SEDOO-AERIS

GIRAFE - REGIONAL REAL TIME FIRE PLUMES SIMULATION TOOL

TECHNICAL DOCUMENT & USER MANUAL

	Name	Organisation	Date	Visa
Written by:	Daria Malik	Magellium	22/12/2023	
Checked by:	Vanessa Pedinotti	Magellium		
Approved by:	Béatrice Berthelot	Magellium		

Document reference:	
Edition.Revision:	1.0
Date Issued:	22/12/2023
Customer:	
Ref. Market, consultation:	

Distribution List

	Name	Organisation	No. copies
Sent to :	D. Boulanger P. Henry	OMP OMP	1 1
Internal copy :	V. Pedinotti B. Berthelot	Magellium Magellium	1 1

Document evolution sheet

Ed.	Rev.	Date	Purpose evolution	Comments
1	0	22/12/2023	Creation of document	

Dissemination level

PU	Public	x
PP	Restricted to other programme participants	
RE	Restricted to a group specified by the consortium	
CO	Confidential, only for members of the consortium	

Contents

1. GIRAFE overview

GIRAFE is a simulation tool that provides estimated trajectories of the fire plumes or other CO emissions. These trajectories are computed by the Lagrangian particle dispersion model FLEXPART which is suitable for the simulation of a large range of atmospheric transport processes and atmospheric components. In the GIRAFE tool we are interested only in the CO emissions. The model uses the European Centre for Medium-Range Weather Forecasts (ECMWF) meteorological data and the emission inventories or fire detection data. The trajectories are represented by the concentration of atmospheric components on the latitude-longitude-altitude grid for date and times of simulation defined by the user. The visualization of results allows to study and see where the particles will be potentially drifted and carried on (Fig.1).

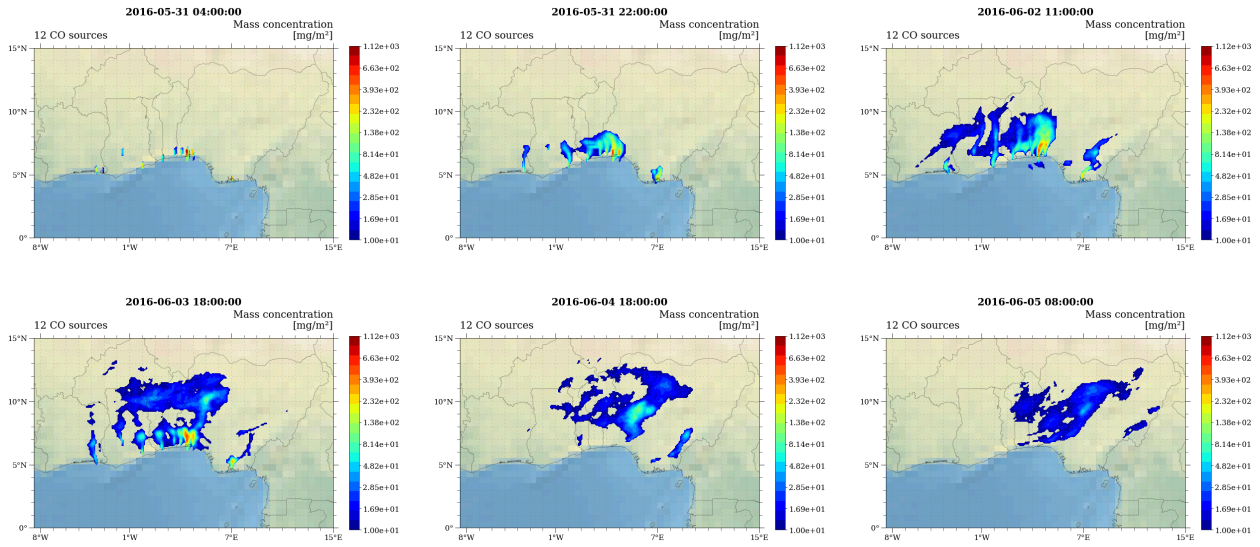

Figure 1: Example of the simulation output for fire plumes trajectories in West Africa

For the detection of emissions or fires GIRAFE uses external files: "emission inventories" or MODIS Fire products respectively. The emission inventories contain concentrations of different emissions on the latitude-longitude grid, for various emissions sources (aviation emissions, volcanic SO2 emissions, biomass burning etc). The MODIS fire products contain thermal anomalies / active fires latitude-longitude locations with their respective confidence levels. Current version of GIRAFE supports the majority of the emission inventories. The requirements are : the file needs to be in the netCDF format and have variables "time", "lat" and "lon" with these exact names. The supported MODIS fire product is the MCD14DL collection, which can be with the txt or csv extension.

