Learning aggregation for 3D anisotropic diffusion on a structured grid

$$-\nabla \cdot (\boldsymbol{D}\nabla u) = 0, \tag{1}$$

$$\boldsymbol{D} := \boldsymbol{R}^T \begin{bmatrix} \varepsilon_x & & \\ & \varepsilon_y & \\ & & 1 \end{bmatrix} \boldsymbol{R}, \tag{2}$$

$$\boldsymbol{R} := \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{bmatrix}. \tag{3}$$

For both the *isotropic* and *anisotropic* cases we generate two sets each of:

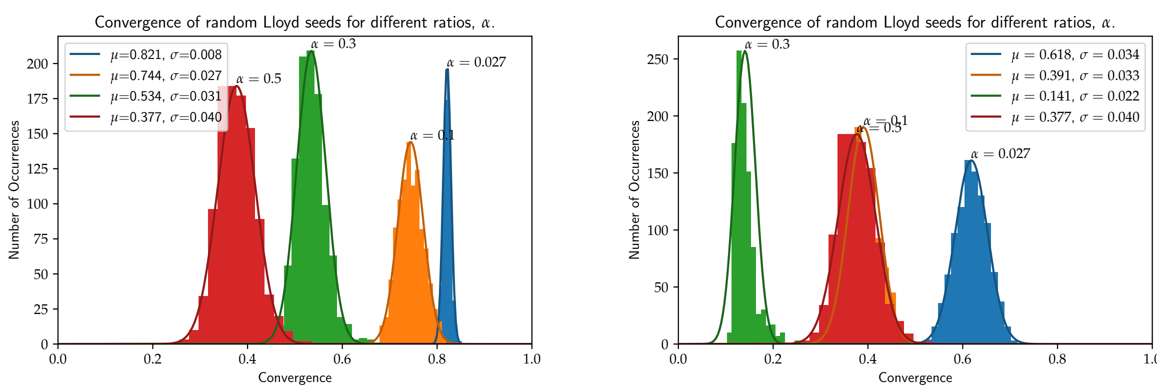
- 500 training problems, and
- 250 testing problems

to train the ML method. For the anisotropic case, these problems randomly have the parameters $N_x = N_y = N_z = N \sim \mathcal{U}\{8, 14\}$; $\theta_z, \theta_y \sim \mathcal{U}(0, 2\pi)$; $\log_{10}\varepsilon_x, \log_{10}\varepsilon_y \sim \mathcal{U}(-4, 4)$. The isotropic problems have the same parameter distribution for N, while $\theta_z = \theta_y = 0$ and $\varepsilon_x = \varepsilon_y = 1$.

Each problem was discretized in Firedrake [1] using piecewise linear tetrahedral finite elements with homogeneous Dirichlet boundary conditions. Afterwards, the degrees-of-freedom corresponding to Dirichlet boundary conditions were removed from the system.

The network was then trained on the anisotropic *and* isotropic problems with a coarsening ratio of $\alpha = 0.1$, as opposed to $\alpha = 0.027$ from last week. I swept over a few different values of α (fig. 1) and $\alpha = 0.1$ seems to provide a good range for the anisotropic problems.

Existing results for the 2D diffusion problems are shown in table 1 for comparison purposes. Results for the untrained (existing 2D model) and trained (specifically on 3D) networks are given in table 2.



(a) Anisotropic problem. The conditioning of the diffusion tensor is $\|\boldsymbol{D}\| \left\|\boldsymbol{D}^{-1}\right\| = 464.93$.

(b) Isotropic problem.

Figure 1: A sweep over various coarsening ratios, α, for two example problems. These are generated by running the solver on random seedings of Lloyd clustering.

