AI in Hybrid Systems

Efficient Differential Equation solving using Physically-informed Neural Networks

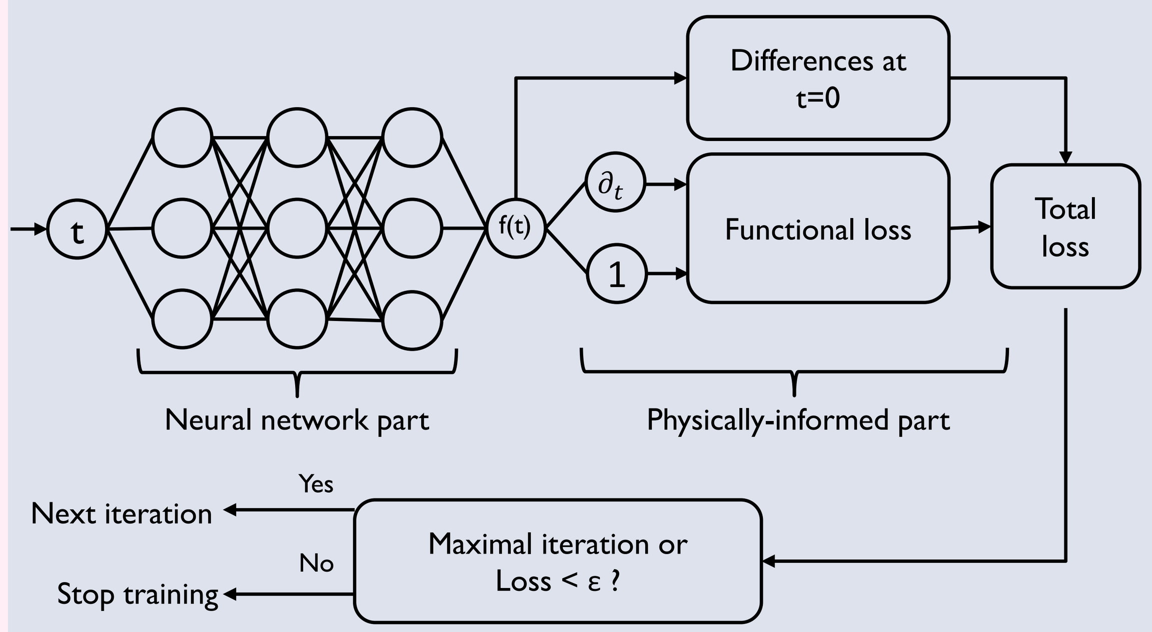
Tariq Berrada, Géraud Faye, Niels Hendrickx, Lila Sainero, Antoine Siraudin, Lina Ye

CentraleSupélec

Curse of dimensionality

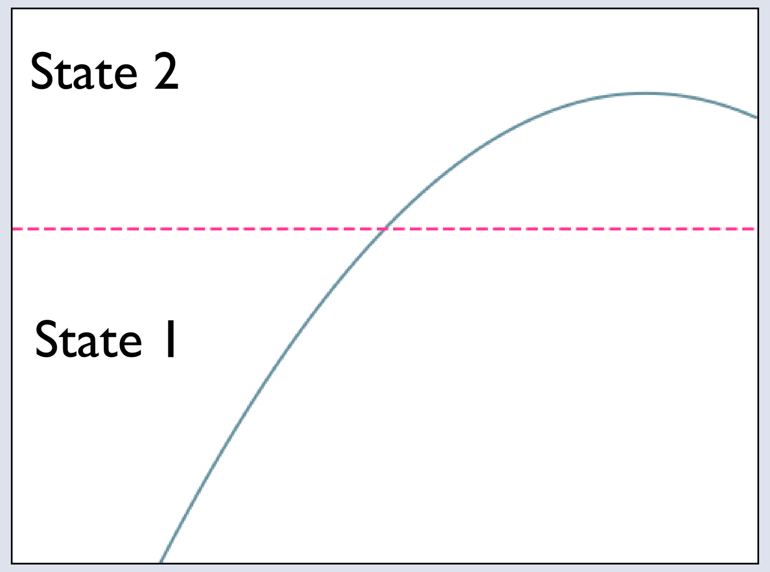
- Solving **differential equations (DE)** is essential to the study of **hybrid systems**. Yet, it is excessively time-consuming in **high dimensions**.
- We use a particular **Neural Network (NN)** class to learn solutions to the DEs then compare our method to classical approaches in terms of **speed**, **precision** and **computational complexity**.

