Introduction to Air Dispersion Modeling with PyAERMOD

A hands-on guide for students who have never used a dispersion model before. No programming experience is required — Tutorials 1--4 use the PyAERMOD graphical interface (GUI) in your web browser, and Tutorial 5 has you copy and run one short Python script.

Work through the pages in order; each tutorial builds on the one before. The whole guide is also one PDF for printing.

---

1. What Is Air Dispersion Modeling?

When a factory, power plant, or vehicle emits pollutants into the air, those pollutants don't stay in one place. Wind carries them downwind, and turbulence spreads them out in all directions. An **air dispersion model** is a computer program that predicts *where* those pollutants go and *how concentrated* they are at ground level.

Why Does It Matter?

- **Protecting public health.** Regulatory agencies (like the U.S. EPA) use dispersion models to decide whether a proposed factory will cause air quality problems in nearby neighborhoods.
- **Designing better facilities.** Engineers use models to determine how tall a smokestack needs to be, or whether adding a scrubber will bring concentrations below safe limits.
- **Emergency planning.** If there's an accidental release of a toxic gas, models help predict which areas might be affected.

What Is AERMOD?

AERMOD (AMS/EPA Regulatory Model) is the U.S. EPA's preferred model for near-field (< 50 km) regulatory air quality assessments. It has been the standard since 2005 and is used worldwide.

AERMOD is a **steady-state Gaussian plume model**, which means:

- **Steady-state**: it assumes meteorological conditions are constant during each hour of simulation.
- **Gaussian plume**: it models the spreading pollutant cloud as a bell-shaped (Gaussian) distribution in both the horizontal and vertical directions.

PyAERMOD is a Python wrapper that lets you set up, run, and analyze AERMOD models through a graphical interface or Python code, instead of editing text files by hand.

2. Key Concepts

Before diving into the tutorials, here are the core ideas you'll need.

2.1 Emission Sources

An **emission source** is anything that releases pollutants into the air. AERMOD supports several types:

Source Type	Real-World Example
Point	Smokestack, exhaust vent
Area	Storage pile, parking lot, landfill
Volume	Conveyor transfer point, building vent
Line	Roadway, railway, pipeline
Open Pit	Quarry, surface mine

For these tutorials, we'll focus on **point sources** (smokestacks) because they're the most common and the easiest to understand.

Key Point Source Parameters

- **Stack height** (meters): How tall the smokestack is. Taller stacks release pollutants higher up, giving them more time to dilute before reaching the ground.
- **Stack diameter** (meters): The inner diameter of the stack opening.
- **Exit velocity** (m/s): How fast the exhaust gas leaves the stack. Faster gas shoots the plume higher (this is called **momentum rise**).
- **Exit temperature** (Kelvin): Hot exhaust gas is buoyant — it rises because it's lighter than the surrounding air. This is called **buoyancy rise**. The combination of momentum and buoyancy rise is called **plume rise**.
- **Emission rate** (g/s): The mass of pollutant released per second.

> **Intuition:** Imagine holding a lit candle. The hot smoke rises (buoyancy). If you blow on it, the smoke bends sideways (wind). If you raise the candle overhead, the smoke takes longer to reach the floor (stack height). These are the same physics AERMOD captures mathematically.

2.2 Receptors

Receptors are the locations where AERMOD calculates the ground-level pollutant concentration. Think of them as virtual air quality monitors placed across a map.

Common receptor layouts:

- **Cartesian grid**: A regular rectangular grid of points (like graph paper laid over a map). You specify the boundaries and spacing. A 2 km x 2 km area with 100 m spacing gives 441 receptors.
- **Polar grid**: Rings of points radiating out from a central location (like a dartboard). Useful for studying how concentration changes with distance from a source.
- **Discrete receptors**: Individual points at specific locations of interest (a school, a hospital, a property boundary).

2.3 Meteorology

Wind and atmospheric stability are the two most important factors controlling dispersion:

- **Wind speed and direction** determine where the plume goes.
- **Atmospheric stability** determines how much the plume spreads:
- **Unstable** (sunny daytime): strong vertical mixing, plume spreads rapidly, concentrations drop quickly with distance.
- **Stable** (clear nighttime): weak mixing, plume stays narrow and compact, can travel far with less dilution.
- **Neutral** (overcast, windy): moderate mixing.

AERMOD uses **hourly meteorological data** — typically one to five years of observations from a nearby weather station, preprocessed by the AERMET program into `.sfc` (surface) and `.pfl` (profile) files. For tutorials 1 and 2 you won't need actual met data because we'll only generate the input file. Tutorial 3 shows how to connect real meteorological files.

2.4 What AERMOD Produces

For each receptor, AERMOD calculates the **pollutant concentration** (typically in micrograms per cubic meter, ug/m3) for each hour of the simulation. It then computes summary statistics:

- **Annual average** — the mean concentration over all hours. Compared to the annual NAAQS (National Ambient Air Quality Standard).
- **24-hour average** — the highest 24-hour average over the simulation period.
- **1-hour average** — the highest single-hour concentration.

The results let you answer questions like: *"Will this factory cause the annual PM2.5 concentration at the nearest school to exceed 12 ug/m3?"*

2.5 The Modeling Workflow

Every AERMOD analysis follows this sequence:

```
Define the problem
       |
       v
Set up sources, receptors, and meteorology
       |
       v
Generate the AERMOD input file (.inp)
       |
       v
Run AERMOD  -->  Output file (.out)
       |
       v
Analyze results (tables, maps, compliance checks)
```

The PyAERMOD GUI walks you through each of these steps in order.

---

3. Setting Up PyAERMOD

> **Want to see AERMOD run before you install anything?** Go to **https://atmmod.github.io/pyaermod/browser/** and click **Open the notebooks**. In a few minutes, inside your web browser, the notebooks build a model of one smokestack, run AERMOD and draw a map of the concentrations. Nothing is installed on your computer. The notebooks use short Python code rather than the GUI, so they do not replace the tutorials below. The AERMOD they run is an unofficial build of EPA's code, fine for learning but not for regulatory work.

3.1 Install

Open a terminal (Command Prompt on Windows, Terminal on Mac/Linux) and run:

```
pip install "pyaermod[gui]"
```

This installs the core package plus everything needed for the graphical interface (NiceGUI and the matplotlib plots the Results step draws).

To verify the install worked:

```
pyaermod info
```

You should see the PyAERMOD version and a list of the features that can run on this computer. (Keep the quotes in the `pip install` command: without them the default macOS shell, zsh, reads the square brackets as a file pattern.)

3.2 Install the AERMOD and AERMET Executables

PyAERMOD is a *wrapper* — it generates input files, launches the model, and visualizes results, but it does **not** include the AERMOD or AERMET programs themselves. Those are separate Fortran executables maintained by the U.S. EPA.

Which executables do you need?

Executable	What it does	Which tutorials need it
`aermod`	Runs the dispersion model	Tutorial 3 (running AERMOD)
`aermet`	Preprocesses meteorological data	Tutorial 5 (AERMET processing)
`aermap`	Preprocesses terrain elevations	Optional (terrain analysis)

Tutorials 1–2 only generate input files, so you can **skip this step initially** and come back when you're ready for Tutorial 3.

If you cannot install programs on your computer (a shared lab computer, for example), the browser notebooks run AERMOD without these executables.

Option A: Compile from source (macOS and Linux)

PyAERMOD's build script downloads EPA's public-domain Fortran source and compiles **all three executables** (AERMOD, AERMAP and AERMET). EPA does not publish macOS or Linux executables, and PyAERMOD releases do not include any.

Prerequisites: You need `gfortran` (the GNU Fortran compiler).

