Nonlinear Elliptic Problem

Model Order Reduction and Machine Learning
Master’s Degree in Mathematical Engineering

Lucia Ghezzi, Elisabetta Roviera

June 14, 2025

Table of Contents

Nonlinear Elliptic Problem (NEP)

Problem definition Given $\Omega = (0,1)^2$, given $\mu = (\mu_0, \mu_1) \in \mathcal{P} = [0.1, 1]^2$, find $u(\mu)$ such that

$$-\Delta u(\mu) + \frac{\mu_0}{\mu_1}(e^{\mu_1 u(\mu)} - 1) = g(x;\mu)$$

with homogeneous Dirichlet condition on the boundary. The source term g is defined as:

1. For NEP1:
$$g(x;\mu) = g_1 = 100 \sin(2\pi x_0)\cos(2\pi x_1), \quad \forall x = (x_0, x_1) \in \Omega.$$
2. For NEP2:
$$g(x;\mu) = g_2 = 100 \sin(2\pi \mu_0 x_0)\cos(2\pi \mu_0 x_1), \quad \forall x = (x_0, x_1) \in \Omega.$$

Weak formulation and Newton scheme Integrating on the domain, multiplying by a general function $v \in V$ and recalling the boundary condition, we get the weak formulation: given $\mu \in \mathcal{P}$, find $u(\mu) \in V$ such that for every $v \in V$

$$F(u)[v] = \int_\Omega \nabla u \cdot \nabla v \, dx + \int_\Omega \frac{\mu_0}{\mu_1}(e^{\mu_1 u} - 1) v \, dx - \int_\Omega g v \, dx = 0.$$

To solve $F(u)[v] = 0$ at each Newton iteration, we solve for δu

$$\left(\int_\Omega \nabla \delta u \cdot \nabla v \, dx + \int_\Omega \mu_0 e^{\mu_1 u_k} \delta u v \, dx \right) = -\left(\int_\Omega \nabla u_k \cdot \nabla v \, dx + \int_\Omega \frac{\mu_0}{\mu_1}(e^{\mu_1 u_k} - 1) v \, dx - \int_\Omega g v \, dx \right)$$

and update $u_{k+1} = u_k + \delta u$.

Check of theoretical results We know from theory that for mesh size h it holds

$$Err_{L^2}(h) = Err_{L^2}(h_0)\left(\frac{h}{h_0}\right)^{s+1}, \quad Err_{H^1}(h) = Err_{H^1}(h_0)\left(\frac{h}{h_0}\right)^{s}.$$

We check if the expected behavior is observed experimentally.
Choice of the mesh size The two most suitable mesh sizes are 0.00312 and 0.00019, so we evaluate the tradeoff between accuracy and cost:

Performance metrics for different mesh sizes

Metric	Mesh = 0.00312	Mesh = 0.00019
Average snapshot computation time (s)	0.5948	11.2261
Average relative error (L^2 Norm)	0.0089	0.0005
Average relative error (H^1 Norm)	0.0937	0.0224

Table of Contents

2 Methods

1. **POD**: the reduced dimension for NEP1 is $N = 3$ and $N = 9$ for NEP2
2. **PINN**: frase sul pinn mooolto breve

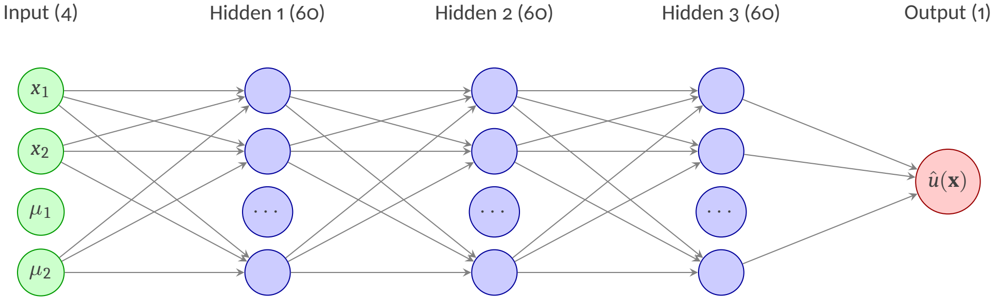

Figure: PINNHardBC Architecture Diagram.

3. **POD-NN**: fully connected network with 4 hidden layers of 40 neurons, $\tanh$ activation, Adam optimizer (`lr`=0.001), up to 500,000 epochs, early stopping at 10^{-6}.

Table of Contents

Comparison of Methods – NEP1

Performance comparison: Accuracy vs computational cost for NEP1

NEP1 Summary		POD (N=3)	PINN	PODNN
Error w.r.t. HF	L2 relative	2.77×10^{-5}	4.19×10^{-2}	6.05×10^{-4}
	H1 relative	3.07×10^{-5}	2.18×10^{-1}	6.04×10^{-4}
Execution Time	Avg. eval. time (s)	8.04×10^{-4}	1.10×10^{-3}	2.18×10^{-4}
	Avg. speed-up vs HF	15.66	8.42	68.62
Training	Iterations	–	10,689	119,274
	Training time (s)	–	718.11	133.36

Plots

image1.png

Figure: Comparison of High Fidelity and PINN solutions

Animated plot

Comparison of Methods – NEP2

Performance comparison: Accuracy vs computational cost for NEP2

NEP2 Summary		**POD (N=9)**	**PINN**	**PODNN**
Error w.r.t. HF	L2 relative	5.1432×10^{-4}	2.4579×10^{-2}	2.8012×10^{-2}
	H1 relative	9.8641×10^{-4}	1.5484×10^{-1}	2.5705×10^{-2}
Execution Time	Avg. eval. time (s)	5.5382×10^{-4}	1.1493×10^{-3}	1.8587×10^{-4}
	Avg. speed-up vs HF	22.327	10.0273	70.8458
Training	Iterations	–	17,415	500,000
	Training time (s)	–	1068.91	570.41

Animated plot

Thank you for your attention!