MUSICA Tutorial:
How to use MUSICAv0 output

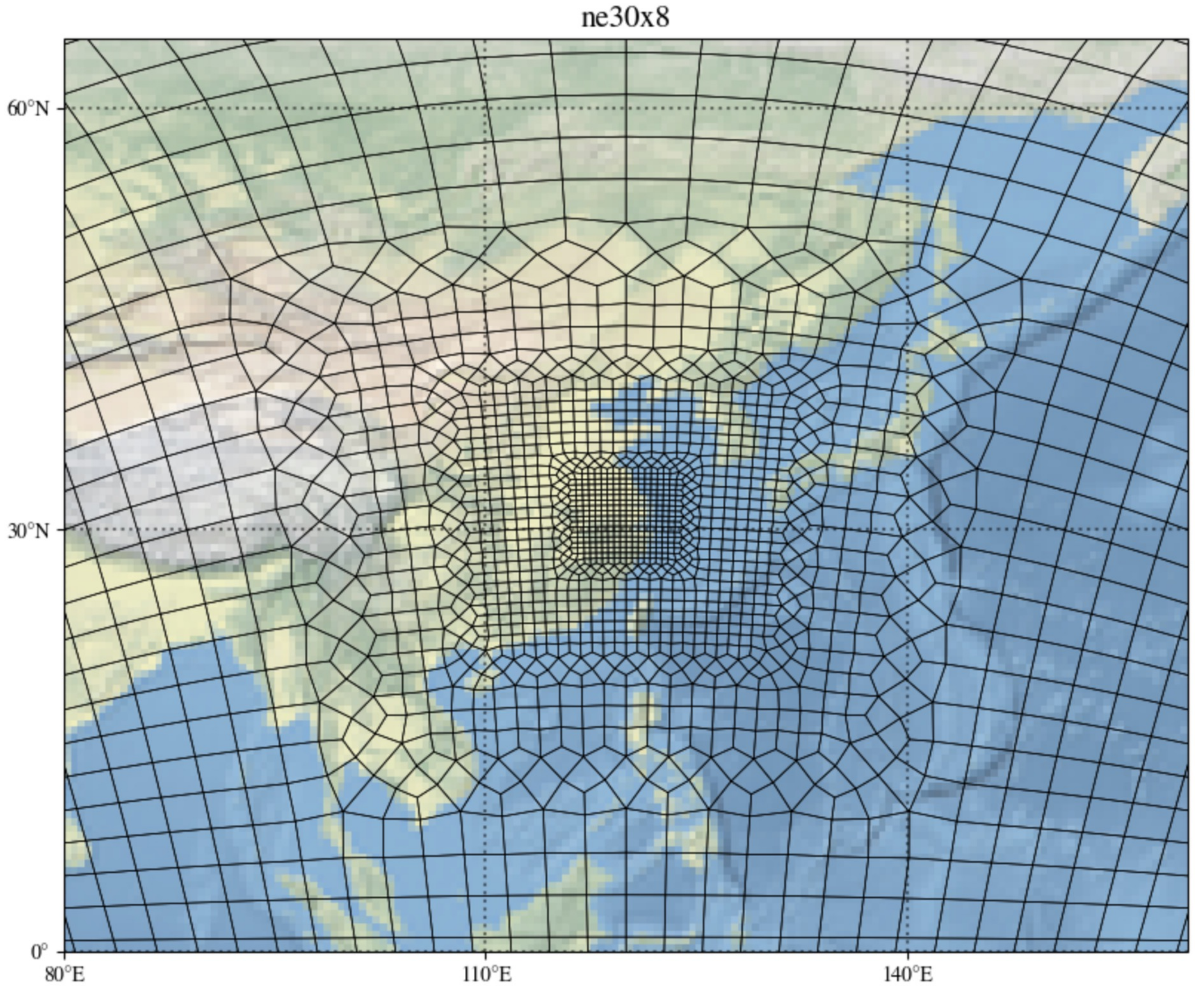
MUSICA: MUlti-Scale Infrastructure for Chemistry and Aerosols

