

Machine Learning-Based Preconditioner to Solve Poisson Equation

Ekaterina Chekmeneva[1], Tatyna Khachova[2], and Vadim Lisitsa[3(✉)]

[1] Novosibirsk State University, Pirogova st. 2, 630090 Novosibirsk, Russia
e.chekmeneva@g.nsu.ru
[2] Institute of Petroleum Geology and Geophysics SB RAS, Koptug ave. 3, 630090 Novosibirsk, Russia
[3] Institute of Mathematics SB RAS, Koptug ave. 4, 630090 Novosibirsk, Russia
lisitsavv@ipgg.sbras.ru

Abstract. In this paper, we present an attempt to construct a preconditioner based on the machine learning to solve Poisson equation. We use the Conjugate Gradient method. To precondition the algorithm we suggest approximating the inverse Laplace operator with using the U-Net. We consider the supervised learning where the vector of unknowns and right-hand sides are known; thus, we use the relative L^2 error as the loss function of the network training. We illustrate that U-Net with five convolutional layers provide insufficient accuracy of inverse Laplace operator approximation, so that the constructed conjugate gradient method stabilizes and possesses irreducible residual.

Keywords: Poisson equation · Conjugate gradient · preconditioner · Machine Learning

1 Introduction

The numerical solution of the Poisson equation in heterogeneous media has broad applications, including modeling: electric potential distribution in heterogeneous materials [10,25]; steady-state temperature fields [3]; species distribution governed by diffusion processes. Additionally, solving the Poisson equation is a critical step in fluid flow simulations, particularly in: multi-phase flows in open channels when projection-based methods are employed [11,12], multi-phase flows in porous media [14].

Upon discretization, the Poisson equation yields a system of linear algebraic equations (SLAEs) with a self-adjoint, positive-definite matrix—provided the original differential operator retains these properties. Such systems are efficiently solved using Krylov subspace methods, with the Conjugate Gradient (CG) method being particularly widespread [21]. However, the convergence rate of CG deteriorates with increasing matrix condition number, which grows with problem size and the contrast in material heterogeneity.

The research was supported by the Russian Science Foundation grant no. 22-11-00004-П.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2025 Workshops, LNCS 15888, pp. 376–387, 2026.
https://doi.org/10.1007/978-3-031-97596-7_25

To accelerate convergence, various preconditioning techniques have been developed:

- Algebraic preconditioners, including Jacobi, incomplete LU factorization [9, 13,21], and low-rank approximations of the matrix or its LU factors [24].
- Geometrically inspired methods, such as domain decomposition [18] and multigrid techniques [4,23], later generalized to algebraic formulations.
- Physics-based preconditioners, which approximate the inverse of a simplified Laplace operator. A key challenge is ensuring efficient inversion at each iteration [10].

The development of the machine and deep learning methods gave rise to the new direction of the preconditioners construction [1,15,16]. The general principle of the ML-based preconditioenrs design is the speed-up of the incomplete LU and Cholesky factorization construction or the approximation of the inverse operator directly. Following our previous works on physics-based preconditioners [10] we suggest focusing on the ML-based approximation of the inverse Laplace operator as preconditioner. The remaining of the paper has the following structure. We formulate the mathematical problem in Sect. 2. The principles of the ML-based preconditioner are discussed in Sect. 3. The numerical experiments are presented in Sect. 4.

2 Statement of the Problem

2.1 Mathematical Formulation

Assume the Poisson equation with variable coefficients stated in a rectangular domain $\Omega = [X_1, X_2] \times [Y_1, Y_2]$:

$$\nabla \cdot (\sigma(x, y) \nabla \psi(x, y)) = f(x, y). \tag{1}$$

Coefficient $\sigma(x, y)$ is assumed to be strictly positive and bounded. In this work, we consider the mixed boundary conditions, so that Dirichlet conditions are stated for $x = X_1$ and $x = X_2$, but Neumann conditions are stated for $y = Y_1$ and $y = Y_2$:

$$\begin{gathered} \psi(X_1, y) = \psi^1, \qquad \psi(X_2, y) = \psi^2, \\ \nabla \psi \cdot \vec{n}(x, Y_1) = 0, \, \nabla \psi \cdot \vec{n}(x, Y_2) = 0. \end{gathered} \tag{2}$$

In these notations, $f(\vec{x})$ is the right-hand side, ψ^1 and ψ^2 are boundary conditions.