On macOS, install it with Homebrew:

```
brew install gcc
```

On Ubuntu/Debian Linux:

```
sudo apt-get install gfortran
```

Build steps:

```
git clone https://github.com/atmmod/pyaermod.git
cd pyaermod
./scripts/build_aermod.sh all
```

This creates executables in the `bin/` directory. Move them to your PATH:

```
mkdir -p ~/bin
cp bin/aermod bin/aermap ~/bin/
```

and AERMET the same way:

```
cp bin/aermet ~/bin/
```

Then add `~/bin` to your PATH so PyAERMOD can find them. Add this line to your `~/.zshrc` (or `~/.bashrc` on Linux):

```
export PATH="$HOME/bin:$PATH"
```

Then **restart your terminal** (close and reopen it), or run `source ~/.zshrc` in your current session, and check:

```
which aermod
aermod
```

`which` should print the path to `aermod`. Running `aermod` alone prints a version header and then stops with an error about a missing input file: **that is normal** and means it is installed.

To build straight into another directory instead of `bin/`, set `BIN_DIR` (a relative path is taken from the directory you run the script in):

```
BIN_DIR=~/bin ./scripts/build_aermod.sh all
```

After each executable the script prints a short build record: its path, its SHA-256 checksum, the compiler version and the compile and link flags, and for AERMOD the version number in its banner (26135 for the current EPA source; "unknown (banner not found)" means the check could not read it). Keep it with your results if you need to say exactly which executable produced them.

A build replaces an executable of the same name in that directory (the script tells you when it does). To keep a second AERMOD, for example one built from source you have changed, give it another name with `AERMOD_EXE_NAME=aermod_test` or build it into its own `BIN_DIR`.

Option B: Windows

The EPA provides official Windows executables:

1. Go to the EPA SCRAM website.
2. Download the **AERMOD Modeling System** zip file.
3. Extract the zip. Inside you'll find `aermod.exe`, `aermet.exe`, and `aermap.exe`.
4. Place them in a folder (e.g., `C:\AERMOD\`) and add that folder to your system PATH: - Search for **"Environment Variables"** in the Start menu. - Under **System variables**, find `Path`, click **Edit**. - Click **New** and add `C:\AERMOD\` (or wherever you put the files). - Click **OK** to save.
5. Open a **new** Command Prompt and verify:

```
aermod
```

Verifying your installation

After installing by any method, you can verify everything is set up with:

```
pyaermod info       # Should show the PyAERMOD version and feature list
which aermod        # Should print the path to the aermod executable
aermod              # Should print AERMOD version (error about input is OK)
```

> | **The GUI's Review & Run step looks for the `aermod` executable on your PATH.** If it can't find it, its readiness checklist says "No 'aermod' binary on PATH. Install AERMOD and re-launch." and the **Run AERMOD** button stays disabled. The GUI has no field for the executable's path: put `aermod` on your PATH and start the GUI again.

3.3 Launch the GUI

```
pyaermod-app
```

Your web browser will open automatically to `http://127.0.0.1:8080`. You'll see the PyAERMOD interface with a list of steps on the left and the **Project** step in the main area. (If port 8080 is already in use on your computer, run `pyaermod-app --port 8081` instead.)

> | **Tip:** The GUI runs as a local web app. It never sends your data to the internet — everything stays on your computer.

3.4 GUI Layout

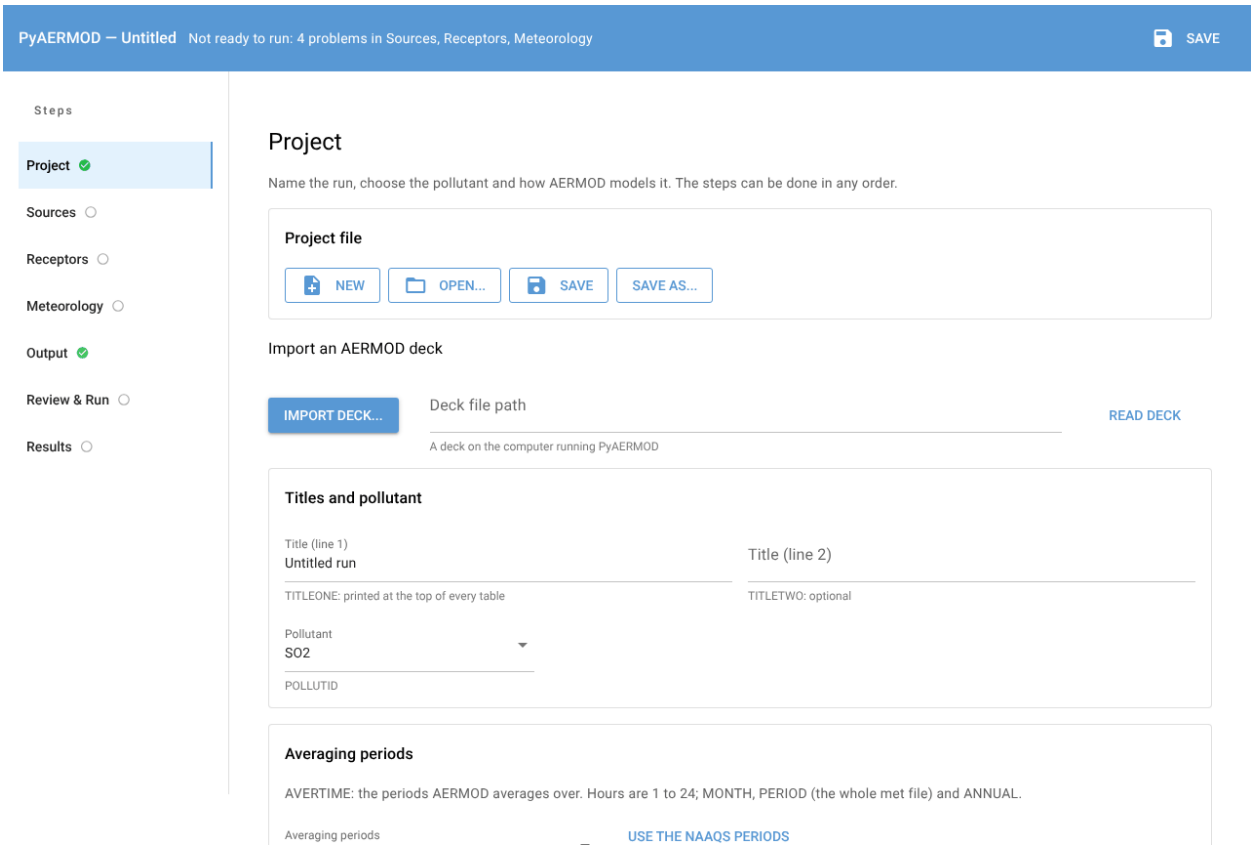
- **Header (top):** The project's name, a "(modified)" marker while there are unsaved changes, a one-line readiness summary ("Ready to run" or "Not ready to run: ...") and a **Save** button.
- **Step list (left):** Navigation between the 7 workflow steps. Each step carries a badge: not started, complete, warning or error. In a narrow window the list is hidden behind the menu button in the header.
- **Main area (center):** The current step — forms, tables, and plots.

The 7 steps, in workflow order:

1. **Project** — open and save project files; titles, pollutant, averaging periods, terrain
2. **Sources** — add and edit emission sources
3. **Receptors** — define where concentrations are calculated
4. **Meteorology** — point to weather data files
5. **Output** — choose the files and tables AERMOD writes
6. **Review & Run** — readiness checklist, input-file preview, and the run
7. **Results** — maxima, concentration map, NAAQS comparison, downloads

You can visit the steps in any order. The GUI calls an AERMOD input file a **deck**, the modelers' word for it.

All coordinates in the GUI are AERMOD's model coordinates: easting and northing in UTM meters. There is no map to click and no latitude/longitude setting.



The GUI when it opens: the header with the readiness line and Save, the step list on the left with a badge per step, and the Project step

---

4. Tutorial 1 — Your First Point Source

Goal: Create a simple AERMOD input file for a single smokestack emitting PM2.5, and understand what each setting means.

