Fast Iterative Solver for Neural Network Method: II. 1D General Elliptic PDEs

César Herrera[1]
Zhiqiang Cai[1] Anastassia Doktorova[1] Robert D. Falgout[2]

[1]Department of Mathematics, Purdue University
[2]Lawrence Livermore National Laboratory

The Finite Element Circus
October 2024

Table of Contents

- C^0 **piecewise linear functions on a fixed mesh in** $[0,1]$**:**

$$S_1^0(\Delta) = \left\{ \sum_{i=1}^{n} c_i \phi_i(x) : c_i \in \mathbb{R} \right\},$$

where

$$\phi_i(x) = \begin{cases} (x - x_{i-1})/(x_i - x_{i-1}), & x \in (x_{i-1}, x_i), \\ (x_{i+1} - x)/(x_{i+1} - x_i), & x \in (x_i, x_{i+1}), \\ 0, & \text{otherwise.} \end{cases}$$

- C^0 **piecewise linear functions on a moving mesh in** $[0,1]$**:**

$$S_1^0(n) = \left\{ \sum_{i=1}^{n} c_i \phi_i(x; x_{i-1}, x_i, x_{i+1}) : c_i \in \mathbb{R}, , x_i \in [0,1] \right\}$$

- **One-dimensional shallow neural network:**

$$\mathcal{M}_n([0,1]) = \mathcal{M}_n(I) = \left\{ c_{-1} + \sum_{i=0}^{n} c_i \sigma(x - b_i) : c_i \in \mathbb{R}, b_i \in [0,1] \right\},$$

where $\sigma(t) = \max\{t, 0\}$.

- **Important relation[1]:**

$$S_1^0(n) \subset \mathcal{M}_n(I) \subset S_1^0(n+1)$$

[1] I. Daubechies et al. "Nonlinear Approximation and (Deep) ReLU Networks". English (US). in: *Constructive Approximation* 55.1 (Feb. 2022), pp. 127–172. ISSN: 0176-4276. DOI: 10.1007/s00365-021-09548-z.

Table of Contents

Shallow Neural Network

Let

$$\mathcal{M}_n(I) \;=\; \left\{ c_{-1} + \sum_{i=0}^{n} c_i \sigma(x - b_i) \;:\; c_i \in \mathbb{R}, 0 \le b_i \le 1, b_i < b_{i+1} \right\}$$

Given $\mathbf{b} = (b_0, b_1, \ldots, b_n)^T$, let

$$\mathbf{H}(x) := (\sigma'(x - b_0), \sigma'(x - b_1), \ldots, \sigma'(x - b_n))^T.$$

Coefficient matrix: $A(\mathbf{b}) = \int_0^1 \mathbf{H}(x)\mathbf{H}(x)^T dx$

Mass matrix

Let

$$\Sigma(x) := (\sigma(x - b_0), \sigma(x - b_1), \ldots, \sigma(x - b_n))^T.$$

Mass matrix: $M(\mathbf{b}) = \int_0^1 \Sigma(x)\Sigma(x)^T dx$

Lemma

The condition number of the mass matrix $M(\mathbf{b})$ is bounded above by $\mathcal{O}\left(n/h_{min}^3\right)$.

Table of Contents

1D Diffusion Reaction Problem

We consider the 1D problem

$$\begin{cases} -u''(x) + u(x) = f(x), & x \in I = (0,1), \\ u(0) = \alpha, \quad u(1) = \beta \end{cases}$$

Ritz formulation: find $u \in H^1(I)$ such that

$$u = \underset{\substack{v \in H^1(I) \\ v(0)=\alpha, v(1)=\beta}}{\arg\min} \left\{ \frac{1}{2}\int_0^1 (v'(x))^2 dx + \frac{1}{2}\int_0^1 (v(x))^2 dx - \int_0^1 f(x)v(x)dx \right\}$$