To solve the stated problem we suggest using the conservative finite differences with harmonic averaging of the coefficient σ when used on the cells faces, as described in [10,17,22].

The discretized problem can be represented as a system of linear algebraic equations:

$$A\vec{\psi} = \vec{g}, \tag{3}$$

where matrix A is self-adjoin positive definite of the size $N_xN_y \times N_xN_y$, $\vec{\psi}$ is the vector representation of the unknowns, and $\vec{g}$ is the vector containing the discretized right-hand sides and the boundary conditions, N_x and N_y are the numbers of grid points in each spatial direction.

2.2 Conjugate Gradient Method with Preconditioner

Matrix A of the system (3) is symmetric and positive definite; thus, one may apply the Conjugate Gradient method to solve the system. However, convergence rate of the method strongly depends on the matrix condition number [21]. The condition number of the system may be high; moreover, it increases with the problem size and the contrast of the coefficient σ. To decrease the condition number and to speed up the convergence rate the preconditoning is used; so that modified problem

$$B^{-1/2}AB^{-1/2}B^{1/2}\vec{\psi} = B^{-1/2}\vec{g}$$

is solved. Operator B is called the preconditioner. The new matrix $B^{-1/2}AB^{-1/2}$ is also symmetric and positive definite if B possesses the same properties. It is clear, that construction of $B^{-1/2}$ may be troublesome, so a simple algebraic manipulations allow to rewrite the CG algorithm for preconditionnered system in the following form:

1. $\vec{r}_0 = \vec{g} - A\vec{\psi}_0$, $\vec{p}_0 = B^{-1}\vec{r}_0$, $\vec{q}_0 = \vec{p}_0$;
   do until converge
2. $\alpha_j = \frac{(\vec{q}_j, \vec{r}_j)}{(A\vec{p}_j, \vec{p}_j)}$;
3. $\vec{\psi}_{j+1} = \vec{\psi}_j + \alpha_j\vec{p}_j$
4. $\vec{r}_{j+1} = \vec{r}_j - \alpha_j A\vec{p}_j$, $\vec{q}_{j+1} = B^{-1}\vec{r}_{j+1}$;
5. $\beta_j = \frac{(\vec{q}_{j+1}, \vec{r}_{j+1})}{(\vec{q}_j, \vec{r}_j)}$;
6. $\vec{p}_{j+1} = \vec{q}_{j+1} + \beta_j\vec{p}_j$

In these notations $\vec{\psi}$ is the solution vector, $\vec{g}$ is the original right-hand side, $\vec{r}$ is the residual vector, and $\vec{q}$, $\vec{p}$ are auxiliary vectors.

The most intense part of the algorithm is the computation of the preconditionner action $B^{-1}\vec{r}_{j+1}$. There are several different ways to construct the preconditioner B and compute $B^{-1}\vec{r}_{j+1}$. The most universal approach is based on the algebraic principles. For example, the Jacobi preconditioner requires inversion of the main diagonal of the original matrix. It is a fast but not very efficient approach. Incomplete LU and Cholesky decomposition-based preconditioners improves the convergence but require storage of factors, which leads to high memory consumption. An efficient approach to precondition Poisson solvers is the use of multi-grid methods [21,23]. However, multi-grid efficiency strongly depends on the contrast in the coefficients σ. In fact, the ideal preconditioner is $B = A$. In this case, the process would converge in one iteration. However, the inversion of matrix B is troublesome. So, the set of approaches was suggested,

where matrix $B \approx A$ in some sense, but to be relatively easy to invert. In particular, the operator corresponding to a homogeneous model $\sigma = const$ can be considered, after that it is inverted using pseudo-spectral method [2,10,19], or the spectral 1D matrix decomposition may be applied. This approach strongly improves the convergence of the CG for Poisson equation, however its efficiency degenerates with the increase of the coefficient contrast. Moreover, use of the pseudo spectral and spectral approach requires intense computations at each iteration. To speed up the computation of the preconditioner action; i.e., $B^{-1}\vec{r}_{j+1}$ for any matrix B we suggest using the Machine Learning methods.

