# Convergence acceleration of a nonlinear solver using statistical learning

Rémy Vallot[1-2], Thibault Dairay [1-2], Florian De Vuyst[3], Mathilde Mougeot[2-4]

[1] Michelin [2]Centre Borelli UMR 9010, École Normale Supérieur Paris-Saclay [3]BBMI UMR 7338, Université de Technologie Compiègne [4]ENSIIE

Contact : remy.vallot@michelin.com

## Introduction

- Numerical simulations in industry : crucial to reduce costly testing, conception, decision making, comprehension.
- Traditional iterative algorithms : convergence heavily dependant on initialization.
- This work explores alternative methods to improve initialization, accelerating convergence while preserving physical accuracy.

## 1 Problem definition

Nonlinear PDE $\rightarrow$ high-dimensional nonlinear algebraic equations, solved using a Quasi-Newton method.

**Finding $U^*$ such that $F(U^*, \boldsymbol{\lambda}) = 0$**

where $\boldsymbol{\lambda}$ is a vector of parameters.

## 2 Strategy

**Use machine learning to find a new initialization for the nonlinear solver generating less iterations to converge.**

Following the work of Jin et al.[2], Aghili et al.[1].

## 3 Applications

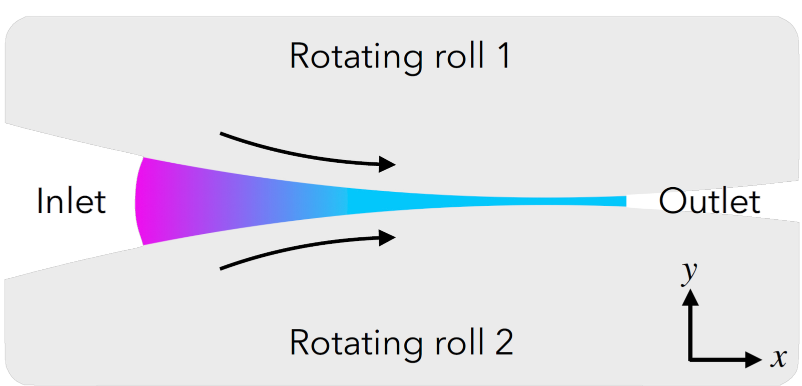
### 3.1 1D Nonlinear Poisson equation

Let $\Omega$ be an open domain of $\mathbb{R}$. The nonlinear Poisson equation is formalised as follows :

$$\begin{cases} -\frac{\partial}{\partial x}\left[q(u)\frac{\partial}{\partial x}u\right] = g(x,\lambda) & \text{in } \Omega \\ u = u_D & \text{on } \partial\Omega \end{cases}.$$

### 3.2 2D calendering process

- Rubber process manufacturing : compress material between two counter-rotating rolls.
- Focus on mechanical fields. Thermal effects are not considered in this study.




**Governing equations :**
Mass and momentum conservation for incompressible highly viscous flow : divergence free velocity and nonlinear Stokes equations.
Nonlinear power-law viscosity model : $\boldsymbol{\lambda}$ are the coefficients of the viscosity model.

## 4 Data & baseline strategies

**Consider :**

- a set of vectors $\{\lambda_k\}$ of parameters
- a set of associated discretized solutions $\{U_k^*\}$ computed using the solver.

**Sampling strategy :**

- **Train :** regular grid $10 \times 10$ in the parameter space.
- **Test :** 100 pairs via Latin Hypercube Sampling in the same intervals.

**2 baseline strategies :**

- **Classical initialization :** constant field set to boundary value.
- **Nearest initialization :** use the solution from the closest parameter in the database.

