A Machine Learning Minimal Residual Method for Solving Quantities of Interest of Parametric PDEs

Ignacio Brevis, *Ignacio Muga, David Pardo, Oscar Rodríguez, and Kristoffer G. van der Zee*

University of Nottingham UK | CHINA | MALAYSIA — IMA Instituto de Matemáticas — Pontificia Universidad Católica de Valparaíso — bcam basque center for applied mathematics — Universidad del País Vasco / Euskal Herriko Unibertsitatea — UKRI Engineering and Physical Sciences Research Council

Problem definition

We want:
- Solve parametric families of PDEs
- High precision in terms of a Quantity of Interest (QoI)
- Employing coarse meshes in Finite Elements

We consider the abstract variational formulation:
Given $\lambda \in \Lambda \subset \mathbb{R}^p$, find $q(u) \in \mathbb{R}$ such that

$$\begin{cases} u \in \mathbb{U} \text{ solves:} \\ B_\lambda u = \ell_\lambda \quad \in \mathbb{V}^*, \end{cases} \quad \Leftrightarrow \quad \begin{cases} u \in \mathbb{U} \text{ solves:} \\ b_\lambda(u,v) = \ell_\lambda(v), \quad \forall v \in \mathbb{V}. \end{cases}$$

We will use:
- Minimal-residual (MinRes) finite elements
- Artificial neural networks

Introduction to the method (weighted-MinRes)

Given discrete conforming trial $\mathbb{U}_n := \operatorname{span}\{\varphi_1, \dots, \varphi_n\} \subset \mathbb{U}$, and test $\mathbb{V}_m \subset \mathbb{V}$ spaces such that $\dim \mathbb{V}_m > \dim \mathbb{U}_n$, we want to find a discrete solution

$$u_n := \operatorname*{argmin}_{w_n \in \mathbb{U}_n} \|\ell_\lambda - B_\lambda w_n\|_{(\mathbb{V}_m)^*}, \quad \text{(MinRes method)}$$

To solve the above problem is equivalent to solve the following saddle point problem:

$$\begin{cases} \text{Find } (r_m, u_n) \in \mathbb{V}_m \times \mathbb{U}_n \text{ such that:} \\ (r_m, v_m)_{\mathbb{V}} + b_\lambda(u_n, v_m) = \ell_\lambda(v_m), \quad \forall v_m \in \mathbb{V}_m, \\ b_\lambda(w_n, r_m) = 0, \quad \forall w_n \in \mathbb{U}_n. \end{cases} \tag{1}$$

What we do:

We modify the MinRes method by introducing a weighted inner product

$$(\cdot,\cdot)_{\mathbb{V}} \longrightarrow (\cdot,\cdot)_{\mathbb{V},\omega}$$

in the $\mathbb{V}$ space [1]

Example: $H_0^1(\Omega)$

$$(r,v)_{\mathbb{V},\omega} = \sum_{k=1}^{K} \omega_k \underbrace{\int_{\Omega_k} \nabla r(x) \cdot \nabla v(x)\, dx}_{G_k}$$

We seek for the coefficients $\omega := (\omega_1, \dots, \omega_m)$, such that solving the system (1)

$$\begin{bmatrix} G_\omega & B_\lambda \\ B_\lambda^\top & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} L_\lambda \\ 0 \end{bmatrix}, \qquad \text{with } G_\omega = \sum_{k=1}^{K} \omega_k G_k.$$

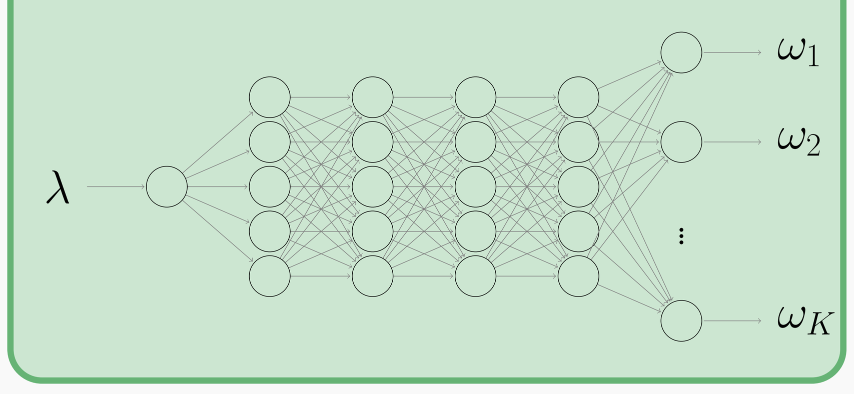
delivers $u_n(x) := \sum_{i=1}^n \beta_i(\omega)\varphi_i(x)$ as a good approximation for the QoI $q(u)$.