Modified Ritz formulation

Given $\gamma > 0$, let $J : H^1(I) \to \mathbb{R}$ be the modified energy functional given by

$$J(v) = \frac{1}{2}\int_0^1 (v'(x))^2 dx + \frac{1}{2}\int_0^1 (v(x))^2 dx - \int_0^1 f(x)v(x)dx + \frac{\gamma}{2}(v(b) - \beta)^2$$

Ritz neural network approximation: find $u_n(x) \in \mathcal{M}_n(I)$ such that

$$J(u_n) = \min_{\substack{v \in \mathcal{M}_n(I) \\ v(0)=\alpha}} J(v)$$

Error estimate

Proposition

Let u be the exact solution and $u_n \in \mathcal{M}_n(I)$ be the Ritz neural network approximation. There exists a constant C depending on u such that

$$\|u - u_n\|_a \leq C \left(n^{-1} + \gamma^{-1/2} \right),$$

where $\|v\|_a^2 = \int_0^1 (v'(x))^2 dx + \int_0^1 (v(x))^2 dx + \gamma (v(1))^2$.

Systems of algebraic equations

Let

$$u_n = u_n(x) = u_n(x; \mathbf{c}, \mathbf{b}) = \alpha + \sum_{i=0}^{n} c_i \sigma(x - b_i)$$

be a solution of the previous minimization problem. Then the linear and nonlinear parameters

$$\mathbf{c} = (c_0, \ldots, c_n)^T \quad \text{and} \quad \mathbf{b} = (b_0, \ldots, b_n)^T$$

satisfy the following system of algebraic equations

$$\nabla_{\mathbf{c}} J(u_n) = \nabla_{\mathbf{c}} J(\mathbf{c}, \mathbf{b}) = \mathbf{0} \quad \text{and} \quad \nabla_{\mathbf{b}} J(u_n) = \nabla_{\mathbf{b}} J(\mathbf{c}, \mathbf{b}) = \mathbf{0}.$$

Linear parameters $\mathbf{c}$

The equation $\nabla_{\mathbf{c}} J(\mathbf{c}, \mathbf{b}) = 0$ has the form

$$\left(A(\mathbf{b}) + M(\mathbf{b}) + \gamma \mathbf{d}\mathbf{d}^T \right) \mathbf{c} = \mathbf{f}(\mathbf{b}) + \gamma(\beta - \alpha)\mathbf{d}.$$

where

- $A(\mathbf{b})$ is the coefficient matrix
- $M(\mathbf{b})$ is the mass matrix
- $\mathbf{f}(\mathbf{b}) = \left(\int_0^1 f(x)\sigma(x - b_0)dx, \ldots, \int_0^1 f(x)\sigma(x - b_n)dx \right)^T$
- $\mathbf{d} = (b - b_0, \ldots, b - b_n)^T$

Linear parameters $\mathbf{c}$

We derived factorizations of the form

$$M(\mathbf{b})^{-1} = T_1^T T_2^{-1} T_1,$$
$$(M(\mathbf{b}) + A(\mathbf{b}))^{-1} = T_1^T (T_2 + T_3)^{-1} T_1,$$

where T_1, T_2, T_3 are tridiagonal matrices.
Hence, the system of linear equations

$$\left(A(\mathbf{b}) + M(\mathbf{b}) + \gamma \mathbf{d}\mathbf{d}^T\right)\mathbf{c} = \mathbf{f}(\mathbf{b}) + \gamma(\beta - \alpha)\mathbf{d}.$$

can be solved in $30(n+1)$ operations.

Nonlinear parameters **b**

Lemma

The Hessian matrix $\nabla^2_{\mathbf{b}} J(\mathbf{c}, \mathbf{b})$ has the form

$$\mathcal{H}(\mathbf{c}, \mathbf{b}) + \gamma \mathbf{c}\mathbf{c}^T = D(\mathbf{c})D(\mathbf{g}) + D(\mathbf{c})A(\mathbf{b})D(\mathbf{c}) + \mathbf{c}\mathbf{c}^T,$$

where $D(\mathbf{g}) = \text{diag}(u_n(b_0) - f(b_0), \ldots, u_n(b_n) - f(b_n))$ and $D(\mathbf{c}) = \text{diag}(c_0, c_1, \ldots, c_n)$.