## 5 Tested methods

All model diagrams follow this color scheme

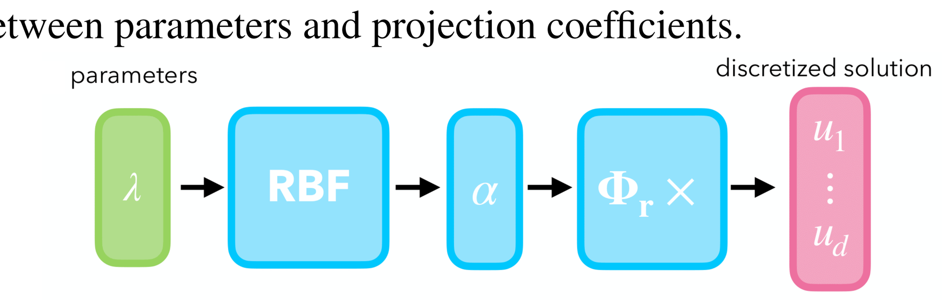
input | model | output

### 5.1 Proper Orthogonal Decomposition (POD)[4]

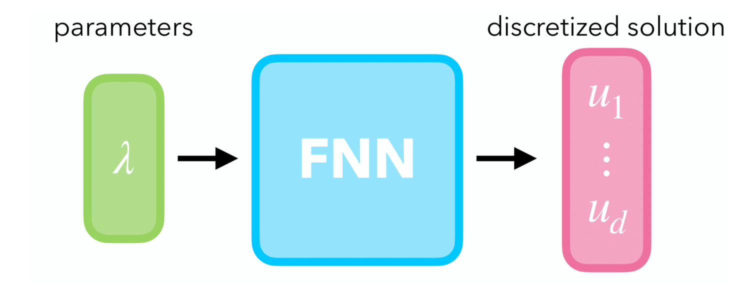
Consider a snapshot matrix $U = \begin{bmatrix} | & | & & | \\ U_1^* & U_2^* & \cdots & U_n^* \\ | & | & & | \end{bmatrix}$.

Compute truncated SVD ($U = \Phi_r \Sigma_r V_r^T$) to obtain $\Phi_r$, the orthonormal reduced basis between projection coefficients $\boldsymbol{\alpha_k}$ and solution $\boldsymbol{U_k}$

Define interpolation method like Radial Basis Function (RBF) between parameters and projection coefficients.




### 5.2 Neural Networks (NN)



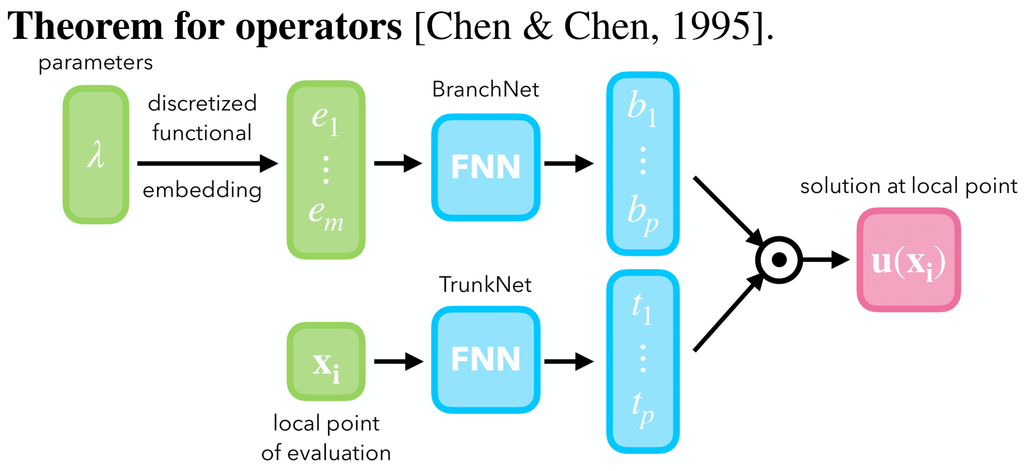

Training objective :

$$\min_{\theta} \mathscr{L}(\theta) = \frac{1}{n}\sum_{k=1}^{n} \|\text{FNN}_\theta(\lambda_k) - U_k\|^2$$

This architecture is **poorly scalable**: one output per discretization point, and prediction points are fixed.

