

# **FlareNet Particle Penetration Calculator (FPPC) User Guide**

Steven Rogak, Keyhan Babaei, UBC, Vancouver

August 2018

## Contents:

---

A-Introduction.....	4	4- Inch_to_Meter (A).....	17
B-How to Use.....	5	5- Deg_to_Radian (A) .....	17
With Python: .....	5	6-Flow_Velocity (Flowrate, Area) .....	18
With EXE File .....	7	7- Kn (l, d) .....	18
C-Excel Data Entry .....	8	8- Cc (Kn).....	18
NO. ....	8	9- visc (T, P).....	19
Name .....	8	10- mfp (visc, T, P).....	19
From & To.....	8	11- KinViscosity (Vis, rho) .....	20
Ignore .....	8	12- Reynolds (rho, Velocity, Diameter, Vis).....	20
ID (inch).....	8	13- SpecificHeat (T) .....	21
Length (inch).....	8	14- Prandtl (C, u, K) .....	21
Bend (Deg) .....	8	15- ThermalCond (T).....	21
$\theta$ (from horizon) (Deg) .....	8	16- rho (T, P) .....	22
Tg (K).....	8	17- Mobility (d, Cc, visc) .....	22
Tw (K) .....	8	18- Particle_Diff_Coeff (B, T) .....	23
T1 (K).....	8	19- Zeta (D, L, Q).....	23
T2 (K).....	9	20- Schmidt (vis, dens, diff) .....	24
Pg (Pa).....	9	21- Effective_Density (K, Dm, dp) .....	24
Q (SLPM).....	9	22- Gravit_Depos (L, vts, d_tube, U).....	25
Material .....	9	23- K_Thermopho (Knud, Cc, Thermal_Cond_Fluid, Thermal_Cond_Particle): .....	25
ID (Second) (Inch).....	9	24- Thermophoresis_Eff (Kth, Re, Pr, Re_t, Tw, Te, k, Cp, Q, rhog, L, D):.....	25
Half_Theta (Second) (Deg) .....	9	25- Settling_Eff (Zg, Re, Re_t, theta): .....	26
Sampling Probe .....	9	26- Terminal_Velocity (Dp, rhop, Cc, vis): .....	26
U0 (m/s).....	9	27- Vt_Turb_Depos (Stk, Re, U):.....	27
Theta probe (deg) .....	9	28- Diffusion_Eff (zeta, Sc, Re, Re_t): .....	27
D-Python Variables .....	11	29- Inertial_Turb_Depos_Eff (d_tube, L, Vt_Turb, Q, Re, Re_t):.....	27
E-Output Data .....	12	30- Inertial_Bend_Depos_Eff (Stk, Bend_Rad, Re, Re_t): .....	27
Excel Outputs .....	12	31- Inertial_Deposit_Contraction_Eff (Stk, Theta_Rad, A_o, A_i):.....	28
Graphs .....	13	32- Stokes (vts, U, L): .....	28
F-Functions .....	16	33- Aspiration_Eff (Stk, U0, U, Theta): .....	28
1- Area_with_Diameter (A) .....	16		
2- LPM_to_M3perSec (A) .....	16		
3- ActualFlowRateLPM (Q_Slpm, Pg_pa, T_0, P_0_pa, Tg_K) .....	16		

## Contents:

---

34- InletTrans_Eff (Stk, U0, U, Theta): .....	29
---	----

## A- Introduction

We developed a Python program "FlareNet Particle Penetration Calculator (FPPC)" to calculate line losses of multi-part sampling lines. Different mechanism cause particles to be lost en route to the sampling device and now with this program user has an estimation about how much of the specific particle diameter has been lost and may correct their measurement. For calculating particle penetration coefficients these loss mechanisms are considered:

- 1- Diffusion
- 2- Sedimentation (inclined lines included)
- 3- Thermophoresis
- 4- Turbulent inertial deposition
- 5- Inertial deposition: bend
- 6- Inertial deposition: contraction
- 7- Sampling probe loss

The line consists of multiple sections, and each section has its properties such as inner diameter, bend angle, gas temperature and flow rate, etc. The user should provide these properties to have the proper calculation. The list of the sections can be reached from the excel input folder. This assumption shows that each part is independent of another and the accuracy of the result depends on the accuracy of the input data.

Please do not hesitate to contact writers if you have any comments or recommendation about the program.

## B- How to Use

### With Python:

#### Step 1

Open the main project directory and in the Input excel folder, open the excel file. Please don't rename any file or directory.



Figure 1 Excel file in the Excel Input directory

#### Step 2

Here the user can enter information of each section in the sampling line. Every column has the label and for more information user can see the "Data Entry" section in the next chapter. Make sure that each row of excel file that used for calculation has all the proper information requested. The user can add many sections to excel file, although, it increases the accuracy of the final result; however, it needs more time to create graphs and output data.

NO	Name	From	To	Ignore	ID(Inch)	Length(Inch)	Bend(Deg)	θ (from horizon) (Deg)	Tg(K)	Tw(K)	T1(K)	T2(K)	Pg(Pa)	Q(SLPM)	Material	ID (Second) (Inch)	Half_Theta (Second) (Deg)	Sampling Probe	U0(m/s)	Theta Probe(Deg)
1	From Main Duct	C11A	C12A	0	0.75	8	90	-45	360	360	360	360	101325	90	Steel	0.75	0	Y	5.5	-90
2	Line	C12A	C13A	0	0.75	10	0	-90	360	360	360	360	101325	90	Steel	0.75	0	N	0	0
3	Line	C13A	C14A	0	0.75	50	0	-90	360	360	360	360	101325	90	Steel	0.75	0	N	0	0
4	Bend	C14A	C15A	0	0.75	30	90	-45	298	298	298	298	101325	90	Steel	0.75	0	N	0	0
5	Line	C15A	C16A	0	0.75	20	0	0	298	298	298	298	101325	90	Steel	0.75	0	N	0	0
6	Enlargement	C16A	C17A	0	1	8	0	0	298	298	298	298	101325	90	Steel	1	0	N	0	0
7	Line	C17A	C18A	0	2	20	0	0	298	298	298	298	101325	90	Steel	2	0	N	0	0
8	Bend	C18A	C18B	0	0.25	4	90	0	298	298	298	298	101325	1	Steel	0.25	0	N	0	0
9	Line	C18B	C19B	0	0.25	20	0	0	298	298	298	298	101325	1	Silicon	0.25	0	N	0	0
10	Bend	C19B	C20B	0	0.25	4	90	-90	298	298	298	298	101325	1	Silicon	0.25	0	N	0	0
11	Line	C20B	C21B	0	0.25	20	0	-90	298	298	298	298	101325	1	Silicon	0.25	0	N	0	0
12	Bend	C21B	C22B	0	0.25	4	90	0	298	298	298	298	101325	1	Silicon	0.25	0	N	0	0
13	PAX Left Line	C22B	C23B	0	0.25	10	0	0	298	298	298	298	101325	1	Silicon	0.25	0	N	0	0

