Modeling and Simulation of Ocean Phenomena Using Physics-Based Machine Learning

Hugo Carrillo Lincopi

Inria Chile Research Center

May 18, 2023

Joint work with:
Emir Chacra (Inria Chile & CMM), Julien Salomon (Inria & Sorbonne Université),
Luis Martí (Inria Chile), Nayat Sánchez-Pi (Inria Chile)

Contents

1. Introduction
   - Mathematical modeling and ML in Oceans and Climate
   - Methods in Physics-Based Machine Learning
2. Physics-Informed ML (I): PINNs
   - State-of-the-art
   - PINNs (I): Optimal hyperparameters search
   - PINNs (II): Applications to ocean models
3. Physics-Based ML (II): Reduced order modeling (ROMs) for PDEs
4. Conclusions, perspectives, challenges

Oceans and Climate: generalities

Many phenomena

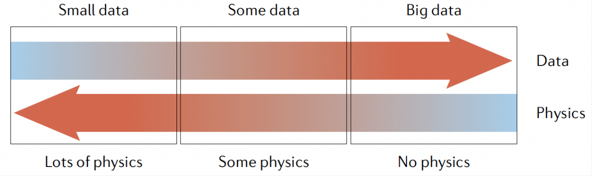
- Ocean circulation.
- Transport of nutrients.
- Plankton dynamics.
- Carbon pump.
- etc.

If we want to predict or simulate, we need:

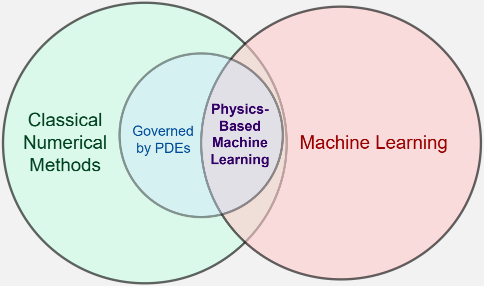
- Mathematical modeling.
- Numerical methods.

Numerical methods. Approaches

Data regimes[1]: Physics-Based vs. Data-Driven approaches.

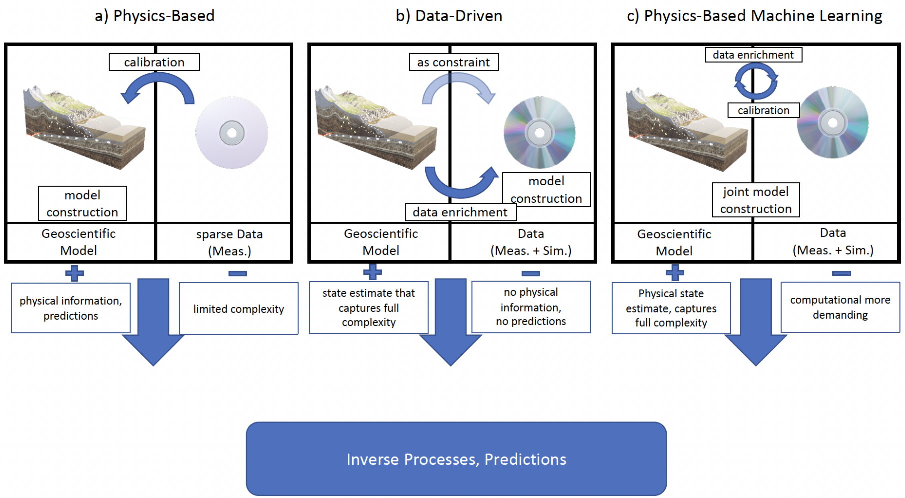


Classical numerical methods vs. Machine Learning (ML) methods:



[1] Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., and Yang, L. (2021). Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440

Physics-Based vs. Data-Driven[2]



[2] Degen, D., Caviedes Voullième, D., Buiter, S., Hendriks Franssen, H.-J., Vereecken, H., González-Nicolás, A., and Wellmann, F. (2023). Perspectives of physics-based machine learning for geoscientific applications governed by partial differential equations. *Geoscientific Model Development Discussions*, 2023:1–50

Physics-Based Machine Learning: Advantages.

Over classical num. methods:

- Reduces computational cost.
- Increases flexibility.
- New patterns.

Over classical ML:

- Incorporates physical laws.
- Reduces data requirements.
- Improves interpretability.
- New phenomena.

Physics-Based Machine Learning: Disadvantages/Challenges.

Over classical num. methods:

- Increases complexity.
- In some cases it could limit applicability.

Over classical ML:

- Increases complexity.
- Increases computational cost.
- Limited flexibility.

