A deep first-order system least squares method for solving elliptic PDEs

F. M. Bersetche[1] , J. P. Borthagaray[2]

1. Departamento de Matemática, Universidad de Buenos Aires (Argentina)
2. Centro de Matemática, Universidad de la República (Uruguay)

October 6, 2022

Problem

- $\Omega \subset \mathbb{R}^d$ open domain, Γ_D, $\Gamma_N \subset \partial\Omega$.
- Find u such that

$$\begin{cases} -\mathrm{div}(A\nabla u) + Bu = f & \text{in } \Omega \\ u = g_D & \text{on } \Gamma_D \\ A\nabla u \cdot \nu = g_N & \text{on } \Gamma_N, \end{cases}$$

- $A\colon \Omega \to \mathbb{R}^{d\times d}$ symmetric and uniformly positive definite.
- B the linear operator of linear order (at most). For instance $Bu = \mathrm{div}(\beta u)$, $Bu = \beta \cdot \nabla u + \gamma u$)

.

We rewrite the problem as: Find $u : \Omega \to \mathbb{R}$ and $\phi : \Omega \to \mathbb{R}^d$, such that

$$\begin{cases} \phi - A\nabla u = 0 & \text{in } \Omega \\ \text{div}(\phi) + Bu + f = 0 & \text{in } \Omega \\ u = g_D & \text{on } \Gamma_D \\ \phi \cdot \nu = g_N & \text{on } \Gamma_N. \end{cases}$$

$$\begin{cases} \phi - A\nabla u = 0 & \text{in } \Omega \\ \operatorname{div}(\phi) + Bu + f = 0 & \text{in } \Omega \\ u = g_D & \text{on } \Gamma_D \\ \phi \cdot \nu = g_N & \text{on } \Gamma_N. \end{cases}$$

We define:

$$\mathcal{A} := \{(u, \phi) \in H^1(\Omega) \times H(\operatorname{div}; \Omega) \colon u = g_D \text{ on } \Gamma_D,\ \phi \cdot \nu = g_N \text{ on } \Gamma_N\}$$

$$\mathcal{L}(u, \phi) := \|\phi - A\nabla u\|_0^2 + \|\operatorname{div}(\phi) + Bu + f\|_0^2$$

We aim to find:

$$\arg \min_{(u,\phi)\in\mathcal{A}} \mathcal{L}(u,\phi)$$

The problem is well posed: There exist an unique minimizer and is a solution of the PDE.

- Cai, Z., Lazarov, R., Manteuffel, T. A., and McCormick, S. F. First-order system least squares for second-order partial differential equations: Part I. SIAM Journal on Numerical Analysis (1994)

Method

Idea: To discretize the space $\mathcal{A}$ (using neural networks) and discretize $\mathcal{L}$ (by means of Montecarlo integration)

Method: Discretization of $\mathcal{A}$

In order to impose the boundary conditions, we consider the auxiliary functions d_D, d_N, G_D, G_N, y n, defined as:

- d_D and d_N smooth distances to Γ_D y Γ_N respectively.
- G_D y G_N smooth functions with $G_D\big|_{\Gamma_D} = g_D$ y $G_N\big|_{\Gamma_N} = g_N$
- $n : \Omega \to \mathbb{R}^d$ a smooth function with $n\big|_{\Gamma_N} = \nu$ (ν the exterior normal vector)

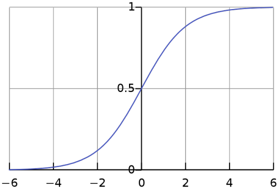
And we consider neural networks $v_\Theta : \mathbb{R}^d \to \mathbb{R}$ y $\psi_\Theta : \mathbb{R}^d \to \mathbb{R}^d$, with parameters Θ. (no particular architecture)

Discretization of $\mathcal{A}$

For example, we can use a (shallow)neural network: $v : \mathbb{R}^d \to \mathbb{R}$

$$v(x) = \sum_{i=1}^{n} \beta_i \sigma\Big(\sum_{j=1}^{d} \alpha_{ji} x_j + c_j \Big)$$

Where $\sigma : \mathbb{R} \to \mathbb{R}$ is a non-linear activation function (for instance a sigmoid function).