20 Sep 2024

This material is based upon work supported by the National Center for Atmospheric Research, which is a major facility sponsored by the National Science Foundation under Cooperative Agreement No. 1852977

Overview of unstructured grids

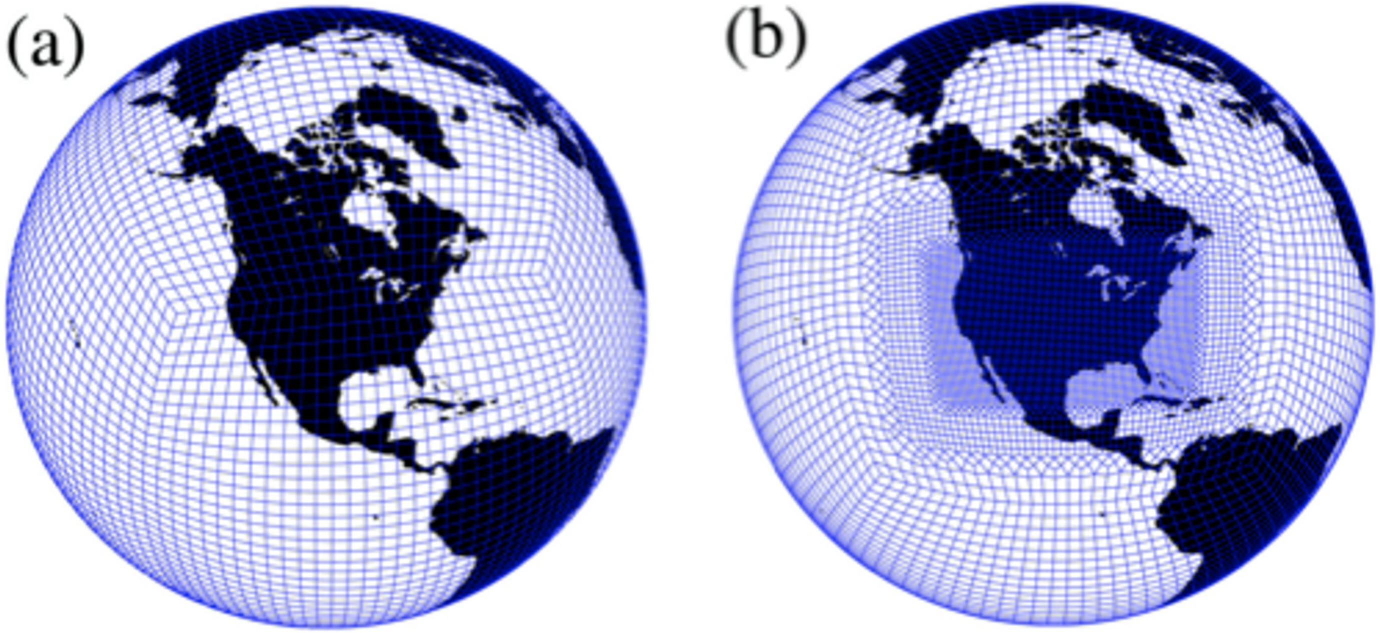
Nanjing Grid



CESM Spectral Element (cubed-sphere)

The spectral element (SE) method has been considered as a numerical method for the fluid flow solver in global weather/climate models

The main motivations were the SE methods' near-perfect scalability, GPU acceleration, high-order accuracy for smooth problems, and mesh refinement capabilities



[*Laurizen et al., 2018; Fig 1*]

CESM Tutorial: https://www.cesm.ucar.edu/events/tutorials/2019/files/Lecture2-lauritzen.pdf

ACOM Fundamentals of Modeling workshop: https://www.acom.ucar.edu/webt/fundamentals/2018/Lecture3_lauritzen2.pdf