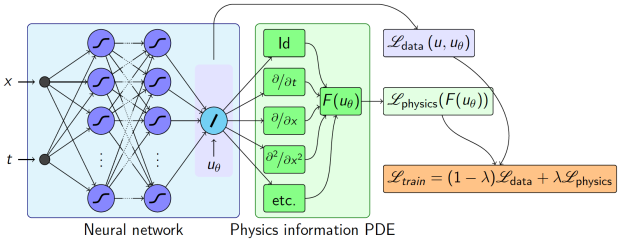
Methods in Physics-Based Machine Learning

- Gaussian processes with physical constraints (GP-PC).
- Regression trees.
- Differential equation neural networks (DENN).
- Physics-informed neural networks (PINNs).
- Hybrid approaches.

Physics-Informed ML (I): PINNs[3]

PDE (residual form):

$F(u) = 0$ in Q, with knowledge of $B(u)$ on a subset of $\overline{Q}$.



Problem:

Find $\theta = (W, b)$ the set of weights and biases that minimizes the loss function

$$\mathscr{L}_{train} = (1-\lambda)\mathscr{L}_{\text{data}} + \lambda\mathscr{L}_{\text{physics}}\,, \tag{1}$$

where

$$\mathscr{L}_{\text{data}} = \frac{1}{N_u}\sum_{i=1}^{N_u}\left|B(u)(x_i^u, t_i^u) - B(u_\theta)(x_i^u, t_i^u)\right|^2, \quad \mathscr{L}_{\text{physics}} = \frac{1}{N_f}\sum_{j=1}^{N_f}\left|F(u_\theta)(x_j^f, t_j^f)\right|^2. \tag{2}$$

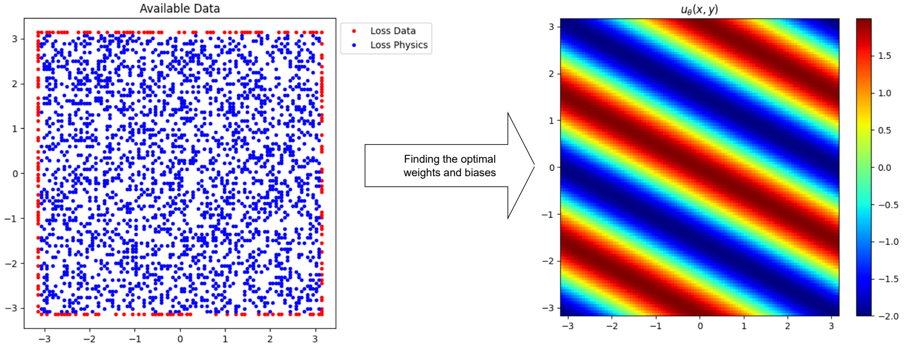
[3] Raissi, M., Perdikaris, P., and Karniadakis, G. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707

Example: Poisson equation (I).

Direct problem

Given $\kappa(x,y)$ and $f(x,y)$ in the interior domain, and $u(x,y)$ on the boundary: Determine $u(x,y)$ if

$$\nabla \cdot (\kappa \nabla u) = f$$

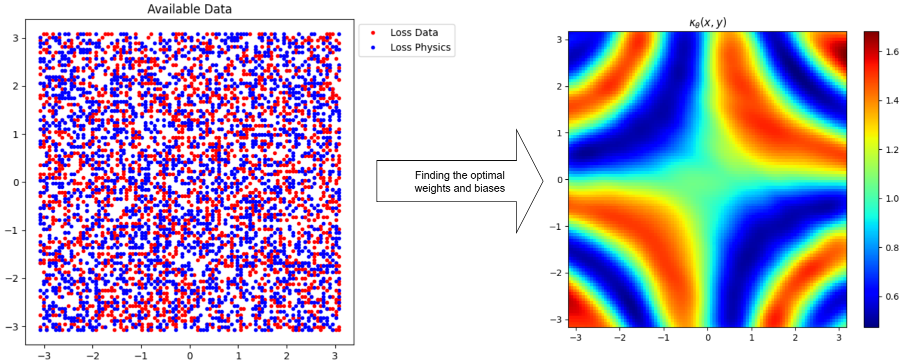


Example: Poisson equation (II).

Inverse problem

Given $f(x, y)$ in the interior domain and $u(x, y)$ in a sampling set:
Determine $\kappa(x, y)$ if

$$\nabla \cdot (\kappa \nabla u) = f$$



Advantages

- It is a meshless method.
- It faces the curse of dimensionality.

