WRFLIB

WRFLIB is a python package that contains a group of functions, developed to help post-processing WRF-ARW output.

open_wrfout (Dirpath, dom, varout=1)

input:

- Dirpath is a path to WRF-ARW output NetCDF data,
- dom must be a interger and indicate the WRF domain to be analyzed,
- varout (optional) must be only one of a sequel values (default=1): 1 or 3 or 0:

output:

Depends of varout. 1 returns list of WRF-ARW NetCDF files data (datetime ordered); 3 add at the previous latitude and longitude 2d numpy.array (XLAT, XLONG variables); 0 return a dict contains: the first 3 output described above, latitude and longitude in 1d numpy.array, bottom to top grid dimension and the domain used for the analysis.

jointime (LISTNC)

input:

- LISTNC must be an ordered list of WRF-ARW NetCDF output files.

output:

The function performs a union of a Time dimension from each output file in LISTNC, and returns a 1d numpy.array data type datetime64[m].

join2time_var3d (LISTNC, var)

input:

- LISTNC must be a ordered list of WRF-ARW NetCDF output files,
- var must be a string indicating one WRF-ARW output variable 3d (see WRF-ARW user guide).

output:

The function performs a join (time dimension) from each output file in LISTNC, and returns a 3d numpy.array.

join2time_var4d (LISTNC, var)

input:

- LISTNC must be a ordered list of WRF-ARW NetCDF output files,
- var must be a string indicating one WRF-ARW output variable 4d (see WRF-ARW user guide) or ‘z’ for compute the model height for mass grid [m].

output:

The function performs a join (time dimension) from each output file in LISTNC, and returns a 4d numpy.array.

nearcell_frompoint (LAT, LON)

input:

- LAT and LON are the geographic coordinates of a generic point of interest. The function only works with lats and lons outputs from open_wrfout.

output:

Return positional indexes for the cell closest to the input point in WRF-ARW domain to south-north and west-east dimension. It also returns the distance in km between input point and nearest cell mass grid.

wrfrun_info (LISTNC, varlist=")

input:

- LISTNC must be a ordered list of WRF-ARW NetCDF output files,
- varlist should be a unique string with different WRF-ARW output variables.

output:

Return different information about WRF specific run: start time simulation, end time simulation, current stydy domain, latitude and longitude length and boundary, number of vertical levels, available variables list, and metadata for each variable in varlist if present (default varlist is empty).

spat_CutFromBox (var, latlim, lonlim)

input:

- var must be a 3d or 4d numpy.array contains one variable from WRF-ARW output, in each case the last two dimensions of var must be spatial dimension (usually latitude and longitude),
- latlim and lonlim must be a list of 2 numeric elements, respectively the smallest and biggest boundary box limits.

output:

Returns a numpy.array with the same shape as the input array var. The output array will have nan values out of the box identified by latlim and lonlim.

spat_ReprojectFromWRF (LISTNC, gdf, flag=2)

input:

- LISTNC must be a ordered list of WRF-ARW NetCDF output files,
- gdf must be a geopandas GeoDataFrame to reprojected from WRF-ARW custom projection,
- flag (optional, default=2) used to decide the function output(s), must be set to 2 or 1 or -1.

output:

if flag is set to 2, the function returns the reprojected GeoDataFrame, and the WRF-ARW projection string. Else if flag is set to 1 the function returns only a reprojected GeoDataFrame. If flag is set to -1 the function returns only a WRF-ARW projection string.

spat_CutFromVect (LISTNC, var, gdf, flag=True)

input:

- LISTNC must be a ordered list of WRF-ARW NetCDF output files,
- var must be a 2d, 3d or 4d numpy.array contains one variable from WRF-ARW output, in each case the last two dimensions of var must be spatial dimension (usually latitude and longitude),
- gdf must be a geopandas GeoDataFrame used to cut the WRF-ARW variable in var,
- flag (optional, default = True) used to decide the function output(s), must be boolean input.

output:

Returns a numpy.array with the same shape as the input array var. The output array will have nan values out of the polygon identified by gdf geometry. If flag input is set to True, is also returned a 2d boolean numpy.array to use as a mask for other cuts.