Useful factorization:

$$\mathcal{H}(\mathbf{c}, \mathbf{b})^{-1} = \left(I + D(\mathbf{c})^{-1} A(\mathbf{b})^{-1} D(\mathbf{g})\right)^{-1} D(\mathbf{c})^{-1} A(\mathbf{b})^{-1} D(\mathbf{c})^{-1}.$$

Table of Contents

A Damped Block Newton (dBN) Method

Let $\left(\mathbf{c}^{(k)}, \mathbf{b}^{(k)}\right)$ be the previous iterate. We then compute the current state $\left(\mathbf{c}^{(k+1)}, \mathbf{b}^{(k+1)}\right)$ by doing the following:

(i) Compute the current linear parameters $\mathbf{c}^{(k+1)}$ solving

$$\nabla_{\mathbf{c}} J(\mathbf{c}, \mathbf{b}^{(k)}) = 0.$$

(ii) Set the search direction

$$\mathbf{p}^{(k)} = -\mathcal{H}(\mathbf{c}^{(k+1)}, \mathbf{b}^{(k)})^{-1} \nabla_{\mathbf{b}} J(\mathbf{c}^{(k+1)}, \mathbf{b}^{(k)}).$$

(iii) Compute the stepsize η_k

$$\eta_k = \underset{\eta \in \mathbb{R}_+}{\operatorname{argmin}} \; J(\mathbf{c}^{(k+1)}, \mathbf{b}^{(k)} + \eta \mathbf{p}^{(k)}).$$

Set the current nonlinear parameters by

$$\mathbf{b}^{(k+1)} = \mathbf{b}^{(k)} + \eta_k \mathbf{p}^{(k)}.$$

Table of Contents

Numerical Experiments

The first test problem involves the function

$$u(x) = x\left(\exp\left(-\frac{(x-\frac{1}{3})^2}{0.01}\right) - \exp\left(-\frac{4}{9\times 0.01}\right)\right)$$