All of the GIRAFE executables and dependencies are wrapped up into a Singularity container.

1.1 GIRAFE architecture

GIRAFE tool is based on the FLEXPART simulation tool which is a Lagrangian transport and dispersion model suitable for the simulation of a large range of atmospheric transport processes. Hence, the main executable that launches simulations is the FLEXPART executable installed in the Singularity container.

The executable needs : 1) FLEXPART input configuration files which follow particular rules and also 2) the access to the meteorological data in the GRIB format (Fig. 1). The syntax of the input files is detailed in the FLEXPART documentation [1]. The user interface of the GIRAFE tool allows to configure the simulation through one configuration xml file, without changing manually every FLEXPART input file, and thus also allows to automate the process in case of multiple simulations. The output of the simulation (concentrations of CO emissions) can either be in the binary or netcdf format (based on your configuration), but the user interface tool supports only the netcdf format for the post-processing.

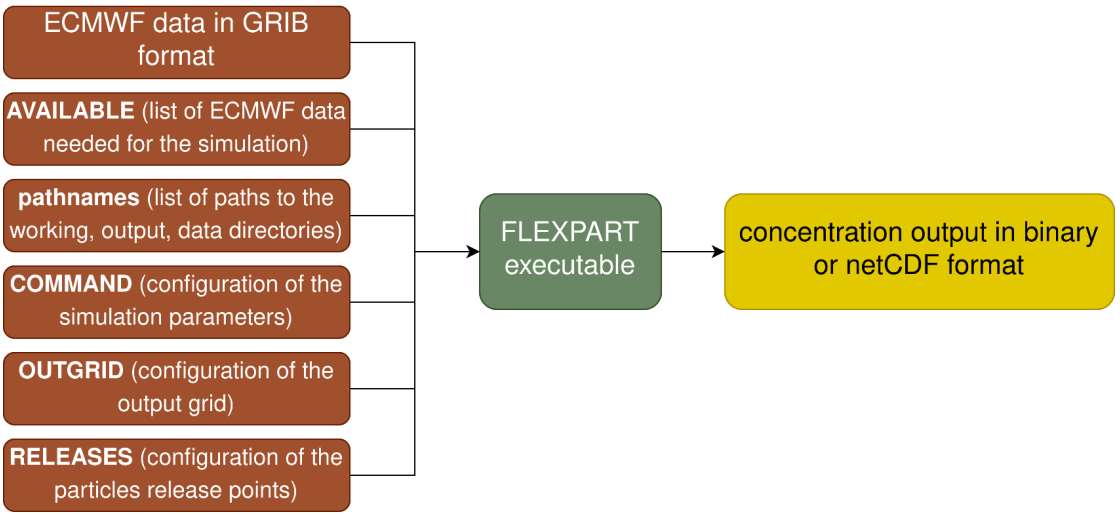

Figure 2 : scheme of the FLEXPART simulation input/output which is the main element of the GIRAFE tool

1.2 FLEXPART input files

FLEXPART needs the following three types of input files [1] :

1. The text file `pathnames` which must be located in the directory where FLEXPART is executed. It must contain four lines:
    - the path to the working directory where the simulation will be executed and where run-defining input files are located (the so-called `options` directory, explained later in this section)
    - the path where output files will be created
    - the path to the directory with meteorological input GRIB files
    - the path to the so-called AVAILABLE file (see point 3)
2. Text files with the run-defining settings located in a subdirectory (given in line 1 of pathnames) called `options`. The brief description of these files is given in the Table 1.

3. The meteorological input data, one file for each input time, stored in GRIB format in a common directory (specified in line 3 of pathnames). To enable FLEXPART to find these files, a file named AVAILABLE (given in line 4 of pathnames) contains a list of all available meteorological input files and their corresponding time stamps.

Table 1 : Alphabetical list of the run-defining input files (upper part) and static input files (lower parts).

