DATASET: Poisson's Equations for Electrostatics

Nathan DeBardeleben, ndebard@lanl.gov, HPC-DES / LANL

September 27, 2025

Contents

Overview of the dataset

This dataset consists of `1000` records in a single HDF5 file generated from the simulation code available at:

https://github.com/lezahlie/esp_simulation. This code implements Poisson's equations for electrostatics and more can be read about it here: https://en.wikipedia.org/wiki/Poisson%27s_equation#Electrostatics.

The files in this repo are:

```
.
|-- README.md
|-- README.pdf
|-- data
|   |-- arguments_electrostatic_poisson_32x32_1-1000.json
|   |-- electrostatic_poisson_32x32_1-1000.hdf5
|   `-- global_statistics_hdf5_electrostatic_poisson_32x32_1-1000.json
|-- demo
|   |-- demo_hdf.py
|   `-- demo_random_record.py
|-- environment.yaml
`-- record_0.png
```

The `demo` directory contains some extremely simply Python code to view contents of the HDF5 file and is explained below. The `environment.yaml` is a sample Conda environment.

In the `data` subdir, the HDF5 file is the raw data of the above explained simulation. The JSON file `arguments_...` details the specific arguments which were used with the above linked GitHub repo to generate this dataset. The `global_statistics_...` file may be useful for dataset normalization for machine learning tasks and details facts of each subrecord (explained below) such as min, max, mean, standard deviation, etc.

Below are some commands to help setup a Python environment to get a quick look at the dataset.

Setting up a conda environment

```
conda env create -f environment.yaml
conda activate electrostatic_dataset_1k
```

Quick look at the HDF5 data

```
python ./demo/demo_hdf.py ./data/electrostatic_poisson_32x32_1-1000.hdf5
```

This should print out:

```
File: ./data/electrostatic_poisson_32x32_1-1000.hdf5
First-level groups (treated as 'records'): 1000
Datasets per record: min=10, max=11

Sub-name presence across records:
- image/charge_distribution: 1000/1000
- image/permittivity_map: 1000/1000
- image/potential_state_1: 1000/1000
- image/potential_state_10: 1000/1000
- image/potential_state_100: 1000/1000
- image/potential_state_final: 1000/1000
- image/potential_state_initial: 1000/1000
- mask/conductive_material_map: 1000/1000
- mask/material_category_map: 1000/1000
- mask/material_id_map: 1000/1000
- image/potential_state_1000: 995/1000

Sub-names missing in some records (showing up to 5 missing record names):
- image/potential_state_1000: missing in 5 records; e.g., record_15, record_156, \
  record_491, record_509, record_63
```

Notice that we have 1000 records, each with 10-11 sub-records including things related to the simulation inputs such as `charge_distribution` and `permittivity_map` (inputs) as well as intermediate states. For this particular simulation, not all inputs run to `1000` time steps which is why the records not all have `potential_state_1000`.

Quick look at the 2d images

The subrecords in this dataset can be visualized as images and are 32x32.

You can quickly visualize a random record with this command:

```
python ./demo/demo_random_record.py ./data/electrostatic_poisson_32x32_1-1000.hdf5
```

See `python ./demo/demo_random_record.py --help` for other parameters such as picking a specific record, saving the image to a file instead, etc.

Final notes

This `README.md` can can be converted into a PDF with:

```
pandoc ./README.md \
  -o README.pdf \
  --pdf-engine=xelatex \
  -V geometry:margin=1in \
```

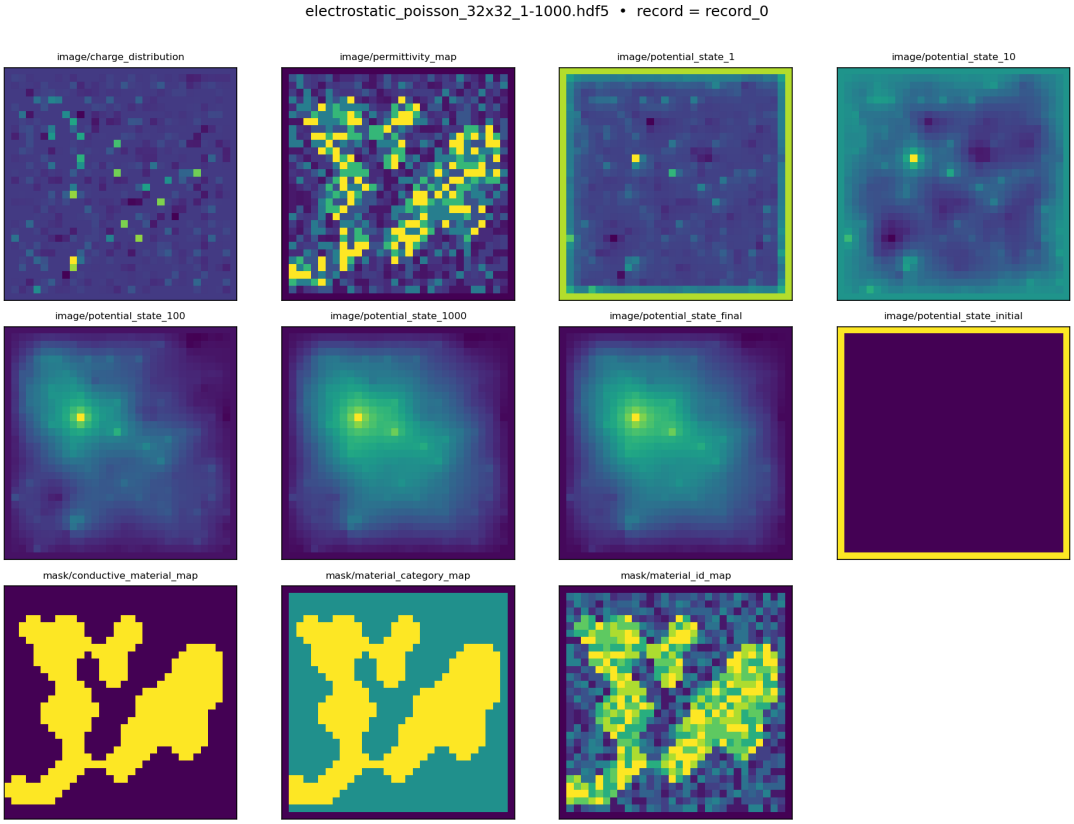

Figure 1: record_0

```
--toc --toc-depth=2 \
--highlight-style=tango
```