Inria

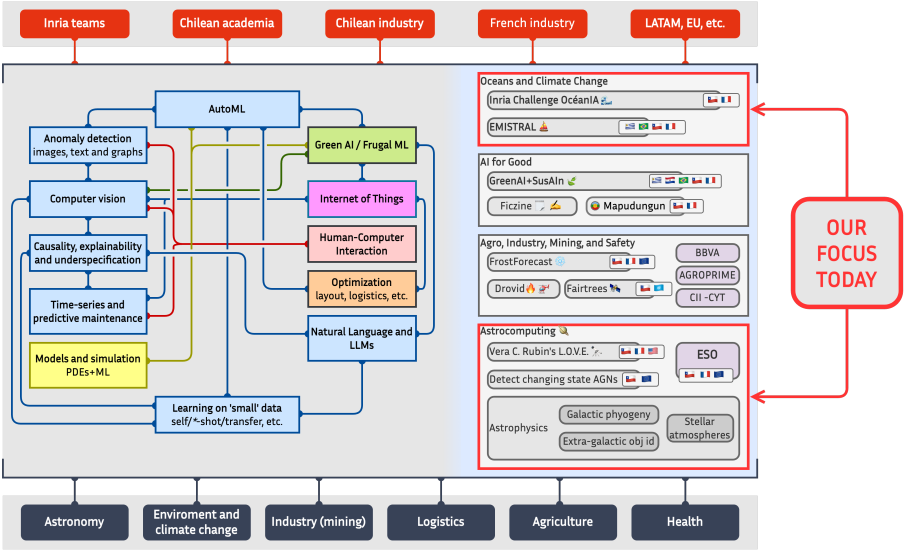
MOPINNs: A Multi-Objective AutoML Approach to PINNs

Understanding the Universe One Epoch at a Time

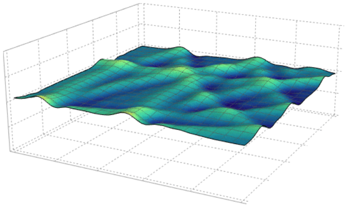
Inria Chile

OcéanIA Annual Meeting
23 February 2024

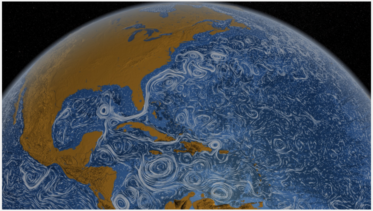
Inria Chile: Science and projects



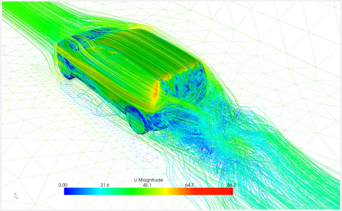
Complex dynamical systems: understanding nature



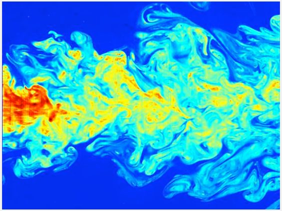
Source: https://soulofmathematics.com/index.php/differential-equations/



Source: NASA.



Source: https://secretofflight.wordpress.com/turbulence/

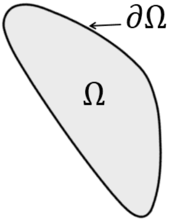


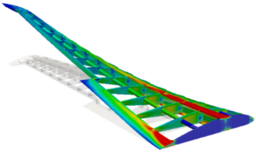
Source: C. Fukushima and J. Westerweel, Technical University of Delft

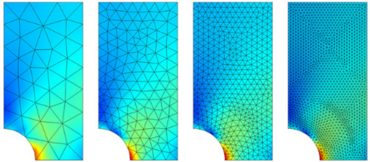
The problem: PDEs are hard!

