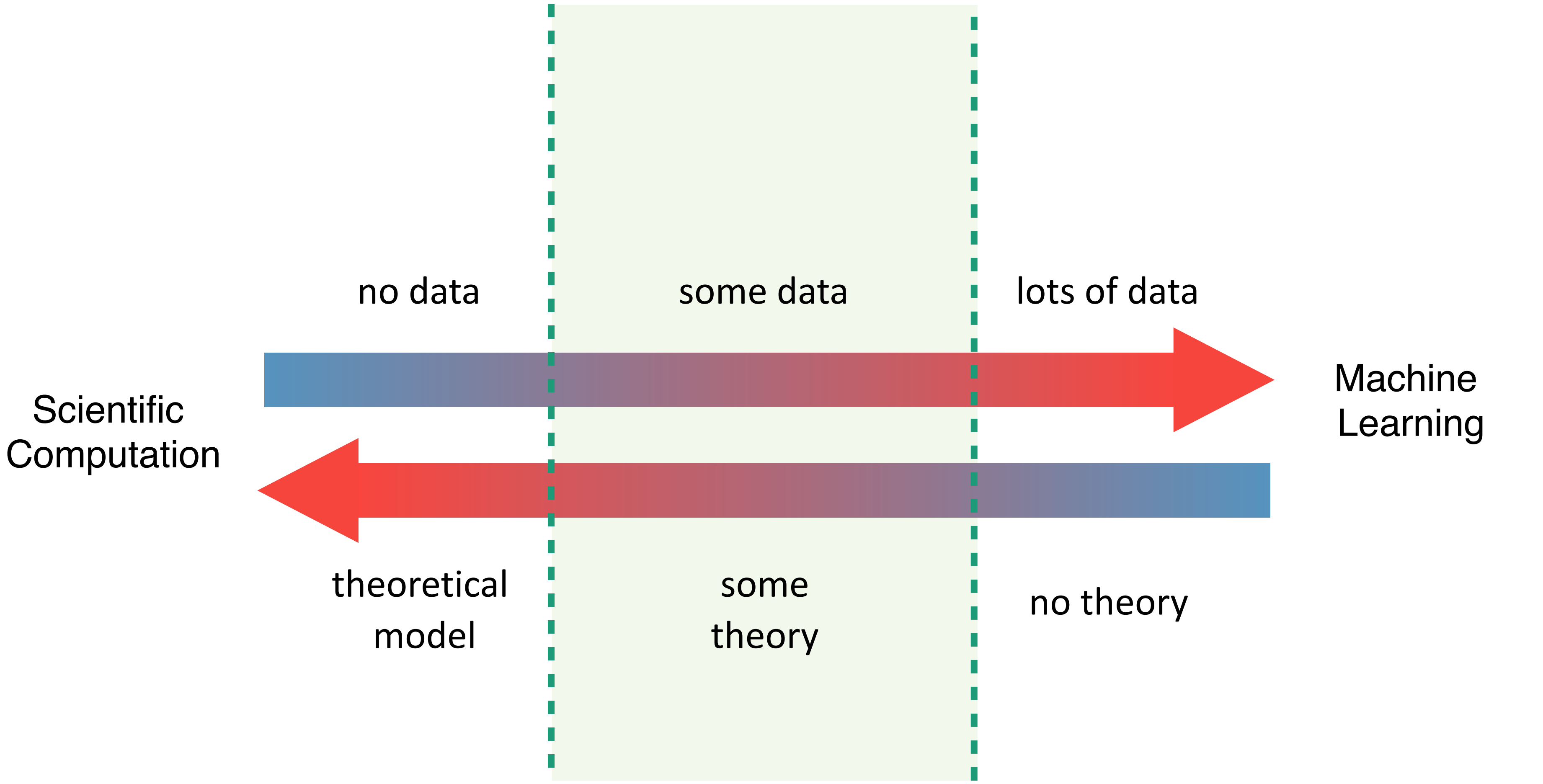
Texas A&M University Institute of Data Science (TAMIDS)

Texas A&M University

10/8/2025 - 13/8/2025
Cenpes, Rio de Janeiro

PETROBRAS

Scientific Machine Learning



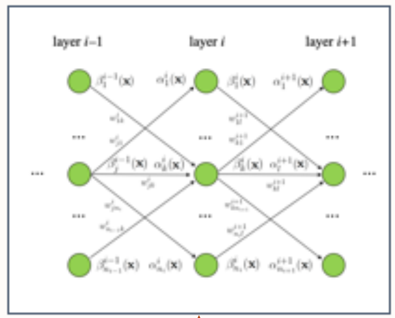
Physics-Informed Machine Learning

forward modeling / prediction

input

physical variables and parameters

machine learning module



$$
\begin{aligned}
&\mathcal{N}_{\boldsymbol{x},t}[u(\boldsymbol{x},t)] = 0\,, \quad \boldsymbol{x} \in \Omega\,,\ t \in [0,T]\,,\\
&u(\boldsymbol{x},t) = g(\boldsymbol{x},t)\,, \quad \boldsymbol{x} \in \partial\Omega,\ t \in [0,T]\,,\\
&u(\boldsymbol{x},0) = h(\boldsymbol{x})\,, \quad \boldsymbol{x} \in \Omega\,.
\end{aligned}
$$

physical laws

output

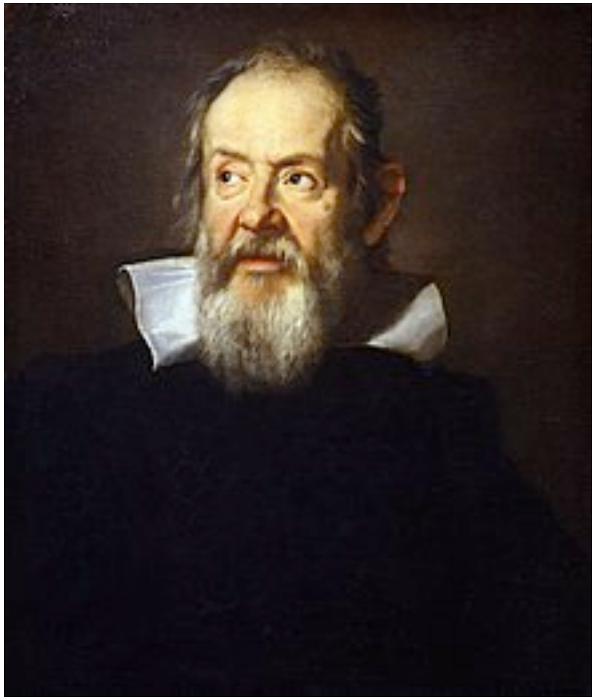
physical quantities of interest

inverse modeling / inference

Scientific Laws are Expressed as Equations

"The book of nature is written in the language of mathematics."

— Galileo Galilei.



The laws of physics are based on principles of conservation (of mass, momentum, and energy), and symmetries, all of which can be written as equations.

Scientific Laws are Expressed as Equations

1. Algebraic equations:

$$u = f(u; \lambda)$$

For example, Newtons' Law of Universal Gravitation:

$$u \; = \; G\,\frac{m_1 m_2}{r^2} \qquad \text{(gravitational force)}$$

In general, one has to solve an implicit equation $F(u) = 0$.

2. Ordinary differential equations (ODE):

$$\frac{du(t)}{dt} = f(u(t), t; \lambda)$$

For example, the discharge of a capacitor:

$$\frac{du(t)}{dt} = \frac{1}{RC}\,u(t) \quad \text{(autonomous linear ODE)}$$

In general, one has to solve an implicit equation $F(u, t, u', u'', \ldots, u^{(n)}) = 0$.
The ODE is linear if F is linear. It is autonomous if F does not depend on t .