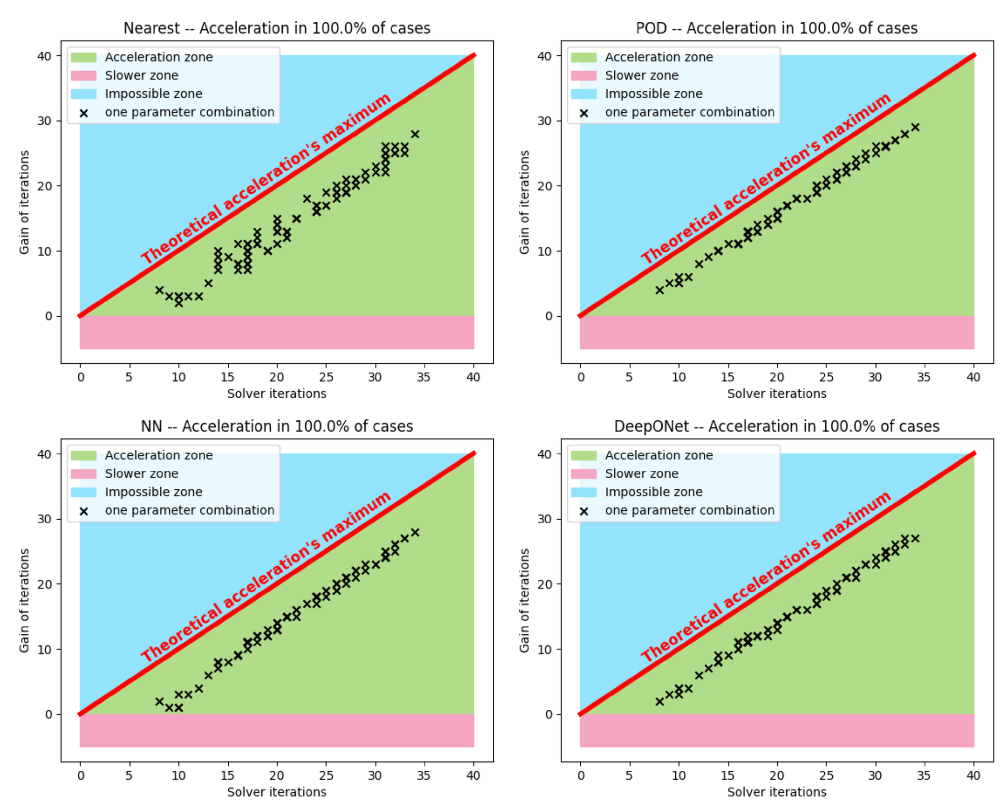
### 5.3 Deep Operator Networks (DeepONet) [3]

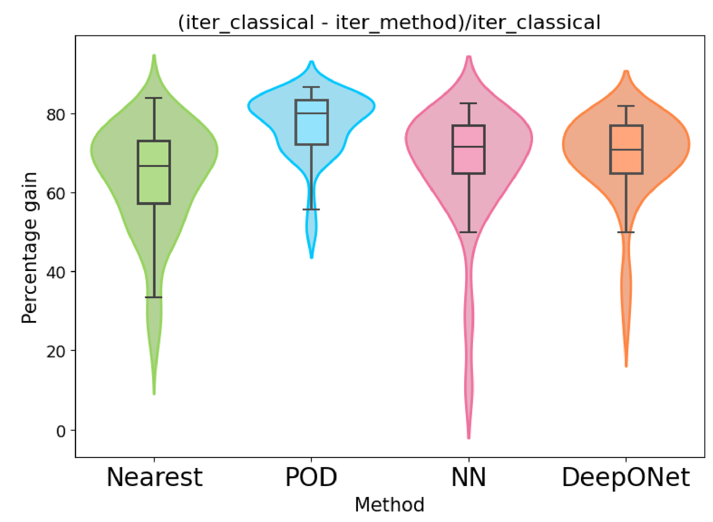
Neural operator architecture to approximate mappings between functions. Supported by the **Universal Approximation Theorem for operators** [Chen & Chen, 1995].




