Gaussian Test-Case

Irène Korsakissok and Meryem Ahmed de Biasi

This document explains how to proceed to perform simulations using the test case for Gaussian models provided with Polyphemus.

When the archive `TestCase-1.0-Gaussian.tar.bz2` is extracted a directory `TestCase-1.0-Gaussian/` is created. It is referred to below as `TestCase`.

```
tar -xjvf TestCase-1.0-Gaussian.tar.bz2
```

The subdirectory `config/` holds all configuration files necessary and the subdirectory `results/` is meant to store the results of simulations. It is divided in three subdirectories (one for each possible simulation) : `puff_line/` for the Gaussian puff model and a gaseous line source, `puff_aer/` for the puff model with point sources of gaseous and aerosol species, and `plume/` for the Gaussian plume model with gaseous species only. In each of those subdirectory a python program which allows to visualize some results easily can be found.

To launch the test cases, you do not need to modify the configuration files. You just have to make sure to replace `Polyphemus/` and `TestCase/` by your path to the last version of Polyphemus and the Gaussian test case respectively.

1 Preprocessing

Prior to use Gaussian models, you need to compute scavenging coefficients and deposition velocities for the various species. This is achieved by using `gaussian-deposition_aer`.

First compile it :

```
cd ~/Polyphemus/preprocessing/dep/
make gaussian-deposition_aer
```

Then run it from the test case directory :

```
cd ~/TestCase/
~/Polyphemus/preprocessing/dep/gaussian-deposition_aer
                        config/gaussian-deposition_aer.cfg
```

The output on screen will be :

```
Reading configuration file... done.
Reading meteorological data... done.
Reading species... done.
Reading diameter... done.
Computation of the scavenging coefficients... done.
Computation of the deposition velocities..done.
Writing data... done.
```

The file `gaussian-meteo_aer.dat` has been created in the directory `TestCase/config/`. It will be used for all simulations.

Note that if your simulation only involves gaseous species, you can use the preprocessing program `gaussian-deposition`. Here we use `gaussian-deposition_aer` because its output can be used for simulations with or without aerosol species.

2 Discretization

This step is only necessary for the simulation with a line source. Its aim is to discretize this source into a series of puffs. To do so, compile the preprocessing program `discretization` :

```
cd ~/Polyphemus/preprocessing/emissions/
make discretization
```

Then run it from the test case directory :

```
cd ~/TestCase/
~/Polyphemus/preprocessing/emissions/discretization
                                config/discretization.cfg
```

The output on screen will be :

```
Reading configuration file... done.
Reading trajectory data... done.
Length of the trajectory: 48.0278
Number of points on the trajectory: 26
Writing source data... done.
```

The file `puff-source-discretized.dat` has been created in the directory `TestCase/config`. It contains a series of puffs representing the discretized line source.

3 Simulations

3.1 Plume

This simulation uses the program `plume`, which is the program for the Gaussian plume model. It uses the following data :

- Gaseous species : Caesium, Iodine.
- Sources : 2 point sources for Iodine, one point source for Caesium.
- Meteorological situations : 4 situations, rotating wind with an increasing speed (0.1m/s, 2m/s, 5m/s et 10m/s).
- Urban environment.

The simulation uses the following files :

- `plume.cfg` gives the simulation domain, the options and the paths to the other files.
- `gaussian-levels.dat` gives the vertical levels.
- `gaussian-species_aer.dat` gives all meteorological data and scavenging and deposition coefficients. It was created during preprocessing (see Section 1).
- `plume-source.dat` contains all the data on stationary sources.
- `plume-saver.cfg` contains the options and paths to save the results.

Compile the program plume :

```
cd ~/Polyphemus/driver/
make plume
```

Then execute it from `TestCase/` :

```
cd ~/TestCase/
~/Polyphemus/driver/plume config/plume.cfg
```

The output on screen will be :

```
        Temperature     Wind angle      Wind velocity   Stability
Case #0
        15              -100            0.5             D
Case #1
        10              -5              2               D
Case #2
```

```
        10              20              5               D
Case #3
        10              60              10              D
```

Results are stored in results/plume/.

