

Progressive Enrichment of Surrogate Models with Guaranteed Error Reduction for Scientific Machine Learning

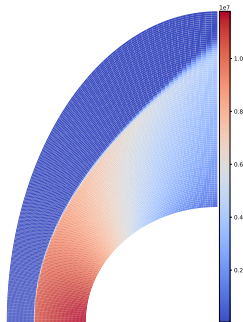
September, 22 2026

B. Poulleau^{1,2}, G. Poette^{1,2}, R. Turpault¹

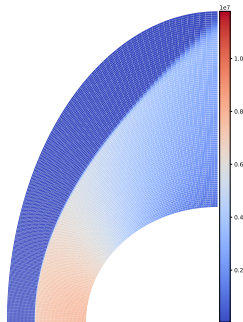
¹INSTITUT DE MATHÉMATIQUES DE BORDEAUX, ²CEA-CESTA

Motivating Example: Atmospheric Reentry Flow

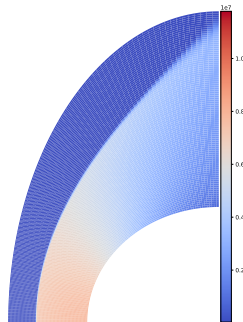
Pressure fields and computational cost comparison [3]



Euler
80 s



Euler + M^{++} [5]
4,090 s ($\times 50$ vs. *Euler*)



Euler + NN [3]
136 s ($\times 30$ speedup vs. M^{++})



Neural Networks for Scientific Machine Learning

Neural networks enable significant advancements in Scientific ML:

- Accelerating expensive simulations
(*e.g.* in aerodynamics [3], multi-material compressible Euler model [2], ...)
- Enabling Bayesian inference under reasonable computational constraints
- Handling shape and topological optimization problems with relative ease
(*e.g.* volume preserving shape optim [1])
- ...

However, major open challenges remain in SciML:

- Guarantees
- Interpretability/explainability
- Frugality



The Reality Behind Neural Network Deployment

During the training and design phase, one must critically choose:

- Number of hidden layers
- Number of neurons per layer
- Activation functions
- Weight and bias initialization strategies
- ...

Open Research Questions:

[Frugality] Do we really need deep architectures? Can we add complexity only when needed?

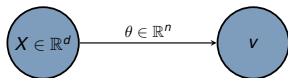
[Interpr.] Are neural networks always the optimal choice?

[Guarant.] Can we guarantee that adding a layer improves the approximation?

[Interpr.] Can we start from an interpretable baseline model and improve it?

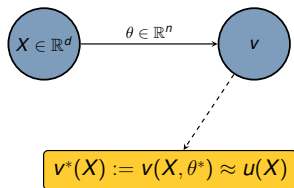
Single-Layer Enrichment: Overview

We want to approximate $u(X) \in \mathbb{R}$.



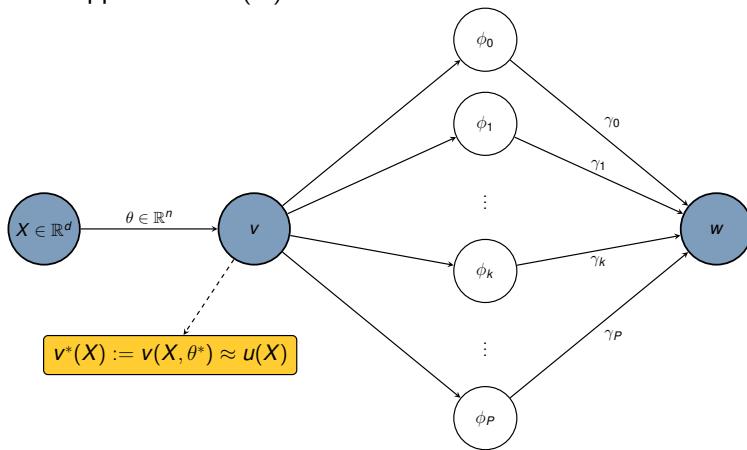
Single-Layer Enrichment: Overview

We want to approximate $u(X) \in \mathbb{R}$.