Finite-element methods

- Mesh based
- Curse of dimensionality
- Parameter change requires reevaluation







Neural networks (or ML) for learning PDEs

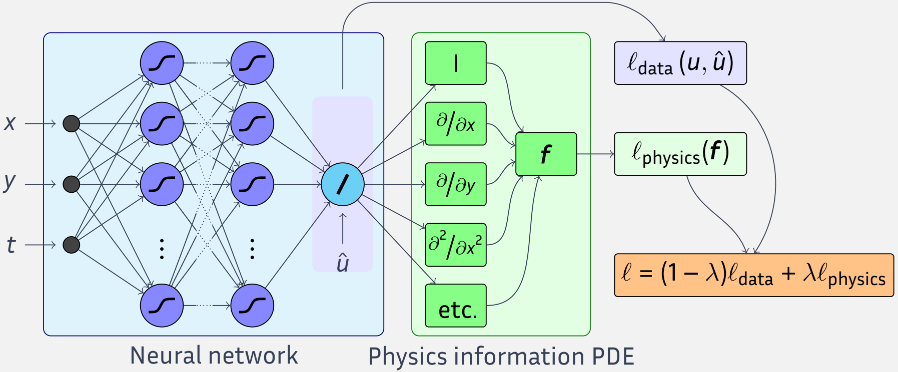
Training a neural network to the solution

- Differentiable everywhere
- Mesh-free approach
- Parameter agnostic
- Requires a lot of data and ignores domain expert knowledge.

Physics-Informed Neural Networks (PINNs)

The physics-informed paradigm

Incorporate deviation from physics law
in the loss function with a λ parameter to weigh the influence of the physics.



MOPINNs: Evolutionary multi-objective PINNs[1]

Problem in 'regular' PINNs

$\lambda \in [0, 1]$ is overloaded with different functions:

- It is meant to express a **preference** between physics and data, but
- different physical scales between losses → differences in magnitudes,
- different numerical characteristics of losses, such as convergence rates.

Our proposal: MOPINNs

- Introduce a multi-objective formulation $\min_\theta(\ell_{\text{data}}, \ell_{\text{physics}})$
- Evolutionary AutoML to automatically find the best network architecture

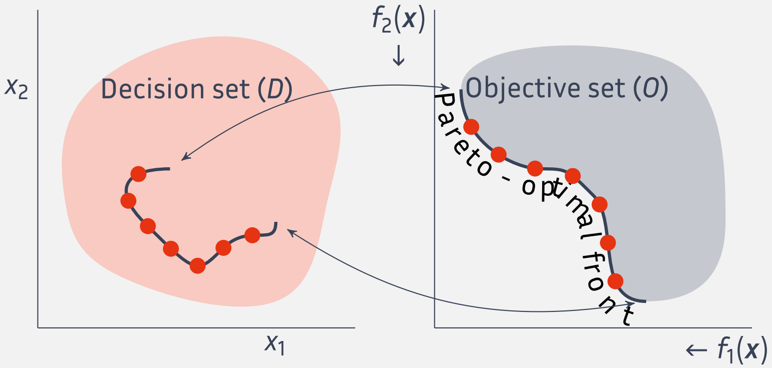
[1] de Wolff, T., Carrillo, H., Martí, L., & Sanchez-Pi, N. (2022). Optimal architecture discovery for physics-informed neural networks. In A. C. Bicharra Garcia, M. Ferro, & J. C. Rodríguez Ribón (Eds.), *Advances in Artificial Intelligence – IBERAMIA 2022* (pp. 77–88). Cham: Springer International Publishing

Multi-Objective Optimization

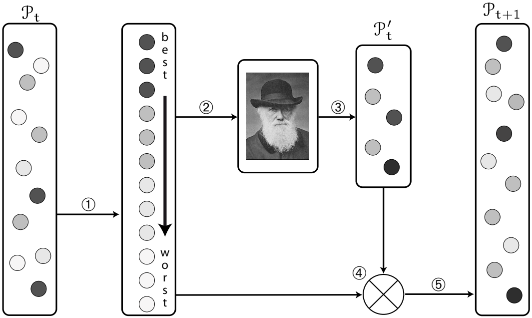
$$\text{minimize } \boldsymbol{F}(\boldsymbol{x}) = \langle f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots f_m(\boldsymbol{x})\rangle \;, \text{ with } \boldsymbol{x} \in D \subseteq \mathbb{R}^n \,. \tag{1}$$

