Photochemistry and aerosol modeling

Polyphemus Training Session

About

Purpose: introduction to the Polyphemus system : preprocessing raw data and launching one simulation. Very basic post-processing is also presented.

Authors: Edouard Debry, `Edouard.DEBRY@ineris.fr`; Meryem Ahmed de Biasi, `Meryem.Ahmed_de_Biasi@inria.fr`; Yelva Roustan, `roustan@cerea.enpc.fr`.

Polyphemus version: 1.7.2

Location: `http://www.enpc.fr/cerea/polyphemus/sessions.html`

Contents

Introduction

Preprocessing, simulation and postprocessing are the three main steps of the Polyphemus system.

Before running one simulation it is needed to change raw data (meteorological fields, emissions, ...) into Polyphemus compliant format. Furthermore some fields, such as gas deposition velocities, can be precomputed in order to lower the computational burden. This is the preprocessing step.

The simulation step consists in compiling an executable with the desired compiler and in filling its configuration file.

Once the simulation is achieved you can display concentrations, either time series or maps, and compute various statistics against observations with postprocessing tools.

The goal of this training session is to make you perform a simulation over a European domain between the 9th of August at 01 a.m and the 13th August of 2004 at noon.

You are advised to extract the archive **aerosol.tar.bz2** in your home directory but you can do otherwise.

```
$ tar -xjvf aerosol.tar.bz2
```

This creates a directory `aerosol/` in `polyphemus-sessions/`, which has also been created if necessary.

Place the source `Polyphemus-1.7.2/` in `polyphemus-sessions/` also so that the paths given in configuration files are correct.

During the practical session, we will work in `polyphemus-sessions/aerosol/`.

In what follows some command lines might be divided by \, they must be typed as a single line and are only divided for clarity's sake.

1 Preprocessing

Ground data and meteorological fields must be computed first (in that order) as they are needed by most of subsequent steps.

All configuration files for preprocessing and simulation are located in `aerosol/config/`.

Almost all programs use the general configuration file `general.cfg`:

```
[general]

Directory_computed_fields: data
Directory_ground_data: <Directory_computed_fields>/ground
Programs: ../Polyphemus-1.7.2/preprocessing/

[domain]

Date: 2004-08-09_01
Delta_t = 1.
x_min = -10.    Delta_x = 2.0    Nx = 16
y_min = 36.     Delta_y = 2.0    Ny = 11
Nz = 3
Vertical_levels: config/levels.dat
```

The [general] section contains several relative paths which you would probably not need to change. You are advised to check them however. The [domain] section contains the characteristics of the simulation domain (grid spacing, number of vertical levels, ...) as well as the beginning date and time step (in hours) used in every preprocessed fields.

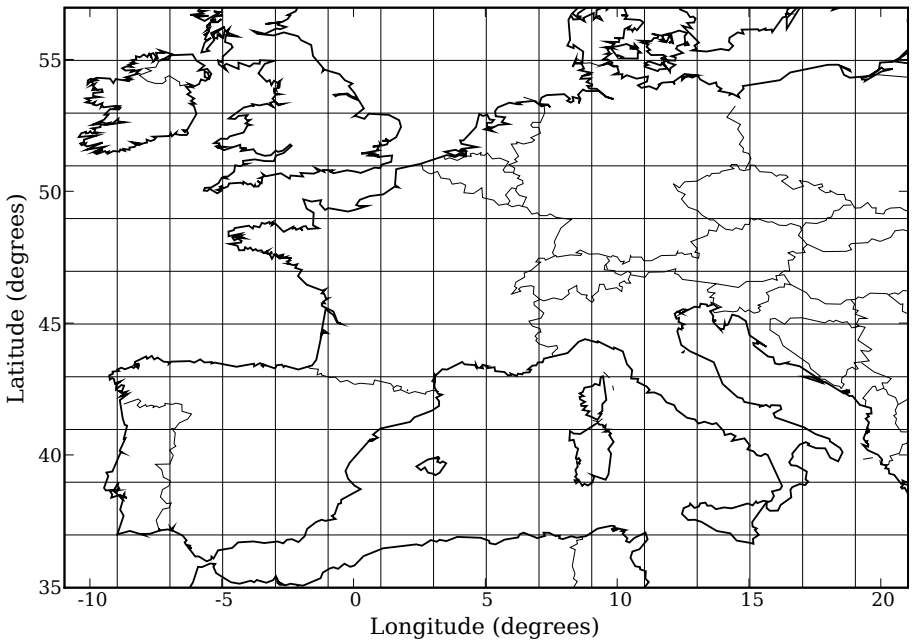

Figure 1: Domain and mesh considered for the simulation.