Figure 2 Data Structure in input file

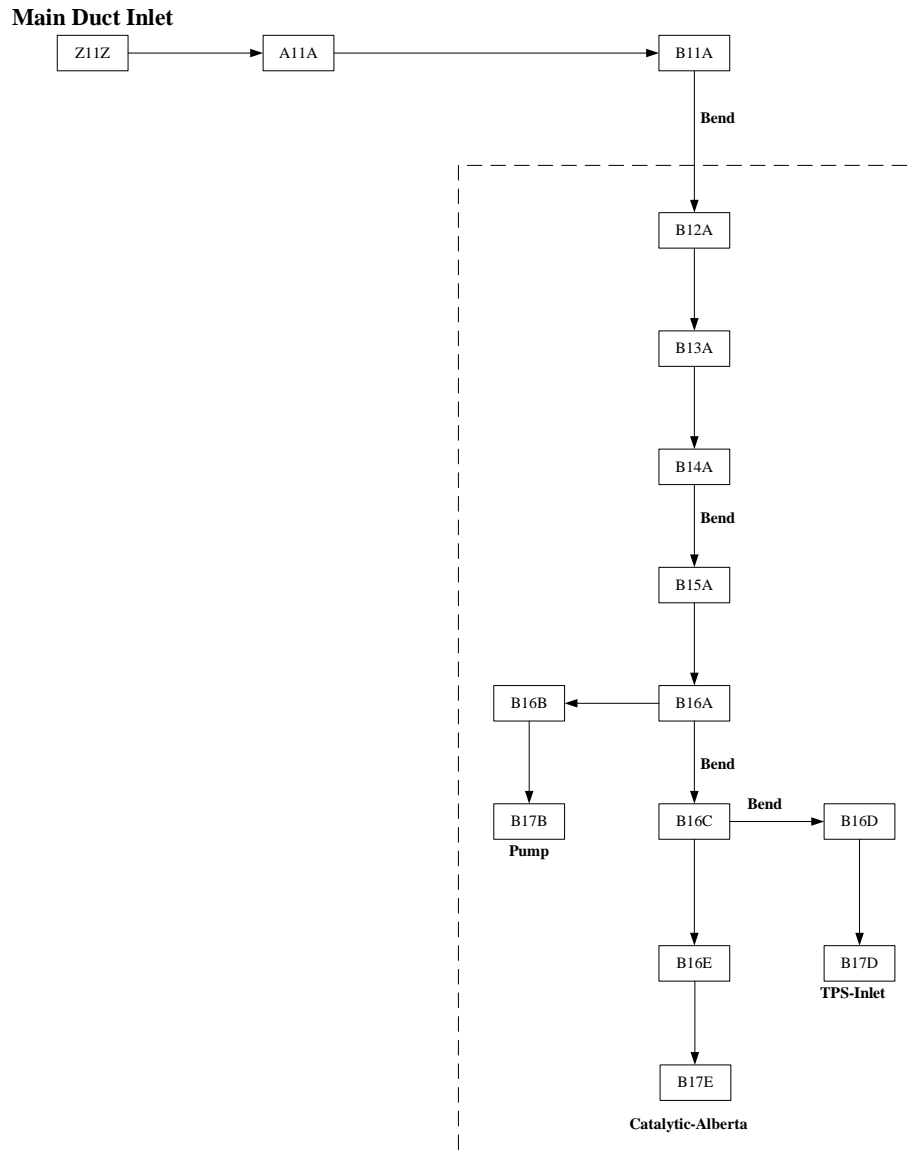


Figure 3 Schematic of the example line

## Step 3

If python 3 installed on user OS, then it just has to run the main python code. In windows, user can hold the shift key and then right click on the folder and open the command prompt on the root directory. Then just add this line and press enter:

```
> python FlareNet_Particle_Penetration_Calculator.py
```

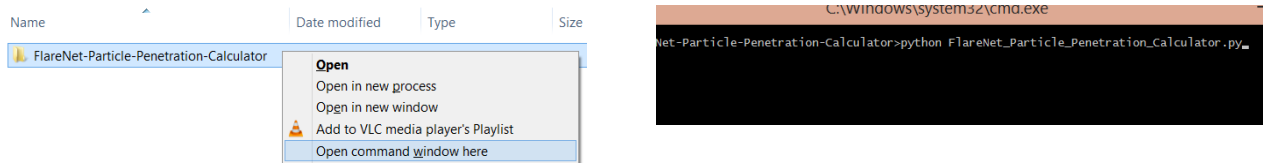


Figure 4 running FPPC

## Step 4

It depends on how many sections that user entered on the excel input file then it started the calculation, and it will create two folders one for graph and another for excel outputs. The output data will be described in the next chapters.

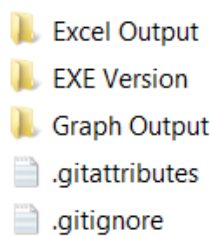


Figure 5 Excel Output and Graph Output directories

## With EXE File

### Step 1

Go to the EXE Version folder and find the address of the zip file which should be downloaded and then unzip that.



Figure 6 GitHub view of the address

### Step 2

Then user should repeat the step 2 of the Python method and enter the data of each section.

### Step 3

After saving the excel input, user should double click on “FlareNet\_Particle\_Penetration-Calculator.exe” and it will do all the calculation, it will create two folders one for graph and another for excel outputs. The output data will be described in the next chapters.

## C- Excel Data Entry

Data input is a description of each section in your lab experiment. Properties of each section should be listed in the table to calculate total penetration. Here is the picture of columns needed to be completed for calculation.

NO	Name	From	To	Ignore	ID(Inch)	Length(Inch)	Bend(Deg)	$\theta$ (from horizon) (Deg)	Tg(K)	Tw(K)	T1(K)	T2(K)	Pg(Pa)	Q(SLPM)	Material	ID (Second) (Inch)	Half_Theta (Second) (Deg)	Sampling Probe	U0(m/s)	Theta Probet(Deg)
1	From Main Duct	C11A	C12A	0	0.75	8	90	-45	330	330	330	330	101325	90	Steel	0.75	0	Y	5.5	-90
2	Line	C12A	C13A	0	0.75	10	0	-90	330	330	330	330	101325	90	Steel	0.75	0	N	0	0
3	Line	C13A	C14A	0	0.75	50	0	-90	330	330	330	330	101325	90	Steel	0.75	0	N	0	0
4	Bend	C14A	C15A	0	0.75	30	90	-45	330	330	330	330	101325	90	Steel	0.75	0	N	0	0
5	Line	C15A	C16A	0	0.75	20	0	0	330	330	330	330	101325	90	Steel	0.75	0	N	0	0
6	Enlargement	C16A	C17A	0	1	8	0	0	330	330	330	330	101325	90	Steel	1	0	N	0	0
7	Line	C17A	C18A	0	2	20	0	0	330	330	330	330	101325	90	Steel	2	0	N	0	0
8	Bend	C18A	C18B	0	0.25	4	90	0	330	330	330	330	101325	1	Steel	0.25	0	N	0	0
9	Line	C18B	C19B	0	0.25	20	0	0	330	330	330	330	101325	1	Silicon	0.25	0	N	0	0
10	Bend	C19B	C20B	0	0.25	4	90	-90	330	330	330	330	101325	1	Silicon	0.25	0	N	0	0
11	Line	C20B	C21B	0	0.25	20	0	-90	330	330	330	330	101325	1	Silicon	0.25	0	N	0	0
12	Bend	C21B	C22B	0	0.25	4	90	0	330	330	330	330	101325	1	Silicon	0.25	0	N	0	0
13	PAX Left Line	C22B	C23B	0	0.25	10	0	0	330	330	330	330	101325	1	Silicon	0.25	0	N	0	0

Figure 7- Input Data Columns

### NO.

The reference number for each section in the lab experiment.

### Name

The brief description of the sections.

### From & To

If there is schematic and you want to define precisely from which point to which point that section means, these columns can be used.

### Ignore

By entering 1, you can ignore that section, and in the code, it assumes that the penetration would be 100%.

### ID (inch)

The inner diameter of the tube in inch.

### Length (inch)

The length of the section in inch.

### Bend (Deg)

If the section has the bend from its initial direction, the bend degree should be recorded here in degree.

