**Flowrates with Flow Meter Calibration (FMC)**

Norense Okungbowa – Jan. 04, 2025

**Introduction**

The package provides basic data analysis to compare 2 flow rate variables and is implemented as a python module ***flowrate.py***

**Requirements** (requires python3)

1. For Command line execution:

* ***flowrate.py***
* ***calibrate.py***

1. For UI execution:

* ***flowrategui.py***
* ***calibrategui.py***
* ***flowrate.ui***
* ***calibrate.ui***

**Command line**

**Usage:**

***flowrate.py***

--input INPUT

[-h]

[--output OUTPUT]

[--table TABLE]

[--from [FRM]]

[--to [TO]] [--dc]

[--graph1]

[--addgroup]

[--graph2]

[--graph3 DISPLAY\_CHARTS\_3]

[--graph4] [--graph5] [--grid]

[--abs]

[--full\_output]

[--resample]

[--bias1 BIAS1]

[--bias2` BIAS2]

[--multiplier1 MULT1]

[--multiplier2 MULT2]

[--dtw]

[--wd]

[--ks]

[--adf]

[--cointegration] [--corr]

[--singleflow SINGLE\_FLOW]

[--flows [FLOWS ...]]

[--baseflow BASEFLOW]

[--match]

[--cluster CLUSTER\_DATA]

[--clustergraph]

[--***calibrate***]

[--synchronize]

[--syncindex]

[--useopt]

[--uselsq]

[--CalibrationSamplePeriodicityInMinutes CALIBRATIONSAMPLEPERIODICITYINMINUTES]

[--CalibrationPeriodicityInHours CALIBRATIONPERIODICITYINHOURS]

[--MinimumFlowToSample MINIMUMFLOWTOSAMPLE]

[--NumberOfSamplesForCalibration NUMBEROFSAMPLESFORCALIBRATION]

[--CalibrationDelayAfterRestartInHours CALIBRATIONDELAYAFTERRESTARTINHOURS]

[--CalibrationOffsetSamplesFromCurrentTime CALIBRATIONOFFSETSAMPLESFROMCURRENTTIME]

Process time series data of two flowrates in a csv file.