Solutions $\boldsymbol{x}^* \in \arg\min_{\boldsymbol{x}\in D} \boldsymbol{F}(\boldsymbol{x})$ are in the *Pareto-optimal set* such that:

$$f_i(\boldsymbol{x}^*) \leq f_i(\boldsymbol{x}),\; \forall\, i \in \{1, \ldots, m\}\,; \boldsymbol{x} \in D\,. \tag{2}$$



Evolutionary multi-objective optimization[2]



Evolutionary algorithms are inspired by the notion of *survival of the fittest* from Darwinian Evolution and modern genetics.

- **Advantages:** inherent parallel search, and lower susceptibility to the shape or continuity of the image of the efficient set
- **Selected algorithms:**
  - Non-dominated sorting genetic algorithm (NSGA-II)
  - Reference-point-based selection NSGA (NSGA-III)
  - Multi-objective evolutionary algorithm by decomposition (MOEA/D)

---

[2] Coello Coello, C., Lamont, G., & van Veldhuizen, D. (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems*. Genetic and Evolutionary Computation. New York: Springer, second edition

The (current) MOPINNs Proposal

Use an EMO algo. to search for individuals that minimize

$$\ell_{\text{data}}(\theta) = \frac{1}{N_u} \sum_{i=1}^{N_u} \left| B(u)\left(x_i^u, t_i^u\right) - B(u_\theta)\left(x_i^u, t_i^u\right) \right|^2 \text{ and} \tag{3}$$

$$\ell_{\text{physics}}(\theta) = \frac{1}{N_f} \sum_{j=1}^{N_f} \left| F(u_\theta)(x_j^f, t_j^f) \right|^2, \tag{4}$$

where individuals express of the network parameters:

- network activation function,
- number of neurons for each layer, and
- λ, the relative trade-off between data and physics losses.

For each individual, train a physics-informed neural network by minimizing

$$\ell(\theta, \lambda) = (1 - \lambda)\ell_{\text{data}}(\theta) + \lambda \ell_{\text{physics}}(\theta)\,. \tag{5}$$

Experiments

Burgers equation:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2}\,,$$
$$u(x,0) = -\sin(\pi x)\,,$$
$$u(1,t) = u(-1,t) = 0\,.$$

Wave equation:

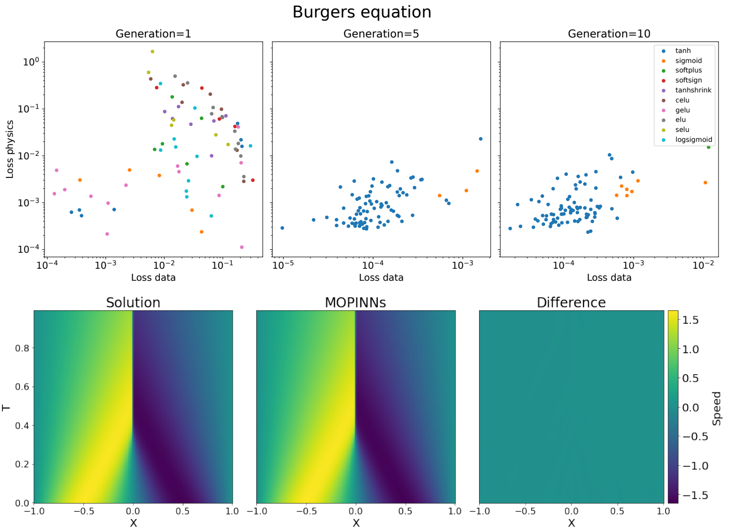
$$\frac{\partial^2 \eta}{\partial t^2} = \nabla\cdot(H\nabla\eta)\,,$$
$$\eta(x,y,0) = e^{-10\left((x-0.5)^2+(y-0.75)^2\right)}\,,$$
$$\frac{\partial \eta}{\partial t}(x,y,0) = 0\,.$$