File name	Content
AGECLASSES	Age class definitions
COMMAND	Main control parameters
OUTGRID	Output grid definition
RECEPTORS	Receptor locations for receptor kernel output
RELEASES	Specification of the sources (forward run) or receptors (backward run)
SPECIES/	Directory containing files with definitions of physical and chemical parameters of species referenced in RELEASES
IGBP_int1.dat	Land cover input data
surfdata.t	Roughness length, leaf area index for different land cover types
surfdepo.t	Seasonal surface resistances for different land cover types

1.2.1 Run-defining settings

This subsection, describes in more detail the run-defining setting files, their contents and their role. These settings control FLEXPART's physics and program flow.

1.2.1.1 File COMMAND

The COMMAND file contains the user settings controlling the simulation and the behavior of the run. Parameters such as simulation begin and end datetime, output type and format, interval of model output and others are defined in this file. A complete listing of all settings with their meaning and preset default values can be found in the FLEXPART documentation [1].

1.2.1.2 File RELEASES

The RELEASES file contains details regarding the introduction of particles in the simulation, including information on the timing, location, and characteristics of release points. The header of this file includes the total number of different species intended for release, accompanied by a corresponding list of FLEXPART species numbers (nnn). The SPECIES_nnn files further define the physical properties of these species. Following the header, an arbitrary number of namelists &RELEASE is entered, each defining a distinct release. For each release, the provided information includes the start and end times, spatial coordinates and dimensions, released masses (with one value per species), the quantity of particles slated for release, and an accompanying comment string.

1.2.1.3 File OUTGRID

The OUTGRID file delineates the domain and grid spacing for the three-dimensional output grid. It is important to note that, in a Lagrangian model, the specifications for the domain and resolution of the gridded output are entirely distinct from those pertaining to the meteorological input. The only requirement is that the output domain must be contained within the computational domain.

1.2.1.5 File RECEPTORS

In addition to gridded model output, it is also possible to define receptor points. This option allows to produce the output for certain points at the surface. The RECEPTORS file contains a list with the definitions of the receptor name, longitude and latitude. If no such file is present, no receptors are written to output.

1.2.2 Meteorological data

FLEXPART is capable of operating with meteorological input data tailored for global domains or more localized, limited-area domains. The computational domain in FLEXPART aligns with the domain established by the input data, while the output domain can be configured to be smaller.

The compilation of FLEXPART version 10.4 results in a single executable that automatically discerns whether the meteorological input data originate from ECMWF IFS or NCEP GFS, and whether they are formatted in GRIB-1 or GRIB-2. However, adjustments to certain parameters in the `par_mod.f90` file may be necessary to accommodate the size of meteorological input files, particularly in terms of array dimensions. Additionally, the input grid might require shifting relative to the output grid (`nxshift` parameter).

1.2.2.1 Extracting ECMWF data

in development…

1.3 FLEXPART output files

At each output time, FLEXPART generates files comprising gridded output, with separate files produced for each species. The file-naming convention follows the pattern `grid_[type]_[date]_[nnn]`. For forward runs, the type can be either *conc* or *pptv*, representing concentrations and mixing ratios, or *flux* for 3-D mass fluxes across grid cell faces. In forward simulations, wet and dry deposition fields are computed on the same horizontal output grid and appended to files named `grid_conc_date_nnn` and `grid_pptv_date_nnn`. The accumulation of deposited matter occurs throughout the model run, generally increasing with model time. However, for species with radioactive decay, losses are possible.

For surface points, concentrations or mixing ratios in forward simulations can be independently calculated from the grid using a kernel method. The results are recorded in files named `receptor_conc` and/or `receptor_pptv`.

If the particle dump option is activated, in addition to the gridded output, particle coordinates, along with additional variables such as pressure, humidity, density, tropopause height, atmospheric boundary layer (ABL) height, and orography height, are recorded in binary files named `partposit_date`. FLEXPART version 10.4 also offers the option to write out time-averaged particle positions and meteorological data, recorded in files named `partposit_average_date`. In plume trajectory mode, for every release, the positions of trajectory clusters representing the centers of mass of all released particles are recorded in the file `trajectories.txt`.

The physical unit used for the output data in the files `grid_conc_date_nnn` and `grid_time_date_nnn` depends on the settings of the switches `ind_source` and `ind_receptor` (Table 2). It's important to note that the unit of mass mixing ratio can also be utilized in `grid_conc_date_nnn`. For forward runs, additional files `grid_pptv_date_nnn` may be generated, containing data such as volume mixing ratios (requiring molar weight in SPECIES_nnn file). Additionally, it's noteworthy that all gridded output quantities in FLEXPART represent grid cell averages, not point values.

