



HERMES_Δ: bridging the gap between official emission inventories and model-ready emission inputs for chemical transport models



Carles Tena¹, Marc Guevara¹, Johanna Gehlen¹, Òscar Collado¹, Paula Camps¹, Luca Rizza¹, Laura Herrero¹

HERMES_Δ (High-Elective Resolution Modelling Emission System – Delta) is an open-source, Python-based, parallel emission processing system. It applies high-resolution spatial, temporal, vertical, and chemical speciation processing to transform official emission inventories into model-ready datasets for air quality and greenhouse gas simulations.

The challenge

Official annual national emission inventories fulfil mandatory reporting obligations and provide robust and consistent data. However, for their use in Chemical Transport Models (CTMs), emission data needs to be spatially distributed over a grid, temporally broken down into hourly resolution, vertical layers, and chemically mapped to the species defined in the chemical mechanism.

HERMES_Δ Inputs

General Inputs

- Annual Emission Inventories**
 - Table with emissions per activity and region
 - Large Point Sources (LPS) Database
- Profiles:** How to distribute emissions of a specific emission activity over time, vertical layers, chemical components and chemical mechanisms.
- Proxies:** Geostructures and gridded files used for the spatial distribution of emissions.

Over the Iberian Peninsula, we are using:

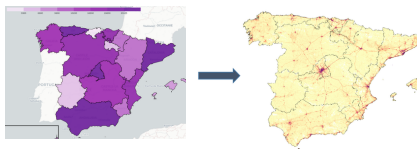
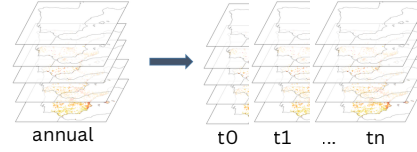
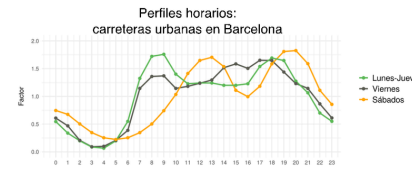
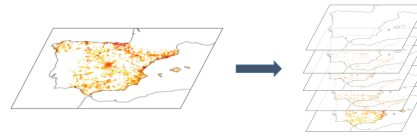
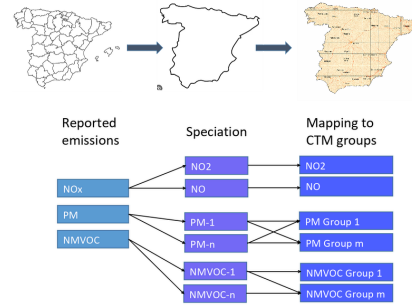
- Annual Emission Inventories:** SEIE (Sistema Español de Inventario de Emisiones) and LPS + Spanish Register of Emissions and Pollutant Sources (PRTR). Both use SNAP nomenclature as activities, and NUTS2 as regions.
- Profiles:** Over 800 unique temporal, vertical and speciation profiles.
- Proxies:** Over 140 proxies.

Flexibility of HERMES_Δ

Input data to HERMES_Δ is completely customisable by the user. Thus, it can be used to simulate various emission scenarios. This enables assessment of the impacts of regulatory measures on local, regional and national air quality.

Methodology

- Initialization of mesh**, optional clipping of the domain, and reading emission inventory input data.
- Chemical Speciation:** splits eg. total NMVOC, NO_x and PM emissions into individual chemical species, then remaps to the grouped species of the selected chemical mechanism, eg: CB5.
- Vertical Distribution** of emissions over layers defined in vertical profiles per emission activity.
- Temporal Distribution** of emissions using temporal local-time profiles, achieving hourly resolution. Outputs are converted from local timezone to UTC time. Holidays are treated as a Sunday. A re-balancing step normalizes the profiles to the specific (month, year) combination.
- Spatial Distribution** per time step using vector and raster proxies.
- Writing of the final emissions as a NetCDF** is done per time step, avoiding memory overload. Available output formats: CF-compliant, CHIMERE, CMAQ, WRF-Chem, MONARCH, MOCAGE



Spatial distribution

We use proxies to spatially distribute the emission values over the grid, that are currently at the level given in the emission inventories.

Proxies contain the normalised fraction of emissions that goes into each grid cell. They are calculated using:

- point, line, polygon geostructures
- categorical and numerical rasters
 - can be combined with polygons.

The geostructures are assigned a weight, and the contribution to the grid cell is calculated based on the weight and intersection with the cell.

In this step, large point sources are also allocated to the correct grid cell according to their coordinates. We avoid double counting by subtracting LPS emissions from the SEIE emissions.

As the proxy calculation can be very resource intensive, proxies are saved as auxiliary files, so they can be reused for any simulation with the same domain and input emissions. For example, the EUCROPMAP has information at a 10mx10m resolution.



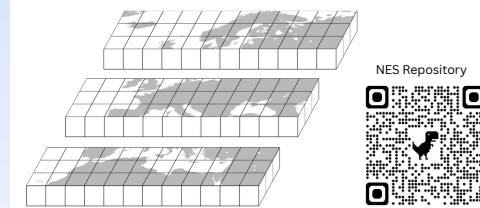
Parallelisation and Performance

The grid is divided into row-wise chunks and each process is assigned to a chunk.

- No communication is required between chunks
- Row-wise division is more efficient, as NetCDFs are written row-wise in C and Python's NetCDF libraries are wrappers of C.

NES as the I/O Engine for NetCDFs

- Parallelisation enabled through the Python library NES, developed at the BSC-CNS.
- Avoids frequent I/O

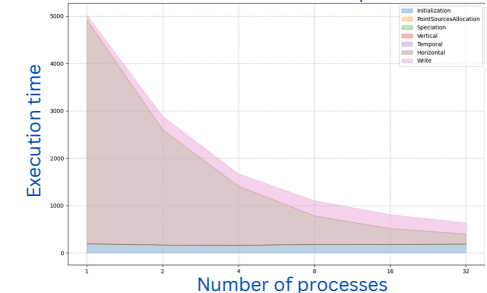


Scalability analysis

- Auxiliary files already created
- 36.5 million 3D cells (24 vertical levels)
- Float64 precision
- 24 time steps, one per hour.
- 10 SNAP sectors
- 38 output pollutants

Results show that performance scales well with the number of processes.

Execution time vs. Number of processes



The project is part of the Recovery, Transformation and Resilience Plan (Plan de Recuperación, Transformación y Resiliencia, PRTyR) funded by the European Union – NextGenerationEU.



1 Barcelona Supercomputing Center (BSC), Barcelona, Spain
carles.tena@bsc.es, johanna.gehlen@bsc.es