**PO**int and **P**uff Concentration Measurements Analysis **P**rogram

Potentielle Namen: POPP, PAPE (beide in verschiedenen zusammenhängen mit WOTAN)

Punkt und Puff Konzentrationsmessungsauswertungsprogramm

Potentielle Namen:

User’s Guide

**Summary:**

This program, written in python, analyses and creates plots and statistical descriptions of boundary layer wind tunnel concentration measurements. The program differentiates between two types of concentration measurements, namely between a continuous release measurement (where the tracer gas is released continuously over time) and a puff release measurement (where the tracer gas is released in many regularly repeated short intervals, called puffs).

1. **Requirements**

Python 3.5.2, with the following packages installed:

numpy v1.13.1

scipy v0.19.1

pandas v0.20.3

logging v0.5.1.2

matplotlib v1.5.3

Newer versions of python and the required packages should not impact the functionality of the program, please contact the developers if you encounter any issues with versions of python or the above packages newer than the ones listed above. However, backwards compatibility (i.e. compatibility with older versions), especially with python 2.x or older, cannot be guaranteed!

The program was tested using Spyder 3.2.3 on a Computer running Windows 10, but should work on any distribution that supports the above.

1. **Getting Started**

Start by opening the file example\_point\_measurement.py (or example\_puff\_measurement.py for puff measurement, if you wish), either in spyder or using a text editor. Enter the path (variable ‘path’) and filename(s) (variable ‘namelist’) of the dataset you wish to analyze. You should also specify the a priori information located below the namelist variable. While the program will in theory run without this, correct a priori information is required for physically correct results. Of particular importance are the variables, x, y, z, pressure, temperature, scaling factor, scale, ref\_length, and ref\_height. Now you can run whichever file you just edited (in spyder just click ‘run’ in the top toolbar), and the program should output you the results of your analysis.

Note that if you want to analyse more than one dataset (i.e. if you have more than one entry in namelist), you may want to consider inputting the a priori information into a csv file. The csv file has several formatting requirements, so a sample csv file containing the a priori information is located in the /windtunnel subfolder of your installation directory (currently //ewtl2/work/Johannes/Konzentrationen/windtunnel). The sample file is called S04\_Ambient\_Conditions.csv; if you rename this file you must also enter the new name into the python script (variable input\_file, line 52 in the point measurement script and line 77 in the puff measurement script). In the csv file, the first line must contain the filename of each dataset, and this must be identical to the filenames of your input datasets. If this is not the case, the program will resort to using the a priori values written in the respective python script. Note further that you must enter (proper) values for all variables located in the sample csv file, you cannot selectively specify some variables from the csv file and others directly from the python file. If the csv file does not contain values for all variables located in the sample csv file, the program will once again resort to using the values from the python script for all the a priori values (thus ignoring the values in the csv file). Everything else is taken care of by the program.

Note that the output is, per default saved in the same location as your original dataset (i.e. under the path you entered for the variable ‘path’). While the point measurement script outputs mainly text files, the puff measurement script outputs several (i.e. many) plots in several subfolder. All necessary subfolders are automatically generated by the program.

1. **Puff and Point Concentration**

A key element of this data analysis package is the ability to analyses both continuous release (point concentration) and puff release (puff concentration) timeseries separately. It is up to the user to determine which type of tracer gas release was used for a particular time series; it is also critical to determine this **before** running this program. The two analysis modes cannot be used interchangeably under any circumstances! As stated above, the continuous release mode implies that the tracer gas was released over the entire timeseries at a steady, continuous rate into the wind tunnel. This type of release is particularly useful when studying, for example, the scenario where harmful chemicals (usually gases) are released from an industrial plant as a result of a malfunction (i.e. leakage). Analyzing this type of data is comparatively straightforward and the program will usually run very quickly (i.e. not more than a few seconds) unless a large amount of input data is specified (>100 mb).

The puff release mode is quite a bit more complicated, but in many cases also more realistic for many applications. Here, instead of being released continuously over time, the tracer gas is released only during short (usually lasting not more than a few seconds), but repeated intervals. Each of these intervals is called a puff. For the purpose of diagnosing the characteristics of the individual puffs, the program plots, among other things, the release of the tracer gas over time into the puff plots. Note that the release signal is binary; i.e at a point in time the gas can either be released or not released. The mean release signal, however (plotted in the plot of the mean puff) can give crucial information about the release length in each puff; if the release lengths for the individual puffs are all the same, the mean release signal appears will appear as a rectangle function (as do all the individual release signals for a particular puff). If the puff release lengths are all different, the mean release function will have a non-rectangular shape. Note that due to the binary nature of the release signal equal release lengths are also necessarily implied if the mean release signal appears as a rectangle function (this could not be generally said if the release signal could take on multiple, in particular a continuous set, of values). The mean puff for an example time series is shown in figure 1.