Obs: See [2] for an extension to Galerkin and least-squares formulations.

Artificial neural networks

We train a neural network nn_θ to learn the inner-product piecewise constant.

Delivers every inner-product coefficient



Training set

$$\{\lambda^{(1)}, \lambda^{(2)}, \dots \lambda^{(N_s)}\}$$

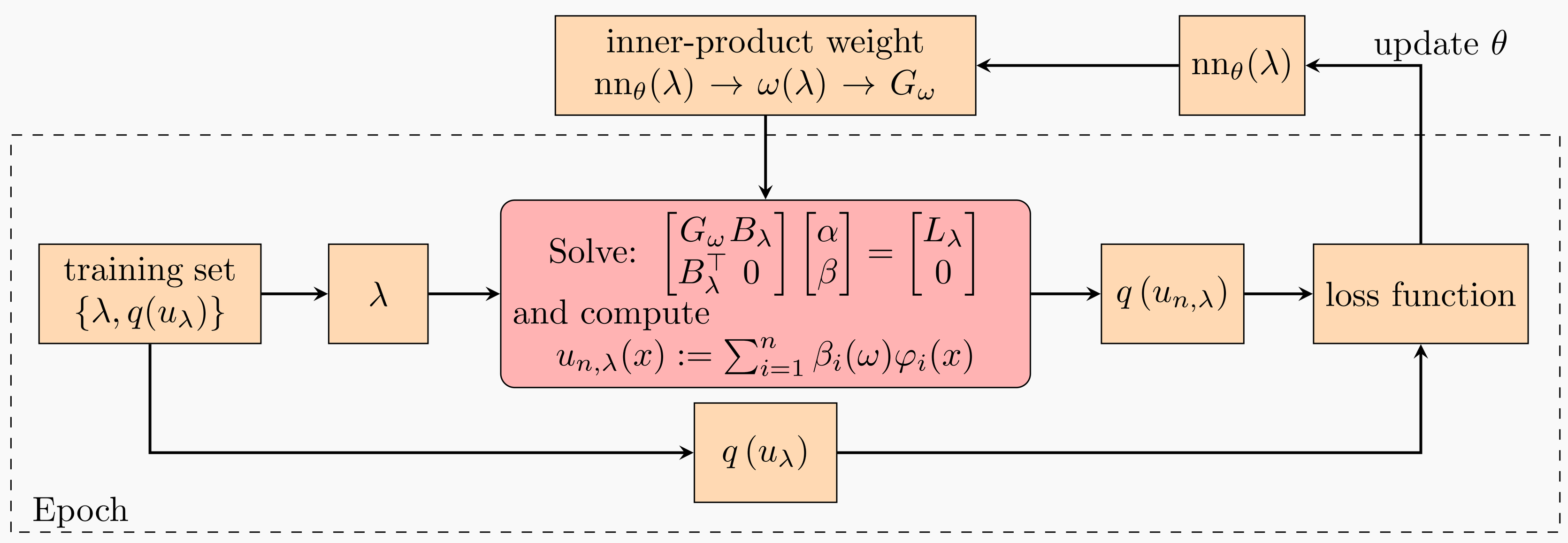
Labels

