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

Samuel Frederick[1], Matin Mohebalhojeh[2], Jeffrey Curtis[1,2], Matthew West[2], and Nicole Riemer[1]

[1]Department of Climate, Meteorology, and Atmospheric Sciences, University of Illinois Urbana–Champaign, 1301 W. Green St., Urbana, IL 61801, USA

[2]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
- `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. 2025.

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)
- 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)
- 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 199). 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 gas mixing ratios, particle species mass fractions: `⋆_subset_t36.nc`
  - Binned number and mass distributions: `⋆_size-dist_subset_t0_t36_z60.nc`
  - Detailed per-particle output (e.g., per-particle mass fractions, kappa): `crosssec_⋆_t36_z40.nc`
  - CCN concentration variables, averaged across each vertical level: `⋆_ccn-vars_subset.nc`
  - Output for ammonia-free scenarios: `⋆-no-nh4_subset_t36.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:
  - Lookup table for the spatial heterogeneity value (η) 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:

- `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 proceededing 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.

- `griddedoutput_helperfuncs.py`: Helper functions for processing and plotting per-particle datasets.

- `griddedoutput_plotting.py`: Plotting functions for per-particle datasets.

- `loaddatastructs.py`: Datasets and associated attributes are housed within the objects `DataStruct` and the inherited class `GriddedOutput` for per-particle datasets.

- `paper-figures-bulk.ipynb`: Python notebook for generating Figures 1-6 and 9-11.

- `paper-figures-particle-resolved.ipynb`: Python notebook for generating Figures 7,8 from per-particle datasets.