1.1 Ground Data

You first have to compute land use data needed by most other programs (e.g. meteo, deposition velocities). The program is luc-usgs which computes ground data from USGS (U.S. Geological Survey) database. The configuration file associated with this program is luc-usgs.cfg.

Note that in order to keep the archive small, USGS raw data is not included. You have to download it from http://edcftp.cr.usgs.gov/pub/data/glcc/ea/lamberte/eausgs2_0le.img.gz and http://edcftp.cr.usgs.gov/pub/data/glcc/af/lambert/afusgs2_0l.img.gz and to put it in raw_data.

Launch this executable:

```
$ cd polyphemus-sessions/aerosol
$ ../Polyphemus-1.7.2/preprocessing/ground/luc-usgs config/general.cfg config/luc-usgs.cfg
```

Here is the output you should get :

```
Reading configuration files... done.
Memory allocation for data fields... done.
```

```
Reading LUC data... done.
Building LUC data on output grid... done.

Writing output data... done.
```

This output shows the general structure of each Polyphemus program : reading configuration, allocating memory, computing, and writing output.

It is generally useful to convert LUC data into different formats. Convert the previously computed LUC data into "zhang" format which is used by the aerosol dry deposition model. The program is ../Polyphemus-1.7.2/preprocessing/ground/luc-convert.

```
$ ../Polyphemus-1.7.2/preprocessing/ground/luc-convert config/general.cfg \
config/usgs_to_zhang.cfg
```

Now compute the roughness data, needed to compute some meteorological fields, with program ../Polyphemus-1.7.2/preprocessing/ground/roughness and configuration file config/roughness.cfg.

Note that when possible the configuration file has the same name as the program.

Finally extract-glcf is used to compute land use using GLCF raw data on EMEP domain. It is necessary for emission computation and needs GLCF raw data from ftp://ftp.glcf.umiacs.umd.edu/glcf/Global_Land_Cover/Global/gl-latlong-1km-landcover/gl-latlong-1km-landcover.bsq.gz which you should put in raw_data.

1.2 Meteo

Meteorological fields (temperature, pressure, cloud attenuation, rain , ...) can be computed from ECMWF or MM5 raw data. We will compute meteorological fields from MM5 raw data with the program ../Polyphemus-1.7.2/preprocessing/meteo/MM5-meteo.

For this we need the MM5 raw data file from Polair3D test-case, which you can find at http://cerea.enpc.fr/polyphemus/test_case/MM5-2004-08-09.tar.bz2.

Put it in raw_data/ then "read" it with program ../Polyphemus-1.7.2/utils/MM5_var_list:

```
$ ../Polyphemus-1.7.2/utils/MM5_var_list raw_data/MM5-2004-08-09
```

Here is the output you should get:

```
Metadata (-999 means unknown):

OUTPUT FROM PROGRAM MM5 V3                                                  : 11
TERRAIN VERSION 3 MM5 SYSTEM FORMAT EDITION NUMBER                          : 1
TERRAIN PROGRAM VERSION NUMBER                                              : 6
TERRAIN PROGRAM MINOR REVISION NUMBER                                       : 0
COARSE DOMAIN GRID DIMENSION IN I (N-S) DIRECTION                           : 76
COARSE DOMAIN GRID DIMENSION IN J (E-W) DIRECTION                           : 86
MAP PROJECTION. 1: LAMBERT CONFORMAL, 2: POLAR STEREOGRAPHIC, 3: MERCATOR   : 1
...
```

From this output you can retrieve the kind of map projection, the number of time step and the time step of MM5 file. These values are needed in the configuration file for MM5-meteo. Check that they are adequate.

Then launch the program:

```
$ ../Polyphemus-1.7.2/preprocessing/meteo/MM5-meteo config/general.cfg \
config/MM5-meteo.cfg 2004-08-09_01 2004-08-13_01
```

The two last parameters are the initial date and end date for meteorological computation. The end date is excluded and you have to make sure that both dates are covered by the same MM5 file. The first time step of MM5 files usually gives unrealistic fields so it is better not to process it, which is why we start the computation at 01 am. After having processed raw data, it is generally a good idea to see how the results look like:

```
$ ../Polyphemus-1.7.2/utils/get_info_float data/meteo/Temperature.bin
```

This will output statistics about the temperature field.

Furthermore the size of most of meteorological files should be $4 \times Nt \times Nz \times Ny \times Nx$, where 4 is the size of a "float". As Nx, Ny and Nz are known from general.cfg, you can check the number of time steps which have been computed.