3 ML-Based Preconditioner

The main assumption of the suggested approach is that one needs to solve Poisson equation for fixed model $\sigma(x, y)$ for numerous different right-hand sides. In this case, it is reasonable to train neural network that approximate $B^{-1}\vec{r}$ and then apply it multiple times as a preconditionner. In this research we suggest constructing the preconditioner to approximate the inverse operator, rather than reducing matrix condition number or matrix norm directly, as suggested in [16]. So, we need to construct a map

$$\mathcal{F}[\vec{r}] = \vec{q}_F \approx B^{-1}\vec{r}.$$

In this research, we use the U-net [20] which is a convolution neural network being widely used for image processing, seismic data processing [5–8]. In particular the U-net with five convolutional layers is constructed as presented in Fig. 1. To train the U-net we considered a series of images $U_1, ..., U_m$ of the size 128×128 pixels. These images can be represented as vectors $\vec{u}_1, ..., \vec{u}_m$ of the length of 128^2. We computed the action of the Laplace operator on the dataset to get the second series of vectors $\vec{v}_1, ..., \vec{v}_m$, so that $\vec{v}_k = A\vec{u}_k$ and represented them as images $V_1, ..., V_m$. We trained the U-net to map the set $V_1, ..., V_m$ to the set $U_1, ..., U_m$, thus we train it to invert matrix A. The loss function was

$$Loss = \sum_{k=1}^{m} \left[\lambda_{rel} \frac{\|\vec{u}_k - \mathcal{F}[\vec{u}_k]\|_2^2}{\|\vec{u}_k\|_2^2} + \lambda_{pos} \max\{0, -(\vec{u}_k, \mathcal{F}[\vec{u}_k])\} \right],$$

where λ_{rel} and λ_{pos} are weights, and $(,)$ denotes standard inner product of vectors. The second term is sued to enforce the mapping $\mathcal{F}[\vec{u}_k]$ being positive definite, because it is crucial property of the preconditioner of the conjugate gradient method.

First, we trained the network using a set of 768 pairs of random pictures as a training dataset and 256 pairs as a validation dataset. We applied the Laplace operator to those pictures and trained the network to recover the images. An example of original picture, the result of Laplace operator action and the recovered image are presented in Fig. 2. Use of such pictures simplify visual interpretation of the U-Net results, however, these plots may not be representative if used in conjugate gradient method, where preconditioner is applied to the residual vector.

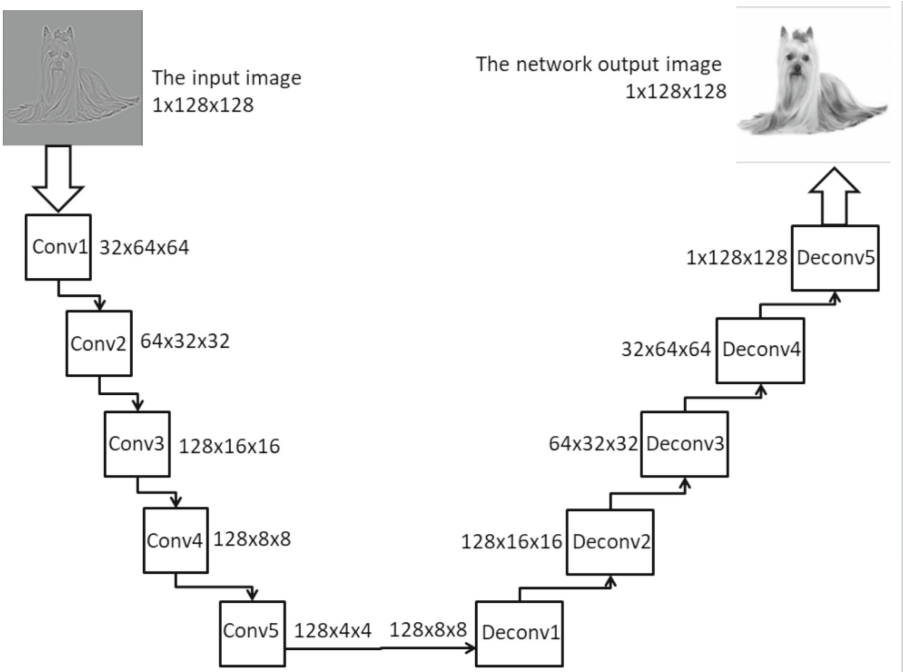

Fig. 1. The architecture of the U-net.