Scientific Laws are Expressed as Equations

3. Partial differential equations (PDE) of order k (for scalar u):

$$\frac{\partial u(\mathbf{x}, t)}{\partial t} = f(u(\mathbf{x}, t), x, t, Du(\mathbf{x}, t), D^2u(\mathbf{x}, t), \ldots, D^k u(\mathbf{x}, t); \lambda)$$

Here, $\mathbf{x} \in R^d$ and $D^k u$ are all spatial derivatives of order k . E.g. with $d = k = 2$, $D^2 u = \{u_{x_1 x_1}, u_{x_1 x_2}, u_{x_2 x_2}\}$. As an example, the heat equation with $d = 1$ and $k = 2$:

$$\frac{\partial u(x, t)}{\partial t} = \kappa \frac{\partial^2 u(x, t)}{\partial x^2}$$

(This is a 2nd-order linear PDE.)

In general, one has to solve the implicit equation (where $\mathbf{x}$ may include time):

$$F(u, \mathbf{x}, Du, D^2u, \ldots, D^k u; \lambda) = 0$$

A PDE of order k is:

(a) *linear:* F is linear in all its arguments (e.g., the diffusion equation).

(b) *semilinear:* F is linear in $D^k u$. The coefficients in $D^k u$ can depend on $\mathbf{x}$.

(c) *quasilinear:* F is linear in $D^k u$. The coefficients in $D^k u$ can depend on $u, \mathbf{x}, Du, D^2u, \ldots, D^{k-1}u$.

(d) *nonlinear:* F is not linear (this includes semilinear and quasilinear PDES).

Scientific Laws are Expressed as Equations

Reaction-Advection-Diffusion PDEs describe most physical phenomena.

Example ($d = 1$, no sources):

$$\frac{\partial u(x,t)}{\partial t} = \underbrace{\frac{\partial}{\partial x} f(u(x,t))}_{\text{(advection)}} + \underbrace{\frac{\partial}{\partial x}\left(\kappa(x,t)\frac{\partial u(x,t)}{\partial x}\right)}_{\text{(diffusion)}} + \underbrace{g(u(x,t))}_{\text{(reaction)}}$$

Compact notation: $u_t = (f(u))_x + (\kappa u_x)_x + g(u)$.

This gives rise to another classification scheme for time-evolution PDEs:

(a) hyperbolic: purely advective ("wave") phenomena (can evolve into discontinuous solutions starting from smooth conditions).

(b) *parabolic:* includes diffusion (solution always smooth).

(c) *elliptic:* no time derivative (space-varying steady-state phenomena).

Initial and Boundary Conditions. In order to obtain a unique solution, additional information must be provided in terms of a spatial-temporal *domain*, and the value of the solution on the boundary of the domain, which are called the initial and boundary conditions. These additional conditions make the problem *well-posed.*

Example: SEIRD ODE System

The SEIRD model predicts the time evolution of an epidemic. It models the dynamic interaction of the susceptible (S), the exposed (E), the infected (I), the recovered (R) and the deceased (D), according to the ODE system:

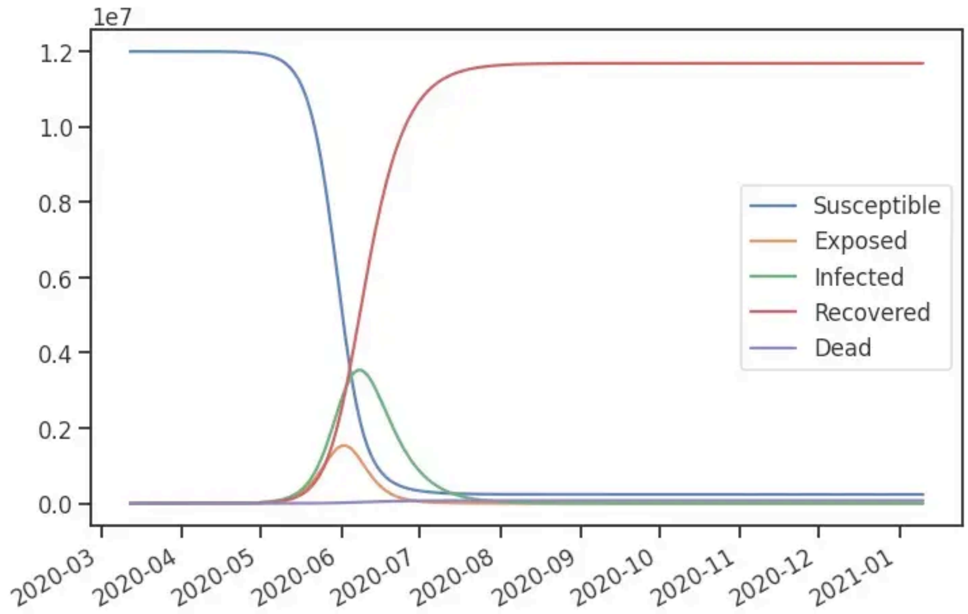
$$\frac{dS(t)}{dt} = -\lambda_S S(t) I(t)$$

$$\frac{dE(t)}{dt} = \lambda_S S(t) I(t) - \lambda_E E(t)$$

$$\frac{dI(t)}{dt} = \lambda_E E(t) - (\lambda_R + \lambda_D) I(t)$$

$$\frac{dR(t)}{dt} = \lambda_R I(t)$$

$$\frac{dD(t)}{dt} = \lambda_D I(t)$$



Source: https://towardsdatascience.com/how-to-actually-forecast-covid-19-778cce27b9d6

where

$$S(t) + E(t) + I(t) + R(t) + D(t) = 1$$

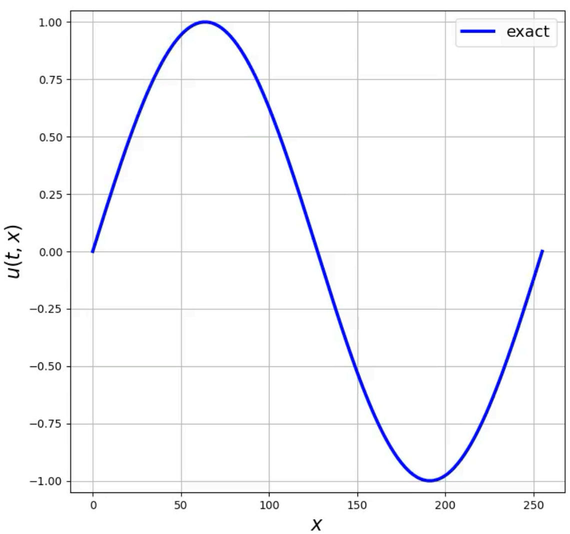
Example: Viscous Burger's Equation

Classical quasilinear PDE in computational fluid dynamics.