- d_D and d_N smooth distances to Γ_D y Γ_N respectively.
- G_D y G_N smooth functions with $G_D\big|_{\Gamma_D} = g_D$ y $G_N\big|_{\Gamma_N} = g_N$
- $n : \Omega \to \mathbb{R}^d$ a smooth function with $n\big|_{\Gamma_N} = \nu$ (ν the exterior normal vector)

And we consider neural networks $v_\Theta : \mathbb{R}^d \to \mathbb{R}$ y $\psi_\Theta : \mathbb{R}^d \to \mathbb{R}^d$, with parameters Θ.

$$\mathcal{A}_m := \left\{ (u_\Theta, \phi_\Theta) : u_\Theta = d_D v_\Theta + G_D \text{ and } \phi_\Theta = \psi_\Theta + \left(G_N - \frac{\psi_\Theta \cdot n}{1 + d_N} \right) n \right\}$$

Method: Discretization of the loss functional

We sample N random points $\{x_\Omega\} \subset \Omega$ (with uniform distribution) and we define:

$$\mathcal{L}_N(u, \phi) := \frac{|\Omega|}{N} \sum_{x \in \{x_\Omega\}} \big(\phi(x) - A\nabla u(x)\big)^2 + \big(\operatorname{div} \phi(x) + Bu(x) + f(x)\big)^2.$$

Method

Then we estimate

$$\arg\min_{(u,\phi)\in\mathcal{A}} \mathcal{L}(u,\phi)$$

with

$$\arg\min_{(u,\phi)\in\mathcal{A}_m} \mathcal{L}_N(u,\phi)$$

- Sirignano, J., and Konstantinos S., "DGM: A deep learning algorithm for solving partial differential equations." Journal of computational physics. (2018)
- Weinan, E., and Bing Y. "The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems." Communications in Mathematics and Statistics. (2018)
- Cai, Z., et al. "Deep least-squares methods: An unsupervised learning-based numerical method for solving elliptic PDEs." Journal of Computational Physics (2020)
- Liu, M., Cai, Z., and Chen, J. "Adaptive two-layer ReLU neural network (part I and II)" Computers and Mathematics with Applications (2022)

Contribution

- Strong imposition of boundary conditions (including Neumann conditions), combined with first order formulation.
- Convergence of the method (taking into account the discretization of the loss function) and generalization to other methods.

Convergence

We consider the discretization of the space $\mathcal{A}$ and the discretization of the functional $\mathcal{L}$.

- Siegel, J., Hong, Q., Jin, X., Hao, W., Xu, J. (2022). Greedy Training Algorithms for Neural Networks and Applications to PDEs. arXiv preprint arXiv:2107.04466.
- Zerbinati, U. (2022). PINNs and GaLS: An Priori Error Estimates for Shallow Physically Informed Neural Network Applied to Elliptic Problems. arXiv preprint arXiv:2202.01059.

Convergence

- Consider $\omega_1, ..., \omega_K$ Borel subsets of $\overline{\Omega}$,
- each ω_i furnished with a finite Radon measure μ_i,
- some given functions $f_1, ..., f_{n_f}$ with $f_i : \Omega \to \mathbb{R}$.

Given some integrable functions $F_i : \mathbb{R}^{n_\gamma n + n_f + d} \to \mathbb{R}$, $1 \leq i \leq K$, we define the loss functional:

$$\mathcal{L}(q) := \sum_{i=1}^{K} \int_{\omega_i} F_i(D^{\alpha_1} q, \ldots, D^{\alpha_{n_\gamma}} q, f_1, \ldots, f_{n_f}, x) \, d\mu_i,$$

Convergence

Given $q_\Theta \in \mathcal{A}_m$, we define $G_i(\Theta, x) : \mathbb{R}^n \times \Omega \to \mathbb{R}$ as

$$G_i(\Theta, x) := F_i(D^{\alpha_1} q_\Theta, \dots, D^{\alpha_{n_\gamma}} q_\Theta, f_1, \dots, f_{n_f}, x)$$

$$\mathcal{L}(q_\Theta) = \sum_{i=1}^{K} \int_{\omega_i} G_i(\Theta, x)\, d\mu_i,$$

As before, we sample random points $\{x_{\omega_i}\} \subset \omega_i$ (with a distribution according to μ_i) and we define:

$$\mathcal{L}_N(q_\Theta) := \sum_{i=1}^{K} \frac{|\omega_i|}{\#\{x_{\omega_i}\}} \sum_{x \in \{x_{\omega_i}\}} G_i(\Theta, x)$$

Convergence

Idea:

$$\mathcal{L}_N \xrightarrow{\Gamma} \mathcal{L} \text{ on } \mathcal{A}_m, \text{ with } N \to \infty$$

$$\arg \min_{(u,\phi)\in\mathcal{A}_m} \mathcal{L}(q) \to \arg \min_{(u,\phi)\in\mathcal{A}} \mathcal{L}(q), \text{ with } m \to \infty$$

Convergence

$$\mathcal{L}(q_\Theta) = \sum_{i=1}^{K} \int_{\omega_i} G_i(\Theta, x)\, d\mu_i,$$

(H1) The map $\mathbb{R}^m \mapsto (\mathcal{A}_m, \|\cdot\|_{\mathcal{A}})$ with $\Theta \mapsto q_\Theta$ is continuous.