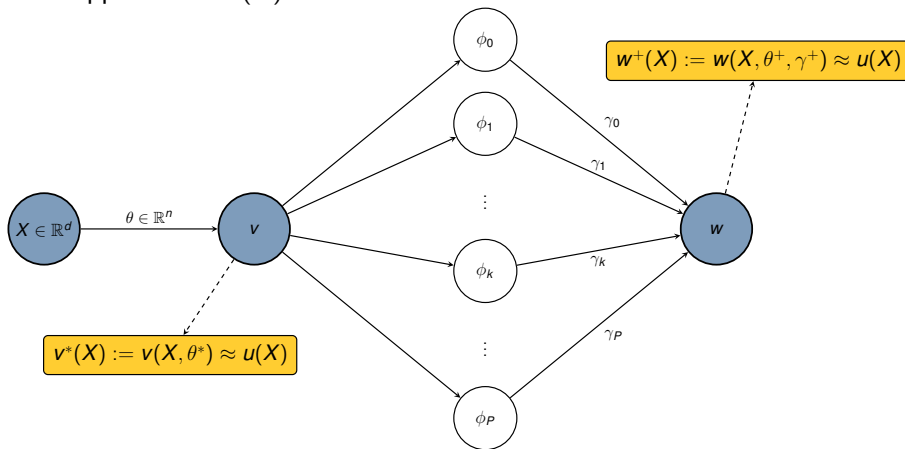
Single-Layer Enrichment: Overview

We want to approximate $u(X) \in \mathbb{R}$.



Single-Layer Enrichment: Overview

We want to approximate $u(X) \in \mathbb{R}$.



How to guarantee that w^+ is more accurate than v^* ?

Single-Layer Enrichment: Mathematical Details

Step 1: Functional Enrichment

- First approx: $v^*(X) := v(X; \theta^*) \approx u(X)$
- Enriched space:

$$V_P^* = \text{Span} \{ \phi_k \circ v^* \}_{k=0}^P$$

where $\{ \phi_k \}_k$ are $\perp$ w.r.t. $\langle \cdot, \cdot \rangle_{v^*(X)}$

- Key property: $v^* \in V_P^*$
- By choosing the new candidate

$$w^* = \Pi_{V_P^*} u = \sum_{k=0}^P \gamma_k^* \phi_k \circ v^*$$

where $\gamma_k^* = \langle u, \phi_k \circ v^* \rangle_X$

- we have a non-degradation guarantee [4]:

$$\|u - w^*\|_X \leq \|u - v^*\|_X$$

Single-Layer Enrichment: Mathematical Details

Step 1: Functional Enrichment

- First approx: $v^*(X) := v(X; \theta^*) \approx u(X)$
- Enriched space:

$$V_P^* = \text{Span} \{ \phi_k \circ v^* \}_{k=0}^P$$

where $\{ \phi_k \}_k$ are $\perp$ w.r.t. $\langle \cdot, \cdot \rangle_{v^*(X)}$

- Key property: $v^* \in V_P^*$
- By choosing the new candidate

$$w^* = \Pi_{V_P^*} u = \sum_{k=0}^P \gamma_k^* \phi_k \circ v^*$$

where $\gamma_k^* = \langle u, \phi_k \circ v^* \rangle_X$

- we have a non-degradation guarantee [4]:

$$\|u - w^*\|_X \leq \|u - v^*\|_X$$

Step 2: Optimization

- We want to minimize

$$\mathcal{R}(\theta, \gamma) := \frac{1}{2} \|u - w(\cdot; \theta, \gamma)\|_X^2$$

where $w^*(X) = w(X; \theta^*, \gamma^*)$

- Let suppose

$$(\theta^*, \gamma^*) \xrightarrow{\text{optimization algorithm}} (\theta^+, \gamma^+)$$

with an optimizer that guarantees

$\mathcal{R}(\theta^+, \gamma^+) < \mathcal{R}(\theta^*, \gamma^*)$, if (θ^*, γ^*) is non critical

- We have a strict improvement guarantee:

$$\|u - w(\cdot, \theta^+, \gamma^+)\|_X < \|u - v^*\|_X$$

Gerschgorin-shifted Newton Algorithm

I SPD regularization: $z^n := (\theta^n, \gamma^n)$, $H^n := \nabla^2 \mathcal{R}(z^n)$, $\varepsilon > 0$,

$$\mu^n = \max \left(0, \max_i \left\{ \sum_{j \neq i} |h_{ij}^n| - h_{ii}^n \right\} + \varepsilon \right) \xRightarrow{\text{Gerschgorin theorem}} \mathcal{H}^n := H^n + \mu^n I \succ 0.$$

■ *Shift refinement*: decrease μ^n by factor 2 while preserving SPD.