Physically-Informed Neural Network



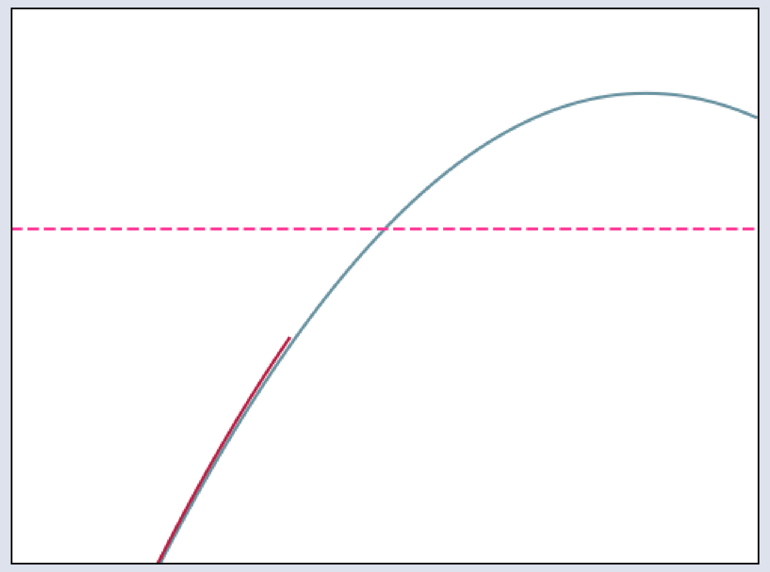
Principle of a Physically-Informed Neural Network (PINN)

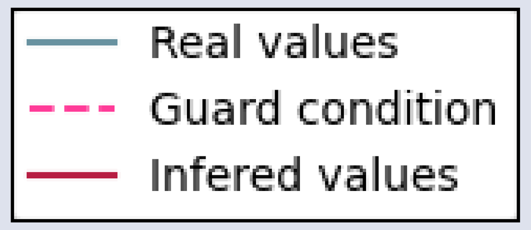
Methodology

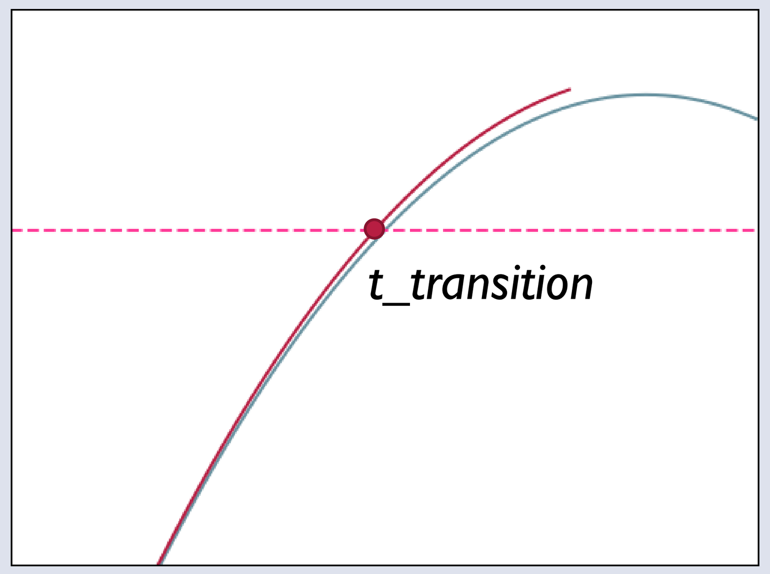


The solution is computed over a time step. The guard condition is not verified.

