How to create a WRF geographical domain

The step of the WRF Preprocessor System (WPS), called ?geogrid, is not included in the WRF4G workflow. Thus, the user has to deal with it by hand, or using another tool, such as ?WRF Domain Wizard. Geogrid does the task of extracting the fixed fields (orography, land use data, etc.) that WRF needs to run at a given resolution and region. In the WRF4G framework, the output of `geogrid` is known as the `domain` of an experiment. Inside the original WRF4G tarball there are two example domains, Santander_50km and wrfuc, located in `$WRF4G_DEPLOYMENT_DIR/repository/domains`. Here, we are going to see how a new one can be added. It is important to note that it exists an ?excellent on-line tutorial for running WPS and WRF by hand. Users of WRF4G are encouraged to work through this tutorial before start running with WRF4G itself. This framework is intended to make life much easier for WRF users, but knowledge about WRF itself is needed to deal with common errors and issues, and to carry on a correct interpretation of the results.

First of all, WPS binaries are needed. If you don't have them in your system, you will need to build them from source. Instructions for doing this are available too in the ?WRF-ARW online tutorial. Once you have the binaries, you will need to prepare a `namelist.wps` file defining your requirements for the domain (e.g. size, location, resolution, etc.).

Here we are going to explain the variables that usually need modification and we will show an example for 2-nesting domain:

- parent_id --> List of integers specifying the domain number of the nest?s parent (One per domain).

```
parent_id=1,1
```

- parent_grid_ratio --> It is a list of integers specifying for each domain the nesting ratio relative to the domain's parent.

```
parent_grid_ratio=1,3
```

- i_parent_start, j_parent_start --> Coordinates of the lower left corner of the nest in the domain's parent.

```
i_parent_start=1,16, j_parent_start=1,34
```

- e_we --> It represents the nest?s full west-east dimension. For nested domains, e_we must be one greater than an integer multiple of the nest's parent_grid_ratio

```
e_we=60,82
```

- e_sn --> It represents the nest?s full south-north dimension. For nested domains, e_sn must be one greater than an integer multiple of the nest's parent_grid_ratio.

```
e_sn=81,112
```

- dx --> Number specifying the grid distance in the x-direction where the map scale factor is 1. It should be in meters for the 'polar', 'lambert', and 'mercator' projection, and in degrees longitude for the 'lat-lon' projection.

```
dx=0.15
```

- dy --> Number specifying the grid distance in the y-direction where the map scale factor is 1. As stated for dx, this value should be in meters for the 'polar', 'lambert', and 'mercator' projection, but for the 'lat-lon' projection, it should be in degrees latitude.

```
dy=0.15
```

- map_proj --> Name of the projections available 'lambert', 'polar', 'mercator', and 'lat-lon'

```
map_proj='lat-lon'
```

- ref_x, ref_y --> Location of the latitude and longitude of reference for the domain

```
ref_x=1, ref_y=1
```

- ref_lat, ref_lon --> Reference coordinates of the domain.

```
ref_lat=-8.85, ref_lon=24.1
```

- pole_lat --> For the latitude-longitude projection, it represents the latitude of the North Pole with respect to the computational latitude-longitude grid in which -90.0° latitude is at the bottom of a global domain and 90.0° latitude is at the top.
- pole_lon --> For the latitude-longitude projection, it represents the longitude of the North Pole with respect to the computational latitude-longitude grid in which 180.0° longitude is at the center.
- stand_lon --> A real value specifying the longitude that is parallel with the y-axis in the Lambert conformal and polar stereographic projections. For the regular latitude-longitude projection, this value gives the rotation about the earth's geographic poles.
- geog_data_path --> Path to the directory where the geographical data directories may be found.

```
geog_data_path='$HOME/WRF/wps_topo/geog'
```

Also, the GEOGRID.TBL file is needed to be present into a folder called geogrid, in the same place where geogrid is going to be executed. The final structure is the following:

```
[user@mycomputer~]$ tree
.
|-- geogrid
|   `-- GEOGRID.TBL
`-- namelist.wps
```

Provided the namelist and the GEOGRID.TBL, the next step is to execute geogrid.exe. This will generate a netCDF file called geo_em_d01.nc, which contains all the fixed fields that WRF needs to run with that domain.

```
[user@mycomputer~]$ ${PATH_TO_GREOGRID}/geogrid.exe >& geogrid.log
[user@mycomputer~]$ ls
geo_em.d01.nc  geogrid  namelist.wps  geogrid.log
```

The next and final step is to copy the geo_em and the namelist.wps files to a directory with the name chosen for the domain (e.g. Europe_15k). This folder has to be located in the path specified by the variable `domain_path` in experiment.wrf4g. After following these steps, the new domain is available for using it in any WRF4G experiment.