Table 2 : Physical units of input/output data in forward runs for various settings of ind_source, ind_receptor

File name	ind_source	ind_receptor	Input unit	Output unit
grid_conc*	1	1	kg	ng.m-3
grid_conc*	1	2	kg	ppt by mass
grid_conc*	2	1	1	ng.m-3
grid_conc*	2	2	1	ppt by mass
grid_conc*	1	1 or 2 (deposition)	kg	ng.m-2
grid_conc*	2	1 or 2 (deposition)	1	ng.m-2
grid_pptv*	1	1	1	ppt by volume

By default, FLEXPART output is written in the native binary format. However, FLEXPART v10.4 can also support output in NetCDF format if the NetCDF libraries are installed. To activate NetCDF support, the option `ncf=yes` must be appended to the compilation `make` command. Only one NetCDF file is written; this file contains all species and all time steps. Since the NetCDF output is specified in the climate and forecast (CF) format, any standard software can be used for displaying and processing the output (Figure 3). NetCDF output data files are compressed. The NetCDF output file contains information on the run settings and the simulation grid from the COMMAND and OUTGRID* files.

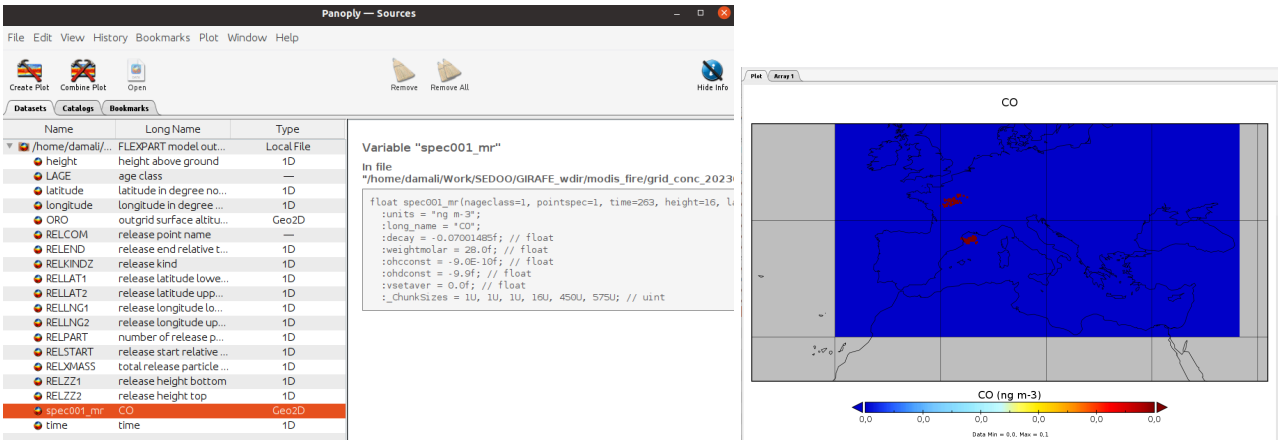

Figure 3 : FLEXPART output in netCDF format, opened with Panoply

2. GIRAFE simulation for user

FLEXPART tool is containerized into a Singularity container along with its dependencies and aux libraries needed for the post-processing or pre-processing of the GIRAFE simulation. Such containerization allows any user to run it without previous libraries installation and source code configurations (except the installation of the Singularity application itself).

The file `girafe-container.def` is a Singularity Definition File, which is a set of blueprints explaining how to build a custom container. It includes specifics about the base OS to build or the base container to start from, software to install, environment variables to set at runtime, files to add from the host system, and container metadata. The command which allows the build of a .sif Singularity image is:

```
sudo singularity build girafe.sif girafe-container.def
```

⚠One must have sudo rights on the system where the container is built. Once it is done, the image can be run anywhere and without sudo rights.

2.1 GIRAFE User interface

GIRAFE User interface encapsulates the FLEXPART architecture seen earlier (Figure 1) into a user-friendly "black box" designed specifically for the GIRAFE scenarios and its purpose of use. GIRAFE functioning can be summarized in following steps :

1. simulation parameters are defined by user through a configuration file `user-config.xml` (which can be renamed, but has to follow a predefined syntaxe)
2. Python script `girafe.py` reads the configuration file, writes input files for FLEXPART, compiles sources codes to obtain FLEXPART executable suited to the current simulation and launches the simulation
3.
4. if the simulation ran successfully, the simulation output is written and the post-processing is performed on the netCDF output (if available) to generate images with results visualization

A log-file is also created for every simulation and allows to back up every log and error message about the simulation running. This log output can also be redirected onto the command windows along with the log-file.