1. **A priori and input values**

This section briefly describes, albeit in some more detail than in the above section, the input and a priori variables. The following a priori variables are specified in both the puff and point concentration analysis:

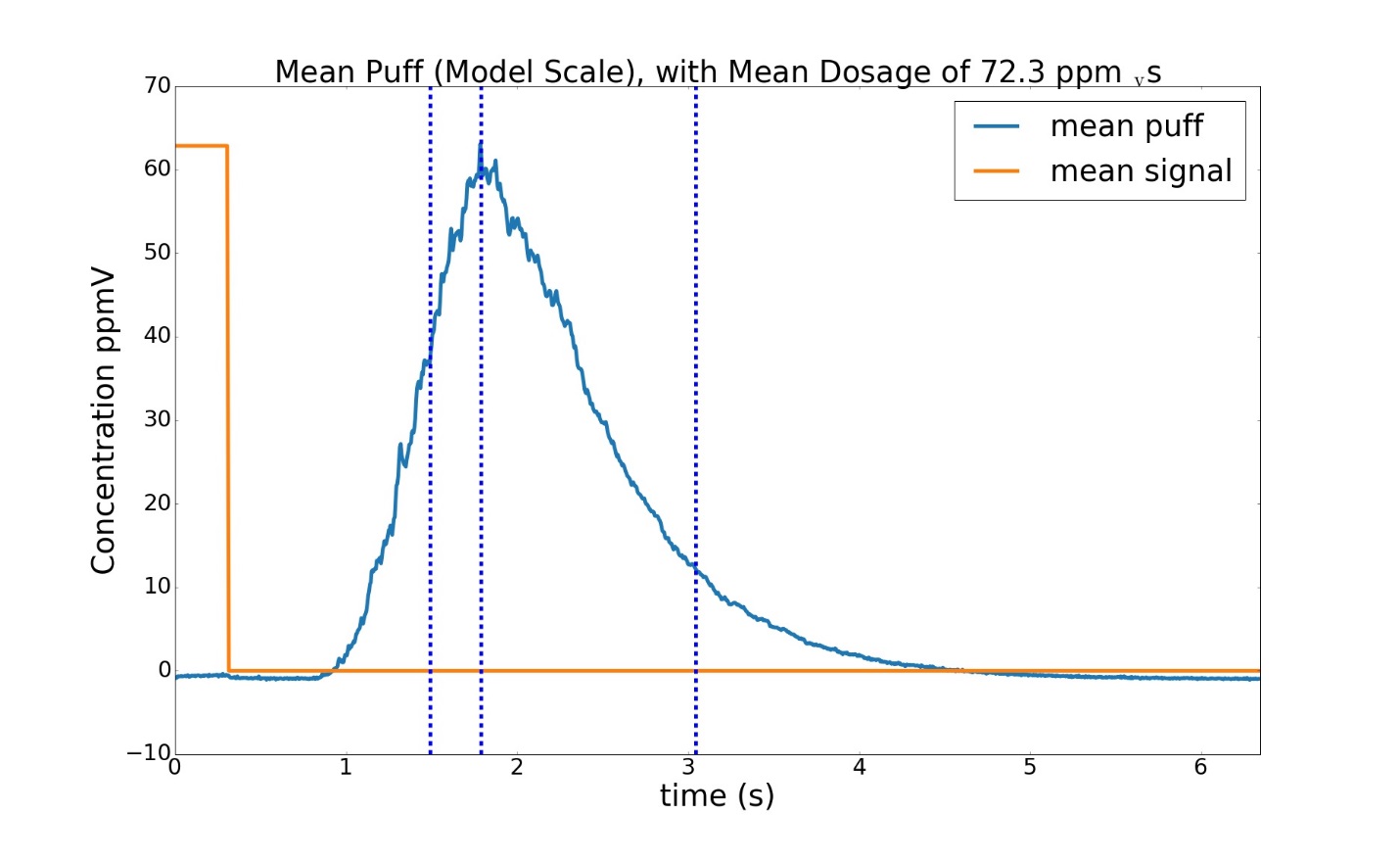


Figure 1: Mean puff for example time series. The puff signal is shown in blue (units ppmV) and the release signal in orange (binary, units adjusted to make signal readable on plot).