Time: 15--20 minutes

What you'll learn:

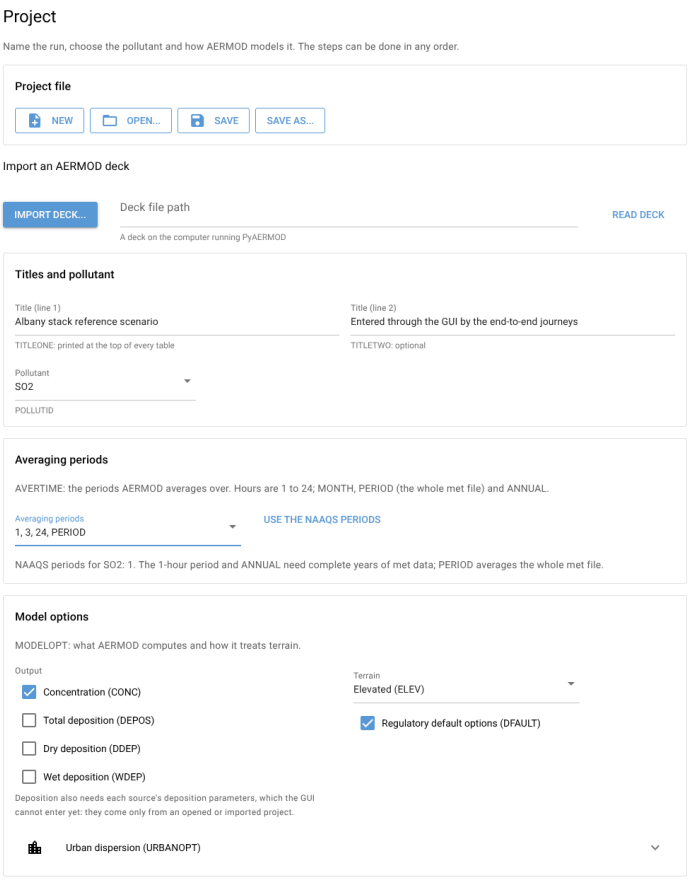
- How to configure a project in the GUI
- How to define a point source (smokestack)
- How to set up a receptor grid
- How to specify meteorology file paths
- How to preview and download the generated AERMOD input file

Step 1: Set Up the Project

1. Launch the GUI (`pyaermod-app`).

2. On the **Project** step, fill in: - **Title (line 1):** `Tutorial 1 - My First Model` - **Title (line 2):** `Single point source, PM2.5` - **Pollutant:** `PM25` - **Averaging periods:** Select `24` and `ANNUAL` (a new project starts with `1` and `ANNUAL`; remove `1`) - **Terrain:** `Flat (FLAT)` - Untick **Regulatory default options (DFAULT)**. AERMOD ignores FLAT while DFAULT is on; the GUI shows a warning on the Project step if you leave it ticked.

| **What you just did:** You told AERMOD what pollutant you're modeling (PM2.5), that the terrain is flat (no hills), and that you want annual and 24-hour average results.



The Project step: titles and pollutant, averaging periods and model options (EPA's Albany example; your values are the ones in the steps above)

Step 2: Add a Point Source

1. Click **Sources** in the step list.
2. Set **Type** to `PointSource` and click **Add**. An editor opens.
3. Fill in the source form:

Parameter	Value	What It Means
Source ID	`STACK1`	A short name for this source
X coord (m)	`500000`	Easting in UTM meters
Y coord (m)	`3870000`	Northing in UTM meters
Stack height (m)	`50`	The stack is 50 meters tall
Stack temp (K)	`400`	Exhaust gas temperature: 400 K (about 127 C)
Exit velocity (m/s)	`15`	Gas exits at 15 m/s
Stack diameter (m)	`2`	Stack opening is 2 meters across
Emission rate (g/s)	`1.5`	Emitting 1.5 grams of PM2.5 per second

Base elevation (m), under **Advanced**	`0`	Ground elevation at the stack base (meters)

4. Click **Save**. (**Close** discards the source.)

> **What you just did:** You defined a medium-sized industrial smokestack. The hot, fast-moving exhaust gas will rise well above the 50 m physical stack height (plume rise), so the "effective" release height might be 80--120 m depending on meteorological conditions. This is why AERMOD needs the temperature and velocity — not just the physical height.

You should see your source appear in the source table and as a marker in the plan view beside it.



The Sources step with one stack in the table and the plan view beside it (EPA's Albany example; your values are the ones in the steps above)

Step 3: Set Up Receptors

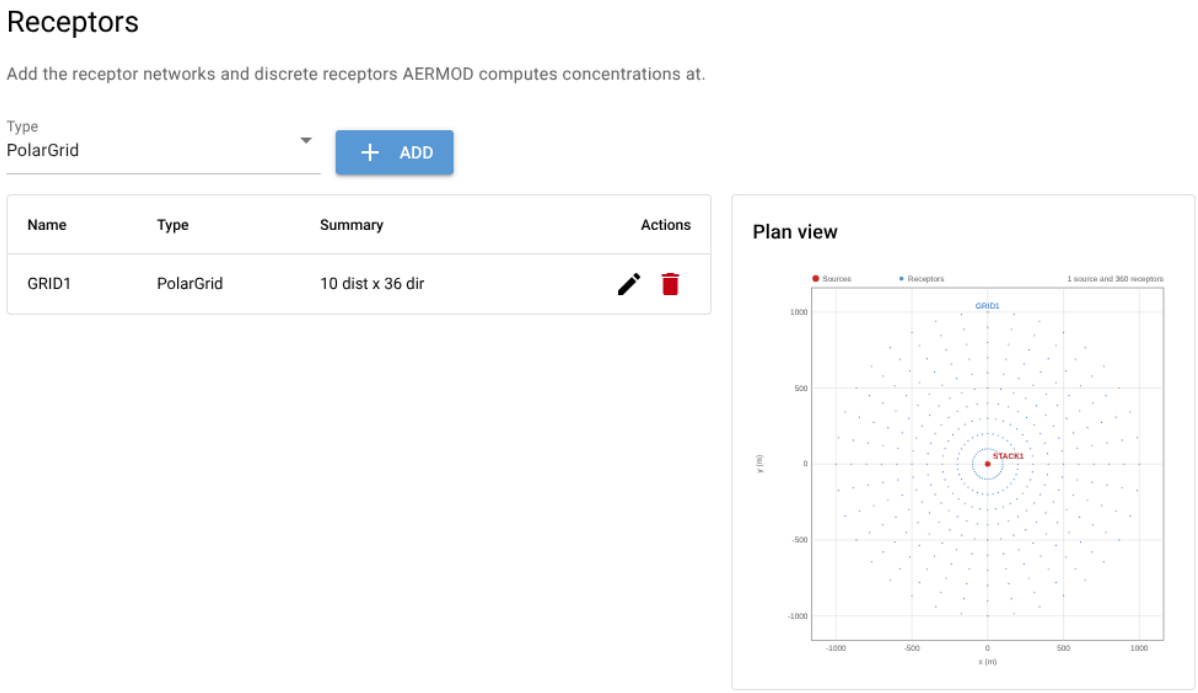
1. Click **Receptors** in the step list.
2. Set **Type** to `CartesianGrid` and click **Add**. In the editor, enter:

Parameter	Value	What It Means
Grid name	`GRID1`	A short name for this grid
X init (m)	`498000`	Western edge of the grid
X num	`21`	21 points across: 4 km total width
X delta (m)	`200`	One receptor every 200 meters
Y init (m)	`3868000`	Southern edge
Y num	`21`	21 points up: 4 km total height

Y delta (m)	`200`	One receptor every 200 meters

3. Click **Save**. The table shows the grid with the summary **21 x 21**: 441 receptors.

> **What you just did:** You placed a 4 km x 4 km grid of virtual air quality monitors centered on your source, with a receptor every 200 meters. AERMOD will calculate the ground-level PM2.5 concentration at each of these 441 points.