⚠The simulation must be launched inside the Singularity container.

2.2 Tool scripts and necessary files

2.2.1 Configuration file

Configuration file `user-config.xml` allows users to define simulation parameters, such as dates of the simulation, latitude-longitude grid of the simulation etc. This file should respect the template scheme (Annexe 1). If the user is not sure about one of the parameters, the default value, when possible, is indicated in the comments of the xml file. These settings summarize

all of the run-defining parameters needed for the FLEXPART input files (RELEASES, OUTGRID etc).

2.2.2 Python script

Python code `girafe.py` is the main script that the user has to launch to perform a simulation. This script has one mandatory input argument (`user-config.xml`) and one optional argument (output shell redirection) (Fig. 4). This script prepares the working directory for the simulation, writes FLEXPART input files, launches the simulation and performs the post-processing.

```
usage: girafe.py [-h] [-gc CONFIG] [--shell-log]

Python code that prepare all FLEXPART inputsand launch FLEXPART simulations
based on your configuration.xml and parameters.xml files where you configure
your simulation time, input data etc

options:

  -h, --help                                show this help message and exit
  -gc CONFIG, --config CONFIG               filepath to your configuration xml file
  --shell-log                               display log also in the shell
```

Figure 4 : Help display for the girafe.py script

In addition to these codes the user must have access to the meteorological data and the emission inventory files. The paths to these files have to be indicated in the configuration file `user-config.xml`.

2.2.3 Particle emissions products

As it was presented earlier, GIRAFE supports two types of the particle releases :

- fire detection via MODIS MCD14DL products
- emission inventories

The emission inventories contain concentrations of different emissions on the latitude-longitude grid, for various emissions sources (aviation emissions, volcanic SO2 emissions, biomass burning etc). Current version of GIRAFE supports the majority of the emission inventories. The requirements are : the file needs to be in the netCDF format, have variables “time”, “lat” and “lon” with these exact names, and the necessary emission variable must have units of `kg.m-2.s-1`. The amount of emissions will then be computed based on releases date/time/duration/location settings indicated in the configuration file and the data offered by the remission inventory. A variety of different inventories can, for example, be found on https://eccad.sedoo.fr [2].

The supported MCD14DL MODIS fire products contain latitude-longitude locations of thermal anomalies and active fires, along with their respective confidence levels. The MCD14DL products are distributed in the txt or csv extension. The Python pre-processing function will find the thermal anomalies within the releases lat-lon domain and with the same start date as the one requested by the user. The start *time* of each release will be taken from the MODIS product. However, the duration of these releases, and the minimum confidence level of the thermal anomalies detection must be indicated by the user. More information about this

collection can be found on https://www.earthdata.nasa.gov/learn/find-data/near-real-time/firms/mcd14dl-nrt [3].

2.3 GIRAFE simulation launch

The python script has to be executed inside the Singularity container where the GIRAFE was installed (`girafe.sif`). There are multiple methods to do so.

2.3.1 Interactive launch

If you want to launch the simulation in the interactive mode and be able to continue to work in the Singularity container, you can follow these steps:

1. launch Singularity container in the shell mode

```
singularity shell path/to/the/container.sif
```

2. launch Bash script with the simulation configuration

```
python3 /my/path/girafe.py \
--config /path/to/the/user-config.xml \
--shell-log
```

If the input data or your working directory are not exactly directories owned by your user, or can't be seen by the container (for example, if you are performing simulations on a server with multiple users and shared folders and volumes), you might need to bind these directories to the container with the `--bind` option. After the `--bind` keyword you can separate by commas a list of absolute paths to the folders that you would like to be seen by the Singularity container :

```
singularity shell \
--bind /root/input_data/, /home/working_folder/simu_x \
path/to/the/container.sif
```

After the bind, you can use same absolute paths to access these folders in the Singularity container. Based on your host system, most of the files and folders owned by your user will be seen, but sometimes special access may be required, and the `--bind` option can help you to link these folders to your container.

2.3.2 Non-interactive launch

It is also possible to launch the script in one go with the singularity exec command :

```
singularity exec path/to/the/container.sif \
python3 /my/path/girafe.py \
--config /path/to/the/user-config.xml \
--shell-log
```

The `--bind` option can be used here in the same manner as in the example above with singularity shell.