### $\theta$ (from horizon) (Deg)

If the section has the angle from the horizon this section should be used for that.

### Tg (K)

Temperature of the gas in kelvin for finding gas properties.

### Tw (K)

Temperature of the wall in Kelvin for thermophoresis calculation.

### T1 (K)

Temperature of the gas in kelvin when it entered the section.



### **T2 (K)**

Temperature of the gas in kelvin when it got out of the section.

### **Pg (Pa)**

Pressure of the gas in Pascal.

### **Q (SLPM)**

Standard volumetric flow rate in liter per minute.

### **Material**

The material of the wall for that section.

### **ID (Second) (Inch)**

If there is contraction on that section the second dimeter of the section should be recorded here in inch and it should be smaller than the former inner diameter.

### **Half\_Theta (Second) (Deg)**

Half theta is the half of the total angle in the contraction. Here is the schematic of the contraction.

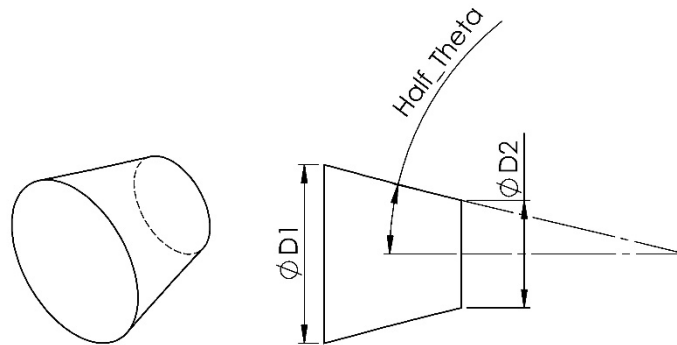


Figure 8- Contraction Schematic

### **Sampling Probe**

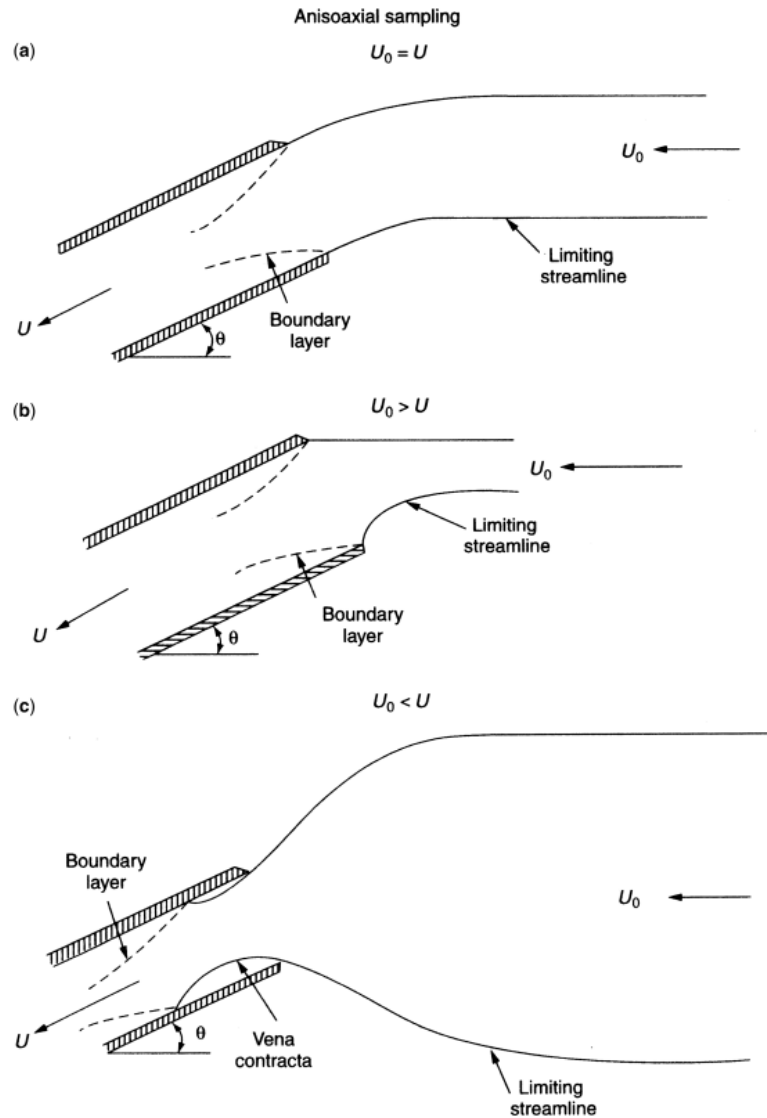
Stating that section is sampling probe or not with Y or N. If the answer is yes the code will calculate aspiration and inlet collection penetration.

### **U0 (m/s)**

For calculating sampling probe penetration, a speed of the main duct that being sampled is needed. The speed of the air in the main duct can be entered here, in meter per second.

### **Theta probe (deg)**

If the sampling probe has been inclined the flow, it can change the particle loss of the system, and I should be considered.



### D-Python Variables

Here are the list variables used for the calculating the particle penetration coefficients. The user can change these variables if its desired.

**T\_0 = 294.26(K)** Standard temperature for calculating actual flow rate

**P\_0 = 101325(Pa)** Standard pressure for calculating actual flow rate

**D\_small = 10e-9(m)** Smallest particle diameter of the computation (10nm)

**D\_large = 10e-6(m)** Largest particle diameter of the computation (10um)

**Nd = 40** Number of points (between above diameters) considered for the computation, points distributed logarithmically

**Reynolds\_Lam\_to\_Turb = 3000** Transition Reynolds between laminar and turbulence flow

**Particle\_thermal\_conduction = 0.2** (w/m/k)

**eff\_k = 0.1074** Effective density variable for calculating particle density for the results to be in  $kg / m^3$ ,

$$\rho_{eff} = K \times d_p^{D_m-3}$$

**eff\_dm = 2.46** Effective density variable for calculating particle density for the results to be in  $kg / m^3$ ,

$$\rho_{eff} = K \times d_p^{D_m-3}$$

**Polynomial\_Deg = 6** Graph's fitting line polynomial degree

## E- Output Data

### Excel Outputs

Here in the excel output directory user can find results for each section. There is the main report file with a summary of all sections in it.

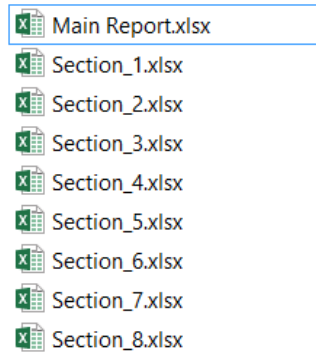


Figure 10 Excel output directory

### Main Report

Main report excel file has different sheets. Each sheet named after the loss mechanisms and the last sheet represents the total penetration variables for the different sections.

Each sheet consists of three parts. First part is the same data that the user entered on the input file. The second portion (separated by empty column) is the calculated variables for each section of the experiment such as flow rate and mean free path, etc. The third portion is the computed penetration coefficients for a specific particle, and the particle diameter can be found at the first row just like the former portion labels.