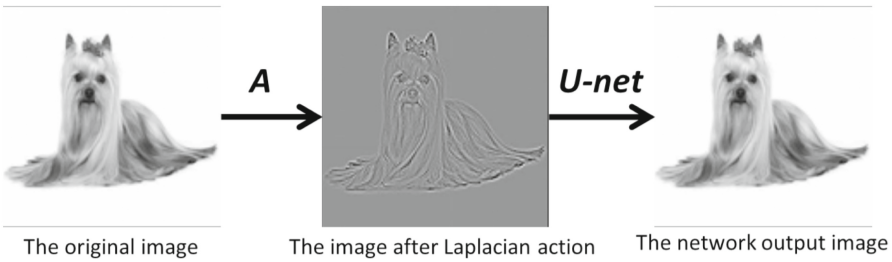

Fig. 2. An example of U-Net application to recover the image. Original image - left, action of Laplace operator - middle, and recovered image - right.

Second, we trained the U-Net on a series of the residuals appeared in the CG algorithm. We used a picture from the set described above, applied Laplace operator to obtain the right-hand side and then started CG iterations without preconditioning. As the result, we obtained a series of vectors $\vec{p}_j$ and $A\vec{p}_j$ which were used to train the U-Net, an example of original and recovered vectors are provided in Fig. 3.

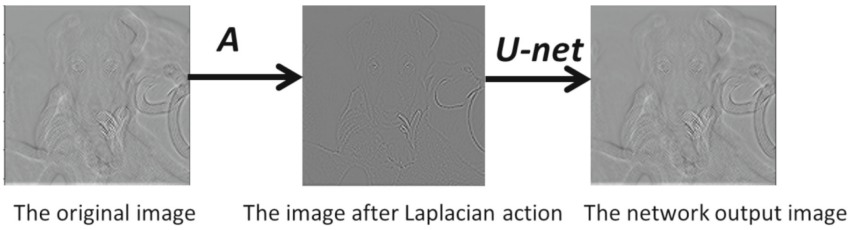

Fig. 3. An example of U-Net application to recover vector $\vec{p}_j$ from CG iterations. Original image - left, action of Laplace operator - middle, and recovered image - right.

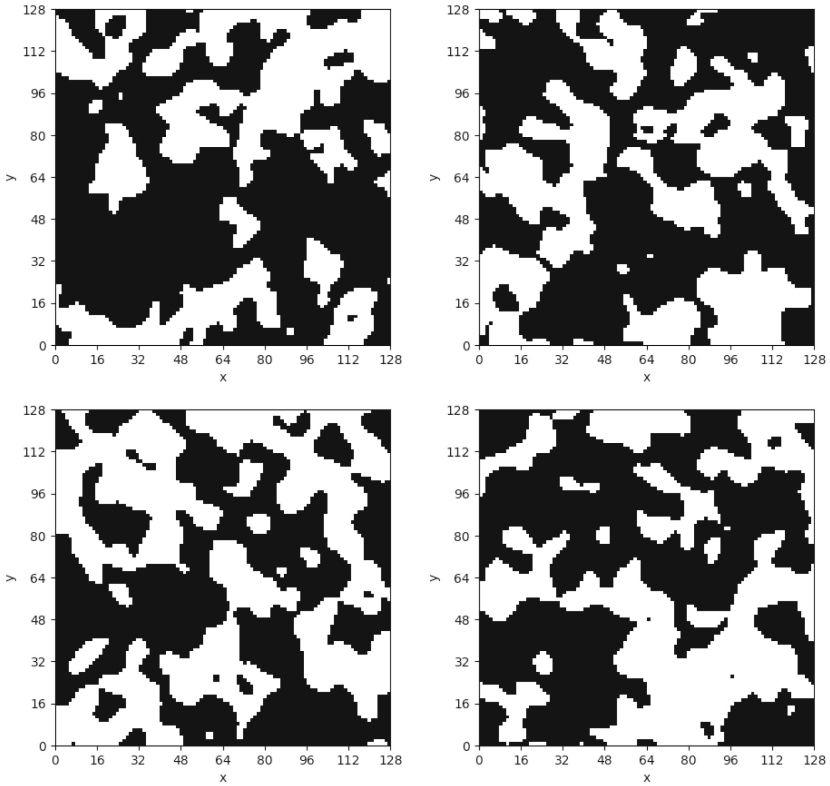

Fig. 4. Examples of models with porosity 35% top left, 40% top right, 45% bottom left, 50% bottom right.