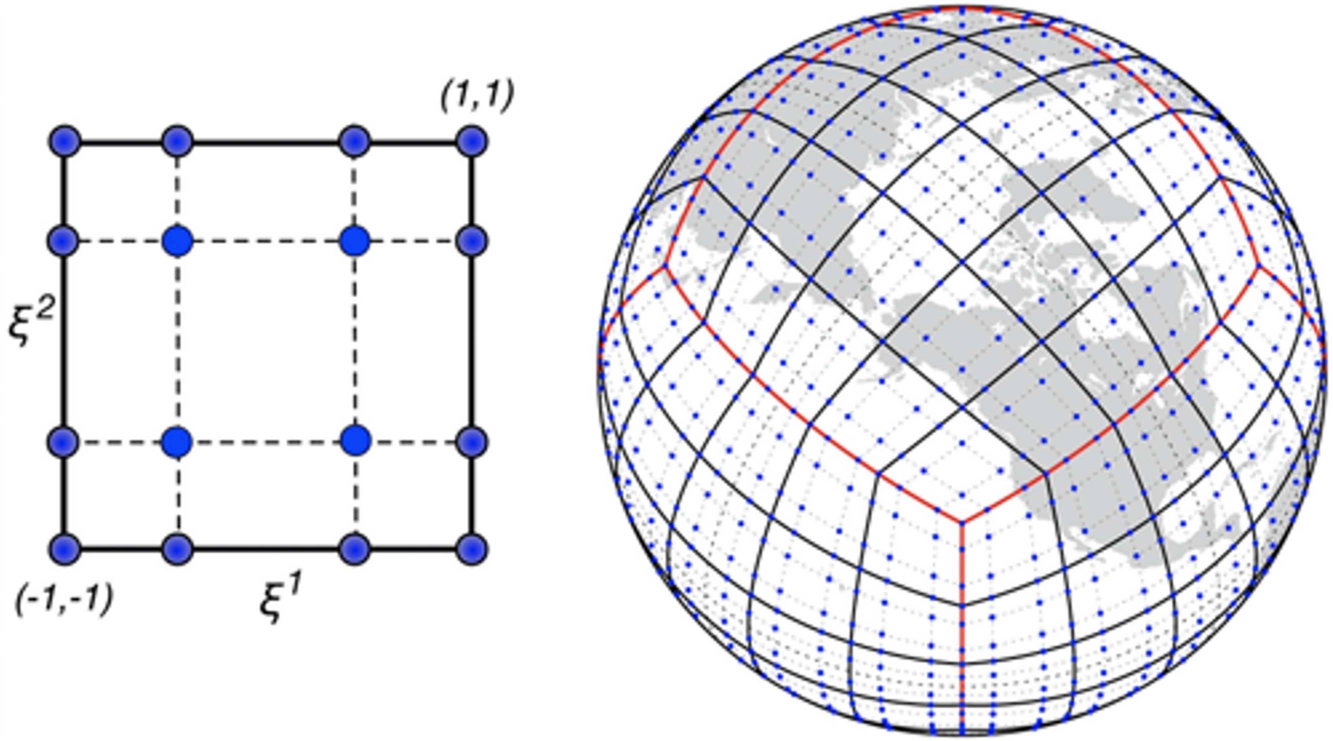
CESM-SE output

The Spectral Element model output is on unstructured grids

Model output provides the lat, lon of the center of each model grid, but the edges and vertices are not

This information is available in “SCRIP” files for each grid

The SCRIP files are read in the plotting examples to allow plotting maps to show the native grid



The left panel shows the Gauss-Lobatto- Legendre (GLL) grid with Np×Np quadrature points defined on a standard element [−1,1]2, where Np=4. The right panel shows the cubed-sphere ( ) grid system tiled with spectral elements Ωe, where Ne is the number of elements in each coordinate direction on a panel (in this case Ne=5). Each element Ωe on has the GLL grid structure.
[*Laurizen et al., 2018; Fig 2*]