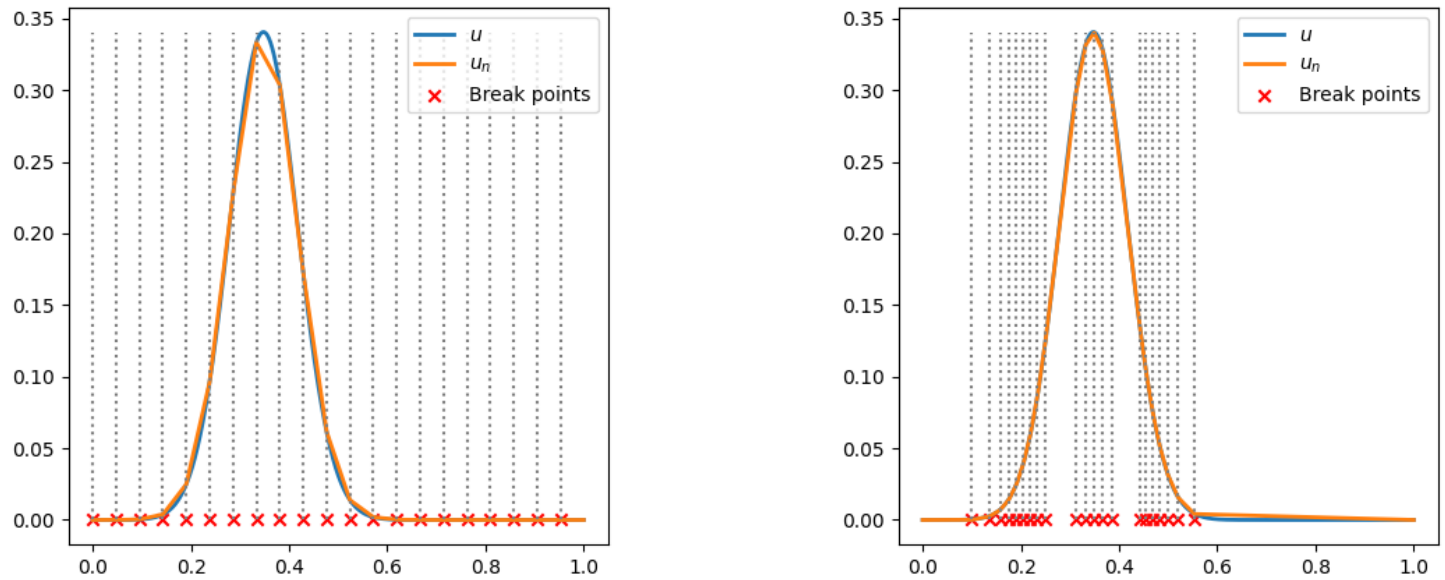


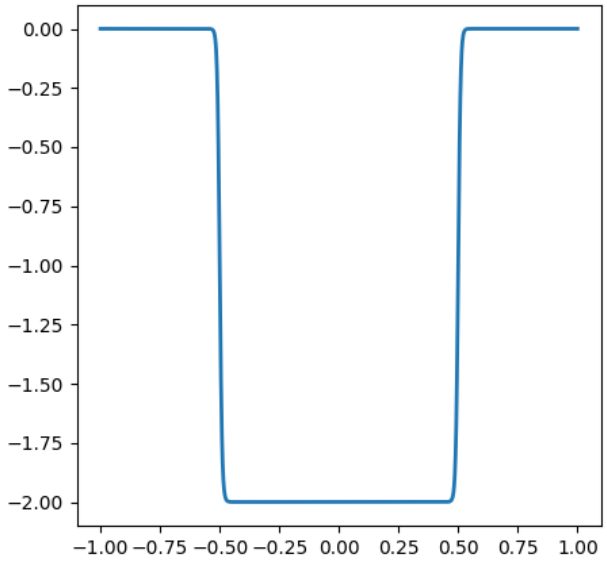
Figure: $u(x)$ approximated by NN. Left: 21 uniform breakpoints, $e_n = 0.238$. Right: optimized NN model with 21 breakpoints, 500 iterations, $e_n = 0.101$.

Singularly Perturbed Reaction-Diffusion Equation

$$\begin{cases} -\varepsilon^2 u''(x) + u(x) = f(x), & x \in I = (-1, 1), \\ u(-1) = u(1) = 0. \end{cases}$$

Figure: Graph of the test function $u(x)$ for $f(x) =$

$$-2\left(\varepsilon - 4x^2 \tanh\left(\tfrac{1}{\varepsilon}(x^2 - \tfrac{1}{4})\right)\right)\left(\frac{1}{\cosh\left(\frac{1}{\varepsilon}(x^2 - \frac{1}{4})\right)}\right)^2 + \tanh\left(\tfrac{1}{\varepsilon}(x^2 - \tfrac{1}{4})\right) - \tanh\left(\tfrac{3}{4\varepsilon}\right)$$