4 Numerical Experiments

For our numerical experiments we considered the inhomogeneous models; i.e., $\sigma = \sigma(x, y)$, is a binary distribution defined as truncated Gaussian distribution with correlation length equal to 10 grid points and total porosity varied in $\{30\%, 35\%, 40\%, 45\%, 50\%\}$. The examples of the models with different total porosity are provided in Fig. 4. We assumed the conductivity of the pore-filling material σ_1 (colored wire in Fig. 4), whereas the conductivity of the matrix σ_2 (black) varied within the set $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$. These models are close to real digital rocks where the conductivity of the pore filled fluid is much higher than that of the insulating matrix. Moreover, the geometry of the pore space of the sandstones is well predicted by the two-point statistics, thus can be approximated by the truncated Gaussian distribution.

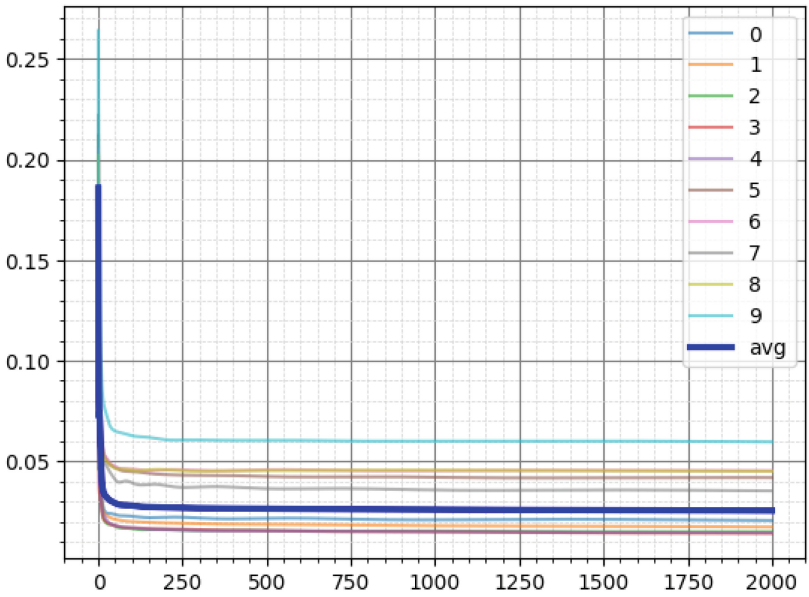

Fig. 5. Residual depending on the iteration number for the model with contrast $\sigma_1/\sigma_2 = 10^1$. Thin lines correspond to experiments for different models, thick line represent averaged values over the model realizations.

