Importance Sampling Methods to Accelerate PINN Calculations for *k*-Eigenvalue Neutron Diffusion Problems

William Strom, Mohamed Elhareef, and Zeyun Wu

Department of Mechanical and Nuclear Engineering, Virginia Commonwealth University, Richmond VA 23284

stromwe@vcu.edu; elhareefmh@vcu.edu; zwu@vcu.edu

INTRODUCTION

The Physics Informed Neural Network (PINN) can be used to approximate a solution to a partial differential equation (PDE) by employing a Neural Network (NN) model [1]. Applying Automatic Differentiation [2] to a NN model with respect to the independent variables allows the PINN to be formed by minimizing the residuals of a PDE model and satisfying boundary and/or initial conditions. In this way, the PINN model can be established to follow physical laws without using a traditional training dataset. Additional computational advantage of the PINN approach includes its ability to obtain a purely mesh-free solution for a non-linear PDE without prior assumptions or approximations of the unknown; however, the computational cost including the time consumed in data sampling process in the approach may limit the efficiency of its applications.

In our previous development, the PINN approach was applied to a two-dimensional (2D) two energy group (2G) *k*-eigenvalue mode neutron diffusion model [3]. The PINN model was used to approximate the solution to the *k*-eigenvalue 2G diffusion equations given by

$$\begin{cases} -\nabla(D_1\nabla\phi_1) + \Sigma_{r,1}\phi_1 = \frac{1}{k}\left(\nu\Sigma_{f,1}\phi_1 + \nu\Sigma_{f,2}\phi_2\right) \\ -\nabla(D_2\nabla\phi_2) + \Sigma_{a,2}\phi_2 = \Sigma_{s,1\to2}\phi_1 \end{cases}, \qquad (1)$$

where ϕ_g, D_g, $\Sigma_{a,g}$, and $\Sigma_{f,g}$ (g=1, 2) stand for the flux, the diffusion coefficient, the macroscopic absorption cross section, and the macroscopic fission cross section of the group-g neutrons, respectively; $\Sigma_{s,1\to2}$ stands for the macroscopic down-scattering cross-section of the fast-group neutrons, and $\Sigma_{r,1} = \Sigma_{a,1} + \Sigma_{s,1\to2}$ is the removal cross section for the fast group neutrons. The best PINN solution we gained, compared to the reference solution provided by the high order finite element method, has 0.914% error in *k*-eigenvalue, 2.16E-5 and 3.67E-5 mean absolute error in flux for the fast and thermal groups respectively. One potential computational obstacle that may contribute to these errors is the inefficient sampling approach employed in the current PINN framework.

In this work, we focus on improving the computational efficiency of the PINN approach by investigating and integrating advanced sampling methods into the PINN framework. Previously, the PINN model was trained with semi-random fixed points in the solution domain generated by Latin hypercube sampling (LHS) to ensure the sampling points were evenly spaced [3]. The importance sampling (IS) method enables new sampling points to be selected for each iteration based on some prior sampling distribution. In the IS method, the new sampling distribution is weighted based on relative contribution to the PDE residual after each training iteration. IS has been demonstrated to accelerate the convergence of similar PINN models in the application of elasticity, diffusion, and plane stress problems [4].

METHOD

A PINN Framework for *k*-Eigenvalue Problem

To construct a PINN framework for the 2D 2G neutron diffusion model, we first construct a NN model with two inputs and two outputs as follows

$$\begin{bmatrix}\phi_1(x.y)\\ \phi_2(x.y)\end{bmatrix} = NN(x,y)\,. \qquad (2)$$

In addition to the weights, *w*, and biases, *b*, of one standard neural network model, we introduce an extra learnable parameter, *k*, to approximate the *k*-eigenvalue in the PINN framework for *k*-eigenvalue problems. Following this consideration in the NN model, we define the PDE residuals based on Eq. (1) as follows

$$\begin{bmatrix}f_1\\ f_2\end{bmatrix} = -\nabla\left(\begin{bmatrix}D_1\\ D_2\end{bmatrix}\nabla NN\right) + \begin{bmatrix}\Sigma_{r,1} - \frac{1}{k}\left(\nu\Sigma_{f,1}\right) & -\frac{1}{k}\left(\nu\Sigma_{f,2}\right)\\ -\Sigma_{s,1\to2} & \Sigma_{a,2}\end{bmatrix} NN\,. \qquad (3)$$