Challenges

- Despite the experimental results, it is necessary to establish in theory that PINNs can accurately simulate/recover the parameters.
- To obtain methods that make the optimization process efficient and accurate (architecture, hyperparameters, etc.)

Some strategies:

- Hypothesis set
  - Hyperparameters (number of neurons and layers, activation functions): manual, gaussian process, evolutionary search.
  - Architectures: FNN, CNN, autoencoders, etc.
- Sampling: grid, hyper-Latin cube, adaptive, evolutionary, etc.
- Optimization strategy
  - Objective function.
    - PDE sense: classical or distributional sense, lagrangian or eulerian, etc.
    - Trade-off for PDE and data loss:
      - Manual, evolutionary, causal, etc.
      - Hard and soft constraints.
  - Gradient descent:
    - SGD, Adam, L-BFGS, etc.
    - Learning rate: constant, scheduled, adaptive, etc.

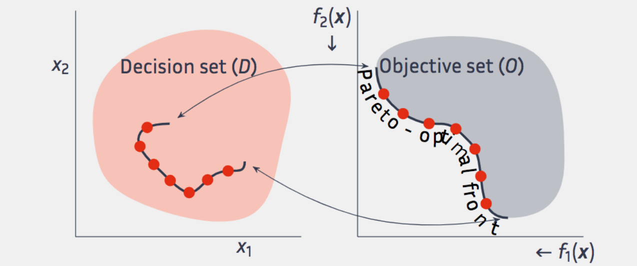
PINNs (I): Optimal hyperparameters search[4] [5]

PINNs as a multi-objective optimization problem

- We seek a solution to the problem

$$\min_{\theta,\boldsymbol{a}} \left(\mathscr{L}_{\text{physics}}(\theta),\ \mathscr{L}_{\text{data}}(\theta)\right). \tag{3}$$

- **Method: Evolutionary algorithms** to obtain an optimal $\boldsymbol{a}$ and a set of λ's such that estimate the Pareto optimal front.



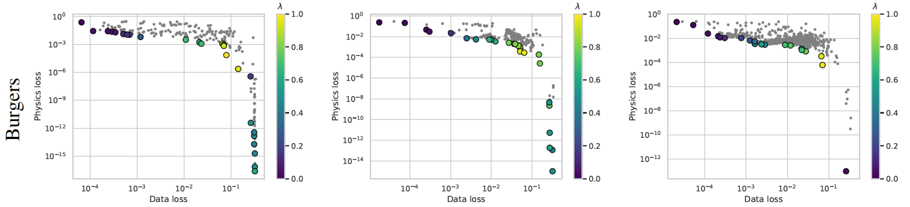
[4] de Wolff, T., Lincopi, H. C., Martí, L., and Sanchez-Pi, N. (2022). Mopinns: an evolutionary multi-objective approach to physics-informed neural networks. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 228–231

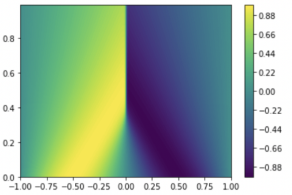
[5] de Wolff, T., Carrillo, H., Martí, L., and Sanchez-Pi, N. (2023). Optimal architecture discovery for physics-informed neural networks. In *Advances in Artificial Intelligence–IBERAMIA 2022: 17th Ibero-American Conference on AI, Cartagena de Indias, Colombia, November 23–25, 2022, Proceedings*, pages 77–88. Springer

Burgers equation

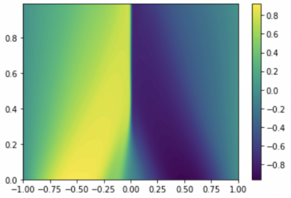
$$\begin{cases} \dfrac{\partial u}{\partial t} + u\dfrac{\partial u}{\partial x} = \nu \dfrac{\partial^2 u}{\partial x^2} & (x,t) \in (0,L)\times(0,T), \\ u(1,t) = u(-1,t) = 0 & t \in (0,1), \\ u(x,0) = -\sin(\pi x) & x \in (-1,1), \end{cases}$$

NSGA-II NSGA-III MOEA/D





(a) Burgers target (exact) solution.



(b) Solution predicted by MOPINNs.

$\boldsymbol{a} = (40, 10, 10)$
$\lambda = 0.048745.$

PINNs (II): Application of PINNs to ocean models

The NPZ model

$$\begin{aligned}
\partial_t N + \boldsymbol{u} \cdot \nabla N &= -f(I)g(N)P + (1-\gamma)h(P)Z + i(P)P + j(Z)Z \\
\partial_t P + \boldsymbol{u} \cdot \nabla P &= f(I)g(N)P - h(P)Z - i(P)P \\
\partial_t Z + \boldsymbol{u} \cdot \nabla Z &= \gamma h(P)Z - j(Z)Z \\
&\text{+Initial conditions}
\end{aligned}$$

Here, u is the velocity of a fluid, which couples the NPZ model to, for instance, Navier-Stokes.