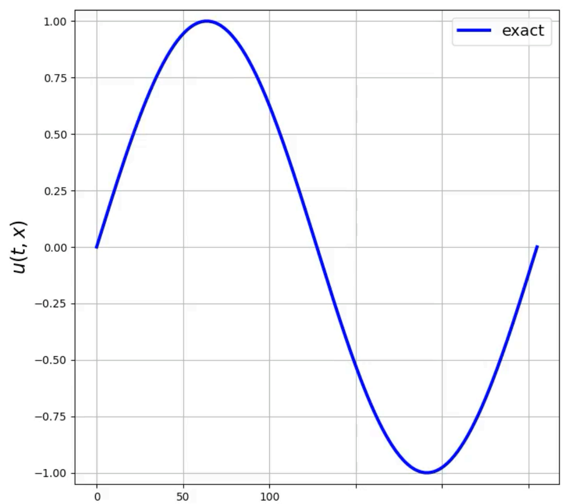
$$
\begin{aligned}
&u_t + u u_x - \nu u_{xx} = 0, \quad x \in [-1,1], t \in [0,1] \\
&u(0,x) = -\sin(\pi x) \ \text{(initial condition)} \\
&u(t,-1) = u(t,1) = 0 \ \text{(Dirichlet boundary condition)}
\end{aligned}
$$

The parameter ν is the kinematic viscosity. With large ν, diffusion dominates:

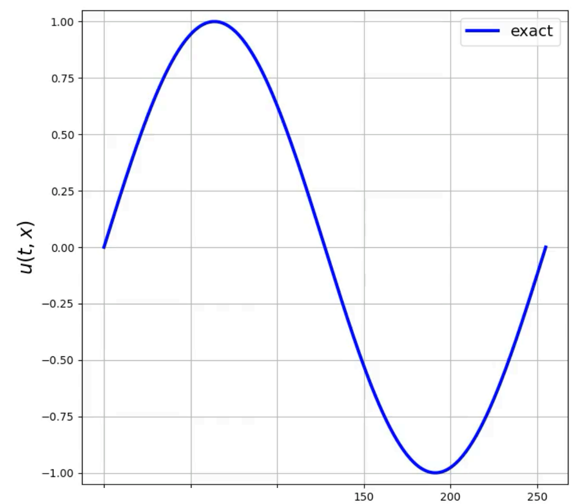
$\nu = 1$



$\nu = 0.01$



$\nu = 0.001$



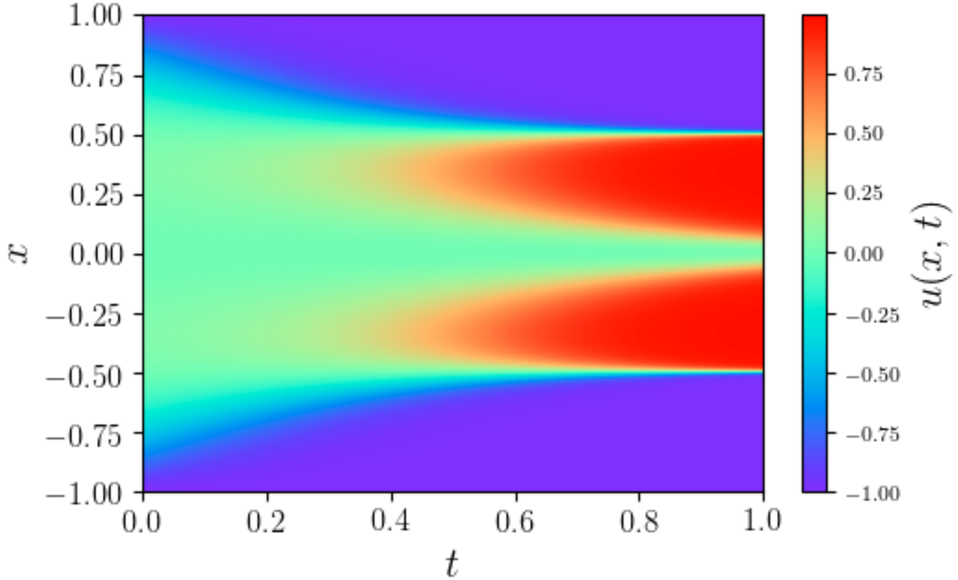
TAMIDS-SciML
DIVISION OF RESEARCH

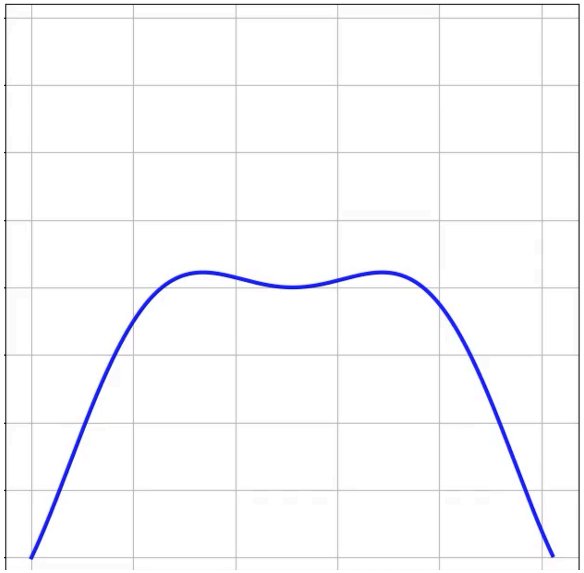
Example: Allen-Cahn Equation

Semilinear reaction-diffusion PDE with double-well potential, used in **phase-field modeling**

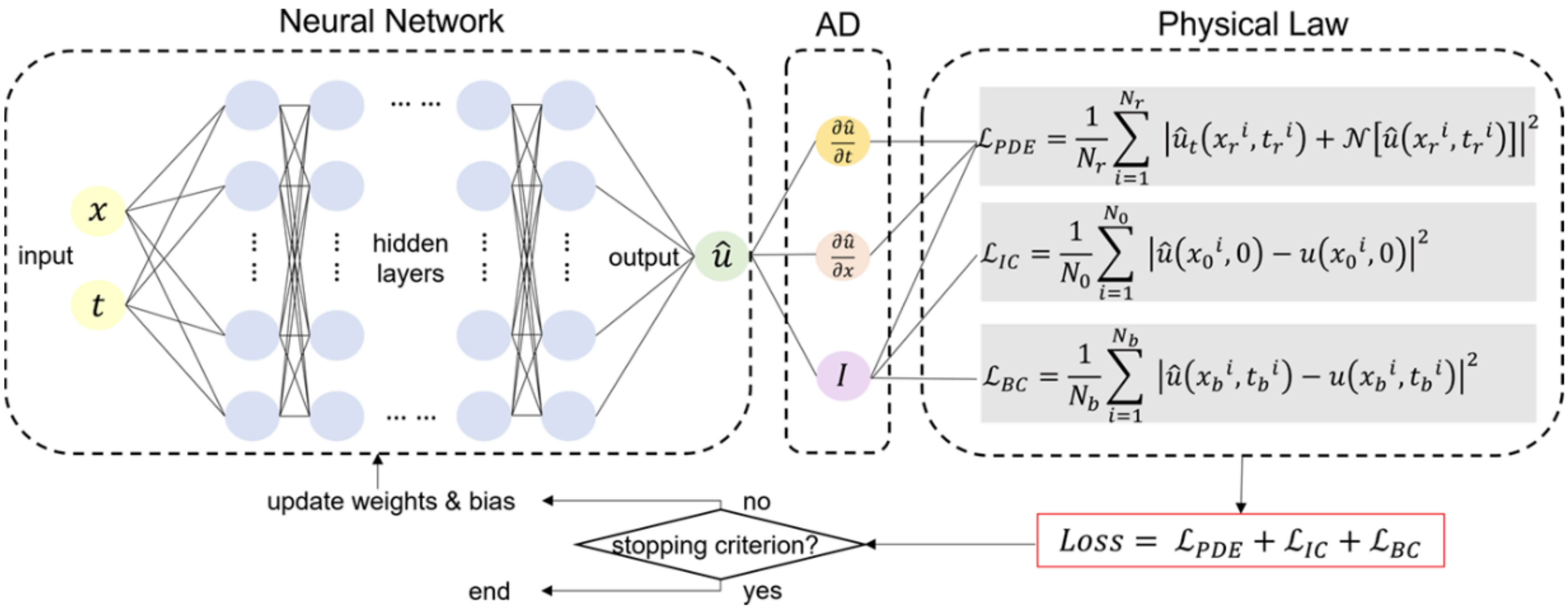
$$
\begin{aligned}
&u_t - 0.0001u_{xx} + 5u^3 - 5u = 0\,, \quad x \in [-1,1],\ t \in [0,1]\,,\\
&u(x,0) = x^2 cos(\pi x)\,,\\
&u(t,-1) = u(t,1)\,,\\
&u_x(t,-1) = u_x(t,1)\,,
\end{aligned}
$$