Next, we sample N_f points within the solution domain and define a loss function in the form of mean-square of the residuals.

$$loss_f = \frac{1}{N_f}\sum_{i=1}^{N_f}\left|\begin{matrix}f_1\left(x_i^f, y_i^f\right)\\ f_2\left(x_i^f, y_i^f\right)\end{matrix}\right|^2\,. \qquad (4)$$

A similar loss function can be defined for each of the boundary conditions. Additionally, a regularization term is needed to avoid convergence to the trivial flux solution.

$$R = \left[\frac{1}{N_f}\left(\sum_{i=1}^{N_f} NN\left(x_i^f, y_i^f\right)\right) - \frac{C}{N_f}\right], \qquad (5)$$

where C is a parameter that can be defined in terms of physical quantities such as reactor power. The sum of the loss function for the residuals, each of the boundary conditions, and the regularization term define our total loss.

$$loss = loss_f + loss_b + R\ . \tag{6}$$

The total loss function is used to iteratively update the learnable parameters (such as k, w, and b) to minimize the loss. This process is continued until the total loss converges or a maximum number of iterations is reached.

Importance Sampling Technique

The focus of this work is on the improvement of the sampling process required to calculate the loss function. The original method used in the PINN model is LHS approach mentioned earlier, in which one produces evenly distributed semi-random sampling points. Ahead of each training, the LHS is performed to create N_f sampling points. The residuals at these selected points are calculated to form the total loss in the PINN training iteration.

A new sampling method, referred to as importance sampling, is employed in this work. The IS method samples the solution domain after every iteration following a prior distribution, Q, which is determined by each point's contribution to the loss function in the previous iteration. In this work, the IS starts with a selection of some number of collocation points, N_c, generated by the LHS method. To avoid the computational cost increases in the calculation of residuals, we also select N_s seed points in the same way. After each training iteration we define the sampling distribution Q such that the probability of collocation point A being selected is equivalent to the nearest seed point's contribution to the total loss for all seed points, that is

$$q_A = \frac{\left(\left|\frac{f_1}{f_2}\right|^2\right)_{closest\_seed}}{\sum_i^{N_S}\left(\left|\frac{f_1}{f_2}\right|^2\right)_{seed\_i}}, \tag{7}$$

where q_A is the probability of collocation point A being selected as a sampling point. The numerator is the magnitude of the residual at the seed point closest to point A. The denominator is the sum of the residuals at every seed point.

2D *k*-eigenvalue Diffusion Problem

We demonstrate the improvement in computational efficiency from employing the IS method by solving a 2D 2G k-eigenvalue diffusion problem with the geometry shown in Fig. 1. The two-group material properties in the problem are summarized in TABLE I.

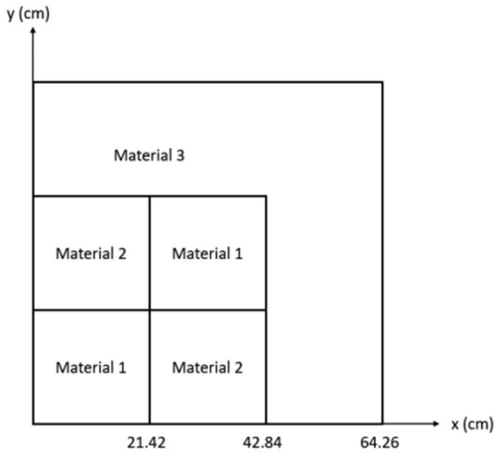

Fig. 1. Geometry of the 2D k-eigenvalue problem.

TABLE I. Materials properties of the 2D diffusion problem.

	Material 1	Material 2	Material 3
D_1 (cm)	1.2	1.2	1.2
D_2 (cm)	0.4	0.4	0.2
$\Sigma_{r,1}$ (cm^{-1})	0.03	0.03	0.051
$\Sigma_{a,2}$ (cm^{-1})	0.3	0.25	0.04
$\Sigma_{s,1\to2}$ (cm^{-1})	0.015	0.015	0.05
$\nu\Sigma_{f,1}$ (cm^{-1})	0.0075	0.0075	0
$\nu\Sigma_{f,2}$ (cm^{-1})	0.45	0.375	0

The boundary conditions of the example problem are as follows: zero flux boundaries are imposed on the right and top edges, while reflective boundary conditions are assumed on the left and bottom sides. These boundary conditions can be described by equations.