II Descent direction:

$$d^n := -[\mathcal{H}^n]^{-1} \nabla \mathcal{R}(z^n) \xRightarrow{\mathcal{H}^n \succ 0} \langle \nabla \mathcal{R}(z^n), d^n \rangle < 0, \quad \text{if } \nabla \mathcal{R}(z^n) \neq 0.$$

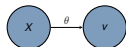
III For sufficiently small $\eta^n > 0$, first-order Taylor expansion gives:

$$\mathcal{R}(z^n + \eta^n d^n) < \mathcal{R}(z^n).$$

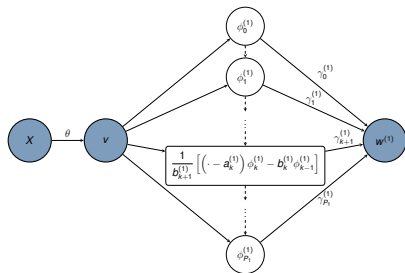
■ *Backtracking*: start from $\eta^n = 1$ and halve $\eta^n \leftarrow \eta^n/2$ until the decrease condition is satisfied.

IV Update: $z^{n+1} = z^n + \eta^n d^n$.

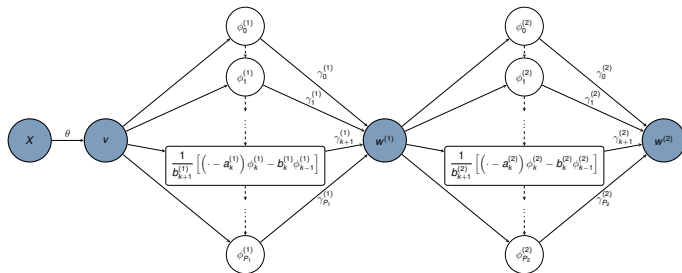
Progressive Enrichment: Multi-Layer Architecture (full model)



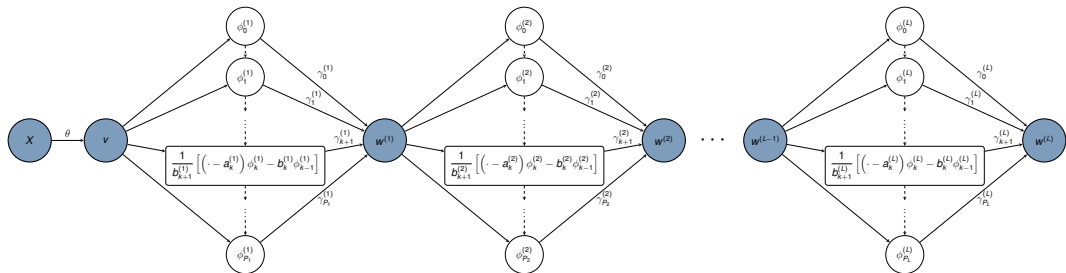
Progressive Enrichment: Multi-Layer Architecture (full model)