Solution:





Physics-Informed Neural Network (PINN)



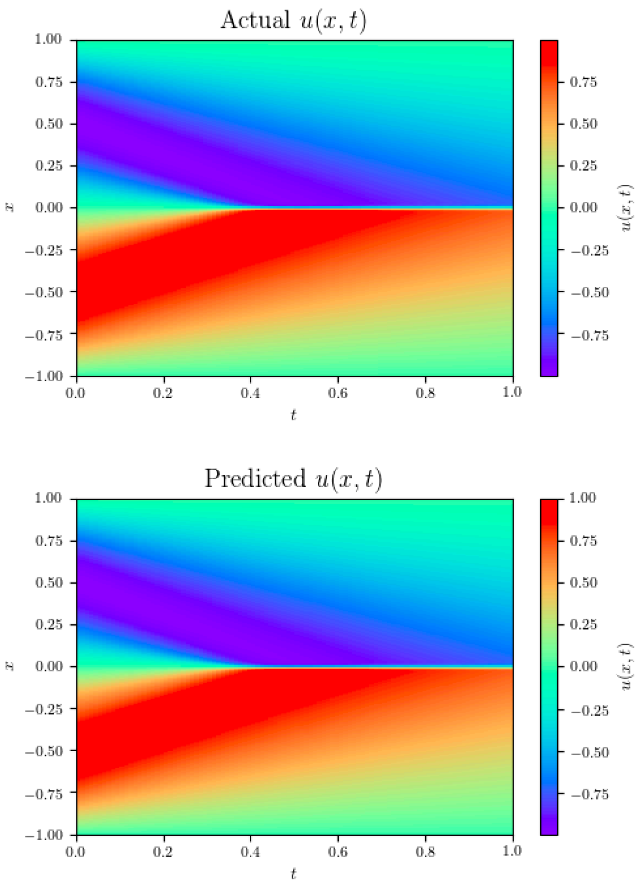
PINNs, introduced in 2017, is the most well-known SciML Algorithm.

Using neural networks to solve PDEs had been known already in the 1990’s.

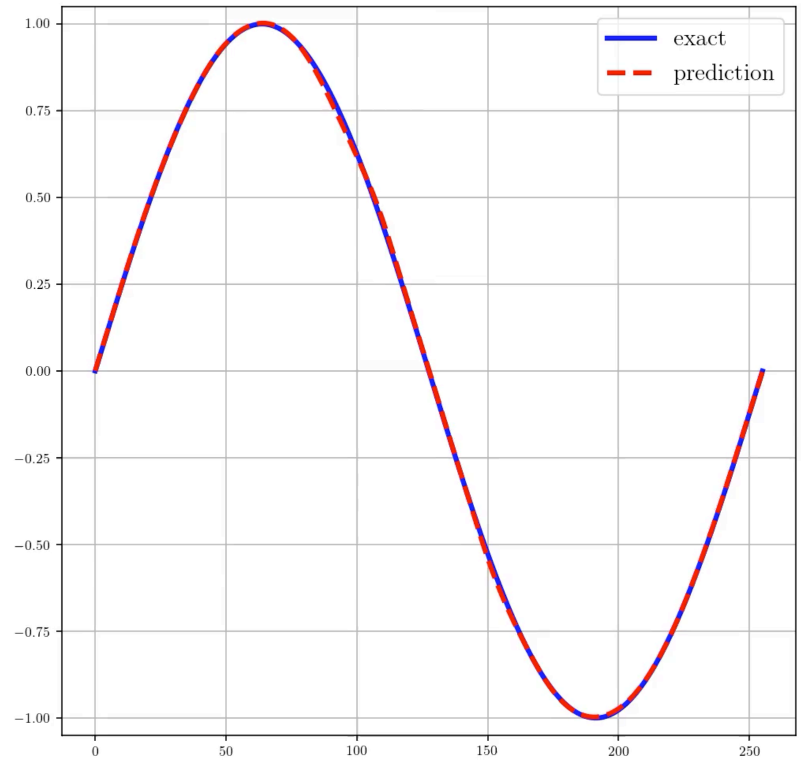
(Diagram from Zhang, Braga-Neto and Gildin, "Physics-Informed Neural Networks for Multiphase Flow in Porous Media Considering Dual Shocks and Interphase Solubility’’, *Energy and Fuels*, 2024.)

Example: Viscous Burger's Equation

Representative PINN solution for $\nu = 0.01/\pi$.
Hyperparameters: 8 layers with 20 neurons each, 200 initial points, 100 boundary points, 20,000 collocation points, 200 Adam + 200 LBFGS iterations.



L_2 error = 1.09 %



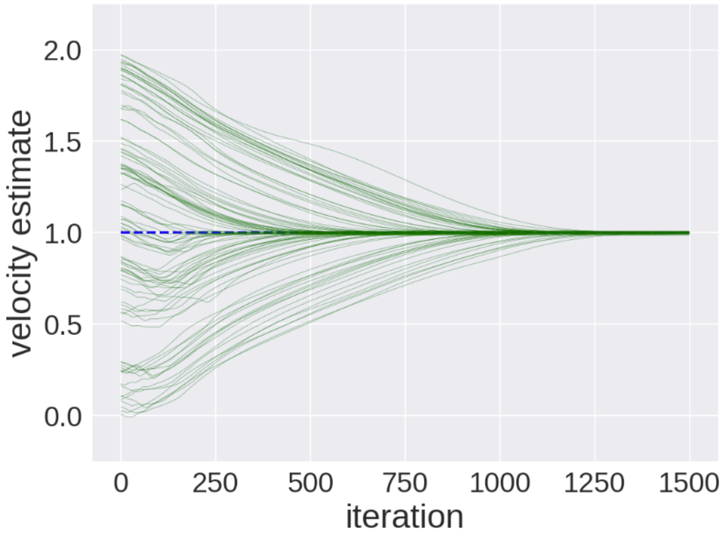
Example (Inverse Problem): Velocity Estimation from Data

Use experimental data to discover the value of unknown quantities (e.g., parameters).

Example: 1D linear advection PDE with unknown velocity (here, $v = 1$):

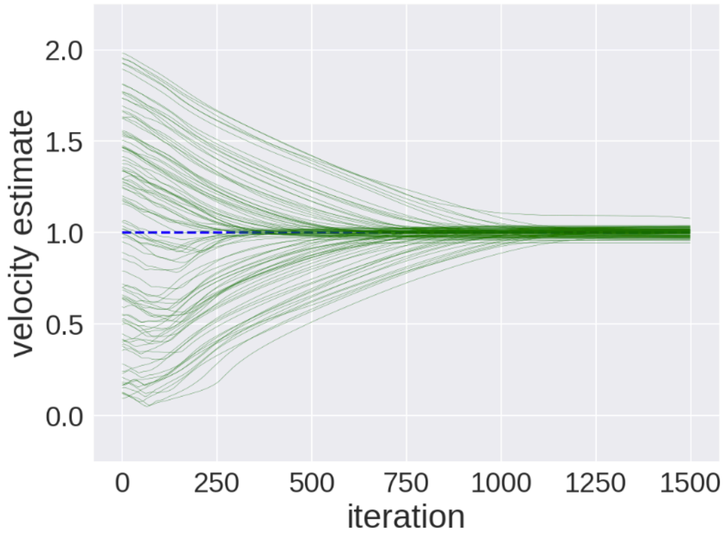
$$\frac{\partial u(x,t)}{\partial t} + v\,\frac{\partial u(x,t)}{\partial x} = 0$$