Navier-Stokes ($w = \nabla \times u$)

$$\begin{aligned}
\partial_t \boldsymbol{w} + \boldsymbol{u} \cdot \nabla \boldsymbol{w} &= \frac{1}{\mathrm{Re}} \Delta \boldsymbol{w} \\
\nabla \cdot \boldsymbol{u} &= 0 \\
&\text{+Initial condition}
\end{aligned}$$

Work in progress: making simulations

- PINNs with several loss functions.
- Hyperbolic nature of the model adds difficulties in training $\longrightarrow$ causal PINNs.

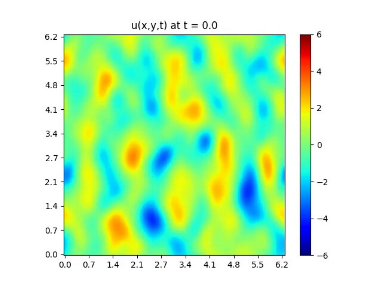
PINNs strategy: Causal PINNs [6]

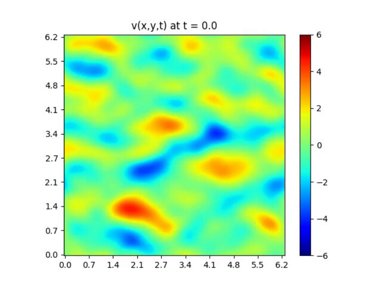
- PINNs can violate physical causality.
- Re-formulation:

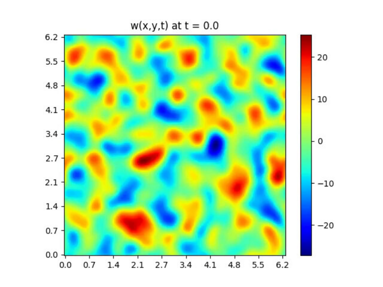
$$\mathscr{L}_r(\theta) = \frac{1}{N_t}\sum_{i=1}^{N_t} w_i \mathscr{L}_r(t_i,\theta), \quad \text{where} \quad w_i = \exp\left(-\varepsilon \sum_{k=1}^{i-1} \mathscr{L}_r(t_k,\theta)\right)$$

[6] Wang, S., Sankaran, S., and Perdikaris, P. (2022). Respecting causality is all you need for training physics-informed neural networks

Navier-Stokes







Physics-Based ML (II): Reduced order modeling (ROMs) for PDEs[7]

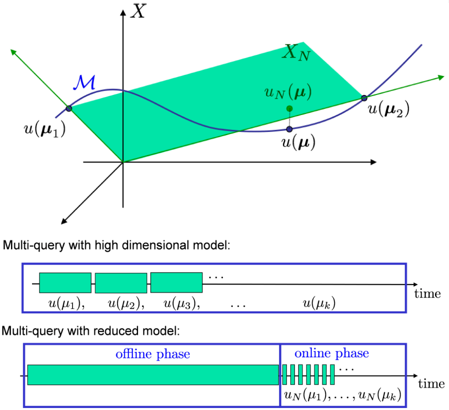
$$\partial_t u_\mu + P_\mu[u_\mu] = f_\mu \quad \text{in } \Omega \times (0, T),$$
$$+BC_\mu + IC_\mu.$$

Problem:

Given an unknown state μ, and measurements (snapshots) for different μ_i's, estimate the solution manifold $\mathscr{M} = \{u(\mu) | \; \mu \in \mathscr{P}\}$.

Method (work in progress):

Reduced Basis method (RB) mixed with NNs.



Application (work in progress)

Optimal location of measurements for inverse problems.

[7] Hesthaven, J. S. and Ubbiali, S. (2018). Non-intrusive reduced order modeling of nonlinear problems using neural networks. *Journal of Computational Physics*, 363:55–78

Conclusions, perspectives, challenges

Application of PBML methods

- Integration to Pisces?
- What method will work better for ocean numerical simulation?
- Could this technique be useful for modeling challenges?
- Multiscale phenomena?
- Multi-physics phenomena
- To determine which parameters influence the state distribution
- Assessment of uncertainty.
- Predictions and real-time applications.

Thanks for your attention!