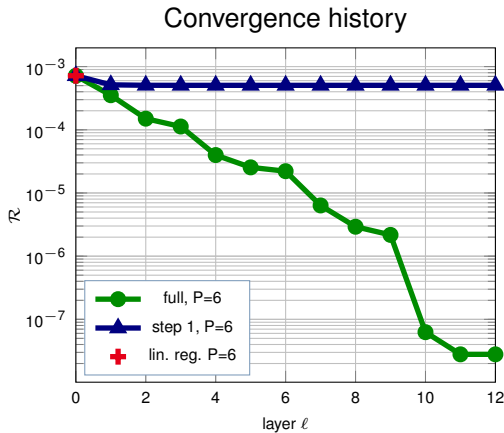
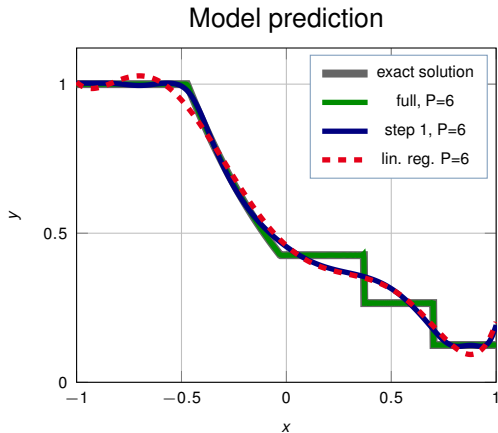
Progressive Enrichment: Multi-Layer Architecture (full model)



Progressive Enrichment: Multi-Layer Architecture (full model)



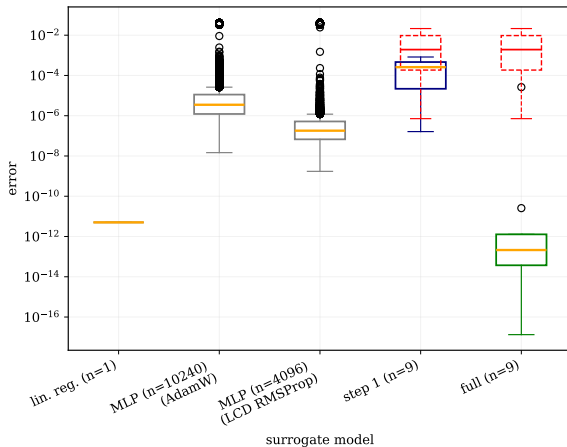
Numerical Application: Sod Shock Tube for Euler



Optimization impact (full vs. step 1) on a polynomial regression baseline (lin. reg.).

Numerical Application: Runge Function

Comparison of surrogate models with 60 degrees of freedom



Error distribution over n independent trainings (60 DoF), baseline v in red.

Conclusion

Proposed Framework & Key Achievements:

- Interpretable enrichment through orthogonal polynomial functions.
- Guaranteed progressive accuracy improvement.
- Controlled complexity throughout the expansion.
- Applicable to a broad class of scalar-output surrogate models.

Core Mathematical Pillars:

- for the architecture (step 1):
 - Inclusion of the baseline in the enriched space leveraging orthonormal polynomial properties.
 - Numerical stability achieved by storing 3-term recurrence coefficients.
- for the optimization (step 2):
 - Optimizer ensuring strict error decay.
 - Certified initialization by orthogonal projection

Perspectives

In progress . . .

- Move beyond the current polynomial regression baseline to integrate other reduced models (*e.g.* Kriging, classical MLPs, etc.).
- Establish a formal proof of the non-triviality of $\nabla \mathcal{R}(\theta^*, \gamma^*)$ under specific mathematical conditions.
- Publication

Future works:

- Multi-Dimensional Extension: Generalize the layer-by-layer expansion to handle multi-variate and vector-valued physical fields.
- Multi-Fidelity Physics: Apply the method to complex physical applications requiring multi-fidelity simulation tools.

Thank you for your attention !

- [1] Amaury Bélières–Frendo, Emmanuel Franck, Victor Michel-Dansac, and Yannick Privat.
Volume-preserving geometric shape optimization of the Dirichlet energy using variational neural networks.
Neural Networks, 184:106957, April 2025.
- [2] Bruno Després and Hervé Jourdain.
Machine Learning design of Volume of Fluid schemes for compressible flows.
Journal of Computational Physics, 408:109275, May 2020.
- [3] Paul Novello.
Combining supervised deep learning and scientific computing: Some contributions and application to computational fluid dynamics.
- [4] Gaël Poëtte, Alexandre Birolleau, and Didier Lucor.
Iterative Polynomial Approximation Adapting to Arbitrary Probability Distribution.
SIAM Journal on Numerical Analysis, 53(3):1559–1584, January 2015.
- [5] James B. Scoggins, Vincent Leroy, Georgios Bellas-Chatzigeorgis, Bruno Dias, and Thierry E. Magin.
Mutation++: Multicomponent Thermodynamic And Transport properties for IONized gases in C++.
SoftwareX, 12:100575, July 2020.