x: x-position of the emission source, in mm. Reference position remain to be determined, comparison with theses of Rasmus Fischer (2007, 2011) and/or Anne Philipp (2010) may be necessary.

y: y-position of the emission source, in mm. . Reference position remain to be determined, comparison with theses of Rasmus Fischer (2007, 2011) and/or Anne Philipp (2010) may be necessary.

z: z-position of the emission source, in mm, relative to wind tunnel surface.

pressure: reference air pressure, **outside wind tunnel**, during measurement, in Pa

temperature, reference air temperature**, outside wind tunnel**, during measurement, in °C.

calibration curve: tbd

mass flow controller: tbd

calibration factor: tbd

scale: scale of model. Note that the number here denotes the real-world units represented by one corresponding model unit; i.e. a scale of 1:250 would equal a value of 250 for the parameter ‘scale’

ref\_length: reference length, in m (model scale), used to make data non-dimensional

ref\_height: reference height, in m (model scale), use (in theory) for constructing wind profiles. Currently not used.

gas\_name: Name of tracer gas. Currently only used for documentation purposes.

mol\_weight: molecular weight, in kg/mol, of gas. Units here are highly doubtful, **to be verified**. More likely units are g/mol.

gas\_factor: used to calculate model scale flow rate

full\_scale\_wtref: full scale reference wind velocity, in m/s, used to calculate full\_scale concentration and time.

full\_scale\_flow\_rate: model scale flow rate out of emission source, in kg/s. Note that this is later converted to full scale flow rate, in m­3/s. Convention of naming both the full scale and model scale flow rate full\_scale\_flow\_rate taken from Benyamin Schliffke’s original script, the reason for this rather odd naming convention is at this point unknown.

The puff analysis uses all of the a priori variables above, plus the following additional a priori variables:

threshold\_concentration: the minimum peak concentration, in ppm­­v­ (model scale) of a puff to be included in the statistical analysis. All puffs with a peak concentration below this value will be filtered out of the analysis.

threshold\_dosage: the minimum dosage, in ppm­­v­­­s (model scale), of a puff to be included in the statistical analysis. All puffs with a peak concentration below this value will be filtered out of the analysis.

full\_scale: variable which controls which scale to perform the analysis in. Accepts ‘ms’ (for model scale data), ‘fs’ (for full scale data), and ‘nd’ (for non-dimensional data) as inputs.

time\_threshold:a number which determines at which percentages of the total dosage the arrival and leaving times should be computed. The number itself is the fraction of the dosage at which to define the arrival time as, with the leaving time determined in a symmetric manner. For example, setting this variable to 0.01 will define the arrival time at 1% of the dosage, and the leaving time at 99% (or 100%-1%) of the dosage. The agreed-upon setting for this variable is 5%, but some puffs cannot be appropriately characterized by this convention. **Warning: changing this variable will substantially modify the results of the program. Do not change this variable unless you know what you are doing!! In all cases, be sure to properly document and justify any change to this variable in any work that results from the usage of this program.**

1. **Output**

*5.1 Point Concentration*

In point concentration mode (i.e. continuous release) the program outputs three txt files containing the results of the analysis. The files begin with \_avg\_ (average data, in full scale), \_fs\_ (time series of full scale data), and \_ms\_ (model scale data). Further, the program outputs two plots of the frequency distributions (both relative and cumulative) of the net concentration values. Both the txt files and the plots are saved in the original path of the data (i.e. whatever is specified under the a priori variable ‘path’).