References

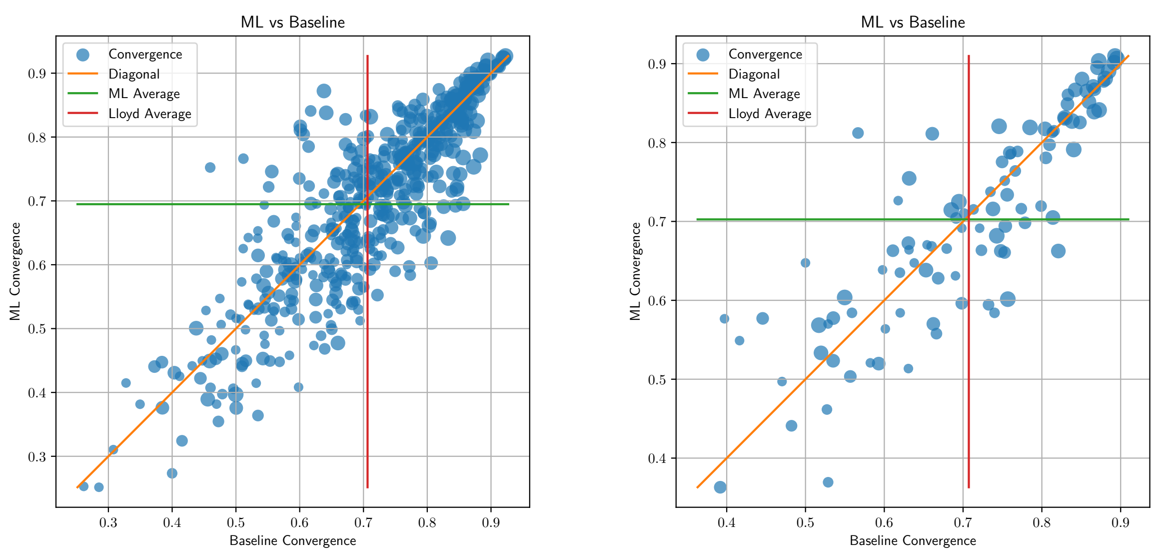
[1] F. Rathgeber, D. A. Ham, L. Mitchell, M. Lange, F. Luporini, A. T. T. McRae, G. Bercea, G. R. Markall, and P. H. J. Kelly, *Firedrake: automating the finite element method by composing abstractions*, CoRR, abs/1501.01809 (2015).

Problem Type	Data Set	Random Conv.	Lloyd Conv.	Full ML Conv.	ML Agg.	ML Int.
Isotropic	Train	0.4652	0.4208	0.3956	0.3974	0.4190
Isotropic	Test	0.4623	0.4177	0.3913	0.3922	0.4172
Anisotropic	Train	0.7680	0.7705	0.7462	0.7532	0.7614
Anisotropic	Test	0.7902	0.7978	0.7727	0.7776	0.7910

Table 1: Existing results for the 2D diffusion problems. This model has been trained on both isotropic and anisotropic diffusion in 2D. The column for *Full ML Conv.* is the method that uses the model for both aggregation and smoothing. *ML Agg.* and *ML Int.* refer to ML aggregation and Jacobi smoothing, Lloyd aggregation and ML smoothing, respectively.

Model	Data Set	Random Conv.	Lloyd Conv.	ML Conv.
Untrained	Iso., Train	0.370	0.393	0.336
Untrained	Iso., Test	0.370	0.387	0.324
Untrained	Aniso., Train	0.706	0.7117	0.7153
Untrained	Aniso., Test	0.707	0.7133	0.7241
Trained	Iso., Train	0.370	0.393	0.319
Trained	Iso., Test	0.370	0.387	0.316
Trained	Aniso., Train	0.706	0.712	0.695
Trained	Aniso., Test	0.707	0.713	0.703