**Code Arguments:**

|  |  |  |
| --- | --- | --- |
| **Mandatory Argument** | **Abbreviated Argument** | **Description** |
| --input INPUT | -i INPUT | Input file name |
|  |  |  |
| **Optional Arguments** |  |  |
| --help | -h | Show this help message and exit |
| --output OUTPUT | -o OUTPUT | Output file name |
| --table TABLE | -b TABLE | Table output file name |
| --from [FRM] | -f [FRM] | Filter record from this time |
| --to [TO] | -t [TO] | Filter record to this time |
| --dc | -dc | Display columns list |
| --addgroup | -ag | Display grouping plot of the results |
| --graph1 | -g1 | Display plot Accumulated volume and flowrates |
| --graph2 | -g2 | Display plot Flow difference and flowrates |
| --graph3 DISPLAY\_CHARTS\_3 | -g3 DISPLAY\_CHARTS\_3 | Display plot of variables (1,2 or 3 (both)) |
| --graph4 | -g4 | Display histograms |
| --graph5 | -g5 | Display FFT plots |
| --grid | -d | Display grid lines |
| --full\_output | -fo | Provide extended output |
| --abs | -a | Use absolute difference |
| --bias1 BIAS1 | -b BIAS1 | Add bias to the first variable |
| --bias2 BIAS2 | -b BIAS2 | Add bias to the second variable |
| --multiplier1 MULT1 | -m1 MULT1 | Multiply by the first variable |
| --multiplier2 MULT2 | -m1 MULT2 | Multiply by the second variable |
| --dtw | -dtw | Use dynamic time warping (very slow processing) |
| --wd | -wd | Use Wasserstein distance |
| --ks | -ks | Use Kolmogorov-Smirnov test |
| --adf | -adf | Use Augmented Dickey-Fuller test |
| --cointegration | -coint | Use Cointegration test |
| --corr | -corr | Display correlation |
| --singleflow SINGLE\_FLOW | -sf SINGLE\_FLOW | Single flow (signal) to process |
| --flows FLOWS FLOWS | -fl FLOWS FLOWS | Flows to compare |
| --baseflow BASEFLOW | -bfl BASEFLOW | Base flow to compare other flows to |
| --match | -m | Check match |
| --cluster CLUSTER\_DATA | -cl CLUSTER\_DATA | Cluster data with given threshold |
| --clustergraph | -cg | Graph of clusters. Only valid if cluster is requested. |
| --calibrate | -fmc | Calibrate one flow meter with another flow meter. Flow Meter Calibration (FMC) |
| --synchronize | -sa | Synchronize Y-axis for the calibration displays (raw and calibrated accumulated flow difference) |
| --syncindex | -si | Synchronize indexes with the beginning of the flow meter factor updates |
| --useopt | -opt | Calculate an optimal flow meter factor from all the flow meter factors (Mean of the flow meter factor samples). Use optimal value calibration for the calibration display. |
| --uselsq | -lsq | Calculate the best fit (slope and intercept) for the flow meter factors using least square.  Use the least square value calibration for the calibration display. |
| --CalibrationSamplePeriodicityInMinutes CALIBRATIONSAMPLEPERIODICITYINMINUTES | -caspm CALIBRATIONSAMPLEPERIODICITYINMINUTES | Calibration sample periodicity in minutes |
| --CalibrationPeriodicityInHours CALIBRATIONPERIODICITYINHOURS | -casph CALIBRATIONPERIODICITYINHOURS | Calibration periodicity in hours |
| --MinimumFlowToSample MINIMUMFLOWTOSAMPLE | -camf MINIMUMFLOWTOSAMPLE | Minimum flow to sample for calibration |
| --NumberOfSamplesForCalibration NUMBEROFSAMPLESFORCALIBRATION | -cans NUMBEROFSAMPLESFORCALIBRATION | Number of samples for calibration |
| --CalibrationDelayAfterRestartInHours CALIBRATIONDELAYAFTERRESTARTINHOURS | -cad CALIBRATIONDELAYAFTERRESTARTINHOURS | Calibration delay after restart in hours |
| --CalibrationOffsetSamplesFromCurrentTime CALIBRATIONOFFSETSAMPLESFROMCURRENTTIME | -cao CALIBRATIONOFFSETSAMPLESFROMCURRENTTIME | Calibration offset samples from current time |

