Computer Practical: Gaussian Plume Model

Paul Connolly, October 2017

1 Overview

In this handout we look at the problem of advection and turbulent diffusion of material from a point source, such as a industrial stack. The result is referred to as a Gaussian Plume model and has been implemented in MATLAB (or Python).

Take note: the best way to work through this practical is to follow the instructions to generate the figures and export the figures to image files as you go along. You can then insert them into a PowerPoint or document so that they can be easily compared later.

2 Gaussian Plume

2.1 Governing Equation and Solution

We start with the advection-diffusion equation in 3-D:

$$\frac{\partial C}{\partial t} + \frac{\partial uC}{\partial x} + \cancel{\frac{\partial vC}{\partial y}} + \cancel{\frac{\partial wC}{\partial z}} = \cancel{K_x \frac{\partial^2 C}{\partial x^2}} + K_y \frac{\partial^2 C}{\partial y^2} + K_z \frac{\partial^2 C}{\partial z^2} \tag{1}$$

we have made an assumption that advection is the dominant term along wind and are able to cancel other terms accordingly. We also make the assumption that the wind, u, is constant and steady:

$$u\frac{\partial C}{\partial x} = K_y \frac{\partial^2 C}{\partial y^2} + K_z \frac{\partial^2 C}{\partial z^2} \tag{2}$$

The above equation can be solved for a point source using advanced mathematical techniques. Thankfully, it has been solved for us and the solution is:

$$C(x,y,z) = \frac{Q}{2\pi u \sigma_y \sigma_z} \exp\left(-\frac{y^2}{2\sigma_y^2}\right)\left[\exp\left(-\frac{(z-H)^2}{2\sigma_z^2}\right) + \exp\left(-\frac{(z+H)^2}{2\sigma_z^2}\right)\right] \tag{3}$$

where, for the standard case, $\sigma_i^2 = \frac{2K_i x}{u}$.

In fact the values of standard deviation, σ_y and σ_z, depend on the atmospheric stability. The form of the functions are not important here but the important point is that *unstable conditions have standard deviations that rapidly increase downwind* and *stable conditions have standard deviations that stay small downwind*. Hence, in stable conditions the pollutant may travel long distances before dispersing. Figure 1 illustrates the situation being modelled.

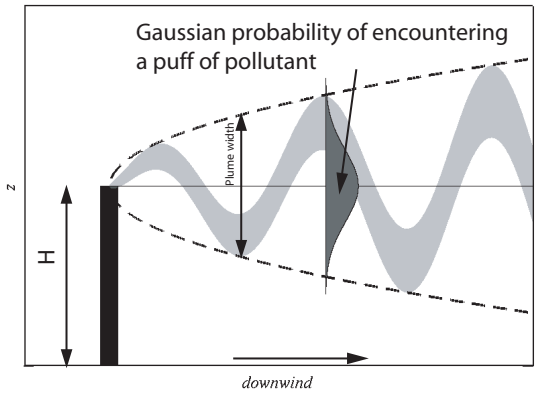

Figure 1: Schematic of the scenario being modelled, where H is the height of the stack.

Table 1: Parameters controlling the behaviour of the model.

Variable	Default value	Description
RH	0.90	Relative humidity of the air
aerosol_type	SODIUM_CHLORIDE	Composition of aerosol particles considered
dry_size	60×10^{-9} m	Dry diameter of aerosol particles assumed
humidify	DRY_AEROSOL	Flag to decide whether to grow the aerosol (Köhler equations)
stab1	1	Vertical stability parameter—set from 1 to 6
stability_used	CONSTANT_STABILITY	Run using a set stability or an annual cycle
output	PLAN_VIEW	How to output results
x_slice	26	If outputting a vertical slice plot along this position
y_slice	1	If outputting a time-series plot at x_slice and this position
wind	PREVAILING_WIND	Assumption for input wind field
stacks	ONE_STACK	Whether to have 1, 2 or 3 stacks
stack_x	[0 1000 -200]	x-position of each stack (m)
stack_y	[0 250 -500]	y-position of each stack (m)
Q	[40 40 40]	mass in grams s^{-1} emitted from each stack
H	[50 50 50]	height (m) of each stack
days	50	model run-time in days