$$\{q(u_{\lambda^{(1)}}), q(u_{\lambda^{(2)}}), \dots q(u_{\lambda^{(N_s)}})\}$$

Loss function

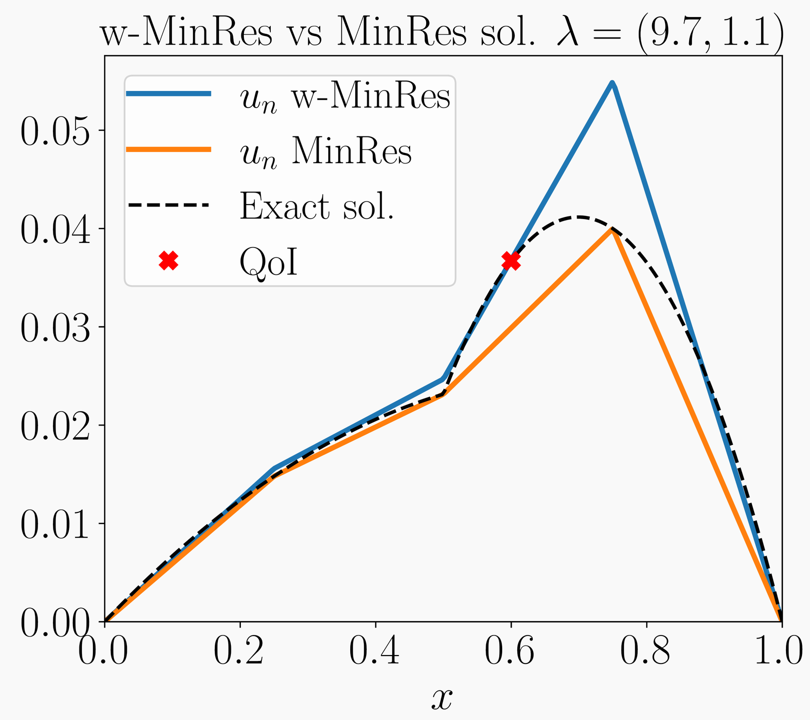
$$\mathcal{L}(\theta) := \frac{1}{N_s} \sum_{i=1}^{N_s} \frac{1}{2} \left| \frac{q(u_{\lambda^{(i)}}) - q(u_{n,\lambda^{(i)}}(\theta))}{q(u_{\lambda^{(i)}})} \right|^2$$

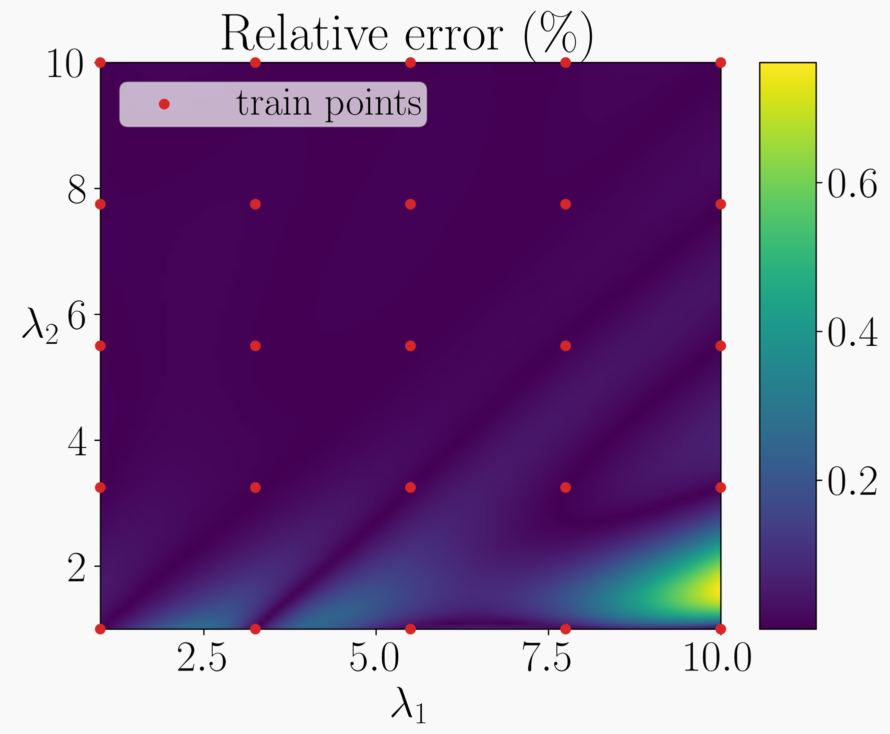
Training process



Ex: Diffusion with two parameters ($\dim \mathbb{U}_n = 4$)