PINN, low noise



Velocity estimate: 0.99949 ± 0.00421

PINN, high noise

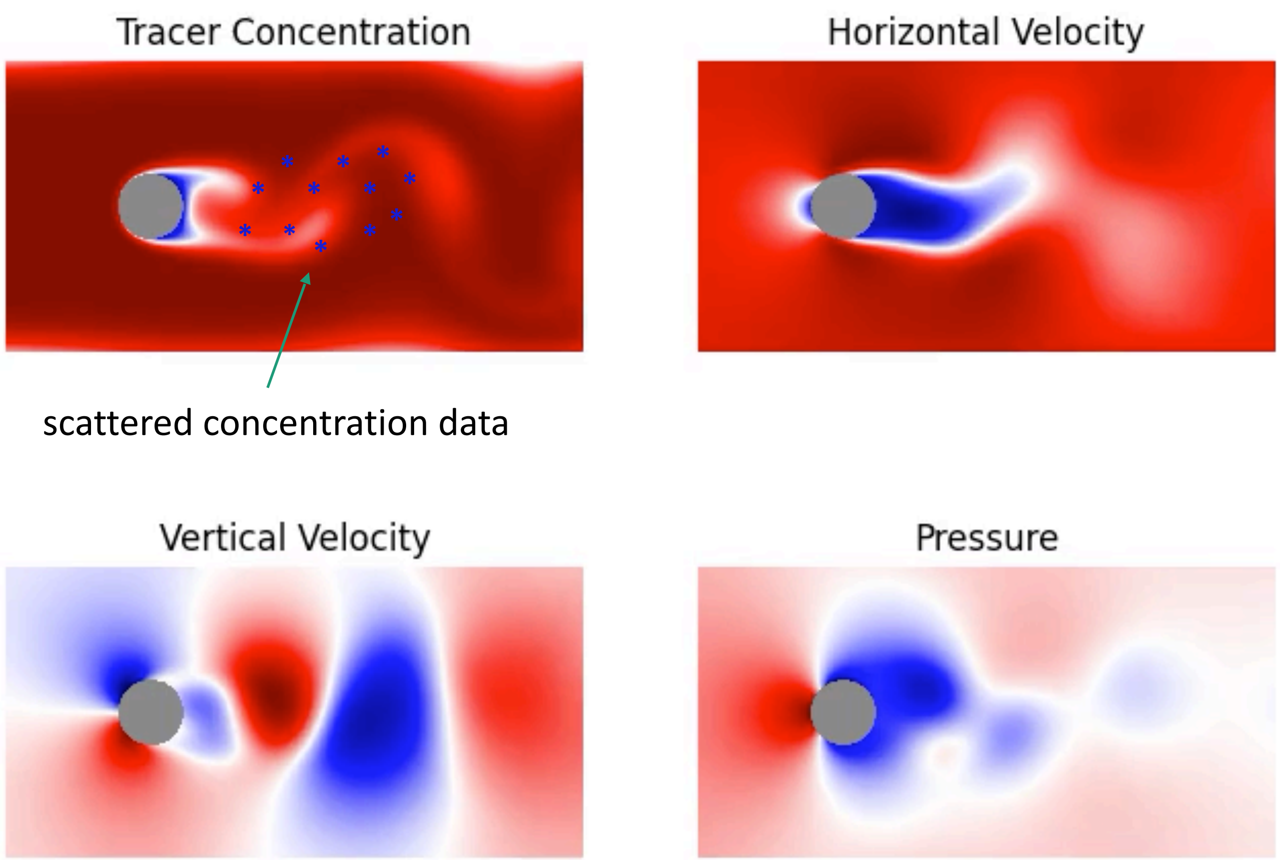


Velocity estimate: 1.00170 ± 0.02117

Example: 2D Flow Around a Cylinder

Navier-Stokes Equations $$\mathbf{u}_t + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla p + R^{-1}\Delta \mathbf{u}$$
$$\operatorname{div} \mathbf{u} = 0$$

Advection of scalar $$c_t + \mathbf{u} \cdot \nabla c = \mathrm{Pec}^{-1}\Delta c$$



Example: Multiphase Flow in Porous Media

Buckley-Leverett: classic nonlinear hyperbolic transport equation in Petroleum Engineering.

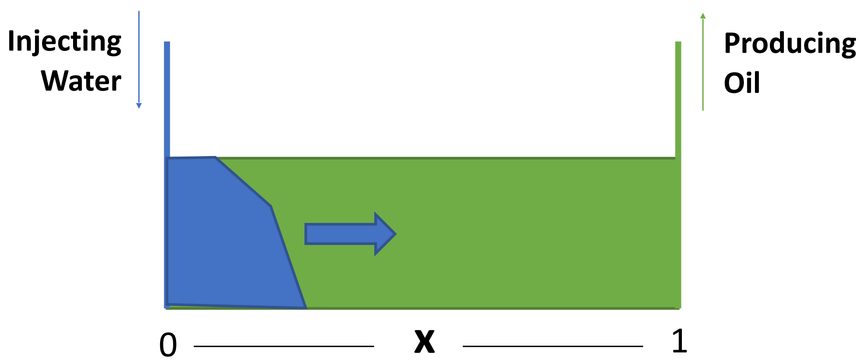
It models the immiscible displacement of one fluid by another in porous media.

For example, oil by water or brine by CO_2.

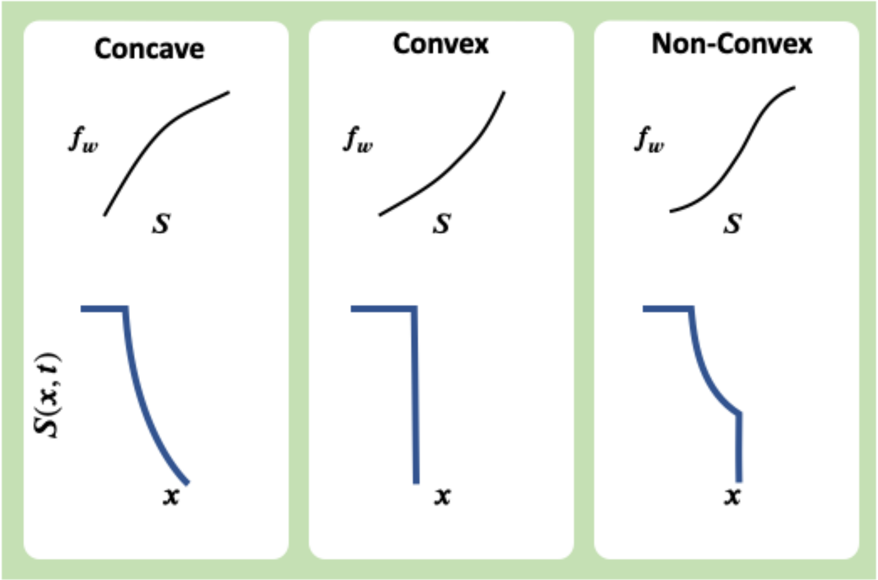
$$\frac{\partial S}{\partial t} + \frac{\partial f_w(S)}{\partial x} = 0$$

$$\text{IC}: S(x,t) = 0, \quad x > 0,\ t = 0$$
$$\text{BC}: S(x,t) = 1, \quad x = 0,\ t > 0$$