**Comparing Flow Rates from two Flow Meters**

**Command Line Processing:**

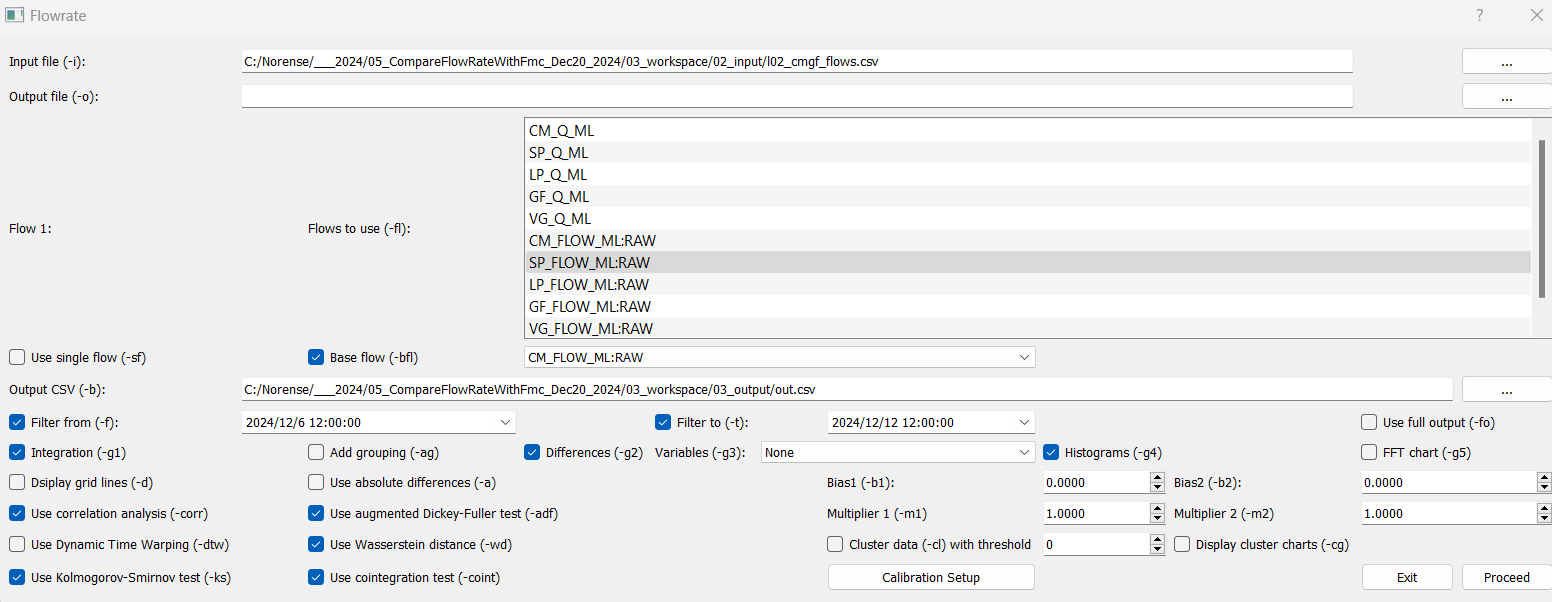
python3 **flowrate.py** **-i** l02\_cmgf\_flows.csv **-f** "2024/12/06 12:00:00" **-t** "2024/12/12 12:00:00"

**-b** out.csv **-g1** **-g2** **-g4** **-corr** **-wd** **-ks** **-adf** **-coint**

Where the code is requested to take l02\_cmgf\_flows.csv file as an input and write the finalized data into out.csv file. Data should be filtered with the range of dates to be from December 06, 2024, 12:00:00 to December 12, 2024, 12:00:00. Output will be displayed in standard output and 3 chart windows will be opened. Correlation, Wasserstein distance, Kolmogorov-Smirnov, Augmented Dickey-Fuller (ADF) and Cointegration test results will be displayed as well.

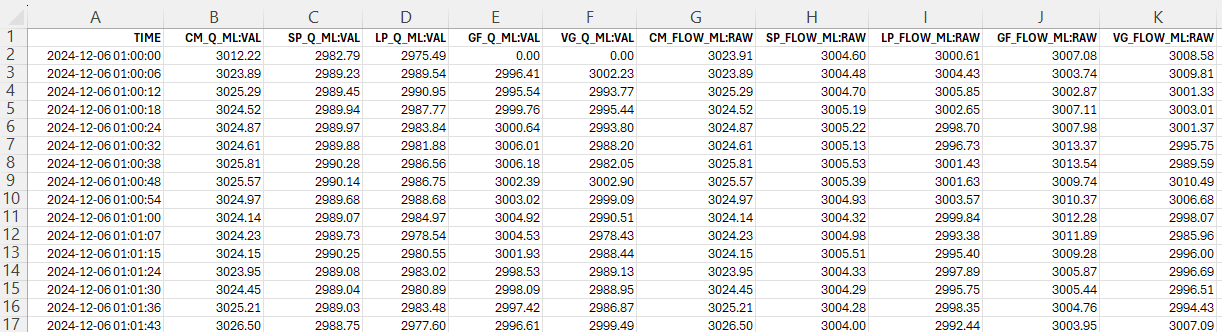
**GUI processing**

The GUI processing of the same command line as above is:



**Sample input data**

Input data consists of a single file in csv format. An example of the content of the input file (l02\_cmgf\_flows.csv) is shown below:



**Sample output data**

Output CSV file (see screenshot below) will contain the original data, filtered within the requested range if given.