NO	Name	From	To	Ignore	ID(Inch)	Length(Inch)	Bend(Deg)	$\theta$ (from horizon) (Deg)	Tg(K)	Tw(K)	T1(K)	T2(K)	Pg(Pa)	Q(SLPM)	Material
1	From Main Duct	B11A	B12A	0	0.25	4	90	-45	400	298	400	298	101325	4.3	Steel
2	Line	B12A	B13A	0	0.25	93	0	-90	400	298	400	298	101325	4.3	Steel
3	Line	B13A	B14A	0	0.25	40	0	-90	298	298	298	298	101325	4.3	Silicon
4	Bend	B14A	B15A	0	0.25	25	90	-45	298	298	298	298	101325	4.3	Silicon
5	Line	B15A	B16A	0	0.25	20	0	0	298	298	298	298	101325	4.3	Silicon
6	Bend	B16A	B16C	0	0.25	2	90	45	298	298	298	298	101325	0.5	Steel
7	Line	B16C	B17E	0	0.25	2	0	90	298	298	298	298	101325	0.3	Steel
8	Catalytic Stripper-DMA-CPMA	B17E	B18E	0	0.25	10	30	20	298	298	298	298	101325	0.3	Silicon

Figure 11 Part of the first portion

	Inner Diameter(meter)	Length(meter)	Bend Angle(Rad)	Theta Angle(Rad)	Actual Flow rate(lpm)	Actual Flow rate(m3/s)	Flow velocity rate(m/s)	Dynamic Viscosity(kg/m/s)
	0.00635	0.1016	1.570796327	-0.785398163	5.845170937	9.74195E-05	3.076157969	2.28516E-05
	0.00635	2.3622	0	-1.570796327	5.845170937	9.74195E-05	3.076157969	2.28516E-05
	0.00635	1.016	0	-1.570796327	4.354652348	7.25775E-05	2.291737687	1.83644E-05
	0.00635	0.635	1.570796327	-0.785398163	4.354652348	7.25775E-05	2.291737687	1.83644E-05
	0.00635	0.508	0	0	4.354652348	7.25775E-05	2.291737687	1.83644E-05
	0.00635	0.0508	1.570796327	0.785398163	0.506354924	8.43925E-06	0.266481126	1.83644E-05
	0.00635	0.0508	0	1.570796327	0.303812955	5.06355E-06	0.159888676	1.83644E-05
	0.00635	0.254	0.523598776	0.34906585	0.303812955	5.06355E-06	0.159888676	1.83644E-05

Figure 12 Part of the second portion

	0.00000001	1.19378E-08	1.4251E-08	1.70125E-08	2.03092E-08	2.42446E-08	2.89427E-08	3.45511E-08	4.12463E-08	4.92388E-08
	0.988408085	0.990829512	0.992747021	0.994263273	0.995460981	0.996406398	0.997152346	0.997740793	0.998204984	0.998571209
	0.913299032	0.929836095	0.943416274	0.954514222	0.963535871	0.970831602	0.976703386	0.981409363	0.985167753	0.988160865
	0.952553212	0.961963386	0.969580345	0.975715551	0.980635805	0.984567194	0.987699043	0.990188096	0.992162813	0.993727564
	0.964792012	0.971888171	0.977594141	0.982163142	0.985808938	0.988709844	0.991012964	0.99283848	0.994283769	0.995427209
	0.969500264	0.975686994	0.980647752	0.984610507	0.987766257	0.990273194	0.992260995	0.993835028	0.995080288	0.996064938
	0.972337577	0.977970659	0.98247956	0.986076051	0.988936669	0.991206961	0.993005767	0.994429327	0.995555063	0.996444929
	0.961573891	0.969285256	0.975497319	0.980479539	0.984460423	0.987631455	0.990151264	0.992149913	0.993733112	0.994986153
	0.896811079	0.91614275	0.932070511	0.945141978	0.955817178	0.964490313	0.971501547	0.977143556	0.981665969	0.985279339
	0.674405123	0.729846064	0.777670739	0.818278472	0.852291949	0.880454223	0.903547338	0.922333794	0.937519097	0.949731629

Figure 13 part of the third portion, first row consists of particle diameter in meter

## Sections

In the section files, the user can find properties of the particles for different diameters such as effective density and particle diffusion coefficients, etc. these numbers based on the conditions that each section had for calculation and these conditions inputted by the user in the excel input file.

	Knudsen Number	Cunningham Correction Factor	Mobility (s/kg)	Particle Diffusion Coefficient (m <sup>2</sup> /s)	Zeta	Schmidt Number	Diffusion Penetration Efficiency
0.00000001	19.14371028	32.29353049	1.49943E+13	8.28048E-08	0.000271302	312.7247396	0.988408085
1.19378E-08	16.03625805	27.14683118	1.05586E+13	5.83091E-08	0.000191044	444.1009488	0.990829512
1.4251E-08	13.4332148	22.83637015	7.44034E+12	4.10886E-08	0.000134623	630.2267697	0.992747021
1.70125E-08	11.25270367	19.22655196	5.2474E+12	2.89782E-08	9.49445E-05	893.6050021	0.994263273
2.03092E-08	9.426138258	16.20381419	3.70456E+12	2.04581E-08	6.7029E-05	1265.764233	0.995460981
2.42446E-08	7.896065251	13.67305332	2.61856E+12	1.44607E-08	4.73792E-05	1790.719833	0.996406398
2.89427E-08	6.614357305	11.55463052	1.85366E+12	1.02366E-08	3.35394E-05	2529.648417	0.997152346
3.45511E-08	5.540699218	9.781862949	1.31453E+12	7.25939E-09	2.37847E-05	3567.120095	0.997740793
4.12463E-08	4.641319845	8.298921143	9.3422E+11	5.15914E-09	1.69034E-05	5019.272146	0.998204984
4.92388E-08	3.887929855	7.059066307	6.65659E+11	3.67603E-09	1.20442E-05	7044.305743	0.998571209
5.87802E-08	3.256831905	6.023171444	4.7578E+11	2.62745E-09	8.60859E-06	9855.605483	0.998860221
7.01704E-08	2.728175265	5.158479547	3.41334E+11	1.88498E-09	6.17597E-06	13737.5694	0.99908839
8.37678E-08	2.285331417	4.437559474	2.45968E+11	1.35834E-09	4.45046E-06	19063.84519	0.999268619
1E-07	1.914371028	3.837426591	1.78177E+11	9.83966E-10	3.22387E-06	26317.0791	0.999411079
1.19378E-07	1.603625805	3.338800876	1.29861E+11	7.17146E-10	2.34966E-06	36108.57291	0.999523784

Figure 14 Part of the Section files, first column consists of particle diameter in meter

## Graphs

Here in the graph output directory user can find results for each section. There is the main report graph with a total penetration of all sections in it.

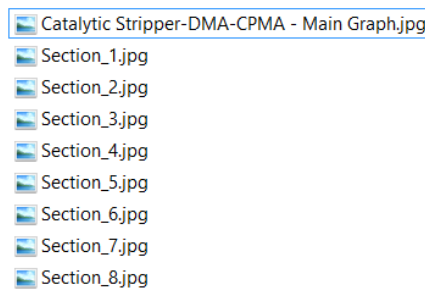


Figure 15 Graph output directory

## Main Graph

In the main graph total penetration of different particle diameters have shown. Different mechanism cause particles to be lost en route to the sampling device and now with this graph user has an estimation about how much of the specific particle diameter has been lost and correct their measurement. The fitted line equation can be seen on the right hand side of the graph. “x” in the fitted line stands for  $\ln(dp)$  and  $dp$  in meter.