Appendix: 3-Term Recurrence Relation

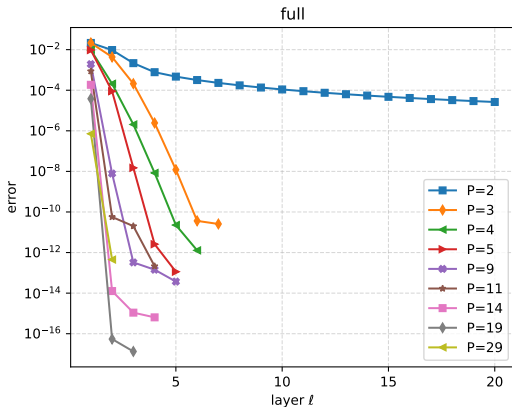
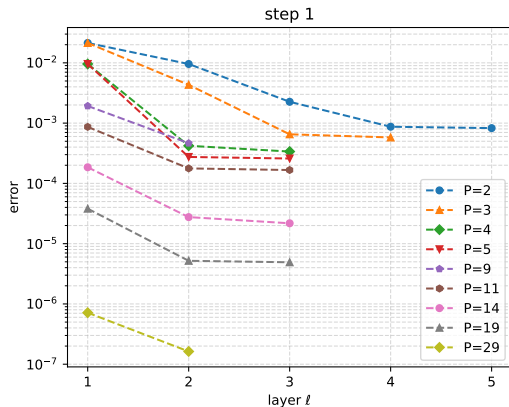
Let $(\phi_n)_{n \geq 0}$ be the sequence of orthonormal polynomials with respect to the probability measure ρ . Then, there exist unique real sequences $(a_n)_{n \geq 0}$ and $(b_n)_{n \geq 1}$, with $b_n > 0$, such that for all $n \geq 0$:

$$\begin{cases} \phi_{-1}(z) := 0, \\ \phi_0(z) := \frac{1}{\sqrt{\langle 1, 1 \rangle_\rho}}, \\ b_{n+1} \phi_{n+1}(z) = (z - a_n) \phi_n(z) - b_n \phi_{n-1}(z), \quad n = 0, 1, 2, \dots \end{cases}$$

where the recurrence coefficients are uniquely defined by:

$$a_n = \langle z \phi_n, \phi_n \rangle_\rho, \quad b_{n+1} = \langle z \phi_n, \phi_{n+1} \rangle_\rho, \quad n = 0, 1, 2, \dots$$

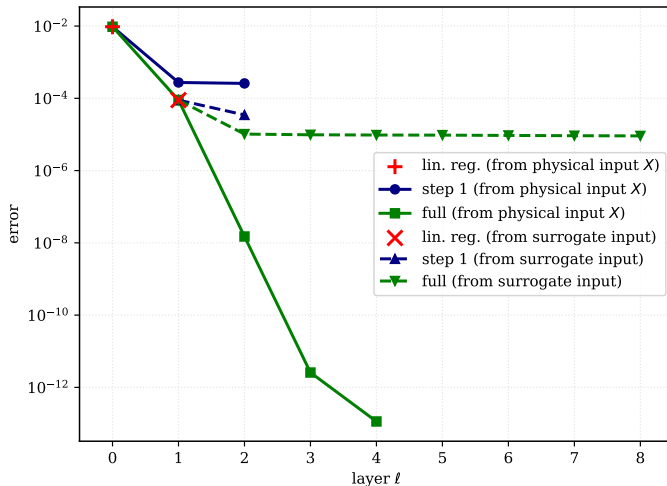
Appendix: Numerical Application: Runge Function



Optimization impact (step 1 vs. full, 60 DoF) on Runge function .

Appendix: Numerical Application: Runge Function

Comparison of physical vs. surrogate input space on the Runge function.



Appendix: Numerical Application: Sod Shock Tube Problem

Comparison of physical vs. surrogate input space on the Sod shock tube problem.