All of the methods described above can be wrapped into cron, bash scripts or slurm jobs depending on one's needs and number of simulations.

2.4 GIRAFE working directory structure

The working directory of the simulation is the directory where input files for the simulation, the executable for simulation launch and the output of the simulation will be stored. This directory follows the structure describe on the Fig.3:

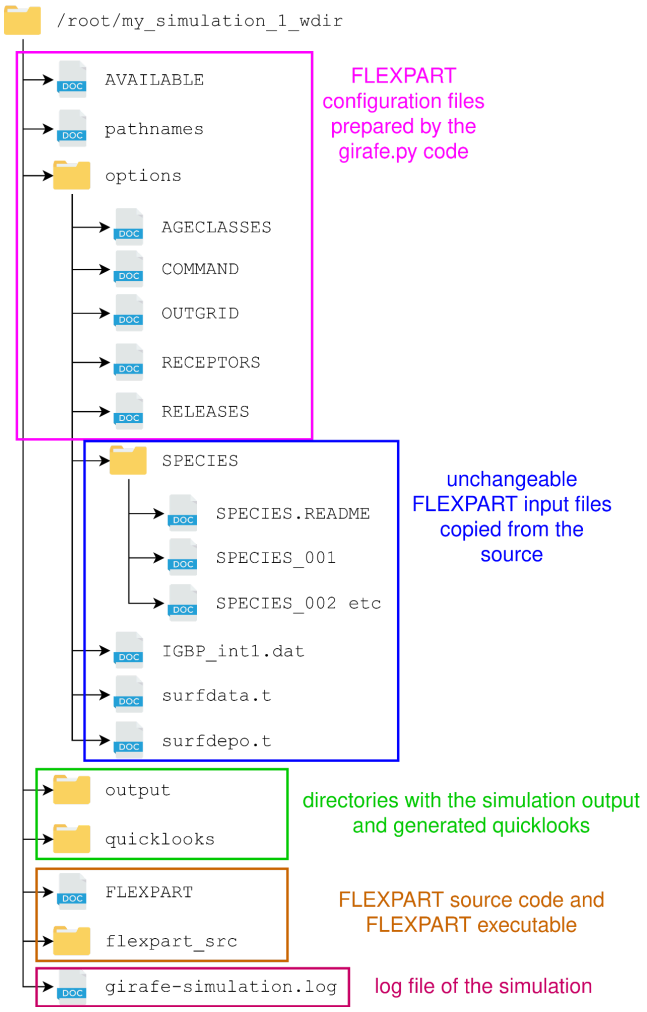

Figure 5 : Working directory structure for GIRAFE simulation

Bibliography

[1] Pisso, I., Sollum, E., Grythe, H., Kristiansen, N. I., Cassiani, M., Eckhardt, S., Arnold, D., Morton, D., Thompson, R. L., Groot Zwaaftink, C. D., Evangeliou, N., Sodemann, H., Haimberger, L., Henne, S., Brunner, D., Burkhart, J. F., Fouilloux, A., Brioude, J., Philipp, A., Seibert, P., and Stohl, A.: The Lagrangian particle dispersion model FLEXPART version 10.4, Geosci. Model Dev., 12, 4955–4997, https://doi.org/10.5194/gmd-12-4955-2019, 2019.

https://gmd.copernicus.org/articles/12/4955/2019/

[2] ECCAD, GEIA Global Emission InitiAtive data portal (AERIS, French data service for Atmosphere)

https://eccad.sedoo.fr

[3] MODIS/Aqua+Terra Thermal Anomalies/Fire locations 1km FIRMS V0061 NRT (Vector data), DOI: 10.5067/FIRMS/MODIS/MCD14DL.NRT.0061