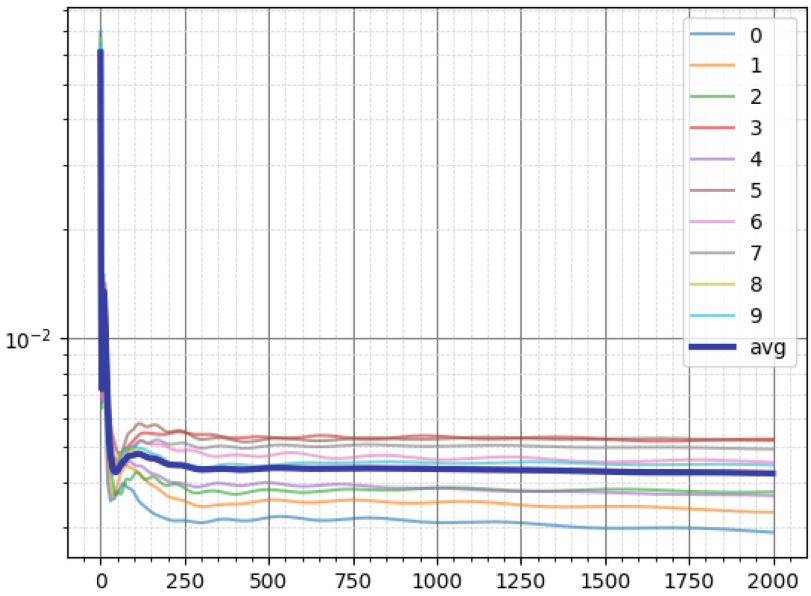

Fig. 6. Residual depending on the iteration number for the model with contrast $\sigma_1/\sigma_2 = 10^2$. Thin lines correspond to experiments for different models, thick line represent averaged values over the model realizations.

To study the effect of the model inhomogeneity on the convergence of the preconditioned conjugate gradient method we considered ten statistical model realizations for each fixed porosity and conductivity contrast. So, in total 200 different models were considered. For each model we trained U-Net which is not a practical way to go, but it is expected to provide the highest possible accuracy. In fact, we trained the net to construct the inverse operator without any additions approximations.

For each of 200 models we applied CG to solve the system of equations for several different right-hand sides. We used pictures from our collection as the true solution, applied the inhomogeneous Laplace operator to obtain the right-hand sides, and then use preconditioned CG to reconstruct the original solution. Unfortunately, the designed algorithm did not converge. So, we considered the residual behavior is ML-based preconditioner was applied to CG for different

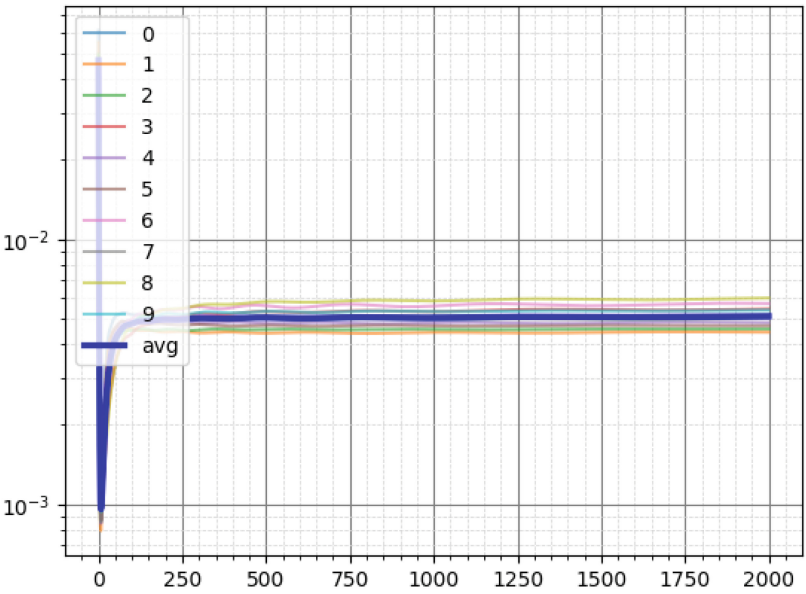

Fig. 7. Residual depending on the iteration number for the model with contrast $\sigma_1/\sigma_2 = 10^3$. Thin lines correspond to experiments for different models, thick line represent averaged values over the model realizations.

models. Figures 5, 6, 7 and 8 provide the residuals over the iteration number for 10 realizations of the model with fixed porosity equal to 45% but with different conductivity contrasts. According to the provided plots, the residual rapidly decreases at the first several iterations, and after that it goes to an asymptotic value far from zero. These value is independent of the model porosity, but sensitive to the conductivity contrast. To estimate these dependence we estimated the irreducible error for all experiments and compute its mean value over all statistical models and realizations, but for fixed conductivity contrast. Moreover, we estimated the mean value of the U-Net loss function over statistical models and realizations. The values of the mean U-Net loss function and the irreducible error are provided in Table 1. Note, that U-Net loss function reaches approximately the same value of about 30% for all conductivity contrasts. However, the irreducible residual decrease with the increase of the contrast.