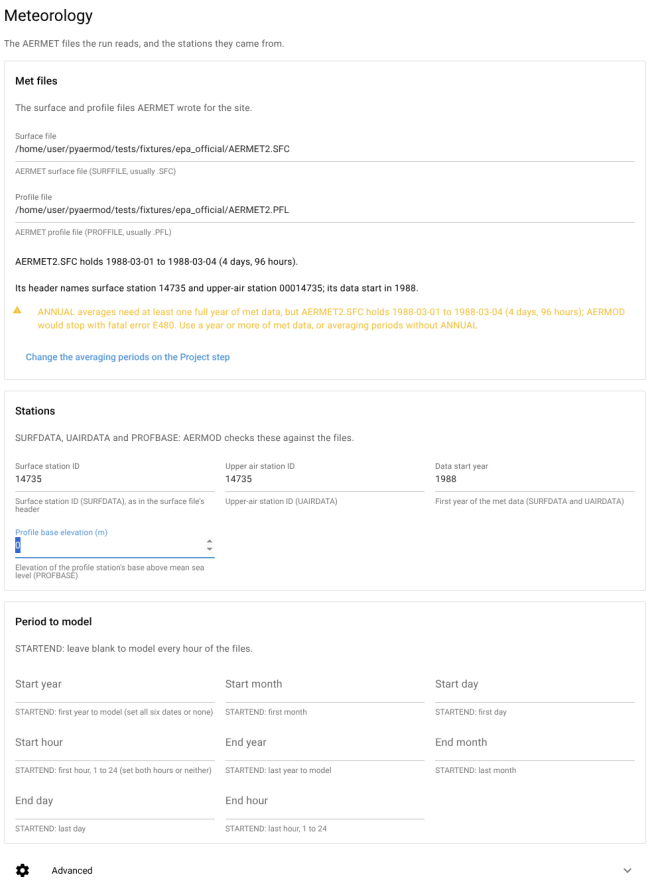


The Receptors step with a polar grid drawn on the plan view (EPA's Albany example; your values are the ones in the steps above)

Step 4: Specify Meteorology

1. Click **Meteorology** in the step list.
2. Under **Met files**, enter: - **Surface file:** `met_data.sfc` - **Profile file:** `met_data.pfl`

> **Note:** You don't need actual meteorological files for this tutorial — we're just generating the input file. Because these files don't exist, the GUI marks the Meteorology step with an error and the header reads "Not ready to run". That is expected here. When you're ready to actually run AERMOD (Tutorial 3), you'll need real `.sfc` and `.pfl` files.



The Meteorology step: the surface and profile files, what the surface file holds, and the stations (EPA's Albany example; your values are the ones in the steps above)

Step 5: Preview the Input File

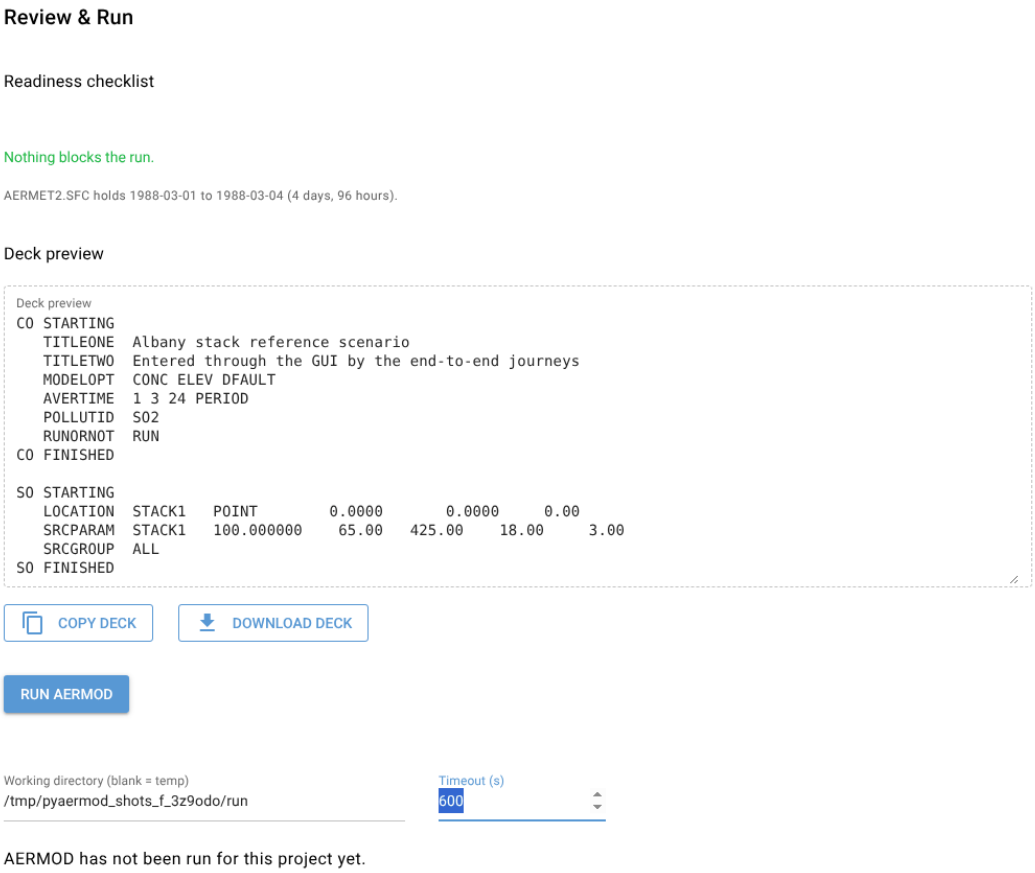
1. Click **Review & Run** in the step list.

2. Look at the **Deck preview** section. You'll see the complete AERMOD input file — a text file organized into five pathways:

- **CO** (Control): Project title, pollutant, options
- **SO** (Source): Your stack parameters
- **RE** (Receptor): Your grid definition
- **ME** (Meteorology): File paths
- **OU** (Output): What results to produce

3. Click **Download deck** to save it. The download is named `pyaermod_gui.inp`; rename it `aermod.inp`.

| **What you just did:** You generated a complete, valid AERMOD input file without editing a single line of text. Open the `.inp` file in a text editor and compare it to what the GUI showed you — they're identical.



Review & Run: the readiness checklist and the deck preview (EPA's Albany example; your values are the ones in the steps above)

Checkpoint

At this point you should understand:

- ✓ The five AERMOD pathways (CO, SO, RE, ME, OU)
- ✓ What stack height, temperature, velocity, and emission rate represent
- ✓ How a Cartesian receptor grid covers an area around the source
- ✓ That AERMOD needs hourly meteorological data in `.sfc` / `.pfl` format

---

5. Tutorial 2 — Comparing Stack Heights

Goal: Create two versions of the same facility — one with a 20 m stack and one with a 60 m stack — to understand how stack height affects ground-level concentrations.

Time: 15--20 minutes

What you'll learn:

- How stack height controls plume rise and ground-level impact
- How to save and reopen project files to create scenario variants
- How to read an AERMOD input file and spot key differences

Conceptual Background

Imagine two identical factories, both emitting 2 g/s of SO2. The only difference is the smokestack:

- **Factory A:** 20 m stack (about 6 stories tall)
- **Factory B:** 60 m stack (about 20 stories tall)

Which factory will cause higher ground-level concentrations? Intuitively, Factory A — its emissions are released closer to the ground, so they have less distance to dilute before reaching people.

But *how much* of a difference does it make? That's exactly what AERMOD calculates.

