

# Data for Idealized Particle-Resolved Large-Eddy Simulations to Evaluate the Impact of Emissions Spatial Heterogeneity on CCN Activity

Samuel Frederick<sup>1</sup>, Matin Mohebalhojeh<sup>2</sup>, Jeffrey Curtis<sup>1,2</sup>, Matthew West<sup>2</sup>, and Nicole Riemer<sup>1</sup>

<sup>1</sup>Department of Climate, Meteorology, and Atmospheric Sciences, University of Illinois Urbana–Champaign, 1301 W. Green St., Urbana, IL 61801, USA

<sup>2</sup>Department of Mechanical Science and Engineering, University of Illinois Urbana–Champaign, 1206 W. Green St., Urbana, IL, 61801, USA

**Correspondence:** Nicole Riemer (nriemer@illinois.edu)

This dataset contains all material required to produce the figures found within the manuscript submitted to Aerosol Chemistry and Physics entitled “Idealized Particle-Resolved Large-Eddy Simulations to Evaluate the Impact of Emissions Spatial Heterogeneity on CCN Activity”. The archived dataset consists of:

- `data.zip`: WRF-PartMC-MOSIAC-LES simulation data in Section 3
- 5    – `scripts.zip`: Python notebooks for generating figures in Section 2 and 3, and scripts used to compile the normalized spatial heterogeneity metric as described by Mohebalhojeh et al. 2026.

## Software requirements/recommendations

All figures in the paper were run with Python 3.9.23. This older version is required due to a dependency of `f2py` (used to compile Fortran code for the spatial heterogeneity metric calculation into an object which can be imported as a Python module) which has since been deprecated. Users should run the `create_env.sh` script in `scripts.zip` to create a conda environment which contains all the necessary packages.

Required packages are as follows with the version used for this manuscript:

- `numpy` (2.0.2)
- `scipy` (1.13.1)
- 15    – `matplotlib` (3.9.2)
- `netCDF4` (1.7.2) - available at <https://unidata.github.io/netcdf4-python/>
- `pandas` (2.3.1)
- `ipykernel` (6.30.1)

– setuptools (59.8.0)

20 – gfortran (15.1.0)

### Directory structure of simulation data for Section 3

Upon downloading and untarring `data.zip`, it may be explored as follows:

– Data files containing a subset of variables, spatial dimensions, and time slices used in generating Section 3 figures. Slices of the domain indicate the index along a given dimension (e.g., time ranges from 0 to 36, vertical height ranges from 0 to 99). Datasets are organized by each emissions scenario (`no-heterogeneity`, `low-heterogeneity`, `medium-heterogeneity`, `high-heterogeneity`). Rather than listing each file here, we replace the scenario title in filenames by `*`:

– Per-grid cell values for gas phase species, aerosol composition, and CCN concentration variables at  $t = 6$  h. Runs without ammonia (discussed in manuscript Section 3.6 contain the scenario followed by `*-no-nh4-`). Datasets end in the following pattern:

`*_subset_t36.nc`

– Binned aerosol number and mass distributions:

`*_size-dist_subset_t0_t36_z60.nc`

– CCN concentration variables, aerosol ammonium, nitrate, and sulfate, and gas phase ammonia and nitric acid, all averaged across each vertical level. The vertical average of each variable is contained at 10-minute increments from  $t = 0$  h to  $t = 6$  h. Runs with emissions turned off at  $t = 4$  h (discussed in Section 3.7) contain the scenario named followed by `*-emis-off-t4`. Scenarios with high RH (discussed in Section 3.8) contain the scenario named followed by `*-high-RH`. Datasets end in the following pattern:

`*_time-height-avg_subset.nc`

– Vertical profiles of environmental conditions (temperature, RH), output at 10-minute increments. At each time, quantities are averaged across each vertical level. This includes “no heterogeneity” and “high heterogeneity” scenarios at lower RH indicated in manuscript Figure 2 and higher RH scenarios (labeled by scenario followed by `*-high-RH-`). Datasets end in the following pattern:

`*_met-vars_subset.nc`

– Detailed per-particle output (e.g., per-particle mass fractions, kappa):

`crossec_no-heterogeneity_t0_z40.nc,`

`crossec*_t36_z40.nc`

– Data in the `spatial-het/` directory consists of lookup tables for spatial heterogeneity values and binary arrays indicating the structure of each emissions pattern:

- 50       – Lookup table for the spatial heterogeneity value ( $\eta$ ) of each emission pattern:  
          `sh_patterns_xres100_yres100_exact.csv`
- Arrays for each emission scenario: `/sh-patterns/xres100yres100/[scenario-name].csv`

### Directory structure of scripts for Section 3

Upon downloading and untarring `scripts.zip`, it may be explored as follows:

- 55       – `create_env.sh`: Shell script for setting up a conda environment with the particular version of Python and associated packages required. **Please run this script first before proceeding to run other files in this directory.**
- `compile_nsh.sh`: Shell script for compiling `nsh.f90` to a Python executable object via the `f2py` package. Once this script runs, you should see a `*.so` shared object library file.
- `nsh.f90`: Fortran module for calculating the discrete normalized spatial heterogeneity metric of Mohebalhojeh et al. 2025. This module contains two subroutines, `normalizedSpatialHet()` which is naive looping routine over all subarray configurations. For large domain sizes, this routine is computationally prohibitive and the Monte Carlo sampling subroutine `monteCarloSpatialHet()` is preferred.
- 60       – `griddedoutput_helperfuncs.py`: Helper functions for processing and plotting per-particle datasets.
- `griddedoutput_plotting.py`: Plotting functions for per-particle datasets.
- 65       – `loaddatastructs.py`: Datasets and associated attributes are housed within the objects `DataStruct` and the inherited class `GriddedOutput` for per-particle datasets.
- `generate_data_subsets.sh`: Shell script for processing WRF-PartMC output files to the data subsets contained in the `data.zip` directory.
- `paper-figures-bulk.ipynb`: Python notebook for generating Figures 1–7 and 10–15.
- 70       – `paper-figures-particle-resolved.ipynb`: Python notebook for generating Figures 8,9 from per-particle datasets.