$$At\ x\ =\ 0: \quad \frac{\partial}{\partial x}\,\phi_g(0,y) = 0, \tag{8}$$

$$At\ y\ =\ 0: \quad \frac{\partial}{\partial x}\,\phi_g(x,0) = 0, \tag{9}$$

$$At\ x\ =\ 64.26: \quad \phi_g(62.46,y) = 0, \tag{10}$$

$$At\ y\ =\ 64.26: \quad \phi_g(x,62.46) = 0. \tag{11}$$

All calculations were performed on the Google Collaborations graphics processing unit (GPU) machine. This problem is identical to that solved in our previous work, however, the NN model used in this work does not include the L-BFGS optimizer which was used in previous work to reduce the mean absolute error (MAE) of the flux [3].

RESULTS

The problem was firstly solved by using the fixed LHS sampling method. This part is essentially a duplication of our previous results [3]. Then, the same problem was solved by the IS method. The two solutions were then compared in terms of accuracy and computational efficiency. For both methods, a NN with 8 layers and 40 neurons was used. COMSOL Multiphysics was used to obtain a high order finite element method (FEM) solution as reference for verification [3].

For the LHS solution, the solution domain was sampled using 15000 points determined by LHS for both the residual and regularization loss term. Similarly, 100 points on each boundary, sampled by LHS, were used to determine the boundary condition loss term. The relative error of the k-eff value was 0.733%. The MAE were 2.48E-4 and 3.49E-5 for the fast and thermal flux respectively. The total runtime to obtain this solution was 107 minutes. The convergence plots for k-eff and the loss terms are shown in Fig. 2 and Fig. 3, respectively.

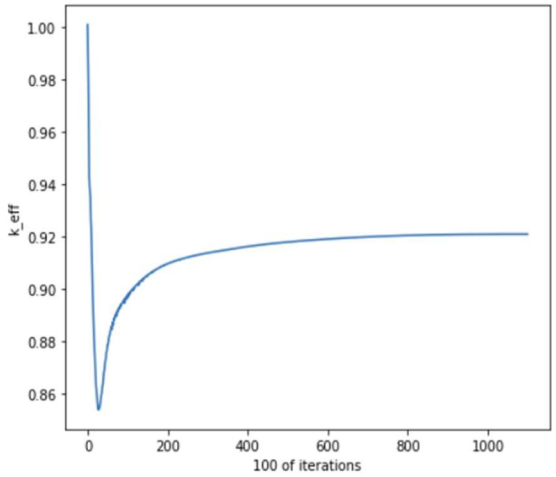

Fig. 2. Convergence of k-eff for the LHS Solution.

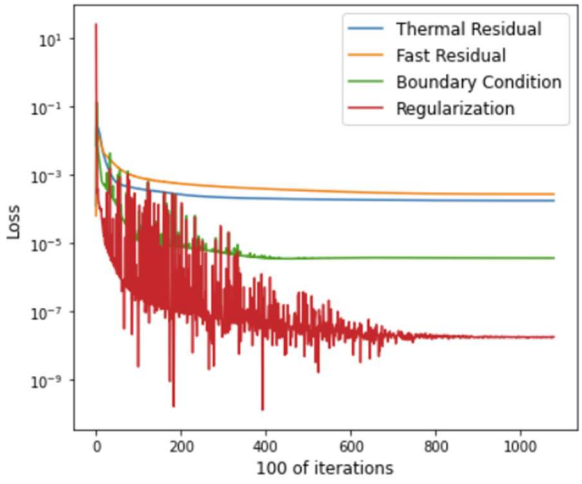

Fig. 3. Convergence of Loss Terms for the LHS Solution.

For the IS solution, we first performed parameter optimization testing to determine the optimal number of seed points and sampling points to achieve the most efficient solution. A 4-layer 5-neuron neural network was used for this testing. The number of seeds were varied while the number of sampling points were held constant at 1000 sampling points. The total runtime results for the first series of tests are shown in Table II.

TABLE II. Seed Point Optimization Testing.

N_S	Run Time (min)
10	32
20	30
50	30
100	35
150	34
200	33

Based on the results shown in Table II, we used 50 seed points for the remainder of this work. The number of sampling points were then varied while the number of seed points were held constant at 50 seed points. The computing time results of the new series of tests are shown in Table III. Based on these results, we use 1000 sampling points for the remainder of this work.

TABLE III. Sampling Point Optimization Testing.