So we can compute the solution over the next time step

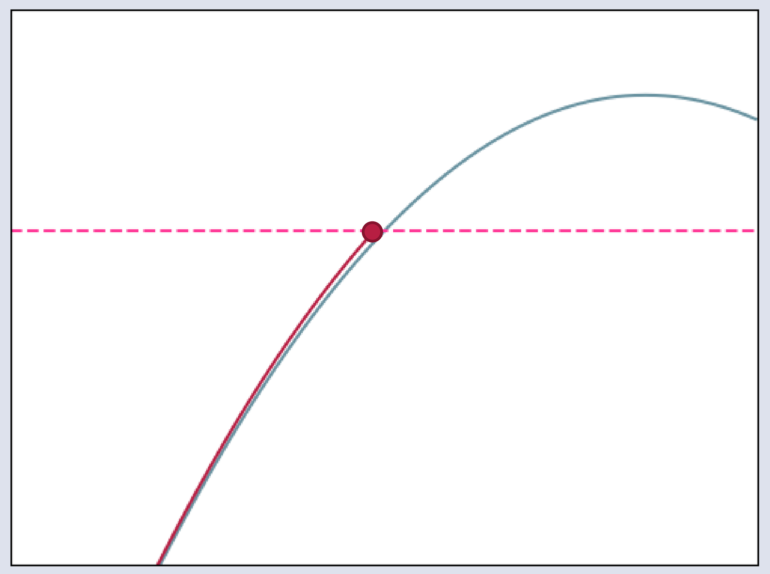




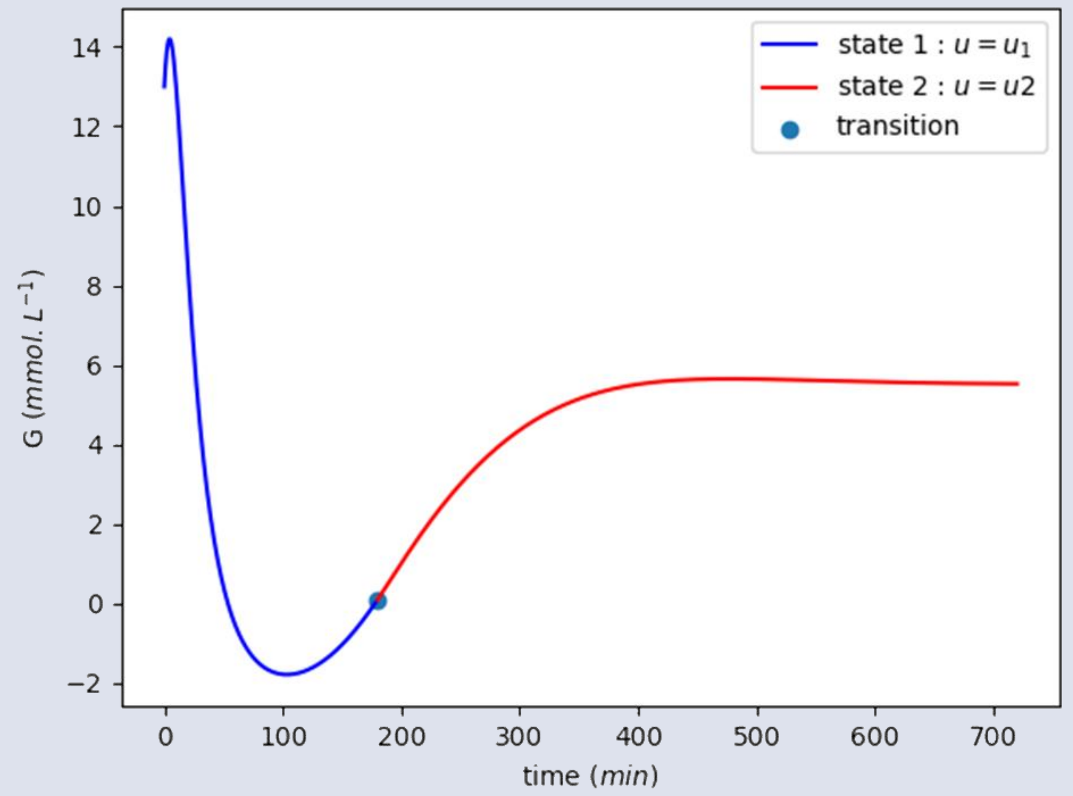


The guard condition has been verified, the associated transition occurs.

A new solution is learned for the new regime while taking transitional conditions into account.