1.3 Vertical Diffusion

The vertical diffusion according to "Louis" parameterization is already computed from previous program, but if you want to use the "Troen and Mahrt" parameterization, you have to launch program `../Polyphemus-1.7.2/preprocessing/meteo/Kz_TM`:

```
$ ../Polyphemus-1.7.2/preprocessing/meteo/Kz_TM config/general.cfg \
config/MM5-meteo.cfg 2004-08-09_01 2004-08-13_01
```

As you can see, Kz_TM requires the same configuration file as MM5-meteo.

1.4 Emissions

Emissions are a crucial part in CTM inputs. They are divided in anthropogenic, biogenic and sea salt emissions.

1.4.1 Anthropogenic Emissions

Anthropogenic emissions are extracted from EMEP raw data, available at `http://www.ceip.at/` in the section "Emissions used in EMEP models". Download emissions for CO, NH3, NMVOC, NOx, SOx, PM2.5 and PMcoarse and make sure to have a file for each species called CO.dat, NH3.dat, NMVOC.dat, NOX.dat, SOX.dat, PM2.5.dat and PMcoarse.dat. Put them in `raw_data/EMEP-2004/`.

Download files with the following options:

- for all countries;
- for the year considered (here, 2004);
- for all activity sectors (SNAP);
- in format "Grid (50 km $\times$ 50 km), Semicolon-Separated";

- for one species at a time

EMEP space grids and speciation have to be mapped to Polyphemus grid and species. By looking into `emissions.cfg`, try to figure out where information about speciation and grid mapping is stored.

Emissions can only be computed for a day at a time. Use program call-dates to launch the program several times successively:

```
$ ../Polyphemus-1.7.2/utils/call_dates ../Polyphemus-1.7.2/preprocessing/emissions/emissions \
config/general.cfg config/emissions.cfg 20040809 20040813
```

Here is the output you should get for the first day:

```
Date: 20040809

Reading monthly factors... done.
Reading daily factors... done.
Grid correspondences EMEP / LUC / POLAIR... done.
Reading aggregation/speciation... done.
Reading vertical distribution... done.
Reading hourly factors... done.
Reading input emissions... done.
Computes emissions on Polair3D grid... done.
Computes final emissions... done.
Writing output emissions... done.
```

It is generally a good idea to check if emissions are realistic. Fields can be displayed with python postprocessing tools.

For example, make a symbolic link to the atmopy library in the current directory,

```
$ ln -s ../Polyphemus-1.7.2/include/atmopy .
```

and enter the *ipython* environment.

```
$ ipython
```

Then issue the following commands:

```
from atmopy.display import *

# Loads the European map
m = getm('config/disp.cfg')
# Loads SO2 surface emissions concentrations d
d = getd('config/disp.cfg', 'data/emissions/surface-emissions/SO2.bin')
# Display 2D field on the European map
disp(m, d[0,0])
```

Lines beginning with a # are comments, do not enter them.

This displays the SO2 surface emissions at first time step and first level.

1.4.2 Biogenic Emissions

Biogenic emissions consist mainly in isoprene, NO, α-pinene (API) and terpene (LIM), which are directly computed from meteorological fields.

```
$ ../Polyphemus-1.7.2/preprocessing/bio/bio config/general.cfg \
config/bio.cfg 2004-08-09_01 2004-08-13_01
```

As for meteorological fields, the second date is excluded. Here is the output you should get:

```
Reading and interpolating meteorological data... done.
Reading biogenic data... done.
Computing biogenic emissions... done.
Writing emissions... done.

-- Analysis


Isoprene.GetMax(): 4.48173
Isoprene.Mean(): 0.102155

NO.GetMax(): 0.00828077
NO.Mean(): 0.00175138

Max API = 0.152403
Mean API = 0.00854339

Max LIM = 0.0750641
Mean LIM = 0.00420796
```

Again you may want to check fields generated. In the ipython environment type:

```
from atmopy import *
# Loads API surface emissions concentrations
d = display.getd('config/disp.cfg','data/emissions/bio/API.bin')
# Computes the mean API surface emission over the domain at each time step
dd = stat.time_evolution(d,'mean')
# Display the API mean as a function of time step
display.plot(dd)
```

1.4.3 Sea Salt Emissions

Sea salt emissions consist in NaCl particles produced at ocean and sea surface by interactions between wind and waves. Several parameterizations exist to compute these emissions, we use that of Monahan. Sea salt emissions may also produce a significant amount a particle sulfate.

```
$ ../Polyphemus-1.7.2/preprocessing/emissions/sea_salt config/general.cfg \
config/sea_salt.cfg 2004-08-09_01 2004-08-13_01
```