3.2 Puff with Aerosol Species

The simulation uses puff_aer, which is the program for puffs with aerosol species, and the following data :

- Gaseous species : Caesium, Iodine.
- Aerosol species : aer1, aer2.
- Sources : 1 point source per species.
- Meteorological situations : 4 situations, rotating wind with an increasing speed (0.1m/s, 2m/s, 5m/s et 10m/s).
- Urban environment.

The simulation uses the following files :

- puff_aer.cfg gives the simulation domain, options and the paths to the other files.
- gaussian-levels.dat gives the vertical levels.
- gaussian-species_aer.dat gives all meteorological data and data on scavenging and deposition. It was created during preprocessing (see Section 1).
- puff-source_aer.dat contains all the data on gaseous and aerosol sources.
- puff-saver_aer.cfg contains the options and paths to save the results.

Compile the program puff_aer :

```
cd ~/Polyphemus/driver/
make puff_aer
```

Then execute it from TestCase/ :

```
cd ~/TestCase/
~/Polyphemus/driver/puff_aer config/puff_aer.cfg
```

Results are stored in results/puff_aer/.

3.3 Puff with Line Source

The simulation uses `puff`, which is the program for puffs with gaseous species only, and the following data :

- Gaseous species : Iodine.
- Source : 1 line source.
- Meteorological situations : 4 situations, rotating wind with an increasing speed (0.1m/s, 2m/s, 5m/s et 10m/s).
- Urban environment.

The simulation uses the following files :

- `puff.cfg` gives the simulation domain, options and the paths to the other files.
- `gaussian-levels.dat` gives the vertical levels.
- `gaussian-species_aer.dat` gives all meteorological data and scavenging and deposition coefficients. It was created during preprocessing (see Section 1).
- `puff-source-discretized.dat` gives data on the discretized source. It has been created using program `discretization` (see Section 2).
- `puff-saver.cfg` gives the options and paths to save the results.

Compile the program `puff` :

```
cd ~/Polyphemus/driver/
make puff
```

Then execute it from `TestCase/` :

```
cd ~/TestCase/
~/Polyphemus/driver/puff config/puff.cfg
```

Results are stored in `results/puff_line/`.

4 Result Visualization

4.1 Gaussian Plume

Python scripts are provided to display easily and quickly the results of a simulation. For the plume simulation, just launch :

```
cd ~/TestCase/
python results/plume/display_plume.py
```

It creates 5 figures in `results/plume/` :

- `plume_gas_max` gives the repartition of the maximum of concentration of Iodine for all meteorological situations.
- `plume_gas_meteo1` gives the concentration for the first meteorological situation.
- `plume_gas_meteo2` gives the concentration for the second meteorological situation.
- `plume_gas_meteo3` gives the concentration for the third meteorological situation.
- `plume_gas_meteo4` gives the concentration for the fourth meteorological situation.

Figure 1 shows the result of `display_plume.py`. Note that you can replace `Iodine.bin` with `Caesium.bin` in `results/plume/disp.cfg` to display the results for Caesium.

4.2 Gaussian Puff with Aerosol Species

Launch the python script to display the results :

```
cd ~/TestCase/
python results/puff_aer/display_puff.py
```

This creates 4 figures in `results/puff_aer/` :

- `puff_aer_meteo1` shows the puff at $t = 0s$, $t = 3s$ and $t = 8s$ for the first meteorological situation.
- `puff_aer_meteo2` shows the puff at $t = 0s$, $t = 3s$ and $t = 8s$ for the second meteorological situation.
- `puff_aer_meteo3` shows the puff at $t = 0s$, $t = 3s$ and $t = 8s$ for the third meteorological situation.
- `puff_aer_meteo4` shows the puff at $t = 0s$, $t = 3s$ and $t = 8s$ for the fourth meteorological situation.

Figure 2 shows what is displayed. It shows how the puff evolves in time. By default, the species displayed is the first aerosol species and the first diameter class (`aer1_0`). Just modify the file `results/puff_aer/disp_puff.cfg` to display other species or diameters.