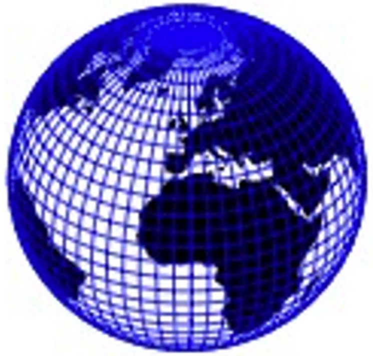
Dimensions and grids

Standard CESM (finite volume) regular grids:

```
dimensions:
        lat = 192 ;
        lon = 288 ;
        time = UNLIMITED ; // (1 currently)
        lev = 32

lat(lat)
lon(lon)
lev(lev)

O3(time, lev, lat, lon)
```



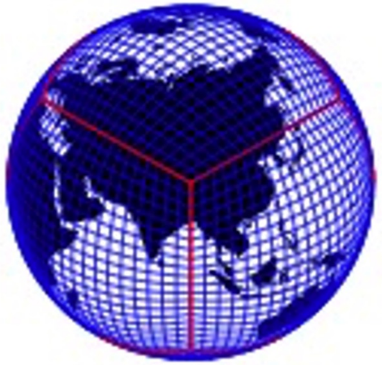
Finite Volume (FV)
[regular lat-lon]

Spectral element output:

```
dimensions:
    ncol = depends on grid numbers;
    time = UNLIMITED ; // (1 currently)
    lev = 32 ;

lat(ncol)
lon(ncol)
lev(lev)
area(ncol)

O3(time, lev, ncol)
```



Spectral Element (SE)
[cubed-sphere]

Nanjing grid **ncol = 60482** ;

Vertical dimension

The dimension **lev** indexes the model layers:

index 0 = top of the model

index 31 = surface layer (if number of lev = 32) [in python can use index = -1]

The variable **lev** is the global mean pressure for each model layer (mid-level), units = hPa

The variable **PMID** is the pressure for each grid box:

```
PMID(time, lev, ncol) ;
                PMID:units = "Pa" ;
                PMID:long_name = "Pressure at layer midpoints" ;
```

Z3(time, lev, ncol) - Geopotential Height (above sea level) of each grid box; units = "m"

PS(time, ncol) - surface pressure; units = "Pa"

Output variables available

Chemical species: find explanations of the chemistry and the species in Emmons et al., JAMES, 2020: https://doi.org/10.1029/2019MS001882

Meteorological & dynamics variables, for example:

- T: temperature (K); U,V: wind speeds (m/s); PRECC, PRECL: precipitation
- CLOUD, CLDTOT: cloud fraction, integrated column cloud fraction
- PBLH: boundary layer height (m)

Emissions diagnostics:

- Total surface emissions: SF{species}
- Biogenic emissions: MEG_{species}
- integrated column of vertical emissions: {species}_CLXF
- Lightning emissions: LNO_COL_PROD (2D), LNO_PROD (3D)

Deposition:

- Dry deposition velocity, flux: DV_{species}, DF_{gas-species}, {aerosol}DDF
- Wet dep, integrated flux: WD_{species}

https://ncar.github.io/CAM/doc/build/html/CAM6.0_users_guide/model-output.html

https://www.cesm.ucar.edu/models/cesm2/atmosphere/docs/ug6/hist_flds_f2000.html

Various output streams (CESM3)

/glade/campaign/acom/acom-weather/emmons/tutorial_nanjing/f.e3beta01.FCnudged.Nanjing_ne30x8.02/atm/hist/