Step 1: Build the Base Scenario

Follow the same steps as Tutorial 1, but with these settings:

Project:

- Title (line 1): `Tutorial 2 - Stack Height Comparison (20m)`
- Pollutant: `SO2`
- Averaging periods: `1`, `24`, and `ANNUAL`
- Terrain: `Flat (FLAT)`, with **Regulatory default options (DFAULT)** unticked

Source (`PointSource`):

- Source ID: `STACK1`
- Stack height (m): `20`
- Stack temp (K): `420`
- Exit velocity (m/s): `15`
- Stack diameter (m): `2`
- Emission rate (g/s): `2.0`

Receptors (`CartesianGrid`):

- 4 km x 4 km area centered on the source, 100 m spacing (X num and Y num `41`, X delta and Y delta `100`)

Meteorology:

- Surface file: `met_data.sfc`
- Profile file: `met_data.pfl`

Step 2: Save the Project

1. Go back to the **Project** step.
2. Click **Save as...**, enter the filename `tutorial2_20m.json` and click **Save**. Your browser downloads the project as a JSON file.

Step 3: Download the 20 m Input File

1. Go to **Review & Run**.
2. Click **Download deck**. Rename the downloaded file `stack_20m.inp`.

Step 4: Create the 60 m Variant

1. Go to **Sources**.
2. Click the **Edit** button on the `STACK1` row of the table.
3. Change **Stack height (m)** to `60` and click **Save**. Everything else stays the same.
4. On the **Project** step, update the title to `Tutorial 2 - Stack Height Comparison (60m)`.
5. Save this project as `tutorial2_60m.json` (**Save as...**).
6. Download the input file and rename it `stack_60m.inp`.

Step 5: Compare the Input Files

Open both `.inp` files in a text editor and find the `SRCPARAM` line of the `SO` pathway:

20 m stack:

```
   SRCPARAM  STACK1     2.000000    20.00   420.00    15.00     2.00
```

60 m stack:

```
   SRCPARAM  STACK1     2.000000    60.00   420.00    15.00     2.00
```

The only difference is the second number (stack height); the title line differs too. Everything else is identical, making this a controlled experiment.

What to Expect When You Run These (Tutorial 3)

When you eventually run both files through AERMOD with real meteorological data:

- The **20 m stack** will produce **higher** maximum ground-level concentrations, especially close to the source (within 500 m).
- The **60 m stack** will produce **lower** maximum concentrations because the plume is released higher and has more room to dilute.
- The **location** of the maximum concentration will also shift — taller stacks push the peak concentration **farther downwind** because the plume takes longer to mix down to ground level.

This is a fundamental principle in air quality engineering: **effective stack height is the single most important factor for ground-level concentrations** from a point source.

Checkpoint

At this point you should understand:

✓ Taller stacks reduce ground-level concentrations

✓ Taller stacks push the peak concentration farther downwind

✓ Saving the project under a new name (**Save as...**) lets you create scenario variants; **Open...** brings one back

✓ The `SRCPARAM` line of the `SO` pathway contains all point source parameters

---

6. Tutorial 3 — Running AERMOD and Reading Results

Goal: Run AERMOD with real meteorological data and interpret the results using the GUI's Results step.

Time: 30--45 minutes

Prerequisites:

- The `aermod` executable must be installed and on your PATH. If you haven't done this yet, go back to **Section 3.2** and follow the instructions for your operating system.
- Meteorological data files (`.sfc` and `.pfl`). Your instructor may provide these, or you can process your own with AERMET (Tutorial 5).

  | **If you don't have met data yet:** Your instructor should provide a pair of `.sfc` and `.pfl` files for your region. These are typically generated from NOAA weather station data using the AERMET preprocessor. Ask your instructor which files to use and where to find them. **If AERMOD is not installed:** Run `which aermod` (Mac/Linux) or `where aermod` (Windows) in a terminal. If nothing is printed, go back to Section 3.2 to install it.

Step 1: Verify AERMOD Is Installed

Open a terminal and type:

```
aermod
```

If AERMOD is installed, you'll see an error about missing input files (that's OK — it just confirms the executable is found). If you get "command not found," you need to install AERMOD or add it to your PATH.

Step 2: Set Up the Model

Open one of your Tutorial 2 project files (**Project** step, **Open...**), or start a new project (**New**):

Project:

- Title (line 1): `Tutorial 3 - Full AERMOD Run`
- Pollutant: `PM25`
- Averaging periods: `24` and `ANNUAL`
- Terrain: `Flat (FLAT)`, with **Regulatory default options (DFAULT)** unticked

Source:

- A single point source with realistic parameters (see Tutorial 1)

Receptors:

- Cartesian grid: 4 km x 4 km, 100 m spacing (41 x 41 = 1,681 receptors)

Meteorology:

- Click **Meteorology** in the step list.
- Under **Met files**, enter the full paths of your `.sfc` and `.pfl` files in **Surface file** and **Profile file**. Give the whole path (for example `/Users/you/met/site.sfc` or `C:\met\site.sfc`): AERMOD does not expand ~, and the GUI tells you under the field when a path is not a file on your computer. There is no upload button; the files stay where they are.
- Below the files, the step says what the surface file holds: the period it covers, the station IDs in its header and the first year of its data.

- Under **Stations**, set **Surface station ID**, **Upper air station ID** and **Data start year** to those values. AERMOD checks them against the files.

> **ANNUAL needs a full year.** If your met files cover less than a year, the GUI warns that AERMOD would stop the run. Replace `ANNUAL` with `PERIOD` on the Project step, which averages over whatever the files hold.

Step 3: Review and Run

1. Click **Review & Run** in the step list.
2. Read the **Readiness checklist**. The GUI checks the project every time you change it and lists here everything that would stop the run, each item with a link to the step that fixes it (common issues: missing met file paths, no sources defined, no receptors defined). When the list is empty the step says "Nothing blocks the run." Warnings that do not stop the run are listed under **Before you run**.
3. Set the **Working directory** — this is where AERMOD will write its output. If you leave it blank, each run gets a new temporary folder.
4. If the checklist says "No 'aermod' binary on PATH", AERMOD is not installed where the GUI can find it: go back to Step 1.
5. Click **Run AERMOD**.

A progress bar follows AERMOD through the days of your met data, next to the elapsed time and a **Cancel** button. Depending on your grid size and the number of hours of met data, this may take anywhere from 10 seconds to a few minutes.

When it finishes, the status line reads either:

- "Succeeded in *N* s. See the Results step.", or
- "Failed: *reason*", above a table of AERMOD's messages and a **Run log**. Read the messages — the most common issue is AERMOD not finding the meteorological files.



A finished run: "Succeeded", AERMOD's message counts and the table of its messages (EPA's Albany example; your values are the ones in the steps above)

Step 4: Explore the Results

Click **Results** in the step list. The step shows the run in several sections, from top to bottom:

Maximum for Each Averaging Period

- One card per averaging period with the **maximum concentration**: the highest value at any receptor, its units, the receptor where it occurred and the date it ended. This is the starting point for a comparison with the NAAQS.

Concentration Map

- Filled contours of concentration (isopleths — areas between lines of equal concentration), with your source and the highest value marked.
- **Map shows** selects the averaging period (e.g., `ANNUAL`).
- Notice the **plume shape**: concentrations are highest downwind of the source and decrease with distance. The elongated pattern reflects the prevailing wind direction in your met data.
- The pattern around the source shows how concentrations decrease with distance — the **concentration gradient**.
- The map is a picture: it does not zoom or pan. The numbers behind it are in the tables and in the plot files under **Downloads**.