$S(x,t)$ = water saturation
$f_w(S)$ = fractional flow of water

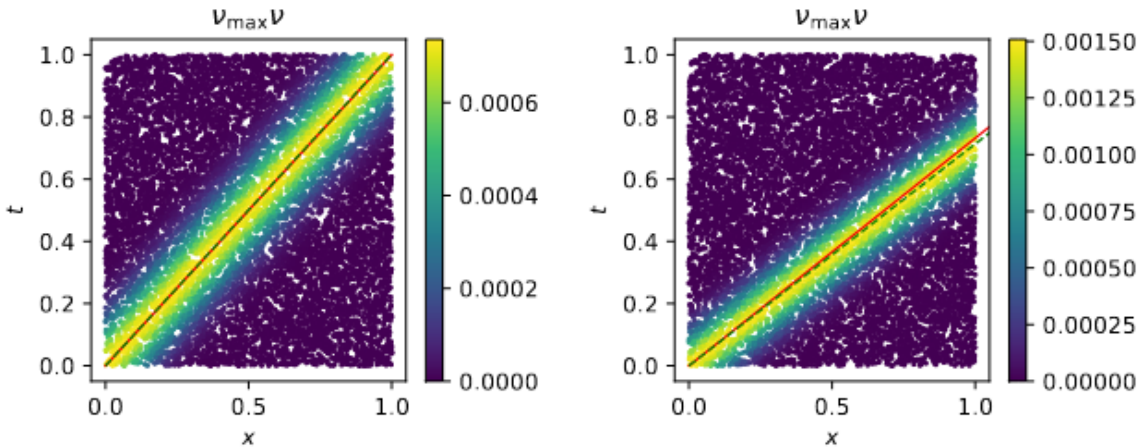


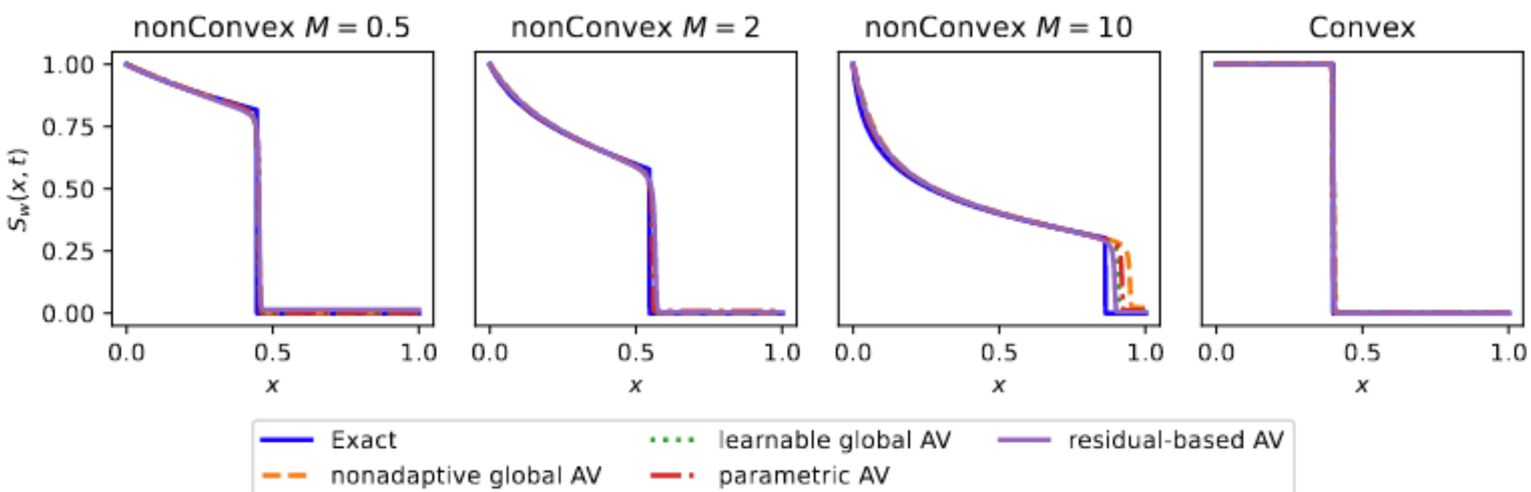
The shape of $f_w(S)$, which depends on rock and fluid properties, determines the behavior of the solution.



PINN with Adaptive Localized Viscosity

The location to add the viscosity can be learned by the neural network.

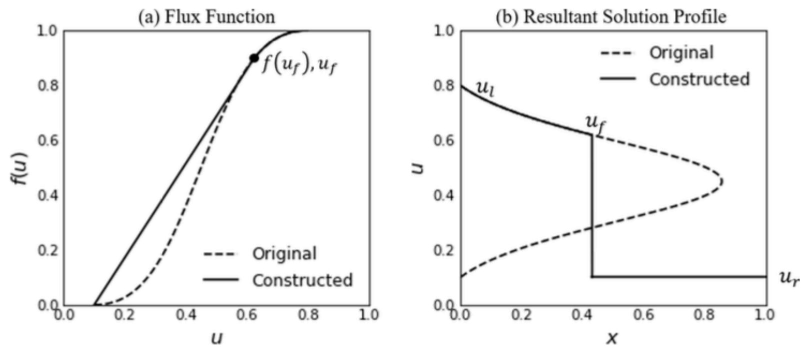


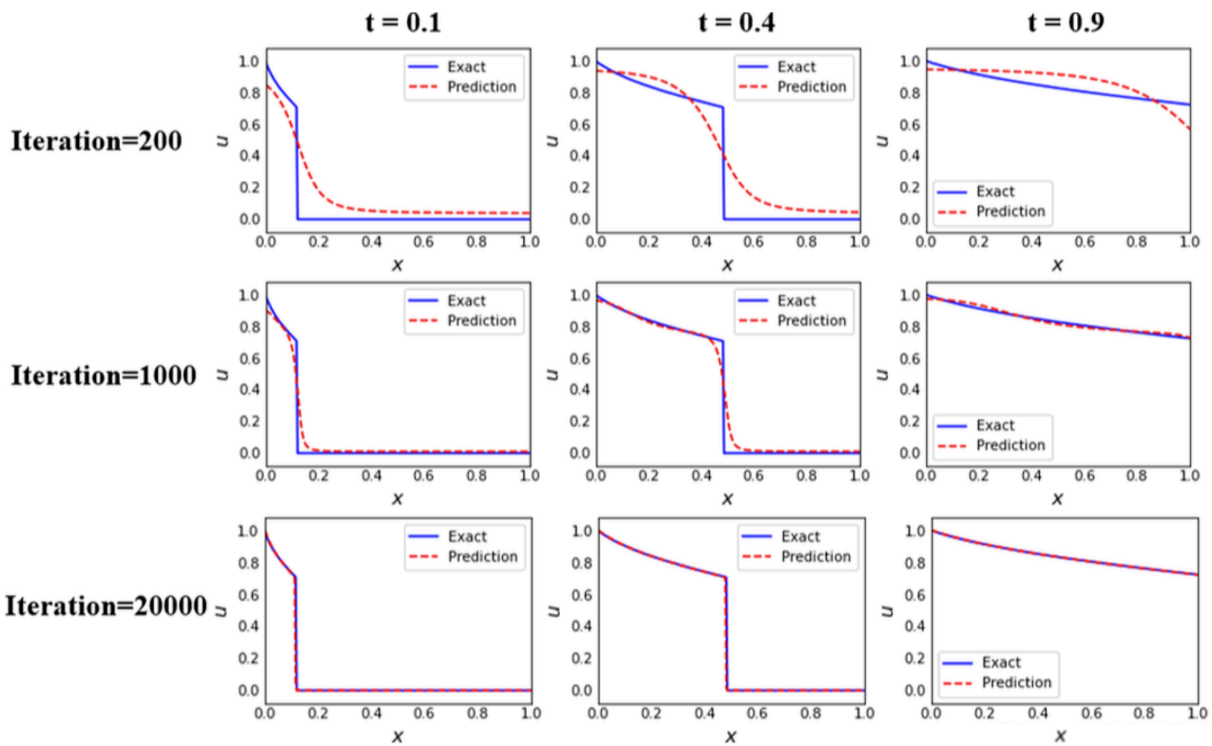


(From Coutinho, Braga-Neto, Gildin, et al. "Physics-informed neural networks with adaptive localized artificial viscosity'', *Journal of Computational Physics*, 2024.)