Here is the output you should get:

```
Reading configuration files... done.
Memory allocation for data fields... done.

Generating sea-salt emissions from 2004-08-09 01:00 to 2004-08-13 01:00.
Reading and interpolating meteorological data... done.
Reading or computing dry radius... done.
Computing fluxes of sea salt aerosol for each size bin... done.
Writing fluxes of sea salt aerosol for each size bin... done.
```

1.5 Deposition Velocities

Gas deposition velocities can be precomputed in order to lower the computational burden at run time. This unfortunately cannot be done for aerosols as their deposition velocities strongly depend on their diameters which are not known before simulation.

```
$ ../Polyphemus-1.7.2/preprocessing/dep/dep config/general.cfg \
config/dep.cfg 2004-08-09_01 2004-08-13_01
```

Here is the output you should get:

```
Reading configuration files... done.
Memory allocation for data fields... done.
Extracting input data... done.
Computing meteorological fields... done.
Computing deposition velocities...
Done for O3
Done for NO
...
```

1.6 Boundary Conditions

1.6.1 Gas Boundary Conditions

Gas boundary conditions are derived from MOZART climatological data. Raw data consists in a series of files labeled `h00xx.nc` where xx is an integer between 40 and 77. Each file stands for ten days. xx can be computed using Equation 1.

$$xx = 40 + int\left[\frac{N_d + 6}{10}\right] \tag{1}$$

with N_d the number of days since the beginning of the year (0 for first January) and $int(x)$ represents the integral part of x.

You have to register on NCAR Community Data portal (`http://cdp.ucar.edu`) in order to download Mozart files. See Polyphemus User's Guide for more details. Registrations have to be assessed that is why it takes about a day.

If you do not want to download these data, initial and boundary conditions have already been generated in data/ic and data/bc.

If you want to generate them (which is advised), check config/ic.cfg and config/bc.cfg (or config/bc-dates.cfg) to make sure that the results are saved in data/ic2 and data/bc2.

The program ../Polyphemus-1.7.2/preprocessing/bc/bc is designed to process MOZART files entirely. So you first have to find the relevant MOZART file according to the simulation dates, download it and put it in raw_data/MOZART/. If the beginning and end dates of your preprocessing are not in the same MOZART file, you might need several MOZART files. MOZART species usually differ from the chemistry model used in Polyphemus. Then MOZART species have to be mapped onto the chemistry module (RACM for instance) ones.

```
$ ../Polyphemus-1.7.2/preprocessing/bc/bc config/general.cfg \
config/bc.cfg raw_data/MOZART/h0062.nc
$ ../Polyphemus-1.7.2/preprocessing/bc/bc config/general.cfg \
config/bc.cfg raw_data/MOZART/h0063.nc
```

You can also use ../Polyphemus-1.7.2/preprocessing/bc/bc-dates, which works essentially like bc, except that you give as arguments a range of dates and the program selects and processes all MOZART files to cover that range.

```
$ ../Polyphemus-1.7.2/preprocessing/bc/bc-dates config/general.cfg \
config/bc-dates.cfg 2004-08-09_01 2004-08-13_01
```

```
INFORMATION :
     ||
     ||     First mozart file : h0062.nc
     ||        The first day of the first Mozart file is : 2004-08-02 00:01
     ||
     ||     The year changes 0 time(s)
     ||
     ||
     ||     Last Mozart file : h0063.nc
     ||
Reading, interpolating and writing concentrations...
Maps input and output species...
Mozart file : raw_data/MOZART/h0062.nc
Memory allocation for data fields... done

Extracting temperature... done
Computing level heights in meter interpolating pressure and temperature... done

 done
Species:
Output: ALD; input: CH3CHO
   (min x: 0.00837364)   (min y: 0.00746624)   (min z: 0.00887433)
   (mean x: 0.110558)   (mean y: 0.0889936)   (mean z: 0.077862)
   (max x: 0.354423)   (max y: 0.418011)   (max z: 0.189317)
Output: API; input: C10H16
   (min x: 3.78604e-09)   (min y: 1.23185e-10)   (min z: 1.73976e-08)
```

```
   (mean x: 0.0267022)   (mean y: 0.0525682)   (mean z: 0.00193847)
   (max x: 0.281594)   (max y: 2.00493)   (max z: 0.0149573)
...
```

Note the first day of the first Mozart file, as it is necessary in the simulation data file.