Comparison with the NAAQS and Summary Tables

This is where you do compliance analysis:

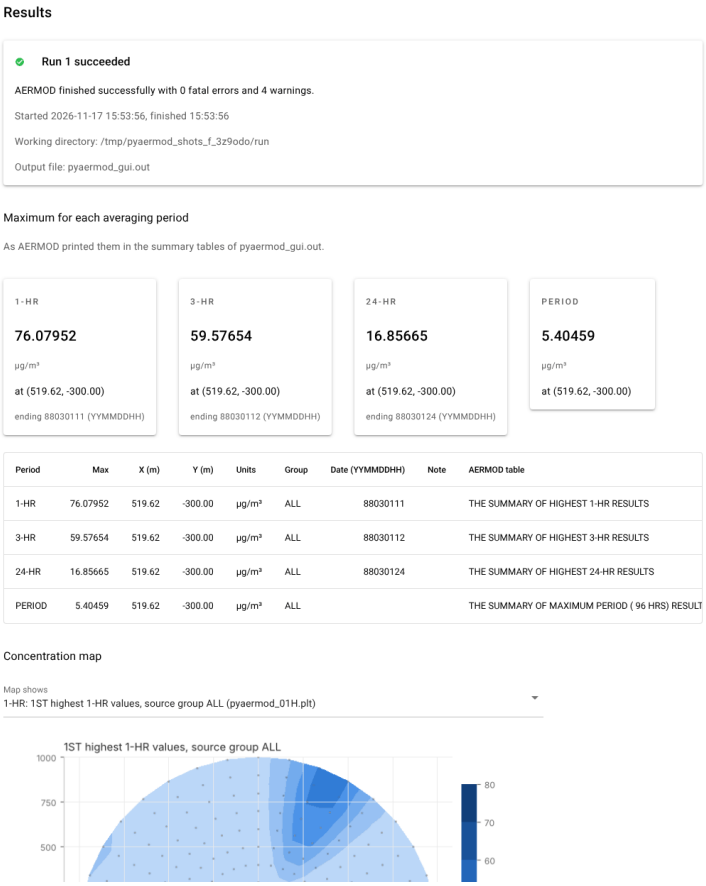
- **Comparison with the NAAQS**: one row for each standard of your pollutant, with the standard's level, this run's value, where it occurred, what was compared and the result.
- **Summary tables**: the ten highest values of each averaging period, with their dates and receptor coordinates. Where are the worst-case locations?
- **Exceedance analysis**: the GUI does not count the receptors above a standard. To count them, download the plot file of that averaging period from **Downloads** (a text file with one row per receptor: X, Y and the highest concentration there) and open it in a spreadsheet.

Common NAAQS values for reference:

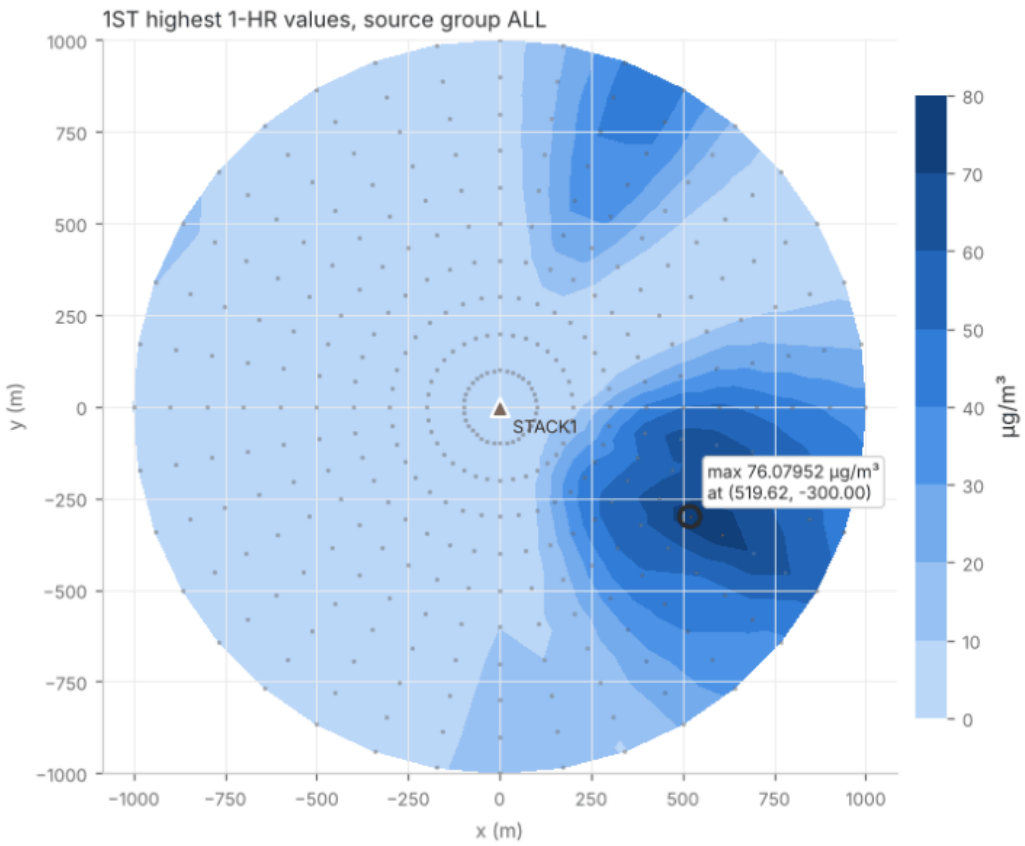
Pollutant	Period	Standard (ug/m3)
PM2.5	Annual	9.0
PM2.5	24-hour	35
PM10	24-hour	150
SO2	1-hour	196
NO2	1-hour	188
CO	8-hour	10,000

Downloads and POSTFILEs (Advanced)

Downloads lists the deck, AERMOD's `.out` file, the plot files and the POSTFILEs of the run. A POSTFILE holds the concentration at every receptor for every averaging interval; the GUI writes one per averaging period unless you untick **POSTFILE for every averaging period** on the **Output** step. The GUI does not animate them. From Python, `pyaermod.postfile.read_postfile` reads one into a table. This is optional for beginners.



The Results step: the run's status card and AERMOD's maximum for each averaging period (EPA's Albany example; your values are the ones in the steps above)



The concentration map: filled contours of the highest 1-hour values, the receptors as dots and the maximum ringed (EPA's Albany example; your values are the ones in the steps above)

Step 5: Interpret What You See

Ask yourself these questions:

1. **Where is the maximum concentration?** It should be downwind of the source, at a distance that depends on stack height and meteorology. For a 50 m stack, the peak is typically 500--2000 m downwind.
2. **What direction does the plume extend?** This reflects the prevailing wind direction in your met data. If the wind usually blows from west to east, the highest concentrations will be east of the source.
3. **Does the maximum exceed the NAAQS?** Read the result in the **Comparison with the NAAQS** table. If the standard is exceeded, the facility would need additional controls (taller stack, emission reduction, scrubbers, etc.).
4. **How quickly do concentrations drop with distance?** Look at the concentration map — concentrations typically drop to 10% of the maximum within 1--2 km for elevated point sources.

Checkpoint

At this point you should understand:

- ✓ How to run AERMOD from the GUI
- ✓ How to read the concentration map and identify the plume direction

- ✓ How to find the maximum concentration and where it occurs
- ✓ How to check whether a facility complies with air quality standards
- ✓ That wind direction and stack height are the dominant factors in where the plume goes and how concentrated it is

---

7. Tutorial 4 — Area Sources: Modeling a Facility with Fugitive Emissions

Goal: Model a construction site with three types of area sources — a rectangular stockpile, a circular staging area, and an irregular polygon site boundary — alongside a point source (generator exhaust).

Time: 25--30 minutes

What you'll learn:

- The difference between point source and area source emission rates
- How to add rectangular, circular, and polygonal area sources in the GUI
- How to combine point and area sources in one model
- How source groups let you analyze contributions separately

Conceptual Background

Not all emissions come from smokestacks. Many facilities also have **fugitive emissions** — pollutants that escape from ground-level or low-height sources rather than through a defined stack. Examples include:

Source	Why It Emits
Dirt stockpile	Wind blows dust off exposed surfaces
Parking lot	Vehicle exhaust, tire/brake wear, re-entrained road dust
Construction site	Earth-moving equipment, exposed soil
Tank farm	Vapors escape through tank vents and seals
Landfill	Gases seep through the surface over a large area

These sources are modeled as **area sources** in AERMOD. The key difference from point sources is how emission rates are specified:

- **Point source**: total mass emitted per second (**g/s**)
- **Area source**: mass emitted per second *per unit area* (**g/s/m2**)

Because area sources spread emissions over a large surface, the per-square-meter rate is typically a very small number (e.g., 0.0001 g/s/m2).

