







User Guide: Lag Time & Pressure Drop Calculation

This tool simplifies and automates the **Lag Time and Pressure Drop** calculations using a structured Excel template and a macro-enabled calculation file. After you fill in the required input fields in `Lag time and PD Datasheet.xlsx`, simply run the Python script. The script will automatically extract the values, feed them into the macro-enabled calculation engine (`Pressure drop & Lag time Calculation File.xls`), and perform all calculations. It then collects the results—such as pump requirement, lag time, and pressure drop values—and inserts them back into your original Excel file. Finally, it generates a clean, ready-to-use output file called `output.xlsx`, containing both your input data and the calculated results, all with no manual intervention needed.

Required Files

Please make sure these two files are in the same folder as the script:

1. ✓ **Lag time and PD Datasheet.xlsx**
 - This is your **input file**. You will enter your data here.
 - This file is structured as a **template**. Use only this format.
2. ✓ **Pressure drop & Lag time Calculation File.xls**
 - This is the **calculation file** that contains Excel macros.
 - The script will automatically convert this to `.xlsm` and process calculations.

Name	Status	Date modified	Type	Size
 code		17-04-2025 12:08	Jupyter Source File	14 KB
 Lag time and PD Datasheet		17-04-2025 10:59	Microsoft Excel Worksheet	13 KB
 Pressure drop & Lag time Calculation File		03-10-2024 18:04	Microsoft Excel 97-2003 Worksheet	948 KB

Step-by-Step Instructions

1. Open the Input Template (Lag time and PD Datasheet.xlsx)

- You must **fill in columns** as described below:

Column	What to Enter
C	Process Supply Pressure (kg/cm ²)
D	Temperature at Sample Point (°C)
E	Density of Sample (kg/m ³)
F	Viscosity of Sample (cP)
G	Return Pressure (kg/cm ²)
H	Bypass Flow (LPM)
L	Sample Line Length (m)
M	Return Line Length (m)

Do NOT change the column letters, headers, or insert extra columns. These columns are mapped directly in the code.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
Analyzer Tag	Stream Name	Process Supply Pressure (kg/cm ²)	Temperature at Sample Point (deg. C)	Density of Sample (kg/m ³)	Viscosity of Sample (cP)	Return Pressure (kg/cm ²)	Bypass Flow (LPM)	Pump Required	Lag Time (seconds)	Pressure Drop (kg/cm ²)	Sample Line Length (mtrs)	Return Line Length (mtrs)	Pressure Drop Supply Line (kg/cm ²)	Pressure Drop Return Line (kg/cm ²)	Pressure Drop in SHS (kg/cm ²)

2. Run the Python Script

- Double-click the .py file or run it via Python from VS Code or terminal.
- The script will:
 - Automatically run the macro-based Excel file.
 - Insert values from your source sheet to the macro sheet.
 - Perform all necessary calculations.
 - Retrieve results and paste them back in your input Excel file.

3. Wait for Completion

- The script will process each row in each sheet (Row 2 to Row ____).
- It will create intermediate .xlsx files (like Target_Sheet1_Row_2.xlsx) — these are **temporary output files**.
- When done, a new Excel file named **output.xlsx** will be created.

What Will Be Automatically Filled in Output

Once completed, these columns will be filled in:

Column	Output Data	Source
I	Pump Required	D36
J	Lag Time (in seconds, rounded)	D35
K	Pressure Drop (Total)	D34
N	Pressure Drop – Supply Line	D31
O	Pressure Drop – Return Line	D32
P	Pressure Drop – SHS	D33

✓ What You Should Do

- ✓ Use only the provided Excel template.
- ✓ Close the excel files before running the script.
- ✓ Place the input and calculation files in the **same folder as the script**.

✗ What You Should NOT Do

- ✗ Do NOT change column headers or their order.
- ✗ Do NOT insert or delete any columns in the source file.
- ✗ Do NOT remove or edit the macro file (.x1s) manually.
- ✗ Do NOT leave required fields blank.

Final Output

After running the script, you will get a file called:

- **output.xlsx** → This contains all your data and calculated results.
- You can now share or save this final output file.

How to Run the Script

1. Install Python dependencies

Open **Command Prompt** and run:

```
pip install openpyxl pywin32
```

2. Place all required files in one folder

Put the following in a single folder:

- a. The .py script
- b. Lag time and PD Datasheet.xlsx
- c. Pressure drop & Lag time Calculation File.xls

3. Run the script

In the same folder, open Command Prompt and type:

```
python scriptname.py
```

(Replace scriptname.py with the name of your Python file.)