4.3 Gaussian Puff with Line Source

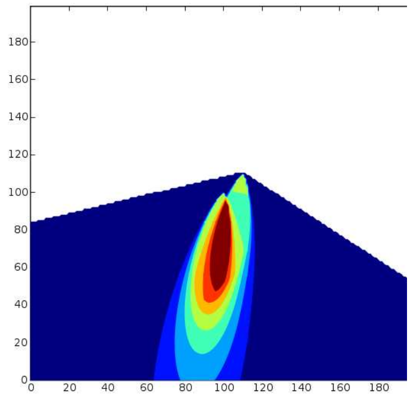
Launch the python script to display the results :

```
cd ~/TestCase/
python results/puff_line/display_puff.py
```

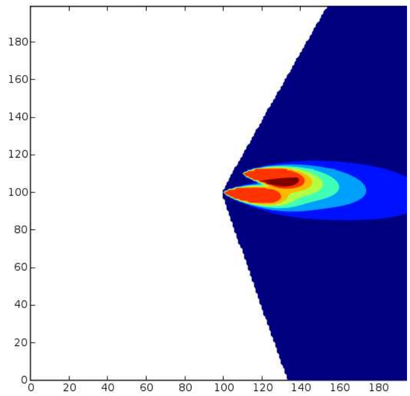
This create 4 figures in `results/puff_line/` :

- `linepuff_meteo1` gives the concentration for the first meteorological situation.
- `linepuff_meteo2` gives the concentration for the second meteorological situation.
- `linepuff_meteo3` gives the concentration for the third meteorological situation.
- `linepuff_meteo4` gives the concentration for the fourth meteorological situation.

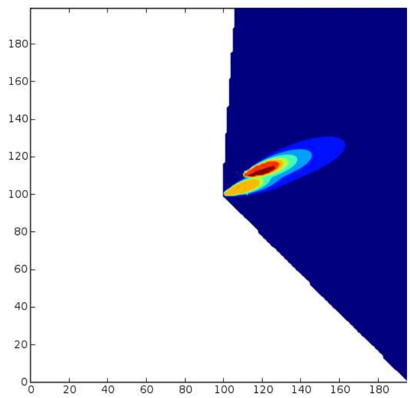
Concentrations are given for Iodine at t = 4s (middle of the simulation). Figure 3 shows what is displayed.



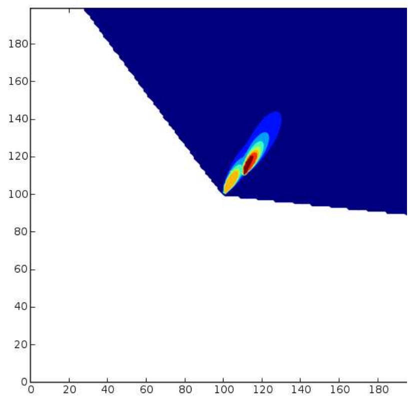
(a) Situation 1



(b) Situation 2

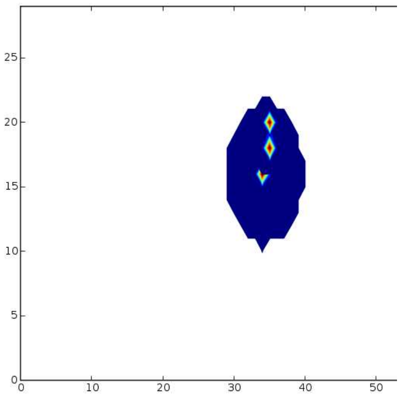


(c) Situation 3

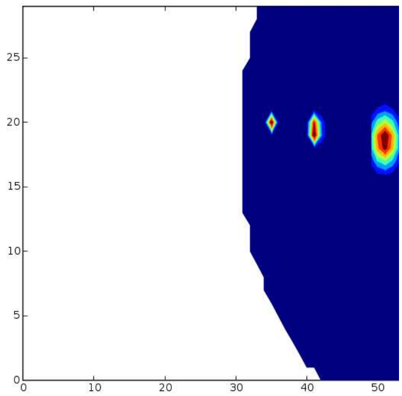


(d) Situation 4

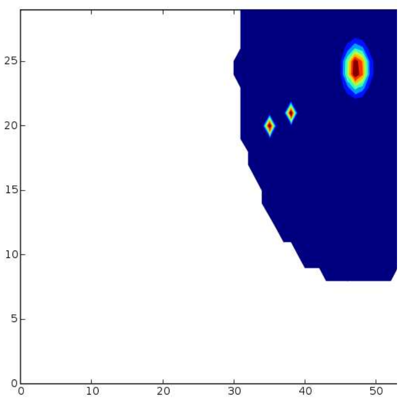
FIG. 1 – Plume for the various meteorological situations.