A screenshot of a computer

Description automatically generated

There will be added columns, covering flow differences, time differences and the results of partial integration for both variables.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a graph

Description automatically generated

**Flow Meter Calibration (FMC)**

Calibrate one flow meter (dependent) with another flow meter (independent) that is more accurate.

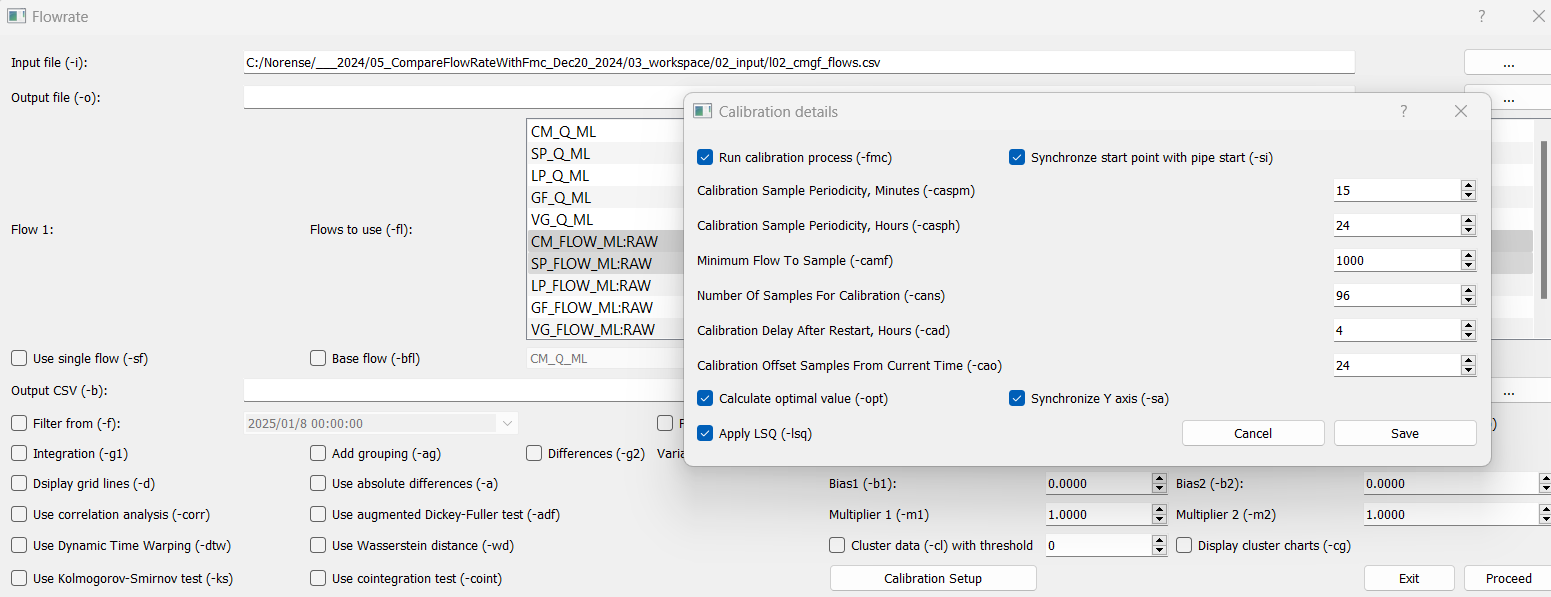
**FMC Command Line Processing:**

python3 **flowrate.py** **-i** l02\_cmgf\_flows.csv **-fl** CM\_FLOW\_ML:RAW SP\_FLOW\_ML:RAW

**-fmc** **-caspm** 15 **-casph** 24 **-cans** 96 **-camf** 1000 **-cad** 4 **-cao** 24 **-opt -lsq -si -sa**

**FMC GUI Processing:**

The UI processing of the above command is shown below.