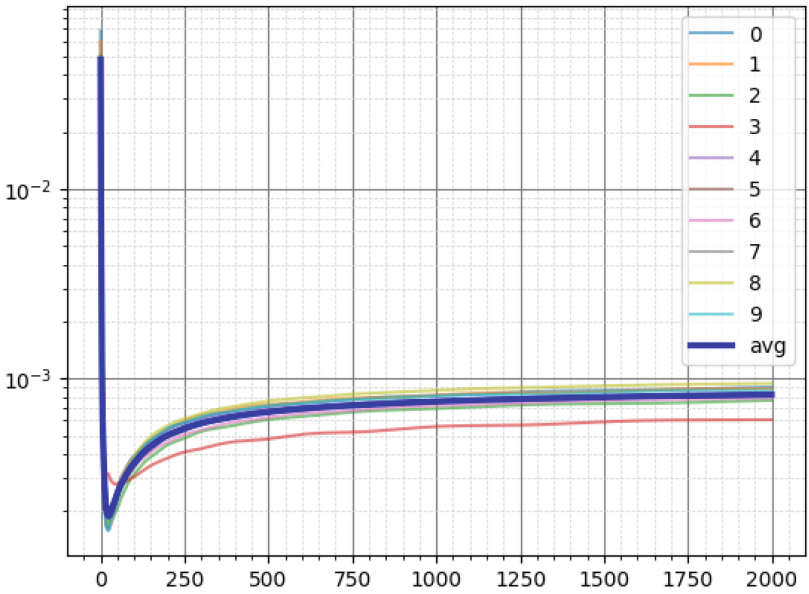

Fig. 8. Residual depending on the iteration number for the model with contrast $\sigma_1/\sigma_2 = 10^4$. Thin lines correspond to experiments for different models, thick line represent averaged values over the model realizations.

Table 1. L^2 error of the ML-recovered images and irreducible residual for different conductivity contrasts.

σ_1/σ_2	L^2 error	Irr. residual
10^4	0.28	$9.48 \cdot 10^{-4}$
10^3	0.24	$5.22 \cdot 10^{-3}$
10^2	0.28	$4.57 \cdot 10^{-3}$
10^1	0.34	$1.76 \cdot 10^{-2}$

5 Conclusions

We presented the approach to construct the machine learning-based preconditioner to solve Poisson equation with conjugate gradient method. We suggested the preconditioner to approximate the inverse Laplace operator for the fixed but spatially varied coefficients. We used the U-Net to approximate the action of the inverse Laplace operator with direct minimization of the relative error (not residual). We managed to achieve the accuracy of the ML-based approximation of only 30%. As the result, this accuracy was not enough to ensure the convergence of the conjugate gradient method. The designed "preconditined" algorithm does not converge to the solution but stabilizes possessing irreducible relative residual

of about 10^{-3}. In this research, we also ignored the fact that the preconditioner in CG method should be linear self-adjoin and positive-definite operator. So, possible way to improve the quality of the ML-based preconditioning is to include these conditions to restrict the ML training loss function.