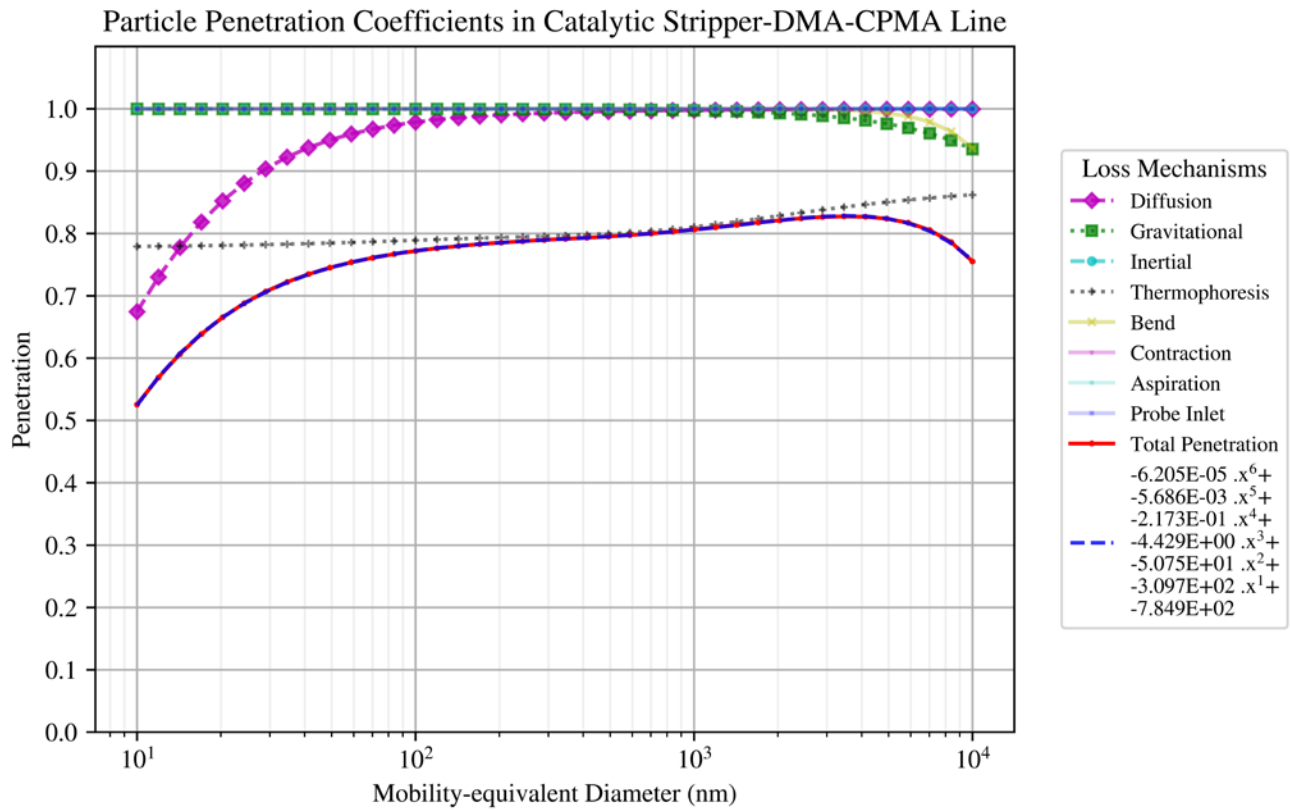


Figure 16 Total penetration graph

## Sections

For each section, the user can find the same graph with fitted line coefficients on it. These graphs can be used in order to find out which section is more responsible for the loss and why.

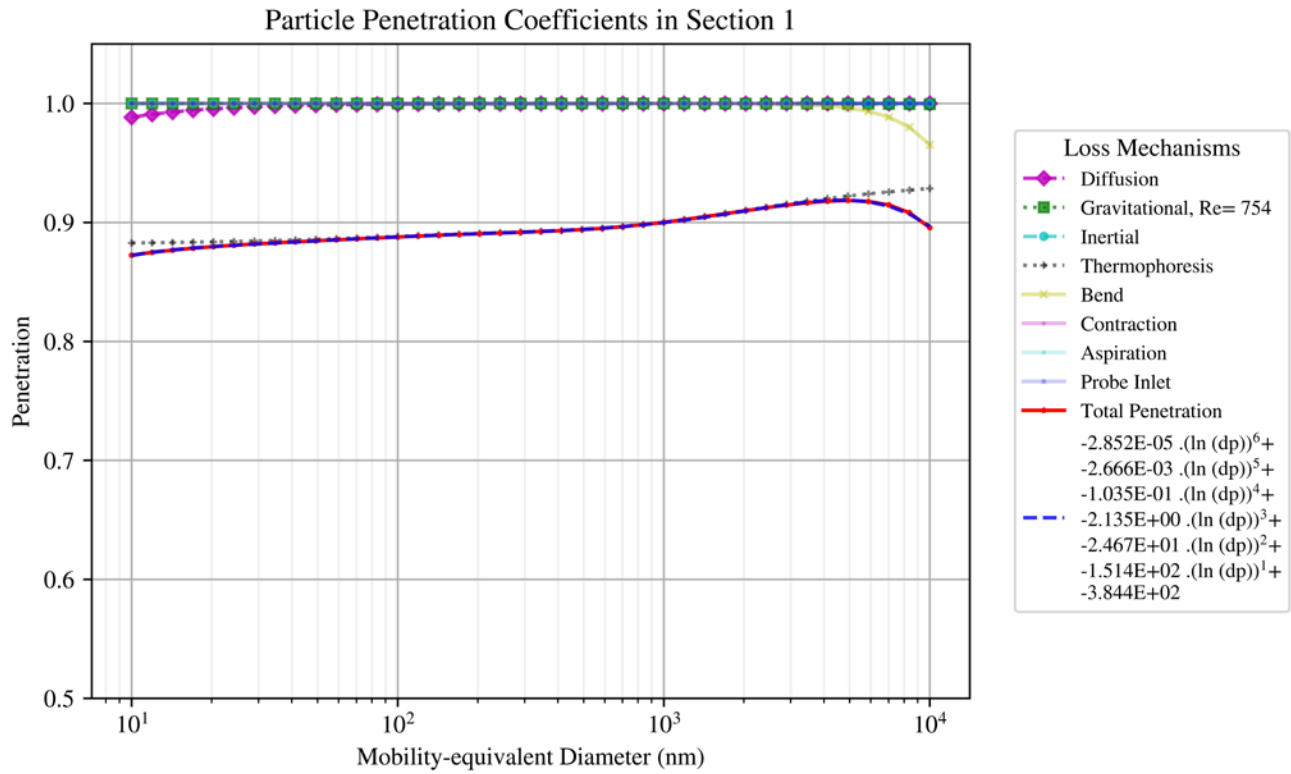


Figure 17 Section 1 penetration graph

## F- Functions

Here user can find what functions writers used for computation of the particle penetration.

### 1- Area\_with\_Diameter (A)

#### Description

Finding the inside area of tube by entering the diameter of the tube ( $m^2$ ).

#### Relation

$$Area = \frac{\pi A^2}{4}$$

#### Code

```
def Area_with_Diameter (A):
    Area = pi * (A ** 2) / 4
    return Area
```

### 2- LPM\_to\_M3perSec (A)

#### Description

Changing the input units of volumetric flow rate to SI value ( $\frac{m^3}{s}$ ).

#### Relation

$$V = \frac{A}{60000}$$

#### Code

```
def LPM_to_M3perSec (A):
    M3persec = A / 60000
    return M3persec
```

### 3- ActualFlowRateLPM (Q\_Slpm, Pg\_pa, T\_0, P\_0\_pa, Tg\_K)

#### Description

Finding Actual flowrate from reported the standard liter per minute (SLPM).