A screenshot of a computer

Description automatically generated

The command line above will use l02\_cmgf\_flows.csv file as an input and run calibration query with the options specified above. The output will be as below:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Flow Rates Clustering (FRC)**

The next example demonstrates clustering of the flat regions

**FRC Command Line Processing:**

python **flowrate.py** **-I** l02\_cmgf\_flows.csv **-fl** CM\_FLOW\_ML:RAW SP\_FLOW\_ML:RAW **-cl** 30 **-cg** **-ag**

As before, l02\_cmgf\_flows.csv is used as input and clustering is requested with slope threshold of 30. Clustering graphs should be generated as part of the output.

**FRC GUI Processing:**

The UI processing of the above command is shown below.

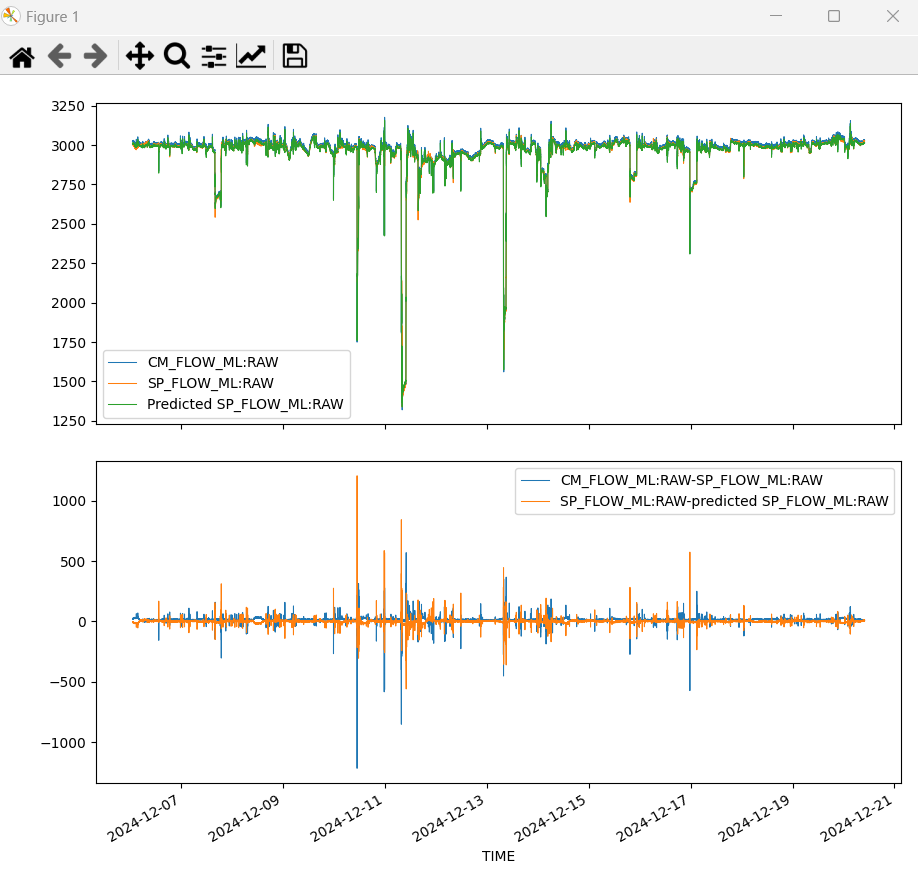
A screenshot of a computer

Description automatically generated

The following output is produced:

A screenshot of a computer

Description automatically generated



**Appendix 1: Detailed Explanation of the Options**

Flowrate.py package supports several options, which provide quantification of the features of the flows:

* **-a** option forces the difference between flows to be taken as an absolute value
* **-fo** adds scipy-based integration details to the output
* **-s** increment first flow by a specified bias value
* **-dtw** calculates Dynamic Time Warping (DTW) distance. In time series analysis, DTW is an algorithm similarity between two temporal sequences, which may vary in speed. For instance, similarities in walking could be detected using DTW, even if one person is walking faster than the other, or if there were accelerations and decelerations during an observation.
* **-wd** calculates Wasserstein distance. The Wasserstein distance or Kantorovich-Rubinstein metric is a distance function defined between probability distributions on a given metric space.
* **-ks** implements Kolmogorov-Smirnov (K-S) test. The K-S test is a non-parametric test of the equality of continuous or discontinuous, one-dimensional probability distributions that can be used to compare two samples with regards to their proximity to each other in statistical terms.
* **-adf** provides the output of Augmented Dickey-Fuller (ADF) test. The ADF test is a common statistical test used to test whether a given time series is stationary or not. It is one of the most used statistical tests when it comes to analysing the stationary of a series. The P-value, reported by the test is compared to a user-selected threshold, and, if the reported p-value is below the threshold, the series can be considered stationary. A stationary time series is one whose properties (mean, variance) do not depend on the time at which the series is observed.
* **-coint** calculates the cointegration distance. A cointegration test is used to establish if there is a correlation between several time series in the long term. Cointegration tests identify scenarios where two or more non-stationary time series are integrated together in a way that they cannot deviate from equilibrium in the long term. The tests are used to identify the degree of sensitivity of two variables to the same average over a specified period of time.
* **-corr** calculates correlation between two measurements (flows in this case). Correlation is a relationship or connection between two things based on co-occurrence or pattern of change. Correlation coefficients are indicators of the strength of linear relationship between two different variables, x and y. A linear correlation coefficient that is greater than zero indicates a positive relationship. A value that is less than zero signifies a negative relationship. Finally, a value of zero indicates no relationship between the two variables x and y.

**Comparing Closeness of the Variable**

1. Mean Square Error (MSE) – This ….
2. Correlation – indicates how well the variables are linearly related (i.e., they change together at a constant rate). It is not the actual closeness but the closeness in behavior.

It is a number between -1 and +1, where -1 means perfectly negative relationship (1 variable changes in the positive direction, the other changes in the negative direction). When correlation is +1, indicates perfectly positive relationship (1 variable changes in the positive direction, the other changes in the positive direction). Zero correlation means no relationship between the variables.

Correlation close to 0 meaning low correlation. That is, they have significance imbalance in how they behave (pretty much useless indicator anyway)

1. T-test – shows how close are the distributions that these variables belong to. It is a statistical test that compares the means of two samples.

A t-value of 0 means that most probably both variables were drawn from the same distributions (A t-value of 0 indicates that the sample results exactly equal the null hypothesis. As the sample data become progressively dissimilar from the null hypothesis, the absolute value of the t-value increases).

1. Dynamic Time Warping (DTW) – which is a way to compare two usually temporal sequences that do not sync up perfectly. It measures similarity between two temporal sequences, which may vary in speed. DTW has been applied to temporal sequences of video, audio, and graphics data

DTW close to 1 suggest close curves. However, DTW should be taken into consideration carefully because it is not a metric and does not follow triangular inequality.

**Appendix 2: Example Command Lines**

**Example Command Lines**

/\*==============================================================================================================================================================================

/\* 2024-11-24 - ver17 -

/\*

/\* directories

/workspace ---------> calibrate.ui, flowrate.ui. Open Cmd window

/workspace/00\_run\_ver18 ---------> flowrate.py, calibrate.py flowrategui.py, calibrategui.py

/workspace/01\_run\_ver18\_from\_exec

/workspace/02\_input ---------> l02\_cmgf\_flows.csv, l02\_qulb\_flows.csv, l09\_cdml\_flows.csv