PINN with Welge's Construction

Regularization of the nonconvex fractional flow to avoid unphysical solutions.





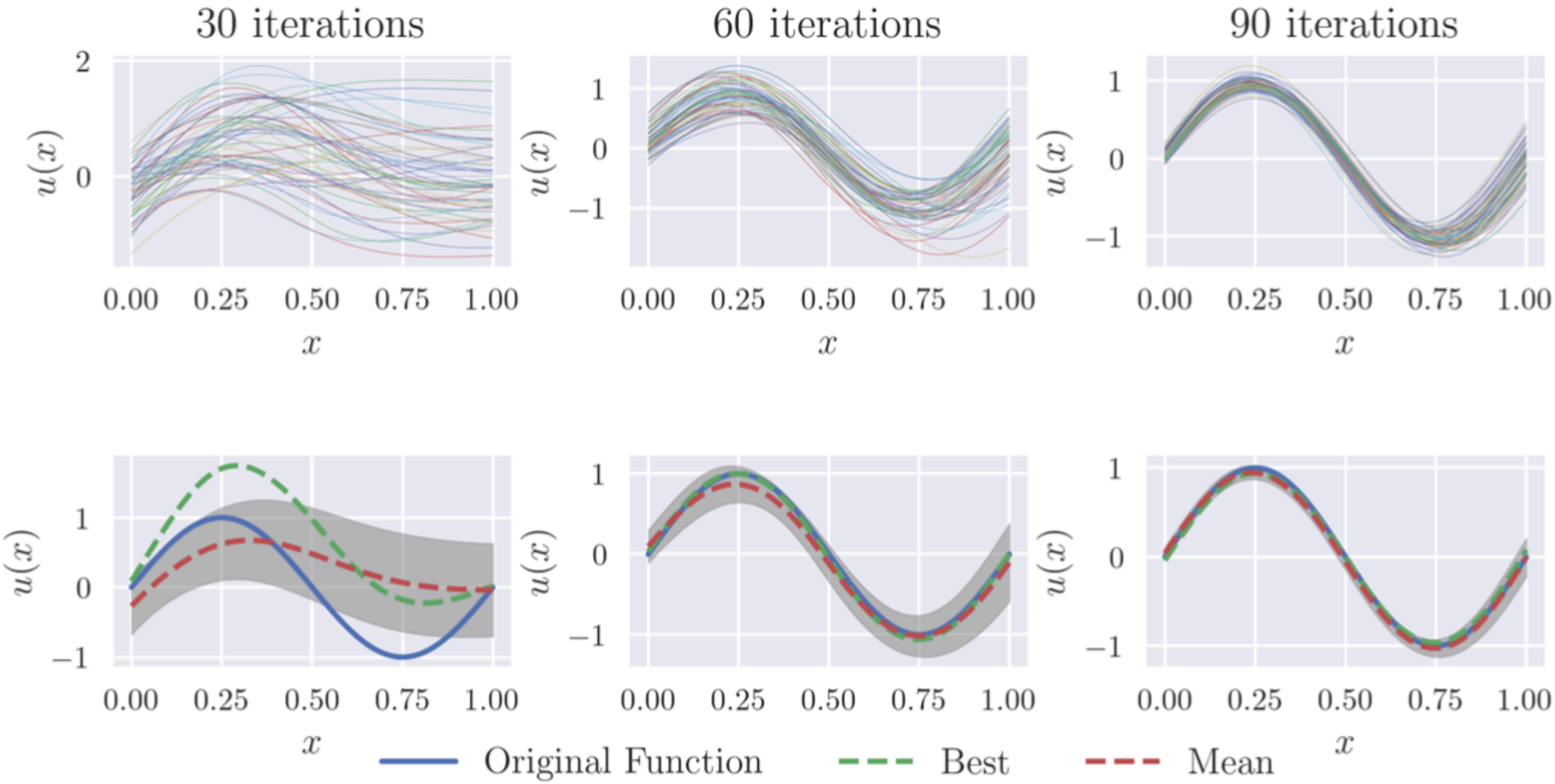
(From Zhang, Braga-Neto and Gildin, "Physics-Informed Neural Networks for Multiphase Flow in Porous Media Considering Dual Shocks and Interphase Solubility'', *Energy and Fuels*, 2024.)

PSO-PINN: Using Particle Swarms to Train the PINN

This is an **ensemble approach**. For example, consider the Poisson Equation:

$$u_{xx} = g(x)\,, \quad x \in [0,1]\,,$$
$$u(0) = u(1) = 0\,.$$

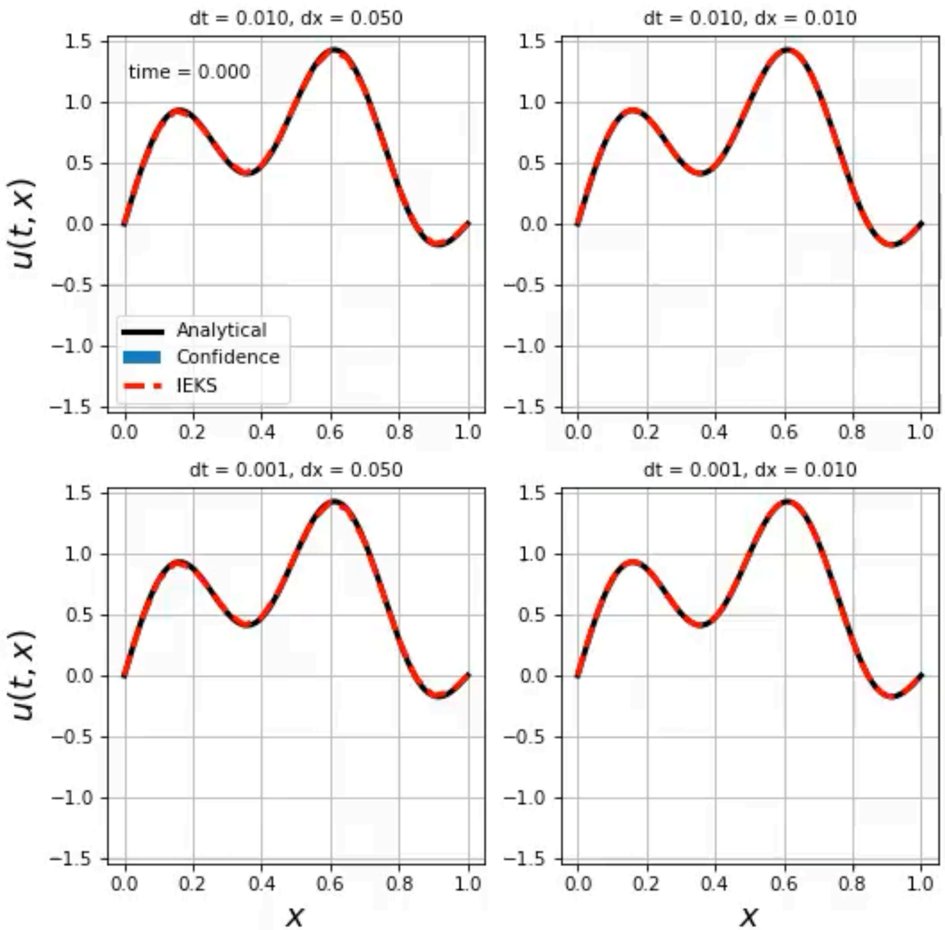
where $g(x) = -(2\pi)^2 \sin(2\pi x)$ is manufactured so that $u(x) = \sin(2\pi x)$.



(From Davi and Braga-Neto, "PSO-PINN: Physics-informed neural networks trained with particle swarm optimization", 2022.)