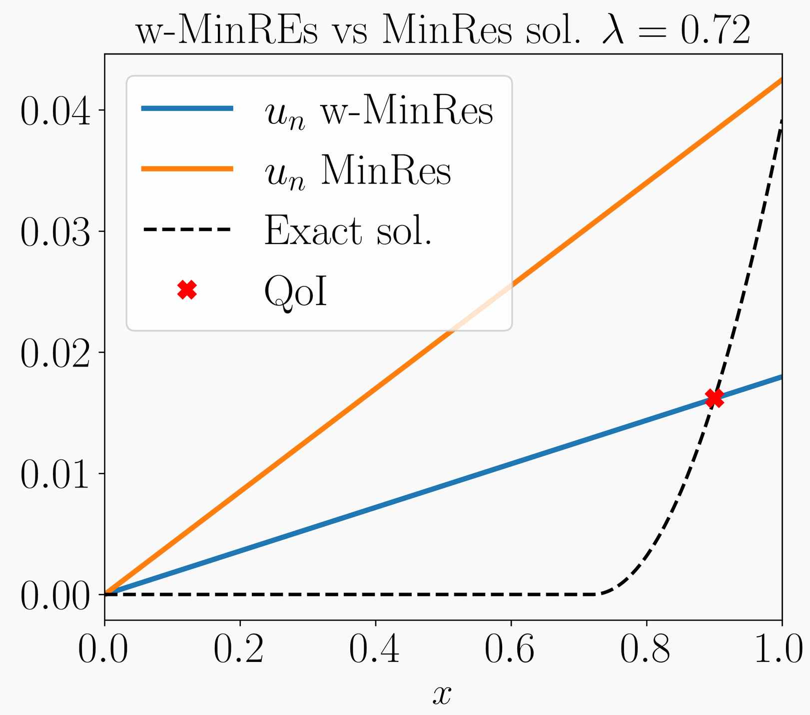
$$\begin{cases} -(a(x)u')' = 1, \text{ in } (0,1), \\ u(0) = u(1) = 0, \end{cases} \quad \text{with} \quad a(x) = \begin{cases} \lambda_1, & \text{if } x \leqslant 0.5, \\ \lambda_2, & \text{if } x > 0.5. \end{cases} \quad \text{and} \quad q(u) = u(0.6)$$

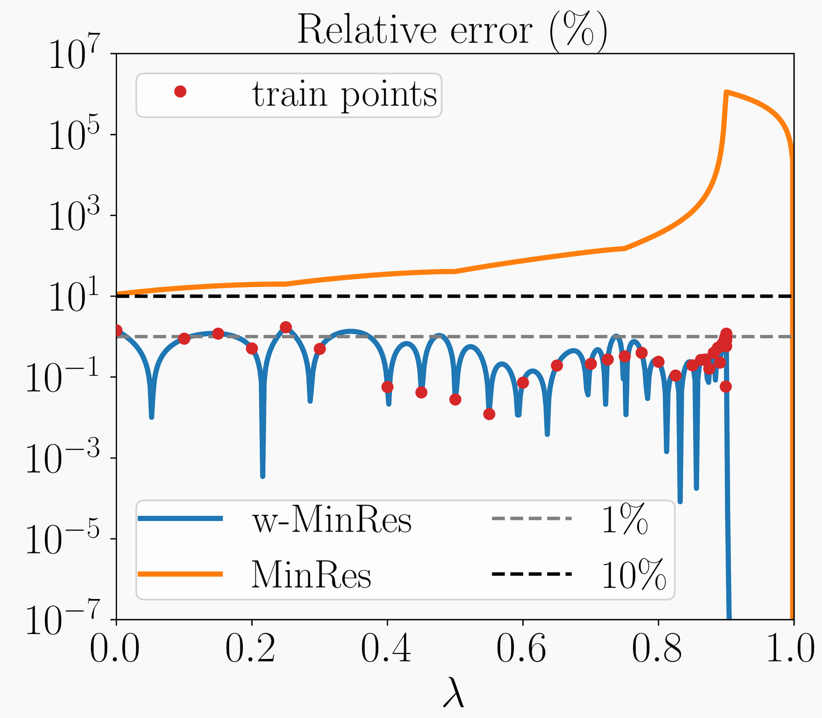




Ex: Advection with parametric rhs ($\dim \mathbb{U}_n = 1$)

$$\begin{cases} u' = f_\lambda, \text{ in } (0,1), \\ u(0) = 0, \end{cases} \quad \text{with} \quad f_\lambda(x) = \begin{cases} 0, & \text{if } x \leqslant \lambda, \\ x - \lambda, & \text{if } x > \lambda. \end{cases} \quad \text{and} \quad q(u) = u(0.9)$$





References

[1] I. Brevis, I. Muga, and K. G. van der Zee, *A machine-learning minimal-residual (ML-MRes) framework for goal-oriented finite element discretizations*, Comput. Math. Appl., 95 (2021), pp. 186–199.

[2] I. Brevis, I. Muga, and K. G. van der Zee, *Neural control of discrete weak formulations: Galerkin, least-squares and minimal-residual methods with quasi-optimal weights*, Comput. Methods Appl. Mech. Engrg., 402 (2022), p. 115716.