#### Relation

$$V = \frac{Q \times P_0 \times T_g}{P_g \times T_0}$$

#### Code



```
def ActualFlowRateLPM (Q_Slpm, Pg_pa, T_0, P_0_pa, Tg_K):
    AFR = Q_Slpm / Pg_pa / T_0 * P_0_pa * Tg_K
    return AFR
```

## Example

$$T_0 = 294.26 \text{ K}$$

$$P_0 = 101325 \text{ pa}$$

$$T_g = 298 \text{ K}$$

$$P = 98000 \text{ pa}$$

$$Q = 20 \text{ SLPM}$$

$$Q_{actual} = 20.941 \text{ LPM}$$

## 4- Inch\_to\_Meter (A)

### Description

Changing the unit of length.

### Relation

$$L = A \times 0.0254$$

### Code

```
def Inch_to_Meter (A):
    if A != 0:
        meter = A * 0.0254
    else:
        meter = 0
    return meter
```

## 5- Deg\_to\_Radian (A)

### Description

Changing the unit of angle.

### Relation

$$Rad = \frac{A \times \pi}{180}$$

### Code

```
def Deg_to_Radian (A):
    Rad = A * pi / 180
    return Rad
```

## 6- Flow\_Velocity (Flowrate, Area)

### Description

Finding flow velocity from flow rate and area.

### Relation

$$V = \frac{Q}{A}$$

### Code

```
def Flow_Velocity (Flowrate, Area):
    Velocity = Flowrate / Area
    return Velocity
```

## 7- Kn (l, d)

### Description

Calculating Knudsen number from mean free path and particle diameter.

### Relation

$$Kn = \frac{2 \times l}{d}$$

### Code

```
def Kn (mfp, dp):
    Kn = 2 * mfp / dp
    return Kn
```

### Example

$$\lambda = 68.64 \text{ nm}$$

$$d_p = 10 \text{ nm}$$

$$Kn = 13.72$$

## 8- Cc (Kn)

### Description

Calculating Cunningham correction factor from Knudsen number (Atmospheric chemistry and physics 2<sup>nd</sup> Ed, Eq. 9.34).

### Relation

$$Cc = 1 + Kn \times (1.257 + 0.4 \times \exp(\frac{-1.1}{Kn}))$$

### Code

```
def Cc (Kn):
    result = 1 + Kn * (1.257 + 4 * exp (-1.1 / Kn))
    return result
```

## Example

$$d_p = 10 \text{ nm}$$

$$Kn = 13.72$$

$$Cc = 23.32$$

## 9- visc (T, P)

### Description

Calculating the dynamic viscosity from Temperature for Air (kg/m/s). (Sutherland Equation)

### Relation

$$\mu = \mu_0 \times \left( \frac{Tg(\text{kelvin})}{T0(\text{kelvin})} \right)^{3/2} \times \left( \frac{T0(\text{kelvin}) + T_{\text{Sutherland}}(\text{kelvin})}{Tg(\text{kelvin}) + T_{\text{Sutherland}}(\text{kelvin})} \right)$$

### Code

```
def visc (T, P): # ASSUME INDEPENDENT OF P HERE, Sutherland Equation
    # Assume Air, https://www.cfd-online.com/Wiki/Sutherland%27s_law
    U0 = 1.716 * 10 ** (-5)
    T_ref = 273.15
    b = 1.458e-6
    S = 110.4
    Viscosity = U0 * ((T / T_ref) ** 1.5) * ((T_ref + S) / (T + S))
    return Viscosity
```

## Example

$$T = 298 \text{ K}$$

$$\mu = 1.84E-5 \text{ (kg / m / s)}$$

## 10- mfp (visc, T, P)

### Description

Calculating mean free path for the air (Atmospheric chemistry and physics 2<sup>nd</sup> Ed, Eq. 9.6).

### Relation

$$Mfp = \frac{2 \times \mu}{P \times (8M_{\text{Air}} / (\pi RT))^{0.5}}$$

### Code

```
def mfp (visc, T, P):
    # For Air
```

```
M = .029 # kg/mol/K
R = 8.314 # J/mol/K
l = 2 * visc / (P * (8 * M / pi / R / T) ** 0.5)
return l
```

## Example

$$\mu = 1.84E-5 (kg / m / s)$$

$$T = 298 K$$

$$P = 98000$$

$$\lambda = 68.64 nm$$

## 11- KinViscosity (Vis, rho)

### Description

Calculating the kinematics viscosity of the fluid.

### Relation

$$A = \frac{\mu}{\rho}$$

### Code

```
def KinViscosity (visc, rho):
    A = visc / rho
    return A
```

## Example

$$\mu = 1.84E-5 (kg / m / s)$$

$$\rho = 1.14 (kg / m^3)$$

$$\nu = 1.6E-5 (m^2 / s)$$

## 12- Reynolds (rho, Velocity, Diameter, Vis)

### Description

Calculating Reynolds number of the section.

### Relation

$$Re = \frac{\rho u D}{\mu}$$

### Code

```
def Reynolds (rho, Velocity, Diameter, visc):
    A = rho * Velocity * Diameter / visc
    return A
```

## Example

$$\rho = 1.14 (\text{kg} / \text{m}^3)$$

$$V = 2.75 (\text{m} / \text{s})$$

$$D = 0.0127 \text{ m}$$

$$\mu = 1.84E-5 (\text{kg} / \text{m} / \text{s})$$

$$\text{Re} = 2183$$

## 13- SpecificHeat (T)

### Description

Calculating specific heat (Cp) of the air (J/kg/k). (Incropera, F. P., & DeWitt, D. P. (2002). Fundamentals of heat and mass transfer)

### Code

```
def SpecificHeat (T):
    # For Air
    A = 287 * (3.653-0.001334 * T+0.000003291 * (T** 2)-0.00000000191 * (T** 3)+0.000000000000275 * (T** 4))
    return A
```

## 14- Prandtl (C, u, K)

### Description

Calculating Prandtl number.

### Relation

$$\text{Pr} = \frac{\mu C_p}{k}$$

### Code

```
def Prandtl (C, visc, K):
    A = C * visc / K
    return A
```

## Example

$$\mu = 1.84E-5 (\text{kg} / \text{m} / \text{s})$$

$$C_p = 1004.3 (\text{J/kg/K})$$

$$k = 0.02604 (\text{W/m/K})$$

$$\text{Pr} = 0.71$$

## 15- ThermalCond (T)

### Description

Calculating thermal conductivity of the air (W/m/K). (Incropera, F. P., & DeWitt, D. P. (2002). Fundamentals of heat and mass transfer)

## Code

```
def ThermalCond (T):
    # For Air
    A = -0.0003933+0.00010184 * T-0.000000048574 * (T** 2)+0.000000000015207 * (T** 3)
    return A
```

## 16- rho (T, P)

### Description

Calculating density of the air from ideal gas relation. ( $\frac{kg}{m^3}$ )

### Relation

$$\rho = \frac{P}{287.05 \times T}$$

## Code

```
def rho (T, P):
    result = P / 287.05 / T # air density kg/m^3, R in J/kg/K T in K
    return result
```

### Example

$$P = 98000 \text{ pa}$$

$$T = 298 \text{ K}$$

$$\rho = 1.14 (kg / m^3)$$

## 17- Mobility (d, Cc, visc)

### Description

Mobility of particular diameter of the particles. ( $\frac{s}{kg}$ )

### Relation

$$B = \frac{Cc}{3\mu d_p \pi}$$

## Code

```
def Mobility (dp, Cc, visc):
    B = Cc / (3 * visc * dp * pi)
    return B
```

## Example

$$d_p = 10 \text{ nm}$$

$$\mu = 1.84E-5 (\text{kg} / \text{m} / \text{s})$$

$$Cc = 23.32$$

$$B = 1.35E+13 (\text{s} / \text{kg})$$

## 18- Particle\_Diff\_Coeff (B, T)

### Description

Particle Diffusion Coefficient. ( $\frac{m^2}{s}$ )