$casename = f.e3beta01.FCnudged.Nanjing_ne30x8.02

./hist/ **{many variables for a number of timesteps in each file}**

$casename.cam.h0a.YYYY-MM.nc : Monthly mean files

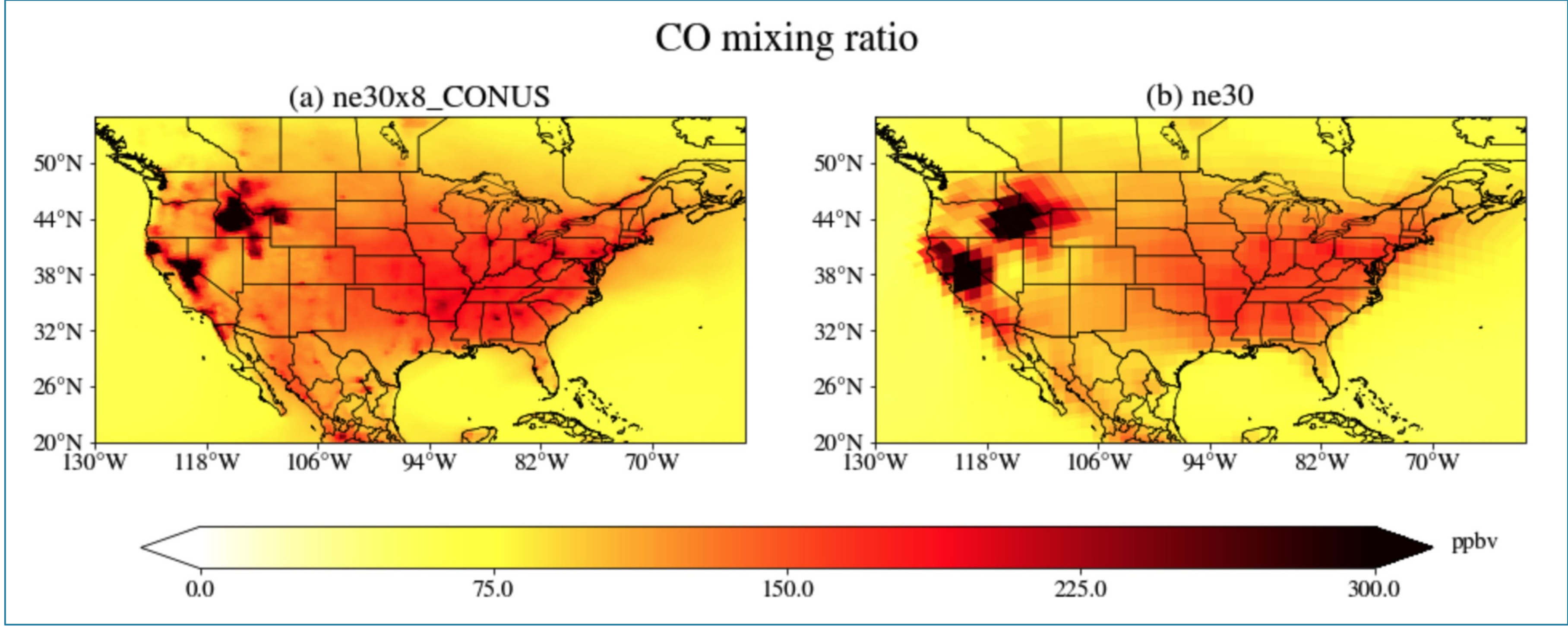
$casename.cam.h0i.YYYY-MM.nc : Monthly instantaneous files

$casename.cam.h1a.YYYY-MM-DD-00000.nc : Daily averages

$casename.cam.h1i.YYYY-MM-DD-00000.nc : Daily instantaneous

$casename.cam.h2i.YYYY-MM-DD-03600.nc : Hourly instantaneous

Use the ‘ncdump’ command on any file from the UNIX command line to get a list of the variables, their dimensions, long names, and units