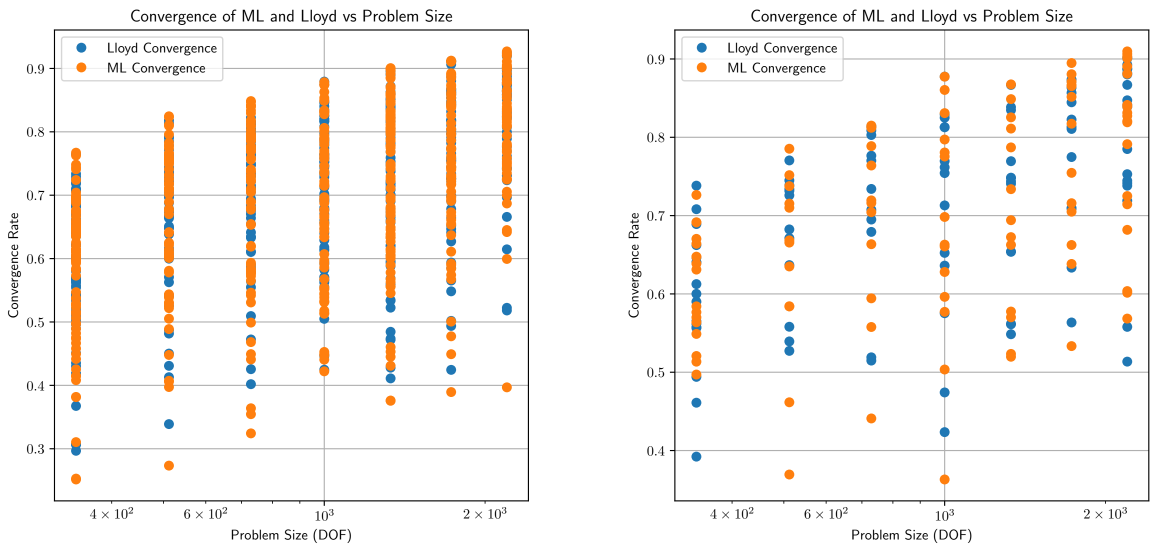
Table 2: Convergence of ML vs baseline, Lloyd methods on training, testing 3D datasets. *Trained* refers to the model trained specifically on the 3D diffusion problems while *untrained* is the existing 2D model (see table 1).



(a) Training convergence

(b) Testing convergence

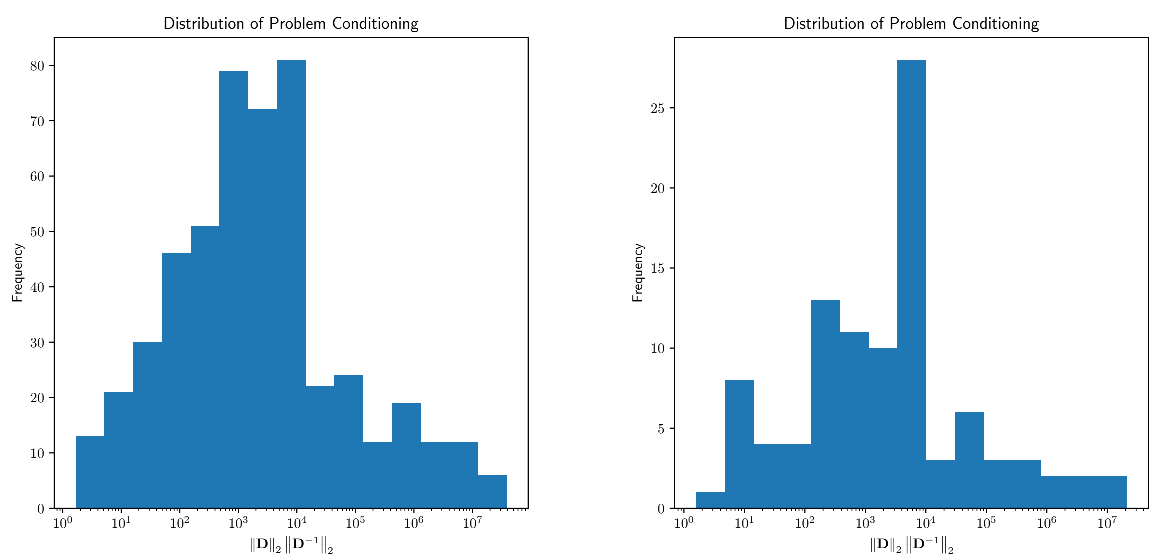
Figure 2: Convergence data for the ML AMG method vs a Lloyd and Jacobi SA method on the *anisotropic* datasets. Values below the diagonal indicate a better convergence for the ML. Markers are scaled by problem size.