3 Running the model

The MATLAB (or Python) files required are gaussian_plume_model.{m,py}, a script that runs the model, gauss_func.{m,py}, a function which implements the Gaussian plume solution, calc_sigmas.{m,py}, a function which calculates the standard deviation of the plume based on distance from the stack and atmospheric stability. Later experiments also require the script overlay_on_map.{m,py} and the data file map_green_lane.mat.
The procedure for running the model is similar for both MATLAB and Python.
If using Python we recommend using IPython (interactive Python) and run the scripts by typing (for example) run -i gaussian_plume_model.py.
The instructions below describe running the model in MATLAB:

1. Edit Section 1 of gaussian_plume_model.m to configure the model on lines 60-81 and save it.
2. Type gaussian_plume_model at the MATLAB prompt to run the model.

Table 1 describes the parameters in Section 1 of gaussian_plume_model.m that can be modified and Table 2 the possible values for the vertical stability parameter, although obviously you can change any part of the code if you want to play. Note that stack_x, stack_y, Q and H are set for each stack even when stacks is set to ONE_STACK. This is because the code selects the ones to use based on the value of stacks.

4 Experiments

4.1 Wind direction

Firstly, lets look at the effect that the assumptions about wind direction have on the dispersion of pollutants. Normally the wind speed and wind direction would be read into an air quality model / Gaussian plume model and be taken from either observational data or from a forecast product; however, here we generate a synthetic dataset by either: (1) having the wind come from a constant direction; (2) having the wind come from a completely random direction and (3) having the wind come from a prevailing direction, with some variation either side. We will do this for *neutral* vertical stability (i.e. stab1 = 4).

Table 2: Possible values of the stability parameter, `stab1`

Value of `stab1`	Vertical stability
1	Very unstable
2	Moderately unstable
3	Slightly unstable
4	Neutral
5	Moderately stable
6	Very stable

The settings to be changed from the defaults are as follows:

- `stab1 = 4;`
- `wind = CONSTANT_WIND;`

put these setting in Section 1 and save the file.

Exercise: Compare the effects of the different wind direction assumptions. Do the following:

First type `gaussian_plume_model` at the MATLAB prompt. The model will run and plot your results for the constant wind assumption (save the figure as an image file for revision later).

Now set the following value:

- `wind = FLUCTUATING_WIND;`

and save the file. Now type `gaussian_plume_model` at the MATLAB prompt to see the results for the random wind direction plotted (again save the figure as an image file).

Lastly set the following value:

- `wind = PREVAILING_WIND;`

and save the file. Finally type `gaussian_plume_model` at the MATLAB prompt to see the results for the prevailing wind direction plotted (save your figure for revision).

Can you explain how the wind direction and variability affects the ground level values of pollutant from the stack?

4.2 Multiple stacks

Often a development can have multiple stacks. Here we will experiment to see the effects that having multiple stacks has on ground level concentrations.

Exercise: To see the effects of this keep `wind = PREVAILING_WIND` and change the following setting:

- `stacks = TWO_STACKS;`

save the file. Then type `gaussian_plume_model` at the MATLAB prompt to see the results for the concentrations due to two stacks (save your figure for revision). Compare this to the result for one stack, with a prevailing wind direction.

Repeat this process for:

- `stacks = THREE_STACKS;`

if you would like to experiment you may want to move the positions and heights of the stacks to different values.

In general terms what is the effect of having multiple stacks on the ground level pollutant concentration?

4.3 Vertical stability—plan view

We want to see what effect vertical stability of the atmosphere has on the ground level concentrations. In the model this is set using the `stab1` parameter and possible values and there meaning are shown in Table 2.

Exercise: Change back to just one stack and set the stability to *Very unstable*:

- `stacks = ONE_STACK;`
- `stab1 = 1;`

save the file. Then run the model by typing `gaussian_plume_model` at the MATLAB prompt. Repeat this process for each of the stability parameters in Table 2 (i.e. change `stab1` to each of the possible values and run `gaussian_plume_model.m`.