Training

- **Multi-objective algorithm:** MOEA/D with 10 generations and a population size of 25.
- **Activation functions:** LeakyReLU, ReLU, Tanh, Sigmoid, Softplus, Softsign, TanhShrink, CELU, GELU, ELU, SELU, and LogSigmoid.
- **Architectures:** three hidden layers of up to 100 neurons per layer, in decreasing order.

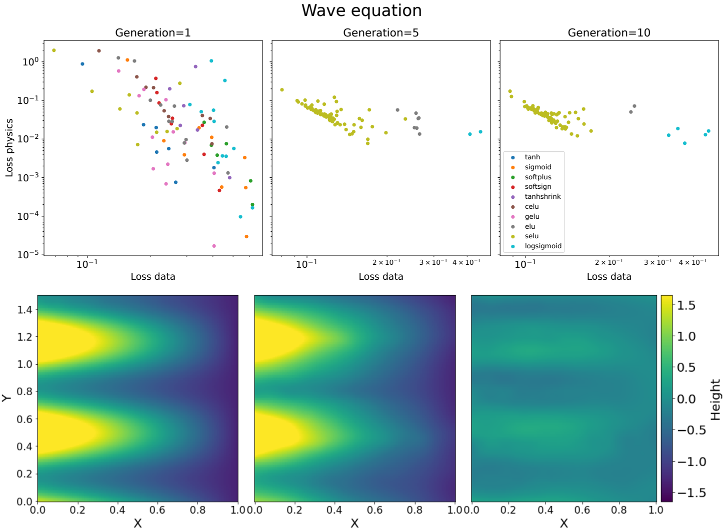
Results - Burgers equation

$50 \times 50 \times 40$ neurons, $\lambda = 0.15$, using the tanh activation function

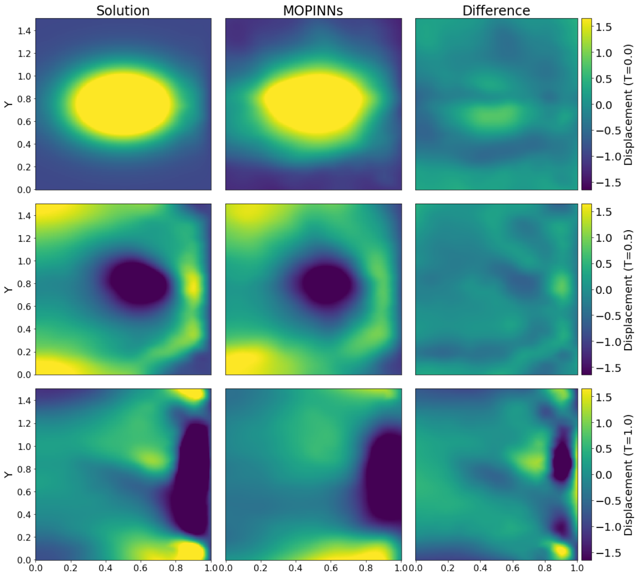


Results - Wave equation

$60 \times 50 \times 50$ neurons, λ = 0.76, using the SELU activation function



Results - Wave equation



Multi-level PINNs: Model nutrients in a fluid (eventually NPZ models)

L1. Physics: Fluid
Two-dimensional decaying turbulence via incompressible Navier-Stokes,

$$\partial_t w + \mathbf{u} \cdot \nabla w = A \Delta w$$
$$\nabla \cdot \mathbf{u} = 0$$

$\mathbf{u} = (u, v)$: flow velocity field, $w = \nabla \times \mathbf{u}$: vorticity, A: eddy viscosity.