(a) Training convergence

(b) Testing convergence

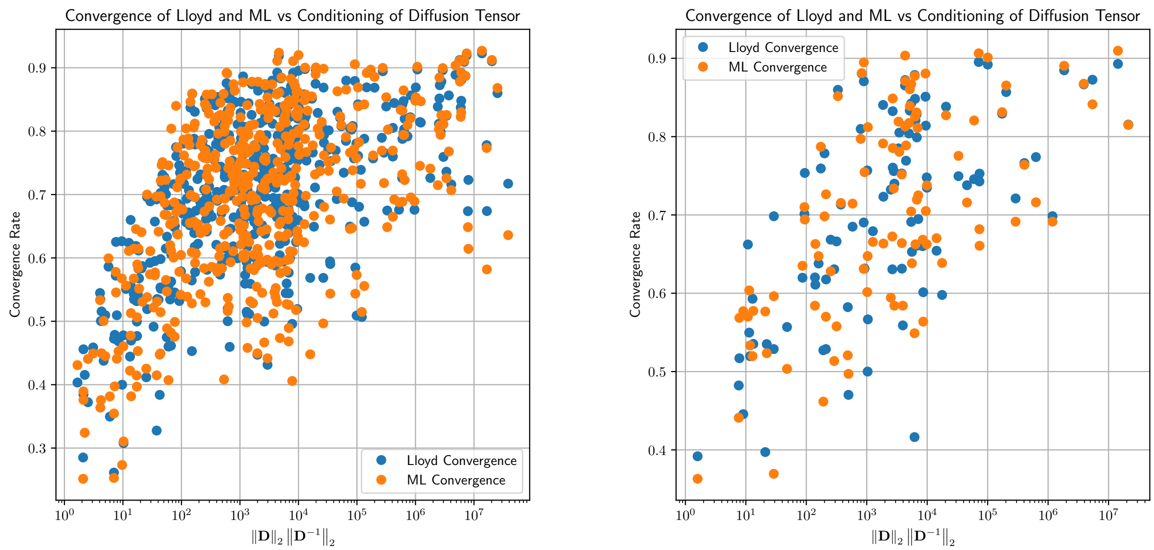
Figure 3: Convergence data for the two methods plotted against problem size (DOF) for *anisotropic* datasets.



(a) Training Conditioning

(b) Testing Conditioning

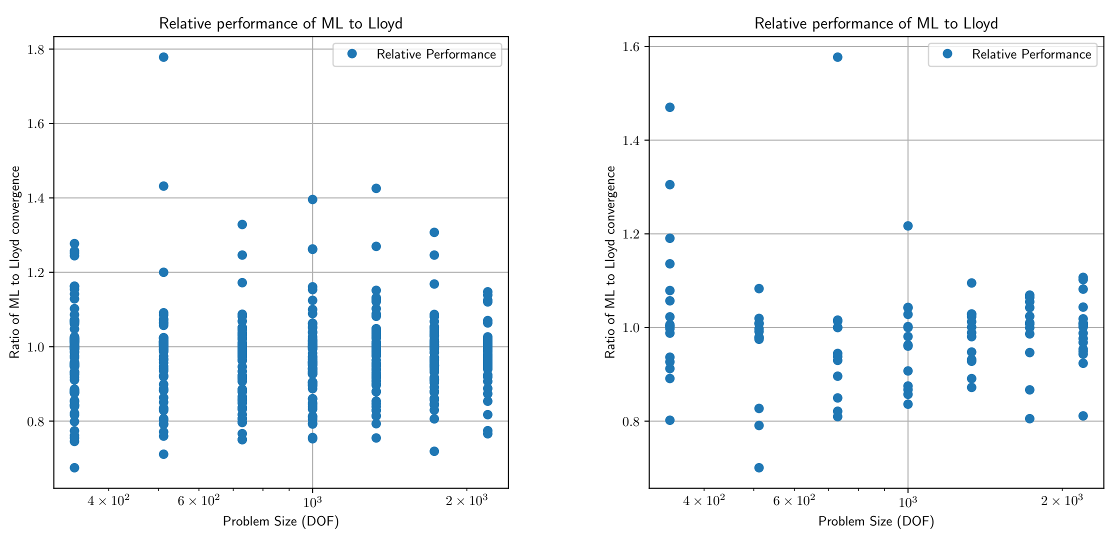
Figure 4: Histograms of problem conditioning for both anisotropic training and testing datasets.



(a) Training Convergence

(b) Testing Convergence

Figure 5: Convergence on both anisotropic datasets per ratio of extremal eigenvalues of the diffusion tensor.



(a) Relative training performance

(b) Relative testing performance

Figure 6: Relative performance of the ML to the Lloyd method, plotted against problem size for the anisotropic datasets. Relative performance is obtained by dividing the ML convergence by the Lloyd convergence for each problem. Values below 1 indicate better ML performance, while values above 1 indicate better baseline performance.