*5.2 Puff Concentration*

In puff concentration mode (i.e. puff release) the program plots outputs quite a few more plots than in point concentration mode. All data is saved in a subfolder titled Puff\_Plots/data\_series under the original path, where data\_series is the name of the data series as specified in the a priori variable name. Note that a separate folder is generated for each time series; this is done to avoid having too many plots in one subfolder. First and foremost a separate plot is plotted for each individual puff, appropriately labeled by puff number, which plots the time series of the net concentration signal, as well as the release signal (see above) and dosage (in the plot title). Further, the arrival time and leaving time are plotted by the two dashed lines (note that the leaving time is always necessarily greater than, or in a few rare cases, equal to, the arrival time). Figure 2 shows a plot from a puff from a sample time series. This is plotted in model scale; in the future a separate plot will be generated for each model scale, full scale, and non-dimensional. Note that time is always the same as specified in the time series, and not the start of the individual puffs! Note that the signal (which is in binary form) is multiplied by the average peak concentration to make sure it is actually readable (see above). By default, the program plots only the first five puffs (to avoid plotting large numbers, sometimes hundreds, of puffs at once). This can be changed by modifying the value for n\_puffs in line 228 of the python script, to either the desired number of puffs, or to ‘all’ to plot all puffs.

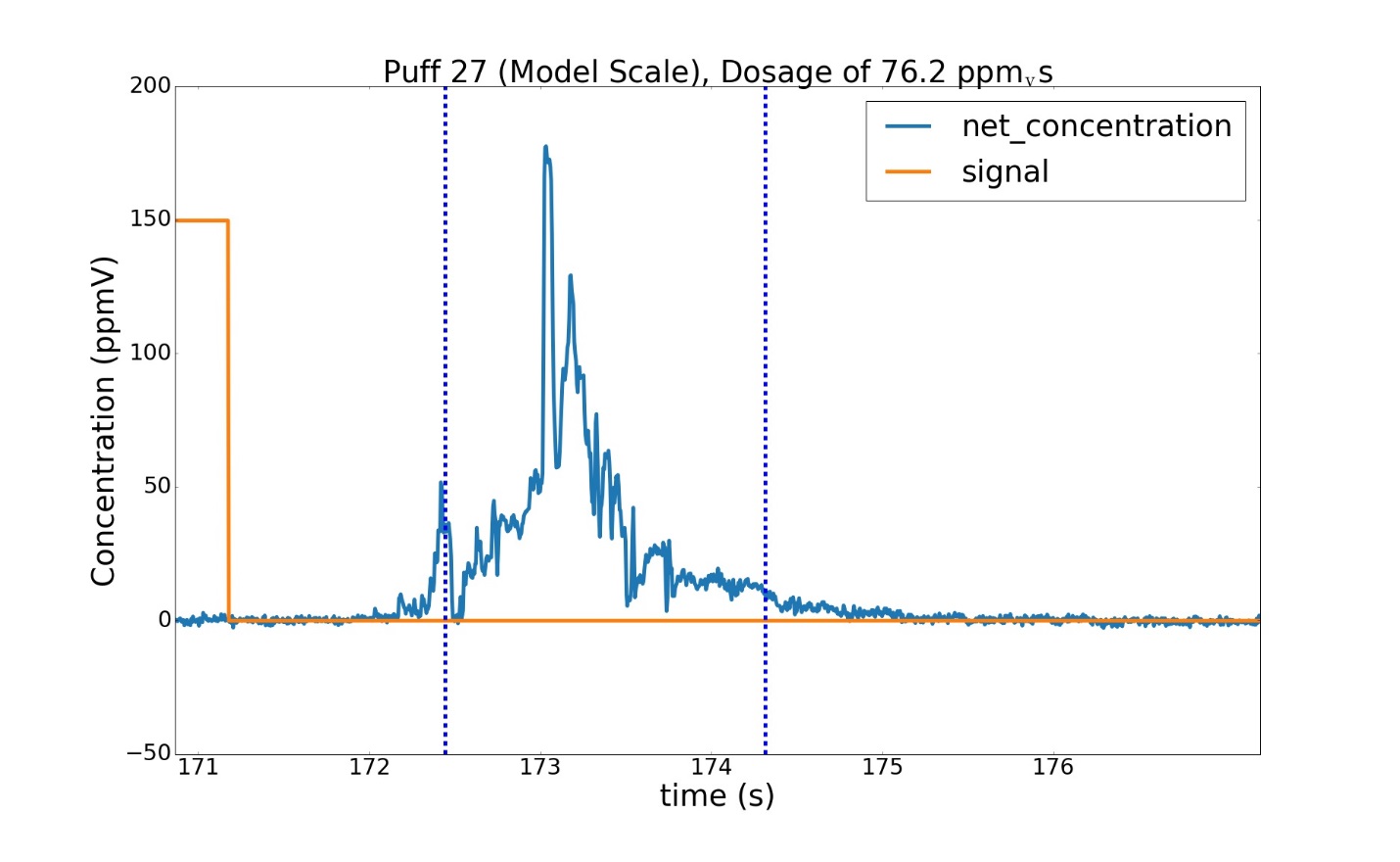


Figure 2: Puff plot of puff from sample time series. The first blue dashed line, at t=172.44 s, denotes the arrival time, while the second blue dashed line, at t=174.31 s, denotes the leaving time.