> **Example calculation:** A 100 m x 50 m stockpile emits 0.5 g/s total. Area = 100 x 50 = 5,000 m2. Emission rate = 0.5 / 5,000 = **0.0001 g/s/m2**.

Step 1: Set Up the Project

1. Launch the GUI (`pyaermod-app`).
2. On the **Project** step: - **Title (line 1):** `Tutorial 4 - Area Source Facility` - **Title (line 2):** `Construction site with mixed source types` - **Pollutant:** `PM10` - **Averaging periods:** Select `24` and `ANNUAL` - **Terrain:** `Flat (FLAT)`, with **Regulatory default options (DFAULT)** unticked

Step 2: Add a Point Source (Generator)

1. Click **Sources** in the step list.

2. Set **Type** to `PointSource` and click **Add**.
3. Fill in the form:

Parameter	Value	What It Represents
Source ID	`GENSET`	Diesel generator exhaust
X coord (m)	`500050`	UTM meters
Y coord (m)	`3870050`	UTM meters
Base elevation (m), under **Advanced**	`0`	
Stack height (m)	`5`	Short exhaust stack
Stack temp (K)	`700`	Diesel exhaust is very hot (K)
Exit velocity (m/s)	`10`	
Stack diameter (m)	`0.3`	Small pipe
Emission rate (g/s)	`0.3`	0.3 g/s of PM10

4. Click **Save**.

| **Why start with the point source?** It gives you a reference marker in the plan view. You'll place the area sources around it.

Step 3: Add a Rectangular Area Source (Stockpile)

1. Set **Type** to `AreaSource` and click **Add**.
2. Fill in the form:

Parameter	Value	What It Represents
Source ID	`PILE1`	Dirt/gravel stockpile
X coord (m)	`500000`	Southwest corner X
Y coord (m)	`3870000`	Southwest corner Y
Base elevation (m), under **Advanced**	`0`	
Release height (m)	`2.0`	Dust lifts off ~2 m above the pile surface
Initial lateral dimension (m)	`100.0`	Xinit, the X side: 100 m in the X direction
Initial vertical dimension (m)	`50.0`	Yinit, the Y side: 50 m in the Y direction
Angle (deg)	`0`	Aligned with the grid (no rotation)
Emission rate (g/(s·m²))	`0.000100`	0.0001 g/s/m2 (see calculation above)

3. Click **Save**.

| **Understanding the dimensions:** AERMOD places a rectangular area source by its *southwest corner* and gives it *full side lengths*: Xinit is the length of the X side and Yinit the length of the Y side (`SRCPARAM`, soset.f APARM). A rotation angle turns the rectangle clockwise about that corner, not about its centre. Enter the actual dimensions, not half of them.

Step 4: Add a Circular Area Source (Staging Area)

1. Set **Type** to `AreaCircSource` and click **Add**.
2. Fill in the form:

Parameter	Value	What It Represents
Source ID	`STAGING`	Equipment staging / laydown area
X coord (m)	`500200`	Center of the circle
Y coord (m)	`3870100`	
Base elevation (m), under **Advanced**	`0`	
Release height (m)	`1.0`	Low-level dust from vehicle traffic
Radius (m)	`60`	60 m radius circle
Num vertices, under **Advanced**	`20`	How many sides to approximate the circle
Emission rate (g/(s·m²))	`0.000050`	Lower rate — less disturbed surface

3. Click **Save**.

| **Num Vertices:** AERMOD approximates circles as polygons. 20 vertices gives a smooth-enough circle for most purposes. More vertices is more accurate but takes slightly longer to compute.

Step 5: Add a Polygonal Area Source (Site Boundary)

1. Set **Type** to `AreaPolySource` and click **Add**.
2. Fill in the form:

Parameter	Value
Source ID	`SITEBND`
Base elevation (m), under **Advanced**	`0`
Release height (m)	`0.5`
Emission rate (g/(s·m²))	`0.000020`

3. In **Vertices (m)**, replace the placeholder with the five corners of the irregular site boundary, one `x, y` pair per line:

```
499900, 3869900
500350, 3869900
500400, 3870100
500300, 3870250
499900, 3870200
```

4. Click **Save**.

| **Vertex order matters:** The vertices should trace the outline of the area in order (clockwise or counterclockwise). AERMOD closes the polygon automatically — you don't need to repeat the first vertex.

Step 6: Set Up Receptors

1. Click **Receptors** in the step list.
2. Set **Type** to `CartesianGrid`, click **Add**, and enter:

Parameter	Value
X init (m)	`499500`
X num	`27`
X delta (m)	`50`
Y init (m)	`3869500`
Y num	`25`
Y delta (m)	`50`

3. Click **Save**. The grid runs from 499500 to 500800 east and from 3869500 to 3870700 north: 27 x 25 = 675 receptors covering the site and a buffer zone around it.

Step 7: Meteorology and Preview

1. Click **Meteorology** and enter the paths to your `.sfc` and `.pfl` files in **Surface file** and **Profile file** (or placeholder names for now).
2. Click **Review & Run** to preview the generated input file.

In the **Deck preview**, look for the `SO` pathway. You should see four sources — one POINT and three area types:

```
  LOCATION  GENSET   POINT   500050.0000 3870050.0000     0.00
  LOCATION  PILE1    AREA     500000.0000 3870000.0000     0.00
  LOCATION  STAGING  AREACIRC  500200.0000 3870100.0000     0.00
  LOCATION  SITEBND  AREAPOLY  499900.0000 3869900.0000     0.00
```

Notice how each source type generates different `SRCPARAM` lines:

- **POINT**: emission rate, stack height, temperature, velocity, diameter
- **AREA**: emission rate, release height, X side length, Y side length (then an optional angle and initial vertical spread Szinit)
- **AREACIRC**: emission rate, release height, radius, number of vertices
- **AREAPOLY**: emission rate, release height, number of vertices (the corners follow on an `AREAVERT` line)

Step 8: Interpret Area Source Results (After Running)

When you eventually run this model, look for these patterns in the Results step's concentration map:

- **The generator (point source)** produces a concentrated plume downwind of the stack — a narrow elongated shape.
- **The stockpile (rectangular area)** produces a broader, lower-concentration pattern centered on the pile. Concentrations are highest at the downwind edge of the pile.
- **The site boundary (polygon)** produces the most diffuse pattern — low concentrations spread over a wide area.
- **Combined impact** is highest where plumes from multiple sources overlap.

Checkpoint

At this point you should understand:

- ✓ Area source emission rates are per unit area (g/s/m2), not total (g/s)
- ✓ Rectangular area sources are placed by their southwest corner and defined by full side lengths, not half-widths
- ✓ Circular sources are approximated by polygons (Num Vertices)
- ✓ Polygon sources are defined by a list of corner coordinates
- ✓ Different source types produce different spatial patterns in results

---

8. Tutorial 5 — Processing Meteorological Data with AERMET

Goal: Use PyAERMOD to generate the two AERMET input files (Stage 1 and METPREP) needed to process raw weather station data into AERMOD-ready meteorological files.

> **This tutorial uses a short Python script, not the GUI.** The GUI's Meteorology step only points AERMOD at finished `.sfc` and `.pfl` files; it has no AERMET page. You don't need programming experience: you copy the script below into a file, change the values, and run it.

Time: 30--40 minutes

What you'll learn:

- What AERMET does and why AERMOD needs it
- How AERMET's two stages, Stage 1 and METPREP, work
- How to configure each stage with PyAERMOD
- What the monthly surface parameters mean and how to choose values

Prerequisites:

- PyAERMOD generates AERMET *input files*, but to actually process them you need the `aermet` executable installed on your computer.
- **macOS/Linux:** If you followed Section 3.2, Option A (compile from source), you already have it.
- **Windows:** The EPA's AERMOD Modeling System download from the SCRAM website includes `aermet.exe` alongside `aermod.exe`.
- Verify it's installed: run `which aermet` (Mac/Linux) or `where aermet` (Windows) in a terminal.