References

1. Azulay, Y., Treister, E.: Multigrid-augmented deep learning preconditioners for the Helmholtz equation. SIAM J. Sci. Comput. **45**(3), S127–S151 (2023). https://doi.org/10.1137/21m1433514
2. Belonosov, M., Kostin, V., Neklyudov, D., Tcheverda, V.: 3d numerical simulation of elastic waves with a frequency-domain iterative solver. Geophysics **83**(6), T333–T344 (2018)
3. Dorn, C., Schneider, M.: Lippmann-Schwinger solvers for the explicit jump discretization for thermal computational homogenization problems. Int. J. Numer. Meth. Eng. **118**(11), 631–653 (2019)
4. Evstigneev, N.M., Ryabkov, O.I., Gerke, K.M.: Stationary stokes solver for single-phase flow in porous media: a blastingly fast solution based on algebraic multigrid method using GPU. Adv. Water Resour. **171**, 104340 (2023)
5. Gadylshin, K., Gondyul, E., Lisitsa, V., Gadylshina, K., Vishnevsky, D.: Mitigation of numerical dispersion in seismic data in spectral domain with neural networks. Soil Dyn. Earthq. Eng. **187**, 109028 (2024). https://doi.org/10.1016/j.soildyn.2024.109028
6. Gadylshin, K., Lisitsa, V., Vishnevsky, D., Gadylshina, K.: Hausdorff-distance-based training dataset construction for numerical dispersion mitigation neural network. Comput. Geosci. **180**, 105438 (2023). https://doi.org/10.1016/j.cageo.2023.105438
7. Gadylshin, K., Vishnevsky, D., Gadylshina, K., Lisitsa, V.: Numerical dispersion mitigation neural network for seismic modeling. Geophysics **87**(3), T237–T249 (2022)
8. Gondyul, E., Lisitsa, V., Gadylshin, K., Vishnevsky, D.: Numerical dispersion mitigation neural network with velocity model correction. Comput. Geosci. **196**, 105806 (2025). https://doi.org/10.1016/j.cageo.2024.105806
9. Haber, E., Ascher, U.M., Aruliah, D.A., Oldenburg, D.W.: Fast simulation of 3D electromagnetic problems using potentials. J. Comput. Phys. **163**(1), 150–171 (2000)
10. Khachkova, T., Lisitsa, V., Reshetova, G., Tcheverda, V.: GPU-based algorithm for evaluating the electrical resistivity of digital rocks. Comput. Math. Appl. **82**, 200–211 (2021)
11. Khachkova, T.S., Lisitsa, V.V., Gondul, E.A., Prokhorov, D.I., Kostin, V.I.: Two-phase flow simulation algorithm for numerical estimation of relative phase permeability curves of porous materials. Russ. J. Numer. Anal. Math. Model. **39**(4), 209–221 (2024). https://doi.org/10.1515/rnam-2024-0020
12. Kim, J.: Phase-field models for multi-component fluid flows. Commun. Comput. Phys. **12**(3), 613–661 (2012)
13. Lee, B., Min, C.: Optimal preconditioners on solving the Poisson equation with Neumann boundary conditions. J. Comput. Phys. **433**, 110189 (2021)
14. Lee, S., Wheeler, M.F.: Enriched Galerkin methods for two-phase flow in porous media with capillary pressure. J. Comput. Phys. **367**, 65–86 (2018). https://doi.org/10.1016/j.jcp.2018.03.031

15. Li, M., Wang, H., Jimack, P.K.: Generative modeling of sparse approximate inverse preconditioners. In: Lecture Notes in Computer Science. Computational Science – ICCS 2024, vol. 14834, pp. 378–392. Springer Nature Switzerland (2024)
16. Li, Y., Chen, P.Y., Du, T., Matusik, W.: Learning preconditioners for conjugate gradient PDE solvers. In: Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., Scarlett, J. (eds.) Proceedings of the 40th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 202, pp. 19425–19439. PMLR (2023). https://proceedings.mlr.press/v202/li23e.html
17. Lisitsa, V., Podgornova, O., Tcheverda, V.: On the interface error analysis for finite difference wave simulation. Comput. Geosci. **14**(4), 769–778 (2010)
18. Loisel, S., Szyld, D.: On the geometric convergence of optimized Schwarz methods with applications to elliptic problems. Numer. Math. **114**(4), 697–728 (2010)
19. Pleshkevich, A., Vishnevskiy, D., Lisitsa, V.: Sixth-order accurate pseudo-spectral method for solving one-way wave equation. Appl. Math. Comput. **359**, 34–51 (2019)
20. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention (MICCAI). LNCS, vol. 9351, pp. 234–241. Springer (2015). http://lmb.informatik.uni-freiburg.de/Publications/2015/RFB15a
21. Saad, Y.: Iterative Methods for Sparse Linear Systems. SIAM (2003)
22. Samarskii, A.A.: The theory of difference schemes. Pure Appl. Math. **240**. CRC Press (2001)
23. Stuben, K.: A review of algebraic multigrid. J. Comput. Appl. Math. **128**(1–2), 281–309 (2001)
24. Xia, J.: A robust inner–outer hierarchically semi-separable preconditioner. Numer. Linear Algebra Appl. **19**(6), 992–1016 (2012). https://doi.org/10.1002/nla.1850
25. Zhan, X., Schwartz, L.M., Toksöz, M.N., Smith, W.C., Morgan, F.D.: Pore-scale modeling of electrical and fluid transport in Berea sandstone. Geophysics **75**(5), F135–F142 (2010)