Physics-Informed Probabilistic Methods

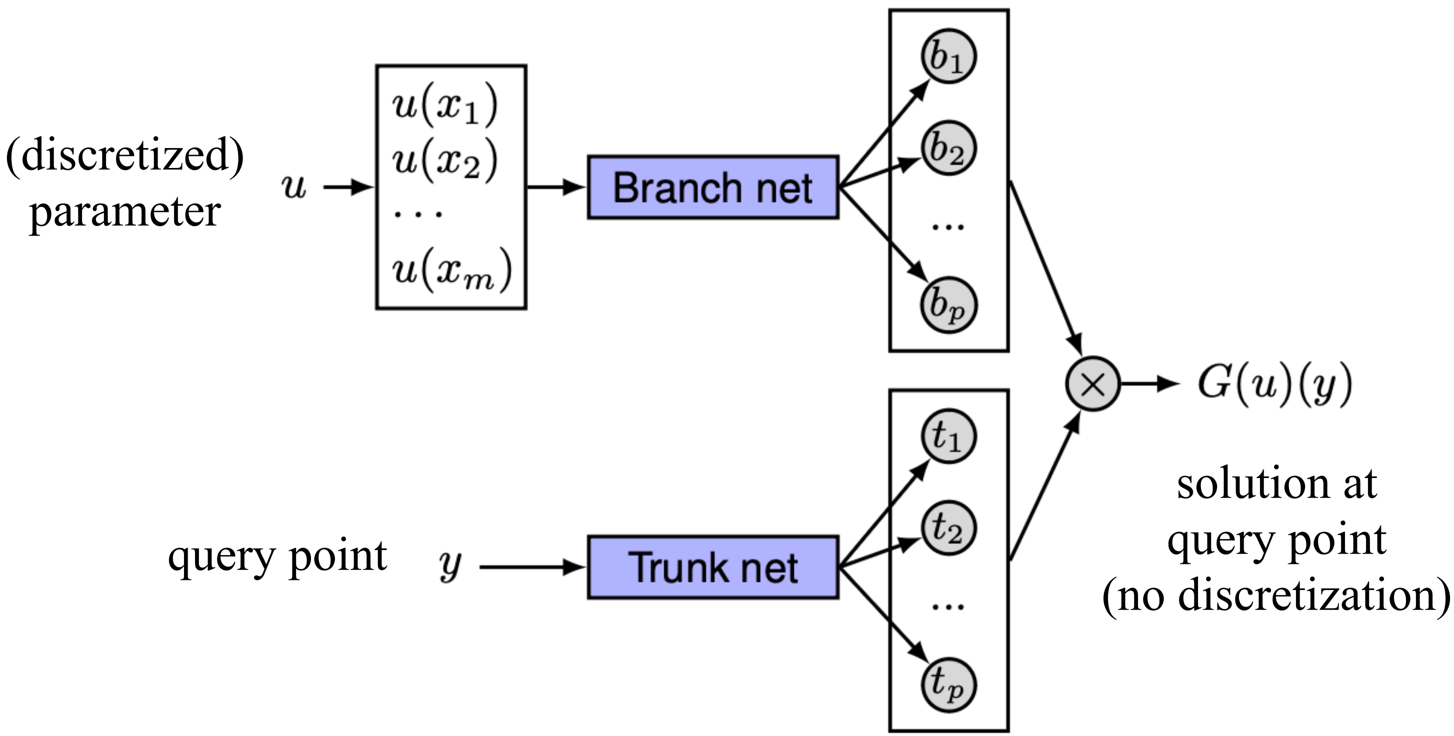
Bayesian inference with Gaussian processes provide a nonparametric approach to quantify prediction uncertainty through the variance of the posterior distribution.
Example: Wave Equation.



(D. Cator, S. Särkkä, and U. Braga-Neto, https://sciml.tamids.tamu.edu/demos/)

Operator Learning for Parametric PDEs

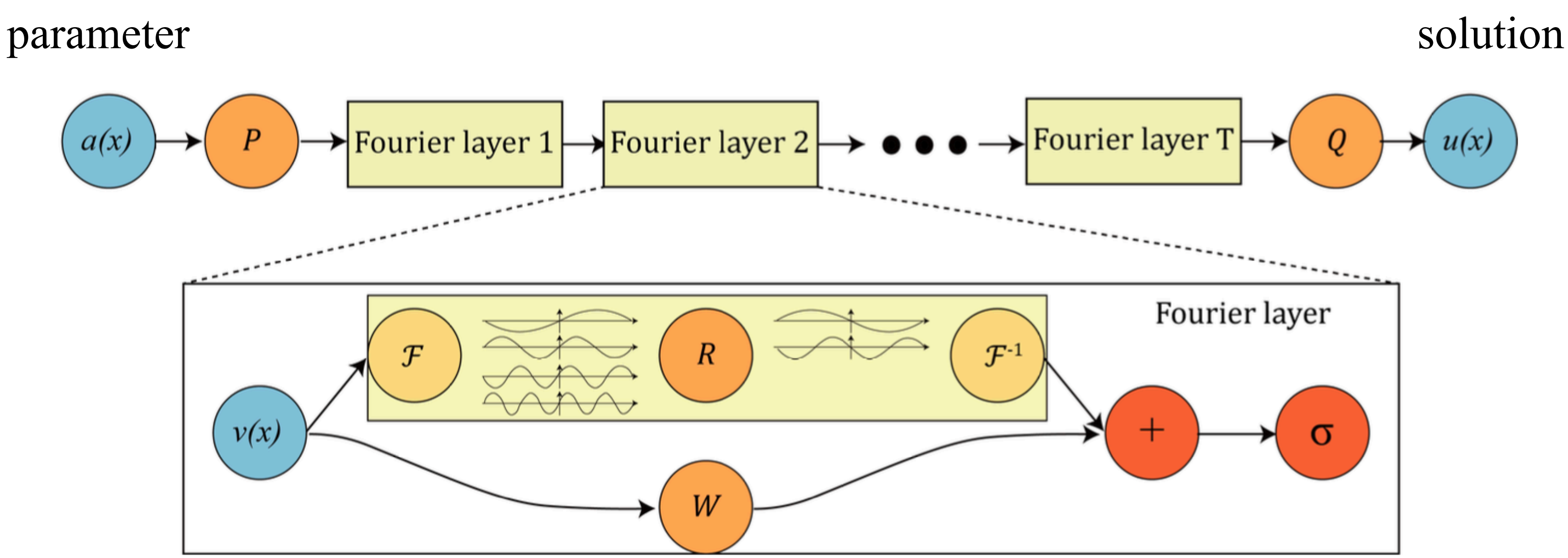
- Neural architectures to learn mappings between **function spaces**.
- As an example, this allows one to map the parameter of a PDE to the solution.
- If the parameter changes, the neural network does not need to be retrained.
- A popular example of neural operator is the Deep Operator Networks (**DeepONets**):



(From Lu et al., "Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators", *Nature Machine Intelligence*, 2021.)

Operator Learning for Parametric PDEs

- Another popular example is the Fourier Neural Operator.
- It is an infinite-dimensional version of a neural network, which cascades linear integral operators and nonlinearities.
- In the FNO, the linear integral operators are discretized and computed in the frequency domain using the discrete Fourier transform (DFT).



(From Li et al., "Fourier Neural Operator for Parametric Partial Differential Equations," ArXiV, 2020.)

TAMIDS Scientific Machine Learning Lab

- The SciML lab was established in Jan 2021 to pilot the **Thematic Data Science Labs** program of TAMIDS.
- The SciML Lab **research mission** is to support and grow a multidisciplinary community of researchers across Texas A&M involved in the development of SciML algorithmic, computational, and applied components.
- The SciML also aims to be a catalyst for accelerating **education** in SciML, through short hands-on courses, seminars, workshops, and case studies.

SciML Lab Website:

https://sciml.tamids.tamu.edu