(H2) For all $1 \leq i \leq K$ and every convergent sequence $\{q_{\Theta_n}\}_{n\in\mathbb{N}} \subset \mathcal{A}_m$, with $q_{\Theta_n} \to q_\Theta \in \mathcal{A}_m$ with respect to the $\mathcal{A}$-norm, there exists a subsequence $\{q_{\Theta_{n_j}}\}_{j\in\mathbb{N}}$ such that $G_i(\Theta_{n_j}, x) \to G_i(\Theta, x)$ μ_i-almost everywhere.

(H3) For every $R > 0$, there are functions $s_i \in L^1_{\mu_i}(\omega_i)$ such that $|G_i(\Theta, x)| \leq s_i(x)$ for all $1 \leq i \leq K$, for all $\Theta \in B(0, R)$, and μ_i-almost every $x \in \omega_i$.

(H4) The loss function $\mathcal{L}$ has a unique minimizer $q_0 \in \mathcal{A}$, and at least one minimizer in $\mathcal{A}_m$ for all $m \in \mathbb{N}$.

(H5) Let $\{q_m\}_{m\in\mathbb{N}} \subset \mathcal{A}$ be a sequence of minimizers of $\mathcal{L}$, namely, $q_m \in \arg\min_{q\in\mathcal{A}_m} \mathcal{L}(q)$ for all $m \in \mathbb{N}$. Then, $\|q_m - q_0\|_{\mathcal{A}} \to 0$ as $m \to \infty$.

Convergence

- Sirignano, J., and Konstantinos S., "DGM: A deep learning algorithm for solving partial differential equations." Journal of computational physics. (2018)
- Weinan, E., and Bing Yu. "The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems." Communications in Mathematics and Statistics. (2018)

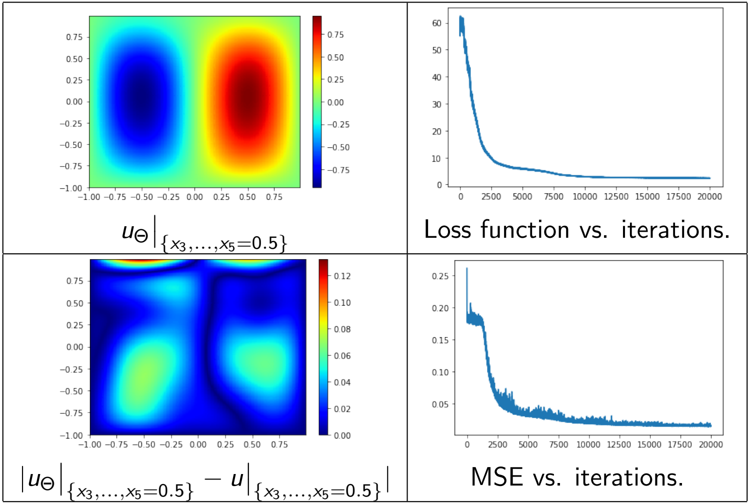
Numerical example

$$\begin{cases} -\Delta u = \prod_{i=1}^{d-1} \sin(k\pi x_i) \left((d-1)k^2\pi^2(1-x_d^2)+2\right) & u \in \Omega, \\ u = 0 & u \in \partial\Omega \setminus \Gamma_N \\ \nabla u \cdot \eta = -2\prod_{i=1}^{d-1} \sin(k\pi x_i) & u \in \Gamma_N, \end{cases}$$

with $\Omega = \{x \in \mathbb{R}^d : \; -1 < x_1, ..., x_d < 1\}$, $\Gamma_N = [-1,1]^{d-1} \times \{1\}$, y $k \in \mathbb{N}$.
Exact solution:

$$u = \prod_{i=1}^{d-1} \sin(k\pi x_i)(1-x_d^2).$$

Numerical example



$u_\Theta\big|_{\{x_3,\ldots,x_5=0.5\}}$

Loss function vs. iterations.

$\left|u_\Theta\big|_{\{x_3,\ldots,x_5=0.5\}} - u\big|_{\{x_3,\ldots,x_5=0.5\}}\right|$

MSE vs. iterations.

Figure: d = 5, 5656 DOF, MSE = 0.019, 20000 optimization steps.

THANKS FOR YOUR ATTENTION!