### Relation

$$D = \text{mobility} \times \text{Boltzmann cons} \times \text{Temperature}$$

### Code

```
def Particle_Diff_Coeff (Mobility, T):
    k = 1.3806e-23 # Boltzmann constant
    D = Mobility * k * T
    return D
```

## Example

$$B = 1.35E+13 (\text{s} / \text{kg})$$

$$T = 298 \text{ K}$$

$$k = 1.38E-23 (\text{m}^2 \text{kg} \text{s}^{-2} \text{K}^{-1})$$

$$D = 5.54E-8 (\text{m}^2 / \text{s})$$

## 19- Zeta (D, L, Q)

### Description

Variable needed to be calculated for diffusion loss. D is particle diffusion coefficient (Willeke, K. and Baron, P.: Aerosol Measurement: Principles, Techniques, and Applications, Van Nostrand Reinhold, New York, 2005)

### Relation

$$\xi = \frac{\pi \times D \times L}{Q}$$

### Code

```
def Zeta (Diffusion, Length, Q):
    A = pi * Diffusion * Length / Q
    return A
```

## Example

$$D = 5.54E - 8 (m^2 / s)$$

$$L = 0.762 m$$

$$Q = 3.49E - 4 (m^3 / s)$$

$$\xi = 3.8E - 4$$

## 20- Schmidt (vis, dens, diff)

### Description

Schmidt number. (Willeke, K. and Baron, P.: Aerosol Measurement: Principles, Techniques, and Applications, Van Nostrand Reinhold, New York, 2005)

### Relation

$$Sch = \frac{\mu}{\rho_f \times D}$$

### Code

```
def Schmidt (visc, density, diffusion):
    A = visc / (density * diffusion)
    return A
```

## Example

$$\mu = 1.84E - 5 (kg / m / s)$$

$$\rho_f = 1.14 (kg / m^3)$$

$$Sch = 289.1$$

## 21- Effective\_Density (K, Dm, dp)

### Description

Effective Density of the particles. K and Dm can be altered at main code. ( $kg / m^3$ )

### Relation

$$\rho = K \times d_p^{D_m - 3}$$

### Code



```
def Effective_Density (K, Dm, dp):
    A = K * dp ** (Dm-3)
    return A
```

## Example

$$K = 0.1074$$

$$D_m = 2.46$$

$$d_p = 10nm$$

$$\rho = 2244 (kg / m^3)$$

## 22- Gravit\_Depos (L, vts, d\_tube, U)

### Description

Gravitational deposition parameter. (Thomas (1958))

### Relation

$$Z = \frac{LV_{ts}}{dU}$$

### Code

```
def Gravit_Depos (Length, V_ts, d_tube, U):
    A = Length * V_ts / (d_tube * U)
    return A
```

## 23- K\_Thermopho (Knud, Cc, Thermal\_Cond\_Fluid, Thermal\_Cond\_Particle):

### Description

This number will be used in the thermophoresis loss calculation. (Talbot et al. 1980)

### Code

```
def K_Thermopho (Knud, Cc, Thermal_Cond_Fluid, Thermal_Cond_Particle):
    Cs = 1.17 # Talbot et al., 1980; Montassier et al., 1991
    Ct = 2.18
    Cm = 1.14
    K_ratio = Thermal_Cond_Fluid / Thermal_Cond_Particle
    K = (2 * Cs * (K_ratio+Ct * Knud) * Cc) / ((1+3 * Cm * Knud) * (1+2 * K_ratio+2 * Ct * Knud))
    return K
```

## 24- Thermophoresis\_Eff (Kth, Re, Pr, Re\_t, Tw, Te, k, Cp, Q, rhog, L, D):

### Description

Depends on the flow regime there are different ways to calculate thermophoresis loss. (Jyh-Shyan Lin, Chuen-Jinn Tsai, 2003)

### Code

```
def Thermophoresis_Eff (Kth, Re, Pr, Re_t, T_wall, T_entrance, k, Cp, Q, rho_g, L, D):
    if T_entrance!=T_wall:
        if Re < Re_t: # laminar
            # Jyh-Shyan Lin, Chuen-Jinn Tsai, 2003, Thermophoretic deposition efficiency in a cylindrical tube taking into account
            # developing flow at the entrance region
            A = 1-0.783 * (Pr * Kth * (T_entrance-T_wall) / T_wall) ** (0.94)
            # Stratmann et al. (1994)
            # A= exp(-0.845*((Pr*Kth+0.025)/((T_wall/(T_entrance-T_wall))+0.28))**0.932)
            # Batchelor and Shen (1985)
            # A = 1-(Pr*Kth*((T_entrance-T_wall)/T_entrance)*(1+(1-Pr*Kth)*((T_entrance-Tw)/T_entrance)))
            # Walker et al. (1979)
            # A= 1-((Pr*Kth/T_wall)*(T_entrance-T_wall))
        else: # Turbulence
            f = (0.79 * log (Re)-1.64) ** (-2) # smooth surface
            Nu = ((f / 8) * (Pr * (Re-1000))) / (1+12.7 * ((f / 8) ** (0.5)) * ((Pr ** (2 / 3))-1)) # fully developed turbulent flow,
            # 0.5<Pr<2000, 3000<ReD<5*10e6 (Incropera and DeWitt, 1996)
            h = Nu * k / D
            A = ((T_wall+(T_entrance-T_wall) * (exp (-1 * ((pi * D * h * L) / (rho_g * Q * Cp)))))) / T_entrance) ** (Pr * Kth)
        else:
            A=1
    return A
```

## 25- Settling\_Eff (Zg, Re, Re\_t, theta):

### Description

Particles settle due to gravitational force and deposit on the lower wall of no vertical lines in a sampling system during transport. (Aerosol measurement principles and techniques and applications, Kulkarni, Baron and Willeke, third Ed.)

### Code

```
def Settling_Eff (Zg, Re, Re_t, theta):
    if Re < Re_t: # laminar flow expression from Brockman
        ep = 0.75 * Zg * cos (theta)
        B = (1-(ep ** (2 / 3))) ** 0.5
        eta = 1-(2 / pi) * ((2 * ep * B)-((ep ** (1 / 3)) * B)+(asin (ep ** (1 / 3))))
    else:
        eta = exp (-4 * Zg * cos (theta) / pi)
    return eta
```

## 26- Terminal\_Velocity (Dp, rho\_p, Cc, visc):

### Description

The particle eventually reaches a terminal velocity that depends on the size of the particle and the viscosity of the air. A drag force is generated, depending on the velocity of the particle that acts in a direction opposite to the direction of motion of the particle. (Aerosol measurement principles and techniques and applications, Kulkarni, Baron and Willeke, third Ed.)

### Code

```
def Terminal_Velocity (dp, rho_p, Cc, visc):
    A = ((dp ** 2) * rho_p * 9.81 * Cc / visc) / 18
    return A
```

## 27- Vt\_Turb\_Depos (Stk, Re, U):

### Description

This number will be used in the turbulent inertial deposition. (Willeke and Baron (2005))

### Code

```
def Vt_Turb_Depos (Stk, Re, U):
    A = U * (((6 * 10 ** (-4)) * (0.0395 * Stk * Re ** (3 / 4)) ** 2) + (2 * (10 ** (-8)) * Re)) / (5.03 * Re ** (1 / 8))
    return A
```

## 28- Diffusion\_Eff (zeta, Sc, Re, Re\_t):

### Description

Small particles undergoing Brownian motion will diffuse from high particle concentrations to low particle concentrations. (Aerosol measurement principles and techniques and applications, Kulkarni, Baron and Willeke, third Ed.)