https://www.earthdata.nasa.gov/learn/find-data/near-real-time/firms/mcd14dl-nrt

Annexes

Annexe 1 : girafe-config.xml

```
<config>
  <girafe>
    <!-- Current GIRAFE version (DO NOT CHANGE)-->
    <version>7.0</version>

    <simulation_date>
      <!-- simulation beginning date (YYYYMMDD) -->
      <begin>20130805</begin>
      <!-- simulation ending date (YYYYMMDD) -->
      <end>20130807</end>
      <!-- Time between requested ECMWF fields (3 or 6 hours) -->
      <dtime>6</dtime>
    </simulation_date>

    <simulation_time>
      <!-- simulation beginning time (HHMMSS) -->
      <begin>000010</begin>
      <!-- simulation ending time (HHMMSS) -->
      <end>140000</end>
    </simulation_time>

    <flexpart>
      <!-- FLEXPART root directory (DO NOT CHANGE)-->
      <root>/usr/local/flexpart_v10.4_3d7eebf/</root>

      <par_mod_parameters>
        <pi>3.14159265</pi>
        <r_earth>6.371e6</r_earth>
        <r_air>287.05</r_air>
        <nxmaxn>1</nxmaxn>
        <nymaxn>1</nymaxn>
        <nxmax>191</nxmax>
        <nymax>101</nymax>
        <nuvzmax>138</nuvzmax>
        <nwzmax>138</nwzmax>
        <nzmax>138</nzmax>
        <maxwf>50000</maxwf>
        <maxtable>1000</maxtable>
        <numclass>13</numclass>
        <ni>11</ni>
        <maxcolumn>3000</maxcolumn>
        <maxrand>2000000</maxrand>
      </par_mod_parameters>

      <outGrid>
        <!-- Longitude of the output grid [-180; +180]-->
```

```
        <longitude>
          <min>0.0</min>
          <max>17.05</max>
        </longitude>
        <!-- Latitude of the output grid [-90;+90]-->
        <latitude>
          <min>36.01</min>
          <max>45.5</max>
        </latitude>
        <!-- Horizontal resolution of the output grid (in degrees) -->
        <resolution>0.1</resolution>
        <!-- Height output grid levels in meters -->
        <height>
          <level>10.0</level>
          <level>100.0</level>
          <level>500.0</level>
        </height>
      </outGrid>

      <command>
        <!-- Factor by which time step must be smaller (put 5.0 for default)-->
        <ctl>-5.0</ctl>
        <!-- -1 = backward simulation, 1 = forward simulation -->
        <forward>1</forward>
        <time>
          <!-- Output every sssss seconds -->
          <output>3600</output>
          <!-- Time average of output every sssss seconds -->
          <averageOutput>3600</averageOutput>
          <!-- Interval of output sampling sssss seconds, higher stat. accuracy with shorter intervals -->
          <sampleRate>900</sampleRate>
          <!-- Time constant for particle splitting every sssss seconds (put 999999999 for default) -->
          <particleSplitting>999999999</particleSplitting>
          <!-- All processes are synchronized to this sssss seconds interval -->
          <synchronisation>900</synchronisation>
        </time>
        <!-- Reduction for time step in vertical transport, used only if ctl>1 (look for "ctl" above, put 4 for default) -->
        <ifine>4</ifine>
        <!-- Output type: 1]mass 2]pptv 3]1&2 4]plume 5]1&4, +8 for NetCDF output (e.g. "13" is 1&4 but in one netCDF output file
instead of binary output files) -->
        <iOut>9</iOut>
        <!-- Particle position output: 0]no 1]every output 2]only at the end 3]time averaged -->
        <ipOut>2</ipOut>
        <!-- Increase of ABL heights due to sub-grid scale orographic variations: 0]off 1]on (put 1 for default)-->
        <lSubGrid>1</lSubGrid>
        <!-- Switch for convection parameterization: 0]off 1]on (put 1 for default) -->
        <lConvection>1</lConvection>
        <!-- Switch for calculation of age spectra (needs AGECLASSES): 0]off 1]on (put 0 for default) -->
        <lAgeSpectra>0</lAgeSpectra>
        <!-- Warm start from particle dump (needs previous partposit_end file): 0]no 1]yes (put 0 for default) -->
        <ipIn>0</ipIn>
        <!-- Separate output fields for each location in the RELEASE file: 0]no 1]yes -->
        <iOfr>0</iOfr>
        <!-- Output of mass fluxes through output grid box boundaries -->
        <iFlux>0</iFlux>
        <!-- Switch for domain-filling, if limited-area particles generated at boundary: 0]no 1]yes -->
        <mDomainFill>0</mDomainFill>
        <!-- Unit to be used at the source: 1]mass 2]mass mixing ratio (current version support only "mass")-->
```