L2. Biological: Nutrients
Nutrients over fluid as coupled advection:

$$\partial_t N + \mathbf{u} \cdot \nabla N = 0$$

"Modified MLP"[3]: avoid gradient issues.
Weighted residual loss:[4]

$$\ell_r(\theta) = {}^1\!/_{N_t} \sum_{i=1}^{N_t} w_i \ell_r(t_i, \theta),$$
$$w_i = e^{-\varepsilon \sum_{k=1}^{i-1} \ell_r(t_k, \theta)}$$

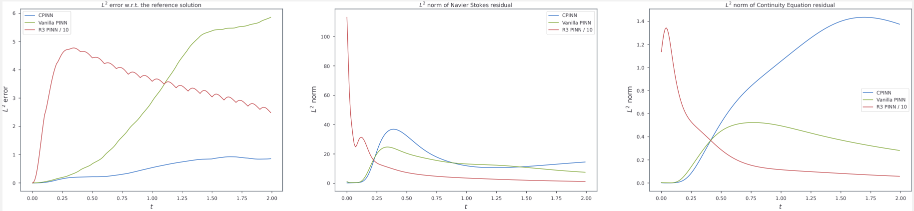
Gate continuous function in the residual loss:[5]

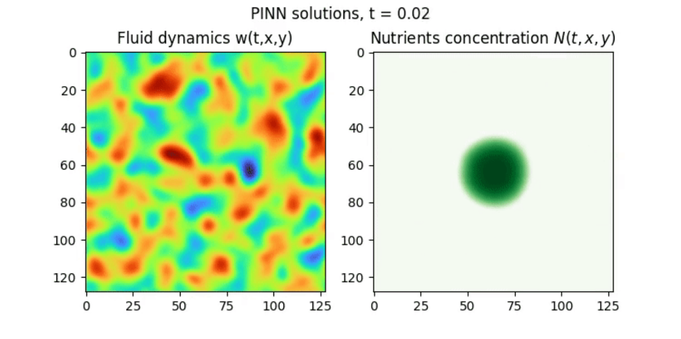
$$\ell_r(\theta) = {}^1\!/_{N_r} \sum_{i=1}^{N_r} \ell_r(x_i, t_i, \theta) g(t_i),$$
$$g(t_i) = 0.5 \left[1 - \tanh(\alpha(t_i - \gamma))\right].$$

[3] Wang, S., Teng, Y., & Perdikaris, P. (2021). Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM Journal on Scientific Computing*, 43, A3055–A3081

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

Multi-PINNs results



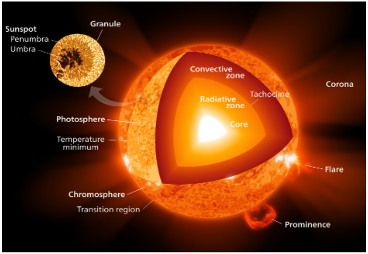


Modelling stellar atmospheres

- Light is created by the interaction of light with the atoms.
- The chemical composition of stars is encoded in their spectra.

Current: MARCS model[6]

- Table with 52.000 entries mapping chemical composition to spectra.
- Observe spectra, find most similar on table -> use that composition.
- Limited to certain types of stars.
- More complex approached “3D” are too computationally expensive.



Source: Wikipedia.

Use PINNs to learn to map from spectra to chemical composition relying on the model(s) of the atmosphere(s).

[6] Gustafsson, B., Edvardsson, B., Eriksson, K., Jorgensen, U. G., Nordlund, A., & Plez, B. (2008). A grid of MARCS model atmospheres for late-type stars: I. methods and general properties. *Astronomy and astrophysics*, 486, 951–970

Thanks! ¡Gracias! Merci !

We have briefly presented some of the work we are doing in as part of project OcéanIA:

- focus on understanding complex natural phenomena,
- create research tools not just better models, and
- problems from completely different contexts share challenges and difficulties → opportunity for knowledge reuse.
- Consolidated code: `https://github.com/Inria-Chile/pypinns`.

Thank you! Obrigado! Merci ! ¡Gracias!

https://inria.cl