Additionally, the program plots a plot similar to the puff plots (e.g. Figure 2), but for a characteristic ‘mean puff.’ This ‘mean puff’ is generated by linearly averaging the net concentration over all puffs in the dataset. In order to do this, the puff time series are truncated to the length of the shortest puff. This should not usually have a substantial impact on the data, except in rare cases where the tracer gas is released at irregular intervals (i.e. the puffs all have different lengths). In this case, the concentration data obtained after the end of the shortest puff will simply be left out of the analysis. Similarly to the regular puff plots, the mean puff plot also includes a characteristic arrival time and leaving time. However, the arrival time and the leaving time plotted on the mean puff plot are not the arrival and leaving time of the mean puff itself, but instead are obtained by calculating the arithmetic mean of all arrival and leaving times in the dataset. An average ‘peak time’, between the arrival and leaving time, is also included in the mean puff plot. By using the average of the characteristic arrival, leaving, and peak times, instead of the arrival, leaving, and peak times of the average puff, an insight of how close the puffs are to being normally distributed can be obtained (if the puffs were all perfectly normally distributed at every point in time, the average arrival, peak, and leaving time, would correspond exactly to the arrival, peak, and leaving time of the average concentration signal. Further, by setting the variable ‘dist’ to one (also in line 228) of the python script, the program will additionally plot the 10th and 90th percentile curve of the puffs. This gives a more quantitative insight into the distribution of the puffs at any given point in time.

Additionally, in the puff release mode, the program also outputs a set of convergence analysis plots. For this, the data is grouped into a set of ensembles, with varying ensemble size, and for each ensemble an ensemble mean is calculated. Note that for each ensemble size there exists as many ensembles as there are puffs. This analysis is performed separately for each variable (arrival time, ascent time, descent time, leaving time, peak concentration, and peak time). A more detailed description of the ensemble analysis and how it works can be found in Anne Philipp’s Bachelor Thesis (2010). The convergence analysis plots then effectively show the ensemble means plotted as a function of ensemble size. Ideally, the convergence analysis plots should have a funnel shape to them, with the top end of the funnel being on the left side of the plot, and the bottom of the funnel being at the right side of the plot.

Lastly, the program also outputs a set of frequency distribution plots. For this, the data are once again grouped into ensembles as for the convergence analysis plots, and the for each ensemble size the ensemble means are grouped into a set number of discrete classes, the frequencies (relative) of which are then output as a histogram. The number of classes varies as a function of ensemble size, and is calculated using a commonly used algorithm, described in more detail in Anne Philipp’s Bachelor Thesis (2010). Note that the analysis is once again performed separately for each of the variables listed above. Note that since the number of ensemble sizes equals the size of the data (in this case the number of puffs), the frequency analysis generates an extremely large number of plots (more precisely 6 times the number of puffs). For a reasonable number of puffs (easily 100 or more) this means that the frequency analysis can easily generate several thousand plots. For this reason, the frequency distribution plots are saved in a separate subfolder inside the data directory, titled ‘Frequency\_Distribution.’

1. **Remarks**

This program is still under development, but was initially developed by Johannes Fischer in the boundary layer wind tunnel working group under the supervision of Bernd Leitl at the Meteorological Institute at the University of Hamburg in 2019. The program is effectively an extension of the windtunnel package by Benyamin Schliffke (2018), some parts (where noted) are also based on an old analysis Program by Anne Philipp, developed for her Bachelor Thesis (2010). Many thanks also go to Frank Harms and Kerstin Surm for providing valuable feedback for improving the program. Some functions were also specifically developed by request from various members of the aforementioned working group (Helen Stricker, Rike Wachsmann, Frank Harms, and Kerstin Surm, among others). If you encounter any errors or have any other feedback please do not hesitate to contact the developers at [johannes.karl.fischer@gmail.com](mailto:johannes.karl.fischer@gmail.com).