How does the vertical stability affect the distribution of pollutant at the ground? Explain your answer with reference to the movement of the air as it exits the stack.

4.4 Vertical stability—vertical structure

Ground level concentrations are one part of the story. To understand the effect of vertical stability in more detail we now look at the pollutant concentration in a vertical y-z slice. In the model this is set using the `output` parameter.

Exercise: Change the model to output a vertical slice and set the stability to *Very unstable*:

- `output = HEIGHT_SLICE;`
- `stab1 = 1;`

save the file. Then run the model by typing `gaussian_plume_model` at the MATLAB prompt. As before repeat this process for each of the stability parameters in Table 2 (i.e. change `stab1` to each of the possible values and run `gaussian_plume_model.m`.

How does the vertical stability affect the vertical distribution of pollutant? Explain your answer with reference to the movement of air, lapse rate and dispersion of the particles. Refer to the phenomena of *looping, coning and fanning*.

4.5 Annual cycle in vertical stability

Until now we have assumed that the vertical stability is a constant for all times. In actual fact the vertical stability is constantly changing with the time of day and the seasons. For instance un-stable air (air that will result in thunderstorms for instance) is more likely in the afternoon and in the summer. Stable air is more likely at night and in the winter. Here we run the model for a whole year and choose the option that sets the stability to change from very stable air in the winter to unstable air in the summer.

Note that the `SURFACE_TIME` ensures pollutant concentrations are plotted at the points specified by `x_slice` and `y_slice`, which correspond to $x = 0$ and $y = -2500$.

Exercise: At this point you may want to return all values to the defaults in Table 1. Then set the following parameters:

- `output = SURFACE_TIME;`
- `stability_used = ANNUAL_CYCLE;`
- `x_slice = 26;`
- `y_slice = 1;`
- `days = 365;`

save the file. Then run the model by typing `gaussian_plume_model` at the MATLAB prompt. It will take a little longer to run because you are now running for a year. You will get two plots: one of the mass loading of pollutant and one of the vertical stability parameter, which is unstable in the middle of the year (summer) and stable at the beginning and end of the year (winter).

How does the vertical stability affect the time series at a point of the pollutant? Refer to *looping, coning and fanning*, but also distance from the stack when explaining your findings.

4.6 Köhler equations and map over-lay

Local councils and governments may receive fines if certain pollutants exceed set thresholds. For particulate matter these thresholds are based on the *mass* of material in the atmosphere and not the number concentration (usually measured in μg m^{-3}.

As particulate matter exits the stack the air is usually warm and hence the humidity can be low. As the air cools through mixing with the environment the humidity in the plume may increase in places. Hence, we will look at the effect of humidity on the growth of the aerosol particles and the resulting particulate matter.

We will also plot the output as contours onto a map and hence you will need the `overlay_on_map.m` and `map_green_lane.mat` files. We will suppress output initially and then plot onto a map using the `overlay_on_map.m` plotting script.

Exercise: Set the following parameters:

- `output = NO_PLOT;`
- `stab1 = 4;`
- `stability_used = CONSTANT_STABILITY;`
- `x_slice = 26;`
- `y_slice = 1;`
- `days = 50;`

save the file. Then run the model by typing `gaussian_plume_model` at the MATLAB prompt. Now plot the data on a map by typing `overlay_on_map` on the MATLAB prompt.

Now look at what the effect of humidity on particulate matter is. Set the following parameters:

- `humidify = HUMIDIFY;`

save the file, run the model: `gaussian_plume_model` and plot the output on a map: `overlay_on_map`.

See what effect the aerosol particle chemistry has by setting the following parameters:

- `humidify = HUMIDIFY;`
- `aerosol_type = ORGANIC_ACID;`

save the file, run the model: `gaussian_plume_model` and plot the output on a map: `overlay_on_map`.

Comment on the effect that humidification has on the particulate matter at the surface and the effect that aerosol chemistry (i.e. the choice of sodium chloride or organic acid) has on particulate matter. Do you know why the aerosol chemistry has the effect it does?