/workspace/03\_output

--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Sample command line to execute the Console application

python.exe 00\_run\_ver18/flowrate.py -h

python.exe 00\_run\_ver18/flowrate.py -i 02\_input/l02\_qulb\_flows.csv -g1 -g2 -wd

python.exe 00\_run\_ver18/flowrate.py -i 02\_input/l02\_qulb\_flows.csv -g1 -g2 -wd -b1 1000

python.exe 00\_run\_ver18/flowrate.py -i 02\_input/l02\_qulb\_flows.csv -g1 -g2 -wd -b2 -1000

python.exe 00\_run\_ver18/flowrate.py -i 02\_input/l02\_qulb\_flows.csv -g1 -g2 -wd -m1 0.50

python.exe 00\_run\_ver18/flowrate.py -i 02\_input/l02\_qulb\_flows.csv -g1 -g2 -wd -m2 0.50

python.exe 00\_run\_ver18/flowrate.py -i 02\_input/l02\_qulb\_flows.csv -g1 -g2 -wd -fl LB\_FLOW\_ML:RAW -bfl QU\_FLOW\_ML:RAW

/\*==============================================================================================================================================================================

Sample command line to execute the Console application

python.exe 00\_run\_ver18/flowrate.py -i 02\_input/l09\_cdml\_flows.csv -fl CD\_FLOW\_ML:RAW ML\_FLOW\_DEL:RAW -bfl CD\_FLOW\_ML:RAW -m2 1.0000 -g1 -g2

python.exe 00\_run\_ver18/flowrate.py -i 02\_input/l09\_cdml\_flows.csv -fl CD\_FLOW\_ML:RAW ML\_FLOW\_DEL:RAW -bfl CD\_FLOW\_ML:RAW -m2 1.0015 -g1 -g2

--------------------------------------------------------

Single variable processing

python.exe 00\_run\_ver18/flowrate.py -i 02\_input/l02\_qulb\_flows.csv -g3 1 -g4 -sf LB\_FLOW\_ML:RAW

python.exe 00\_run\_ver18/flowrate.py -i 02\_input/l02\_qulb\_flows.csv -g3 1 -g4 -g5 -sf LB\_FLOW\_ML:RAW

/\*==============================================================================================================================================================================

FMC - Sample command line to execute the Console application

python.exe 00\_run\_ver18/flowrate.py -i 02\_input/l02\_cmgf\_flows.csv -fl CM\_FLOW\_ML:RAW SP\_FLOW\_ML:RAW -fmc -camf 500

python.exe 00\_run\_ver18/flowrate.py -i 02\_input/l02\_cmgf\_flows.csv -fl CM\_FLOW\_ML:RAW SP\_FLOW\_ML:RAW -fmc -caspm 5 -casph 1 -cans 60 -camf 500 -cad 4 -cao 24

python.exe 00\_run\_ver18/flowrate.py -i 02\_input/l02\_cmgf\_flows.csv -fl CM\_FLOW\_ML:RAW SP\_FLOW\_ML:RAW -fmc -caspm 5 -casph 2 -cans 60 -camf 500 -cad 4 -cao 24

python.exe 00\_run\_ver18/flowrate.py -i 02\_input/l02\_cmgf\_flows.csv -fl CM\_FLOW\_ML:RAW SP\_FLOW\_ML:RAW -fmc -caspm 5 -casph 3 -cans 60 -camf 500 -cad 4 -cao 24

python.exe 00\_run\_ver18/flowrate.py -i 02\_input/l02\_cmgf\_flows.csv -fl CM\_FLOW\_ML:RAW SP\_FLOW\_ML:RAW -fmc -caspm 5 -casph 5 -cans 60 -camf 500 -cad 4 -cao 24

/\*==============================================================================================================================================================================