Singularly Perturbed Reaction-Diffusion Equation

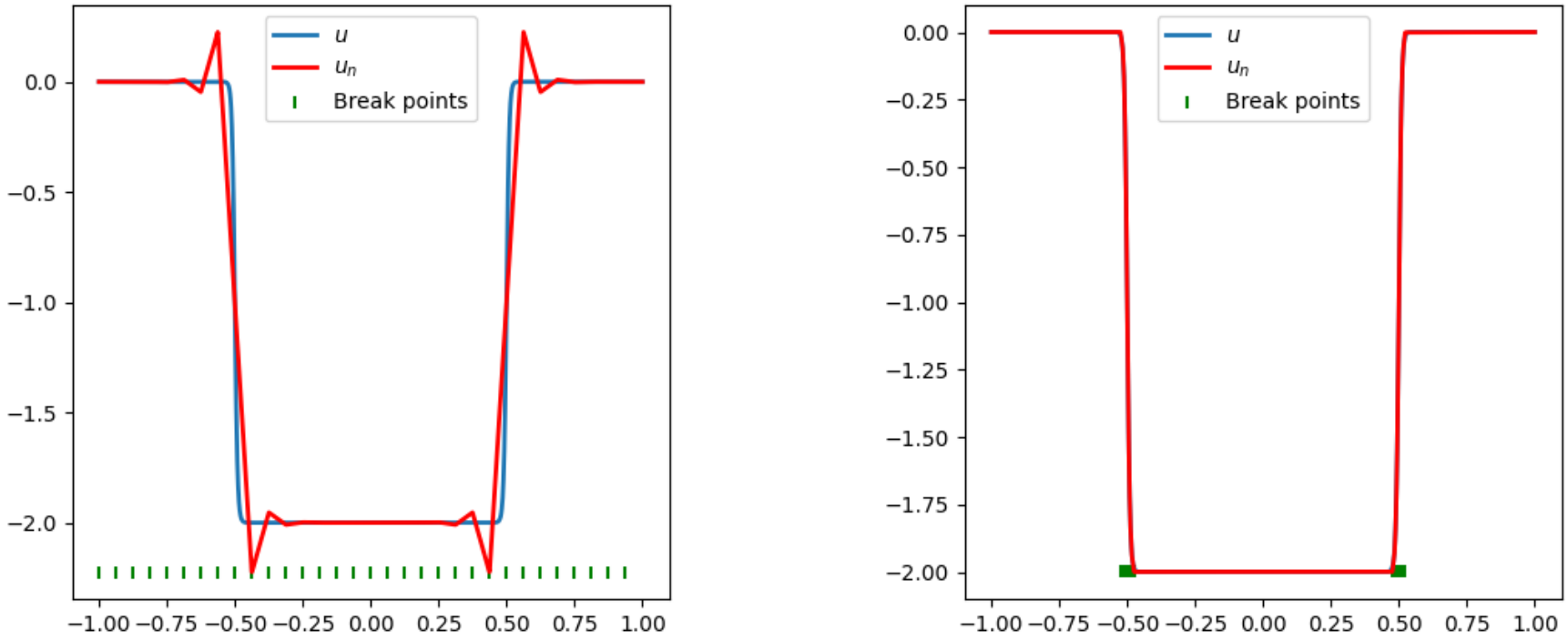
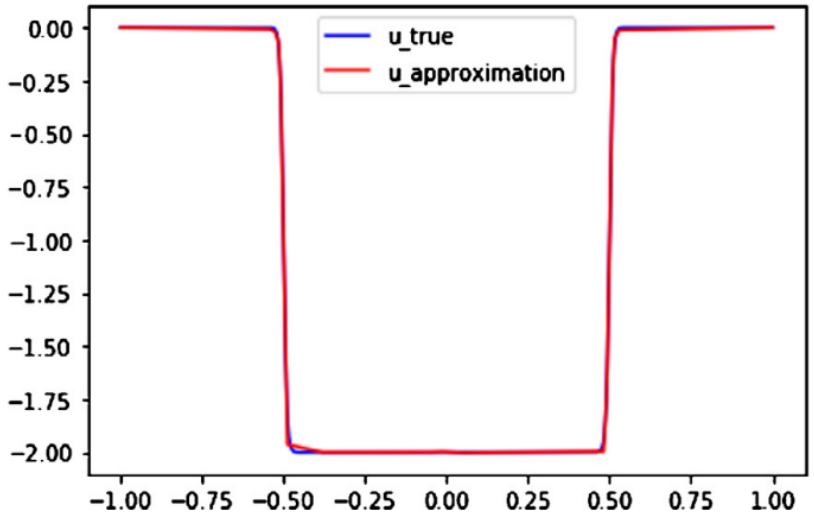


Figure: $u(x)$ approximated by NN. $\varepsilon = 0.01$ Left: 32 uniform breakpoints, $e_n = 0.889$. Right: optimized NN model with 32 breakpoints, 100 iterations, $e_n = 0.090$.