.

```
        <indSource>1</indSource>
        <!-- Unit to be used at the receptor: 1]mass 2]mass mixing ratio 3]wet depo 4]dry depo -->
        <indReceptor>1</indReceptor>
        <!-- Quasi-Lagrangian mode to track individual numbered particles: 0]no 1]yes -->
        <mQuasilag>0</mQuasilag>
        <!-- Output also for a nested domain: 0]no 1]yes -->
        <nestedOutput>0</nestedOutput>
        <!-- Output sensitivity to initial conditions (backward mode only): 0]off 1]concentration 2]mass mixing ratio -->
        <lInitCond>0</lInitCond>
        <!-- Output only for the lowest model layer, used with lInitCond=1 or lInitCond=2 -->
        <surfOnly>0</surfOnly>
        <!-- Skewed, not Gaussian turbulence in the convective ABL, need large CTL and IFINE -->
        <cblFlag>0</cblFlag>
    </command>

    <releases>
        <!-- ID number of FLEXPART species to simulate -->
        <!-- (2) O3, (3) NO, (4) NO2, (5) HNO3, (6) HNO2, (7) H2O2, (10) PAN, (11) NH3, (12) SO4-aero, (13) NO3-aero -->
        <!-- (14) I2-131, (15) I-131, (16) Cs-137, (17) Y-91, (18) Ru-106, (19) Kr-85, (20) Sr-90, (21) Xe-133 -->
        <!-- (22) CO, (23) SO2, (24) AIRTRACER, (25) AERO-TRACE, (26) CH4, (27) C2H6, (31) PCB28, (34) G-HCH, (40) BC -->
        <species> 22 </species>
        <!-- Minimum fire confidence (0-100%) if fire inventory is used -->
        <fire_confidence>70</fire_confidence>
        <!-- Dates and times of releases -->
        <!-- (for each different time and date of release you have to add a new <release> node with :
            name, start_date, start_time, end_time, altitude min&max, zones where to search for emissions (zones can be different for
different releases) -->
        <release name="Release1">
            <!-- Start date of the release YYYYMMDD-->
            <start_date>20130805</start_date>
            <!-- Start time of the emission DDHHMMSS from the above start_date -->
            <start_time>00000010</start_time>
            <!-- Duration of the emission DDHHMMSS from the above start_date and start_time -->
            <duration>00100000</duration>
            <!-- Minimum altitude of the emission -->
            <altitude_min>10</altitude_min>
            <!-- Maximum altitude of the emission -->
            <altitude_max>15</altitude_max>
            <!-- Zones where to search for releases part (CAMS grid is 0.1x0.1 deg so it is very simple to explose the number of releases
points if lat/lon of zones are very big -->
            <zones>
                <zone name="Rome">
                    <latmin>41.8</latmin>
                    <latmax>42.0</latmax>
                    <lonmin>12.2</lonmin>
                    <lonmax>12.6</lonmax>
                </zone>
                <zone name="Naples">
                    <latmin>40.8</latmin>
                    <latmax>40.9</latmax>
                    <lonmin>14.2</lonmin>
                    <lonmax>14.3</lonmax>
                </zone>
            </zones>
        </release>
    </releases>

    <!-- Receptors for which concentration at surface will be computed -->
```


.

```
      <receptor>
        <descriptif
          name="receptor1"
          latitude="6.1333"
          longitude="15.5000"
        >
        </descriptif>
        <descriptif
          name="receptor2"
          latitude="6.5000"
          longitude="15.5000"
        >
        </descriptif>
      </receptor>

      <ageclass>
        <!-- Ages are given in seconds, ages give the maximum time a particle is carried in the simulation (put 172800 for default)-->
        <class>172800</class>
      </ageclass>
    </flexpart>

    <paths>
      <!-- Wokring directory where the input/output FLEXPART files will be stored (except the GRIB data) -->
      <working_dir>/root/user/girafe/wdir</working_dir>
      <!-- Path to ECMWF data -->
      <ecmwf_dir>/ECMWF_data/</ecmwf_dir>
      <!-- Docker path to emission data -->
      <emissions>/data/emissions/CAMS-inventory.nc</emissions>
    </paths>
  </girafe>
</config>
```

Annexe 2 : girafe.py

```
usage: girafe.py [-h] [-gc CONFIG] [--shell-log]

Python code that prepare all FLEXPART inputsand launch FLEXPART simulations
based on your configuration.xml and parameters.xml files where you configure
your simulation time, input data etc

options:

  -h, --help                      show this help message and exit
  -gc CONFIG, --config CONFIG     filepath to your configuration xml file
  --shell-log                     display log also in the shell
```