/\* USE

python.exe 00\_run\_ver18/flowrate.py -i 02\_input/l02\_cmgf\_flows.csv -fl CM\_FLOW\_ML:RAW SP\_FLOW\_ML:RAW -fmc -caspm 15 -casph 0.25 -cans 96 -camf 1000 -cad 4 -cao 24

python.exe 00\_run\_ver18/flowrate.py -i 02\_input/l02\_cmgf\_flows.csv -fl CM\_FLOW\_ML:RAW SP\_FLOW\_ML:RAW -fmc -caspm 15 -casph 0.25 -cans 96 -camf 1000 -cad 4 -cao 24 -opt

python.exe 00\_run\_ver18/flowrate.py -i 02\_input/l02\_cmgf\_flows.csv -fl CM\_FLOW\_ML:RAW SP\_FLOW\_ML:RAW -fmc -caspm 15 -casph 0.25 -cans 96 -camf 1000 -cad 4 -cao 24 -lsq

python.exe 00\_run\_ver18/flowrate.py -i 02\_input/l02\_cmgf\_flows.csv -fl CM\_FLOW\_ML:RAW SP\_FLOW\_ML:RAW -fmc -caspm 15 -casph 0.25 -cans 96 -camf 1000 -cad 4 -cao 24 -opt -lsq

python.exe 00\_run\_ver18/flowrate.py -i 02\_input/l02\_cmgf\_flows.csv -fl CM\_FLOW\_ML:RAW SP\_FLOW\_ML:RAW -fmc -caspm 15 -casph 1.00 -cans 96 -camf 1000 -cad 4 -cao 24 -opt -lsq

python.exe 00\_run\_ver18/flowrate.py -i 02\_input/l02\_cmgf\_flows.csv -fl CM\_FLOW\_ML:RAW SP\_FLOW\_ML:RAW -fmc -caspm 15 -casph 3.00 -cans 96 -camf 1000 -cad 4 -cao 24 -opt -lsq

python.exe 00\_run\_ver18/flowrate.py -i 02\_input/l02\_cmgf\_flows.csv -fl CM\_FLOW\_ML:RAW SP\_FLOW\_ML:RAW -fmc -caspm 15 -casph 6.00 -cans 96 -camf 1000 -cad 4 -cao 24 -opt -lsq /\* good

python.exe 00\_run\_ver18/flowrate.py -i 02\_input/l02\_cmgf\_flows.csv -fl CM\_FLOW\_ML:RAW SP\_FLOW\_ML:RAW -fmc -caspm 15 -casph 12.00 -cans 96 -camf 1000 -cad 4 -cao 24 -opt -lsq

python.exe 00\_run\_ver18/flowrate.py -i 02\_input/l02\_cmgf\_flows.csv -fl CM\_FLOW\_ML:RAW SP\_FLOW\_ML:RAW -fmc -caspm 15 -casph 24.00 -cans 96 -camf 1000 -cad 4 -cao 24 -opt -lsq

/\*--------------------------------- SIMULATE LEAK--------------------------------------------------------------------

python.exe 00\_run\_ver18/flowrate.py -i 02\_input/l02\_cmgf\_flows.csv -fl CM\_FLOW\_ML:RAW SP\_FLOW\_ML:RAW -fmc -caspm 15 -casph 24.00 -cans 96 -camf 1000 -cad 4 -cao 24 -opt -lsq

python.exe 00\_run\_ver18/flowrate.py -i 02\_input/l02\_cmgf\_flows.csv -fl CM\_FLOW\_ML:RAW SP\_FLOW\_ML:RAW -fmc -caspm 15 -casph 24.00 -cans 96 -camf 1000 -cad 4 -cao 24 -opt -lsq -m2 0.9700

/\*==============================================================================================================================================================================

Sample Command Line to open the UI application

Notes:

1) flowrate.ui and calibrate.ui must be in the same directory the cmd was opened (e.g., workspace)

2) flowrategui.py must be in workspace/00\_run\_ver18

python.exe 00\_run\_ver18/flowrategui.py

/\*==============================================================================================================================================================================

**Data Clustering - Sample command line to execute the Console application**

python.exe 00\_run\_ver18/flowrate.py -i 02\_input/l02\_cmgf\_flows.csv -fl CM\_FLOW\_ML:RAW SP\_FLOW\_ML:RAW -cl 30 -cg -ag

/\*==============================================================================================================================================================================