Demonstration of python plotting examples

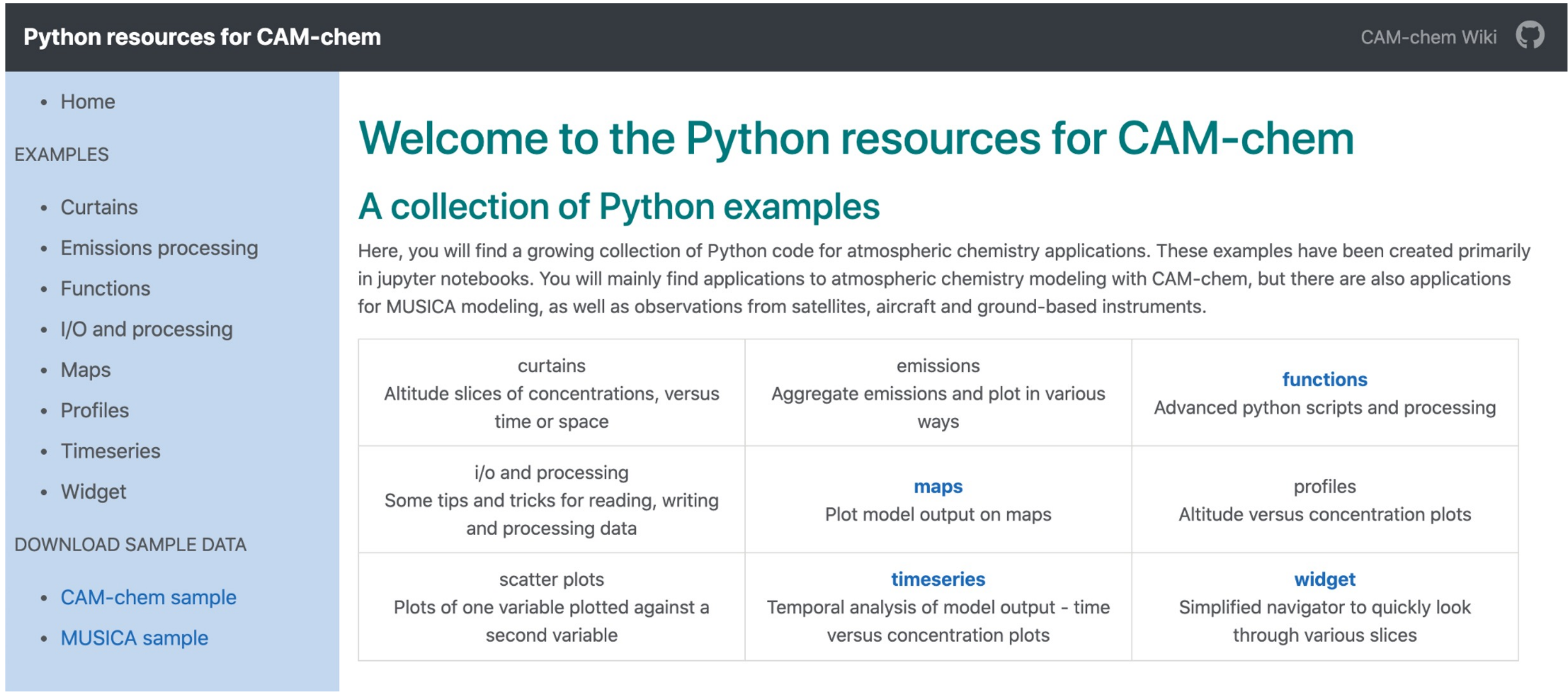
Duseong Jo
Seoul National University
<duseong@snu.ac.kr>

Python Plotting Examples

Tutorial example notebooks:
https://github.com/jzhan166/MUSICAv0_Nanjing_tutorial_2024

Examples of a variety of plots:
https://ncar.github.io/CAM-chem/index.html

Contributions welcome!



Welcome to the Python resources for CAM-chem

A collection of Python examples

Here, you will find a growing collection of Python code for atmospheric chemistry applications. These examples have been created primarily in jupyter notebooks. You will mainly find applications to atmospheric chemistry modeling with CAM-chem, but there are also applications for MUSICA modeling, as well as observations from satellites, aircraft and ground-based instruments.

curtains Altitude slices of concentrations, versus time or space	emissions Aggregate emissions and plot in various ways	**functions** Advanced python scripts and processing
i/o and processing Some tips and tricks for reading, writing and processing data	**maps** Plot model output on maps	profiles Altitude versus concentration plots
scatter plots Plots of one variable plotted against a second variable	**timeseries** Temporal analysis of model output - time versus concentration plots	**widget** Simplified navigator to quickly look through various slices

We are going to learn

- Read MUSICAv0 output file(s)
- View file structure
- Basic array manipulation
- Various 2D map applications with an unstructured grid output (global, regional, custom colorbar, overlaying observations, log-scale plot, multi-panel, gridlines)

- Regrid unstructured grid data to structured grid data and save as a new NetCDF file