## 6 Results

### 6.1 1D example

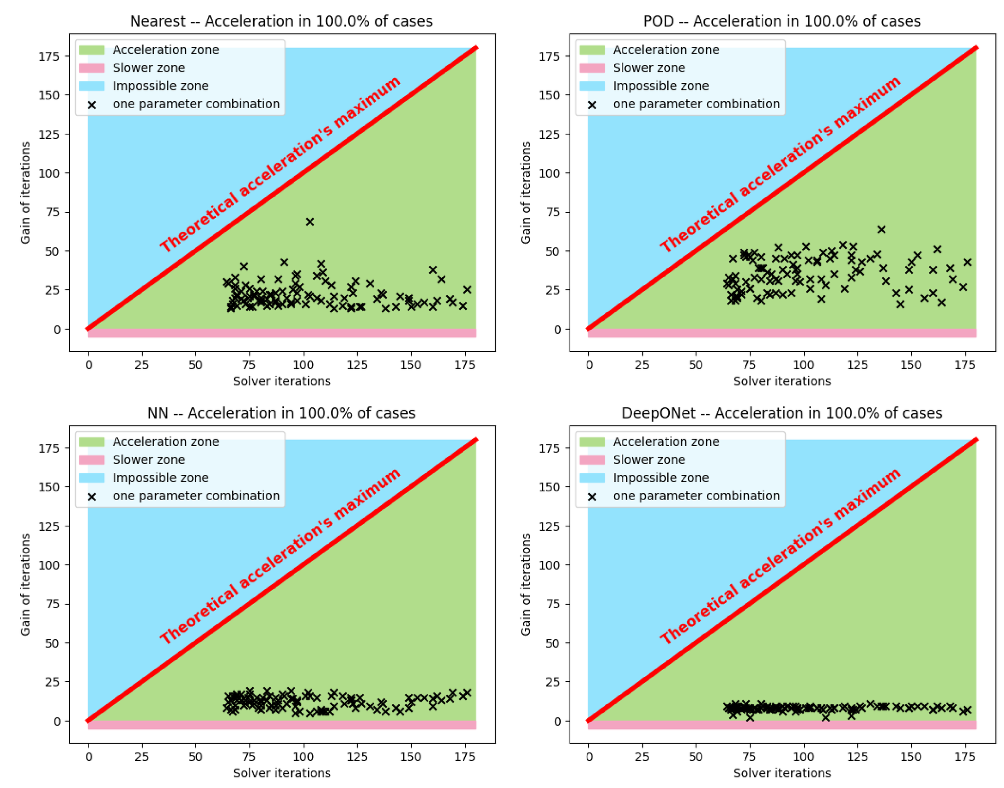


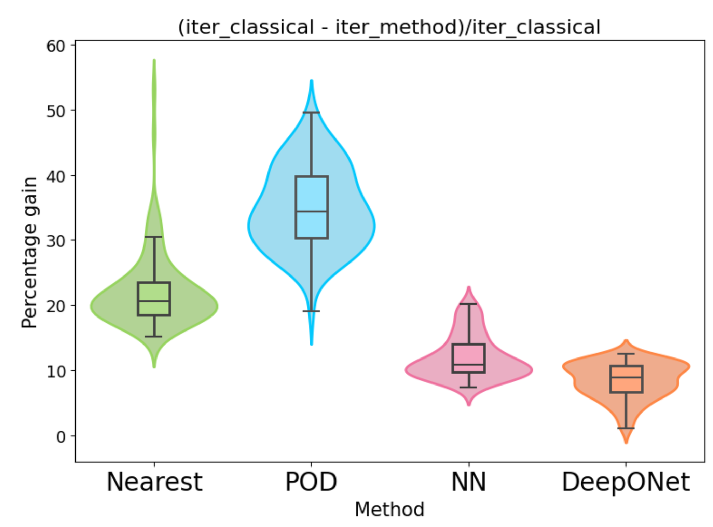




| | Mean | Median |
|---|---|---|
| **Nearest** | 63.47 | 66.67 |
| **POD** | 77.13 | 80.00 |
| **NN** | 66.58 | 71.43 |
| **DeepONet** | 68.89 | 70.83 |

*n. obs = 100*

### 6.2 2D example




| | Mean | Median |
|---|---|---|
| **Nearest** | 22.16 | 20.54 |
| **POD** | 34.89 | 34.31 |
| **NN** | 12.06 | 10.91 |
| **DeepONet** | 8.49 | 8.89 |

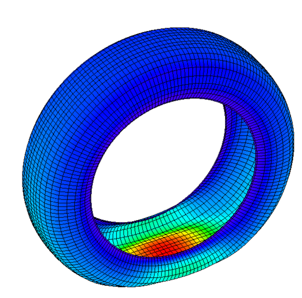
*n. obs = 100*

## 7 Preliminary conclusions

- All initialization methods perform better than using a constant field.
- POD consistently outperforms all other tested methods.
- In 1D, the iteration gain grows with baseline cost; in 2D the iteration gain is constant.
- Why not PINNs ? Longer training time, inefficient for parametric studies

## 8 Future work

- Extension to industrial test cases
- Generalization to geometrical variability



**How to assess confidence in surrogate-based initialization ?**

### References

[1] Aghili, J., Franck, E., Hild, R., Michel-Dansac, V., and Vigon, V. (2025). Accelerating the convergence of newton's method for nonlinear elliptic pdes using fourier neural operators. *Communications in Nonlinear Science and Numerical Simulation*, 140:108434.

[2] Jin, T., Maierhofer, G., Schratz, K., and Xiang, Y. (2025). A fast neural hybrid newton solver adapted to implicit methods for nonlinear dynamics.

[3] Lu, L., Jin, P., Pang, G., Zhang, Z., and Karniadakis, G. E. (2021). Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229.

[4] Lumley, J. L. (1967). The structure of inhomogeneous turbulent flows. *Atmospheric Turbulence and Radio Wave Propagation*, pages 166–178.