1.6.2 Aerosol Boundary Conditions

Aerosol boundary conditions are provided by the GOCART global scale model (Global Ozone Chemistry Aerosol Radiation Transport). If you want to retrieve data from this model, you will have to contact Mian Chin or Paul Ginoux at the web page `http://acdb-ext.gsfc.nasa.gov/People/Chin/gocartinfo.html`. In this case put it in `raw_data/GOCART/`.

GOCART files all stand for one month, the program `../Polyphemus-1.7.2/preprocessing/bc/bc-gocart` is thus designed to process one file at a time in the same way as `../Polyphemus-1.7.2/preprocessing/bc/bc`. Further information about GOCART files and their format is available in the Polyphemus User Guide in section 3.8.3.

Unlike other programs, `../Polyphemus-1.7.2/preprocessing/bc/bc-gocart` has four configuration files, one for each kind of GOCART species:

bc-gocart-CC.cfg for carbonaceous species, including organic matter.

bc-gocart-DU.cfg for dust.

bc-gocart-SU.cfg for sulfate.

bc-gocart-SS.cfg for sea salt species.

These configuration files have the same structure. try for example to understand what `[input_species]` means and how the mapping between GOCART and Polyphemus aerosol species is done.

The program `../Polyphemus-1.7.2/preprocessing/bc/bc-gocart` has to be launched four times for each period. Launch it successively for CC, DU, SU and SS GOCART species.

The GOCART files needed are in `raw_data/GOCART/`. The Gocart date is 200408.

The command line for carboneous species is:

```
$ ../Polyphemus-1.7.2/preprocessing/bc/bc-gocart config/general.cfg \
config/bc-gocart-CC.cfg raw_data/GOCART/200408.CC.STD.tv15.g.day 200408 5
```

You may notice that GOCART does not provide boundary conditions for particle nitrate and ammonia. We provide the program `../Polyphemus-1.7.2/preprocessing/bc/bc-nh4` which computes boundary conditions for particle ammonia, assuming electroneutrality. This program has to be launched after generation of all other boundary conditions over the whole period.

1.7 Initial Conditions

1.7.1 Gas Initial Conditions

Gas initial conditions are derived from MOZART raw data in the same way as boundary conditions, with program `../Polyphemus-1.7.2/preprocessing/ic/ic`.

The Mozart file needed is computed by the program according to Equation 1.

```
$ ../Polyphemus-1.7.2/preprocessing/ic/ic config/general.cfg config/ic.cfg
```

1.7.2 Aerosol Initial Conditions

Aerosol initial conditions are derived from typical aerosol concentrations found in EMEP report "A European Aerosol Phenomenology" available at `http://ccu.jrc.it/Pubblications/Phenomenology.pdf`. Note that the Polyphemus system can run without any aerosol initial conditions, that is to say no aerosols in atmosphere at beginning.

2 Simulation

2.1 Gaseous Simulation

First a simulation without aerosol species will be performed.

Polyphemus has one configuration file `config/polair3d.cfg` which contains all information about the simulation (time, domain, ...), options of the simulations, characteristics of input data (those previously precomputed) and which species to save. For more clarity the last two parts are stored in separate files, respectively `polair3d-data.cfg` and `polair3d-saver.cfg`.

Once you are sure of your configuration files, you can call the main program `../Polyphemus-1.7.2/processing/racm/polair3d` with its main configuration file (`config/polair3d.cfg`). This configuration file contains the path to the other files, so you should not put them in command line.

2.2 Aerosol Simulation

Here we will use program `../Polyphemus-1.7.2/processing/siream/polair3d-sieam`. To have polair3d-sieam working it is necessary to install ISORROPIA (see Polyphemus User's Guide).

The main configuration file for this program is `config/aerosol-polair3d.cfg`.

3 Going Further

3.1 Changing the Domain

The main change you may want to do is to adapt Polyphemus to one particular domain. To change the size or resolution of the domain implies to recompute all preprocessed data.

What are the configuration files you need to change? (*Hint:* there are two of them)

3.2 Changing the Number of Aerosol Bins

When dealing with size-resolved aerosol models, is is necessary to specify the desired number of sections. The default configuration runs with 5 bins. Try to increase the number of bins up to 10. First you will have to recompute aerosol input data. This affects only emissions, boundary conditions and initial conditions. Modify the number of bins to 10 each time it is encountered in preprocessing configuration files. For emissions you should look into `emissions/input/speciation` where EMEP species are directly mapped to aerosol bins and species.

Once you have recomputed input data, you need to reconfigure polair3d in order to use 10 bins. You do not have to recompile polair3d.