Singularly Perturbed Reaction-Diffusion Equation



(a) FOSLS u with Leaky ReLU

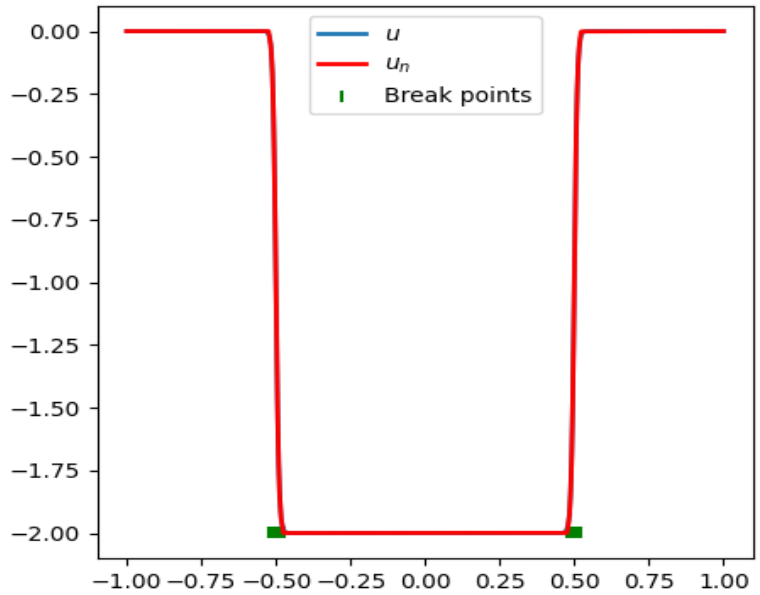


Figure: $u(x)$ approximated by NN. Left: Scientific Machine Learning approach[2],1-32-32-24-24-1, 2962 parameters, about 14 hours. Right: dBN method, 1-32-1, 64 parameters, 2 minutes (100 iterations).

[2] Zhiqiang Cai et al. "Deep least-squares methods: An unsupervised learning-based numerical method for solving elliptic PDEs". In: *Journal of Computational Physics* 420 (2020), p. 109707. ISSN: 0021-9991.

arXiv Preprint

arXiv > math > arXiv:2407.01496

Search... Help | Adv

Mathematics > Numerical Analysis

[Submitted on 1 Jul 2024]

Fast Iterative Solver For Neural Network Method: II. 1D Diffusion-Reaction Problems And Data Fitting

Zhiqiang Cai, Anastassia Doktorova, Robert D. Falgout, César Herrera

This paper expands the damped block Newton (dBN) method introduced recently in [4] for 1D diffusion-reaction equations and least-squares data fitting problems. To determine the linear parameters (the weights and bias of the output layer) of the neural network (NN), the dBN method requires solving systems of linear equations involving the mass matrix. While the mass matrix for local hat basis functions is tri-diagonal and well-conditioned, the mass matrix for NNs is dense and ill-conditioned. For example, the condition number of the NN mass matrix for quasi-uniform meshes is at least $\mathcal{O}(n^4)$. We present a factorization of the mass matrix that enables solving the systems of linear equations in $\mathcal{O}(n)$ operations. To determine the non-linear parameters (the weights and bias of the hidden layer), one step of a damped Newton method is employed at each iteration. A Gauss-Newton method is used in place of Newton for the instances in which the Hessian matrices are singular. This modified dBN is referred to as dBGN. For both methods, the computational cost per iteration is $\mathcal{O}(n)$. Numerical results demonstrate the ability dBN and dBGN to efficiently achieve accurate results and outperform BFGS for select examples.

Subjects: **Numerical Analysis (math.NA)**; Machine Learning (cs.LG)
MSC classes: 65K10, 65F05
Cite as: arXiv:2407.01496 **[math.NA]**
(or arXiv:2407.01496v1 **[math.NA]** for this version)
https://doi.org/10.48550/arXiv.2407.01496

Thanks!