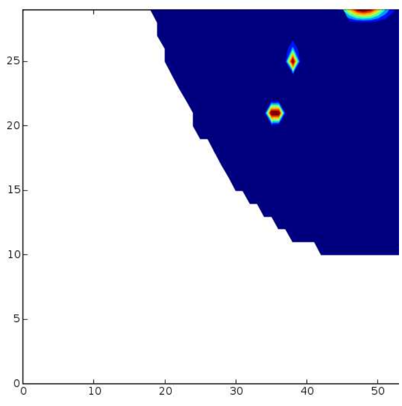
(a) Situation 1 - Concentration at $t = 0s$, $t = 3s$ and $t = 8s$



(b) Situation 2 - Concentration at $t = 0s$, $t = 3s$ and $t = 8s$

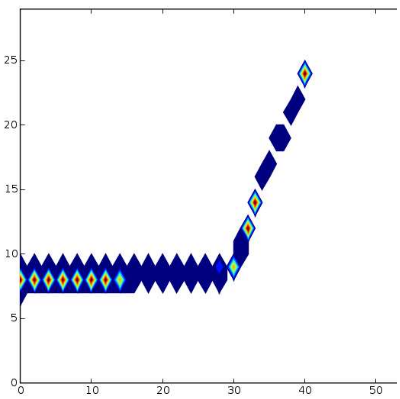


(c) Situation 3 - Concentration at $t = 0s$, $t = 0.5s$ and $t = 2.5s$

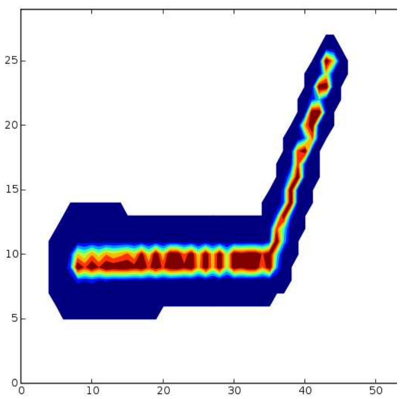


(d) Situation 4 - Concentration at $t = 0s$, $t = 0.5s$ and $t = 2.5s$

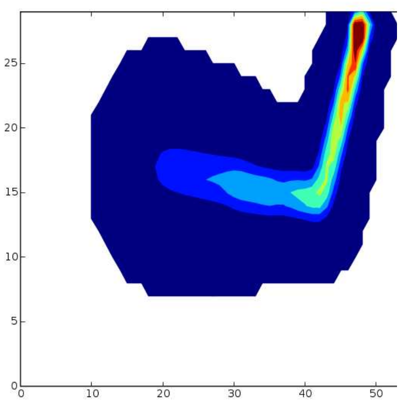
FIG. 2 – Evolution of the puff for the various meteorological situations.



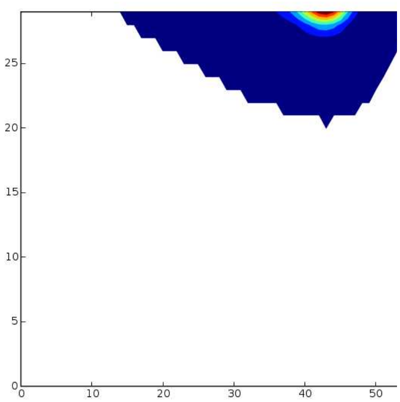
(a) Situation 1



(b) Situation 2



(c) Situation 3



(d) Situation 4

FIG. 3 – Concentration at t = 4s for the various meteorological situations.