### Code

```
def Diffusion_Eff (zeta, Sch, Re, Re_t):
    if Re < Re_t: # laminar flow expression from Brockman
        Sh = 3.66 + (0.2672 / (zeta + 0.10079 * zeta ** (1 / 3))) # Sherwood Number
        eta = exp (-1 * zeta * Sh)
    else:
        Sh = 0.0118 * Re ** (7 / 8) * Sch ** (1 / 3) # Sherwood Number
        eta = exp (-1 * Sh * zeta)
    return eta
```

## 29- Inertial\_Turb\_Depos\_Eff (d\_tube, L, Vt\_Turb, Q, Re, Re\_t):

### Description

Particle loss due to turbulent inertial deposition. (Aerosol measurement principles and techniques and applications, Kulkarni, Baron and Willeke, third Ed.)

### Code

```
def Inertial_Turb_Depos_Eff (d_tube, Length, Vt_Turb, Q, Re, Re_t):
    if Re > Re_t:
        A = exp (-1 * ((pi * d_tube * Length * Vt_Turb) / Q))
    elif Re > 15600: # not valid
        A = 1
    else:
        A = 1
    return A
```

## 30- Inertial\_Bend\_Depos\_Eff (Stk, Bend\_Rad, Re, Re\_t):

### Description

When the direction of sampling gas flow is diverted in a bend, an aerosol particle may deviate from the gas flow due to its inertia and deposit on the wall of the bend. (Aerosol measurement principles and techniques and applications, Kulkarni, Baron and Willeke, third Ed.)

## Code

```
def Inertial_Bend_Depos_Eff (Stk, Bend_Rad, Re, Re_t):
    if Re < Re_t:
        A = (1+(Stk / 0.171) ** (0.452 * (Stk / 0.171)+2.242)) ** (-2 * (Bend_Rad * 180 / pi) / pi)
    else:
        A = exp (-2.823 * Stk * (Bend_Rad * 180 / pi))
    return A
```

## Example

### 31- Inertial\_Deposit\_Contraction\_Eff (Stk, Theta\_Rad, A\_o, A\_i):

#### Description

Particle loss due to the flow constriction (contraction) in the sampling line. (Muyshondt et al. (1996))

## Code

```
def Inertial_Deposit_Contraction_Eff (Stk, Theta_Rad, A_o, A_i):
    # Validity: 0.001≤Stk(1-Ao/Ai)≤100 and 12≤θcont≤90
    A = 1-(1 / (1+((Stk * (1-(A_o / A_i))) / (3.14 * exp (-0.0185 * (Theta_Rad * 180 / pi)))))) ** (-1.24)))
    return A
```

### 32- Stokes (vts, U, L):

#### Description

Stokes number is the ratio of the particle stop distance to the characteristic length of the flow (L).

#### Relation

$$St = \frac{D_p^2 \rho_p C_c u_0}{18\mu L}$$

## Code

```
def Stokes (V_ts, U, Length):
    A = (V_ts / 9.81) * U / Length
    return A
```

### 33- Aspiration\_Eff (Stk, U0, U):

#### Description

The aspiration efficiency is the ratio of the number concentration of particles that enter the sampling probe cross section to the number concentration of particles in the environmental air. (Thin walled probe) (Aerosol measurement principles and techniques and applications, Kulkarni, Baron and Willeke, third Ed.)

## Code

```
def Aspiration_Eff(Stk, U0, U, theta_Rad):
    # Liu, Zhang, and Kuehn (1989) correlation for isoaxial aspiration efficiency for 0.01 < Stk < 100 and 0.1 < U0/U < 10.
    theta = theta_Rad * 180 / pi
    Stk0 = (U0 / U) * Stk # in correlation, Stokes is based on ambient velocity
    eta_asp = 1

    if theta == 0: # isoaxial
        if (U0 / U) >= 1: # sub-isokinetic sampling
            eta_asp = 1 + (((U0 / U) - 1) / (1 + (0.418 / Stk0)))
        else: # super-isokinetic sampling
            eta_asp = 1 + (((U0 / U) - 1) / (1 + (0.506 * ((U0 / U) ** 0.5) / Stk0)))

    # Anisoaxial
    elif fabs(theta) <= 60: # 0.02 < Stk < 4 and 0.5 < U0/U < 2
        Stk0_prime = Stk0 * exp(0.022 * fabs(theta)) # Durham and Lundgren (1980)
        k = 2 + (0.617 * (U / U0))
        eta_asp = 1 + (((U0 / U) * cos(theta_Rad) - 1) * ((1 - ((1 + k * Stk0_prime) ** (-1))) / (1 - ((1 + 2.617 * Stk0_prime) ** (-1))))) *
        (1 - ((1 + 0.55 * Stk0_prime * exp(0.25 * Stk0_prime)) ** (-1)))

    elif fabs(theta) <= 90: # 0.02 < Stk < 0.2 and 0.5 < U0/U < 2
        eta_asp = 1 + (((U0 / U) * cos(theta_Rad) - 1) * (3 * Stk0 ** ((U / U0) ** 0.5)))

    return eta_asp
```

## 34- InletTrans\_Eff (Stk, U0, U):

### Description

Transmission efficiency in the sampling probe. Inertial and gravitational are the most important mechanisms which affect the efficiency of the sampling probe. (Aerosol measurement principles and techniques and applications, Kulkarni, Baron and Willeke, third Ed.)

### Code

```
def InletTrans_Eff(Stk, U0, U, theta_Rad, Zg, Re):
    eta_inert = 1
    theta = theta_Rad * 180 / pi
    Stk0 = (U0 / U) * Stk # in correlation, Stokes is based on ambient velocity

    # transmission efficiency for gravitational settling, Hangal and Willeke (1990b)
    K_theta = (Zg ** 0.5) * (Stk0 ** 0.5) * (Re ** (-0.25)) * cos(theta_Rad)
    eta_grav = exp(-4.7 * (K_theta ** (0.75)))

    # Hangal and Willeke (1990b) give the transmission efficiency for inertia
    if theta == 0: # isoaxial
        if U0 >= U: # sub-isokinetic sampling, 0.01 < Stk < 100 and 1 < U0/U < 10
            r = (U0 / U) - 1
            eta_inert = (1 + r / (1 + 2.66 * (Stk0 ** (-2 / 3)))) / (1 + r / (1 + (0.418 / Stk0)))
        elif U0 < U: # super-isokinetic sampling, 0.02 < Stk < 4 and 0.25 < U0/U < 1.0
            IV = 0.09 * (Stk0 * (U - U0) / U0) ** 0.3
            eta_inert = exp(-75 * (IV ** 2))

    else: # Anisoaxial
        if U0 <= U:
            IV = 0.09 * (Stk0 * cos(theta_Rad) * (U - U0) / U0) ** 0.3
```

```
else:
    IV = 0
    alpha = 12 * ((1-(fabs(theta) / 90))-exp(-1 * fabs(theta)))
    if theta < 0: # downward sampling
        IW = Stk0 * ((U0 / U) ** 0.5) * sin((theta-alpha) * pi / 180) * sin(((theta-alpha) / 2) * pi / 180)
    if theta > 0: # upward sampling
        IW = Stk0 * ((U0 / U) ** 0.5) * sin((theta+alpha) * pi / 180) * sin(((theta+alpha) / 2) * pi / 180)
    eta_inert = exp(-75 * (IV+IW) ** 2)

eta = eta_grav * eta_inert
return eta
```