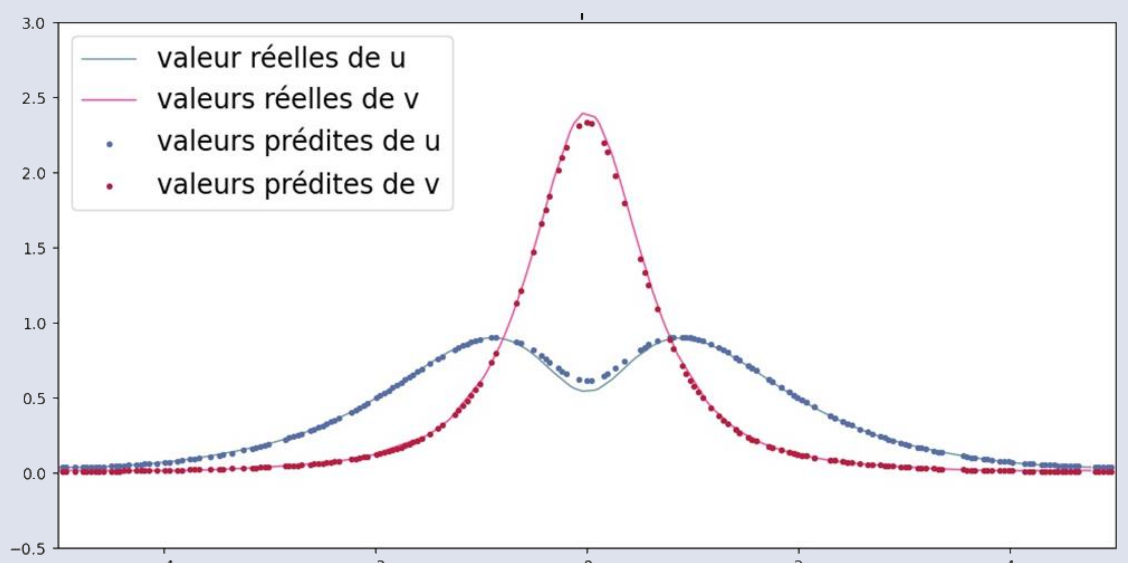


PINN results for a hybrid system



Simulated values of the difference of plasma glucose concentration

PINN results for the Schrödinger equation



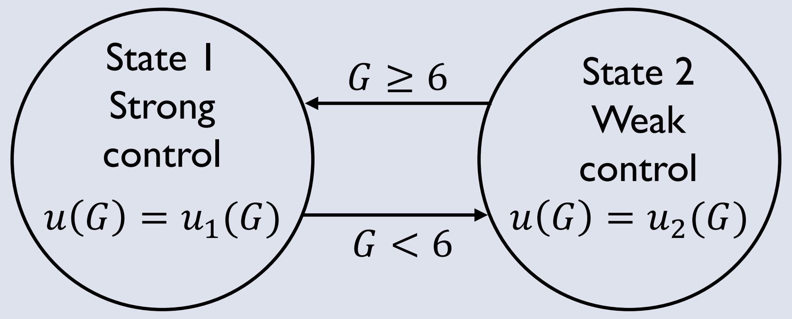
Spatial distribution (time fixed) : PINN predictions vs real values

Hybrid systems

A **hybrid system** combines the behaviors of a **discrete automaton** and a **continuous system.**

Its behavior is defined by the **interaction between both aspects**: the evolution of continuous variables, modeled by DEs, is determined by discrete states. The transition between these states is triggered by conditions on the continuous substate.

$$\begin{cases} \dot{G} = -p_1 G - X(G + G_B) + P(t) \\ \dot{X} = -p_2 X + p_3 I \\ \dot{I} = -n(I + I_B) + \dfrac{u(G)}{V_1} \end{cases}$$



Automaton used for the study

Limits

- Large parameter values in the DE **lead to instability** in the NN, causing **divergence** from the correct solution.
- The model has the same pitfalls as classical hybrid system simulations (zero-crossing detection and location), caused by **Zeno behavior** *(infinite number of transitions during a finite time period)*.

Conclusion

PINNs could be successfully applied to **simulate hybrid systems behavior**.

This method may be promising for hybrid systems which have **high dimensionality**, where PINNs have proven to be **more efficient than classical approaches**.

Link to our github with the full results and references