N_f	Run Time (min)
500	32
1000	30
2000	34
3000	32
4000	35
5000	37
6000	39

For the IS implementation, the solution domain was sampled using 15000 collocation points and 50 seed points determined by LHS. The 1000 sampling points were picked from the collocation points by IS method. These sampling points were used to determine the residual loss terms and were updated every 100 training iterations. The solution domain was sampled using 15000 points by LHS to determine the regularization loss term. Similarly, 100 points on each boundary, sampled by LHS, were used to determine

the boundary conditions loss term. The relative error of the k-value was 0.622%. The MAE were 2.54E-4 and 4.85E-5 for the fast and thermal flux respectively. These error values are not significantly different from those obtained by the LHS solution. However, the total runtime to obtain this solution was 62 minutes, which is considerably reduced compared to the LHS method. The convergence plots of k-eff and the loss terms for the IS solutions are shown in Fig. 4 and Fig. 5, respectively.

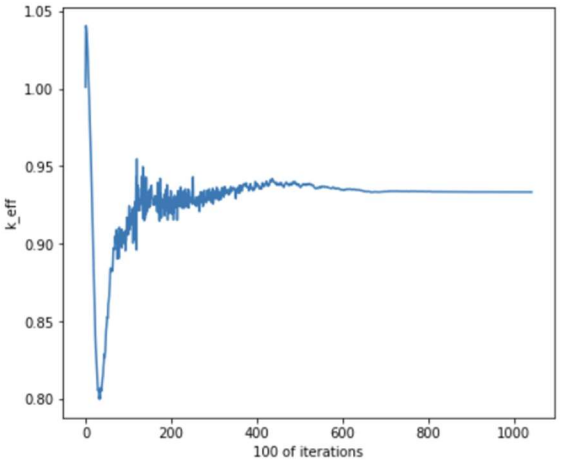

Fig. 4. Convergence of k-eff for the IS Solution.

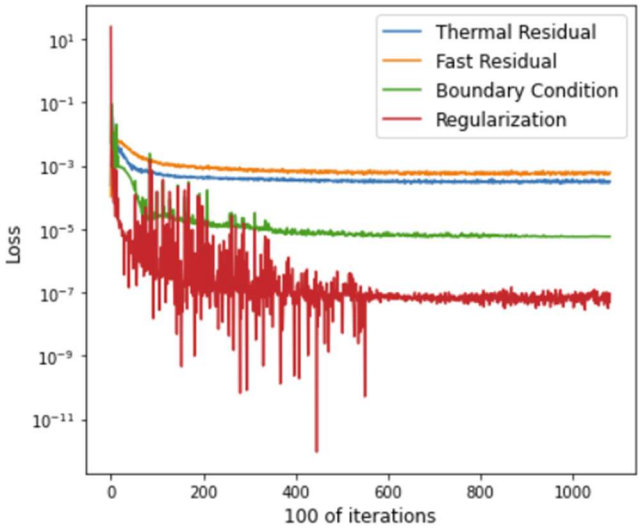

Fig. 5. Convergence of the Loss Terms for the IS Solution.

CONCLUSIONS

While the IS method did not significantly decrease the required number of training iterations, the total runtime required to obtain a solution with similar accuracy was reduced by 42%. This represents a significant decrease in the computational cost of training the NN. This decrease can be attributed to the significantly smaller required sample size since the NN had to be evaluated at an order of magnitude fewer sampling points during each training iteration. However, no significant increase in model accuracy was achieved by implementing this sampling method. Further, when employing IS, it is difficult to determine the loss value convergence since selecting new sample points causes the loss value to fluctuate between iterations. This fluctuation can be observed in Fig. 5. While this work demonstrates a significant step in improving the PINN model efficiency, future work needs to be pursued to obtain a more accurate solution.

REFERENCES

1. M. RAISSI et al., "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, **378**, 686-707 (2019).
2. A. GRIEWANK and A. WALTHER, *Evaluating Derivatives, Principles and Techniques of Algorithmic Differentiation*, 2nd Edition, Society for Industrial and Applied Mathematics, Philadelphia, PA (2008).
3. M. H. Elhareef and Z. WU, "Physics-Informed Neural Network Method and Application to Nuclear Reactor Calculations: A Pilot Study," *Nuclear Science and Engineering*, Published online (2022).
4. M. A. NABIAN, R. J. GLADSTONEG, and H. MEIDANI, "Efficient training of physics-informed neural networks via importance sampling," *Computer Aided Civ Inf.*, **36**, 962–977 (2021).