Conceptual Background

AERMOD cannot use raw weather data directly. Raw data from weather stations comes in formats designed for archival and general forecasting — not atmospheric dispersion modeling. AERMOD needs specially processed files that contain:

- **Hourly surface data** (`.sfc` file): wind speed, wind direction, ambient temperature, atmospheric stability class, mixing height, friction velocity, Monin-Obukhov length, and other boundary-layer parameters.
- **Hourly profile data** (`.pfl` file): vertical profiles of wind speed, wind direction, and temperature at multiple heights above ground.

AERMET is the preprocessor that transforms raw weather station observations into these two files. Current AERMET (versions 24142 and 26135) runs in two stages, each from its own input file:

```
Raw weather data (NOAA archives)
          |
   Stage 1: Extract & QA/QC
          |
   Quality-assured hourly data (QAOUT files)
          |
   METPREP: merge surface + upper air, compute boundary-layer parameters
          |
   aermod.sfc  +  aermod.pfl   (ready for AERMOD)
```

Older AERMET versions had a separate merge stage between the two; current AERMET merges the data inside METPREP. pyaermod calls the METPREP deck "Stage 3" (`AERMETStage3`, `aermet_stage3.inp`) so

older projects keep their file names.

What Data Does AERMET Need?

AERMET requires two types of weather observations:

1. **Surface observations** — hourly weather data from a ground-level station (typically an airport). Includes wind speed, wind direction, temperature, cloud cover, and ceiling height. In the U.S., this data is available from NOAA's Integrated Surface Hourly Data (ISHD) archive.
2. **Upper air soundings** — twice-daily vertical profiles of temperature, humidity, and wind measured by weather balloons (radiosondes). Available from NOAA's Radiosonde Database. The closest upper air station may be 100+ km away — that's normal.

> **Your instructor should provide** the station IDs, data files, and date ranges for your region. If you need to download your own data, NOAA's archives are at https://www.ncei.noaa.gov/.

Step 1: Start the Script

1. Make a folder for this tutorial and put your raw data files in it.
2. In that folder, create a text file named `make_aermet_inputs.py` and start it with these lines:

```
from pathlib import Path

from pyaermod.aermet import AERMETStage1, AERMETStage3, AERMETStation, UpperAirStation
```

You will add to this file in Steps 2 and 3 and run it in Step 4. The script describes two stages, Stage 1 and Stage 3. Current AERMET (11 and later) has no separate Stage 2: METPREP (Stage 3) merges the surface and upper-air data itself.

Step 2: Configure Stage 1 — Extract & QA/QC

Purpose: Stage 1 reads raw weather station data, performs quality assurance checks (flagging missing or suspicious values), and extracts the variables AERMET needs.

Surface Station

Describe the surface station. Your instructor should provide these values. Example for Atlanta, GA:

Parameter	Example Value	What It Means
Station ID	`13874`	The station's WBAN number (Atlanta Hartsfield, ICAO KATL). AERMET reads it as a number, so an ICAO code such as KATL makes the METPREP stage stop
Station Name	`Atlanta Hartsfield`	Descriptive name (for your reference)
Latitude	`33.6300`	Station latitude (decimal degrees, negative = south)
Longitude	`-84.4400`	Station longitude (decimal degrees, negative = west)
Time Zone (UTC offset)	`-5`	Eastern Standard Time = UTC-5
Elevation (m)	`315.0`	Station elevation above sea level

Anemometer Height (m)	`10.0`	Height of the wind sensor (usually 10 m)
Data Format	`ISHD`	NOAA Integrated Surface Hourly Data (most common)

Add the station to your script:

```
surface = AERMETStation(
    station_id="13874",
    station_name="Atlanta Hartsfield",
    latitude=33.6300,
    longitude=-84.4400,
    time_zone=-5,
    elevation=315.0,
    anemometer_height=10.0,
)
```

The data format goes into Stage 1 itself, below.

> | **Data Format options:** - **ISHD** — Integrated Surface Hourly Data (NOAA, post-2006). This is the most common format for recent U.S. data. - **HUSWO** — Hourly US Weather Observations (legacy NCDC format). - **SCRAM** — EPA SCRAM format (older pre-processed data). - **SAMSON** — Solar and Meteorological Surface Observational Network.

Upper Air Station

Describe the upper air (radiosonde) station. Example:

Parameter	Example Value	What It Means
Station ID	`72215`	WMO station number
Station Name	`Peachtree City`	Closest radiosonde launch site
Latitude	`33.3600`	
Longitude	`-84.5700`	
Elevation (m)	`245.0`	Station elevation above sea level; check it against the station list of your data source

```
upper_air = UpperAirStation(
    station_id="72215",
    station_name="Peachtree City",
    latitude=33.3600,
    longitude=-84.5700,
    elevation=245.0,
)
```

> | **Finding your upper air station:** The closest radiosonde station may be far from your surface station. In the U.S., there are only about 90 upper air stations (compared to thousands of surface stations). Use the one that is most representative of your area's upper-atmosphere conditions.

Data Files and Date Range

Parameter	Example Value	What It Means
Surface Data File	`72219013874.dat`	Path to the raw ISHD surface data file

Upper Air Data File	`72215.dat`	Path to the raw upper air data file
Start Date	`2020/01/01`	First day of the period to process
End Date	`2020/12/31`	Last day (typically 1--5 years of data)

Add Stage 1 to your script:

```
stage1 = AERMETStage1(
    surface_station=surface,
    surface_data_file="72219013874.dat",
    surface_format="ISHD",
    upper_air_station=upper_air,
    upper_air_data_file="72215.dat",
    start_date="2020/01/01",
    end_date="2020/12/31",
)
```

What Stage 1's Input File Contains

The file this produces (you will write it in Step 4) contains keywords like:

- `JOB` — identifies the processing job and names its report and message files
- `SURFACE` with a `DATA` line — points to the surface data file and specifies the format
- `UPPERAIR` with a `DATA` line — points to the upper air data file
- `XDATES` — specifies the date range to extract

Step 3: Configure Stage 3 — Boundary Layer Parameters

Purpose: This is the most important stage. Stage 3 (AERMET's METPREP) reads Stage 1's quality-assured files, merges them, and computes the **planetary boundary layer parameters** that AERMOD needs: friction velocity, Monin-Obukhov length, convective velocity scale, mixing height, and more. These parameters depend on both the meteorological data and the **surface characteristics** around the station.

File Paths

Parameter	Example Value
Start Date	`2020/01/01`
End Date	`2020/12/31`
Surface Output (.sfc)	`aermod.sfc`
Profile Output (.pfl)	`aermod.pfl`

Monthly Surface Parameters

This is the part that requires the most judgment. Stage 3 takes three parameters for each month:

Parameter	What It Controls	Typical Range
Albedo	Surface reflectivity (fraction of sunlight reflected). Snow-covered ground has high albedo; dark pavement has low albedo.	0.10 -- 0.60

Bowen Ratio	Ratio of sensible heat to latent heat. Dry surfaces (deserts, cities) have high Bowen ratios; moist surfaces (wetlands, irrigated fields) have low values.	0.1 -- 10.0
Roughness (m)	Aerodynamic roughness length — how "bumpy" the surface is to wind flow. Open water is very smooth (~0.001 m); a city center is very rough (~1.0 m).	0.001 -- 2.0

This tutorial uses **suburban values** — these are reasonable starting values for an area with moderate development (residential neighborhoods, scattered trees):

Month	Albedo	Bowen Ratio	Roughness (m)
Jan	0.35	1.5	0.30
Feb	0.35	1.5	0.30
Mar	0.25	1.0	0.30
Apr	0.18	0.8	0.30
May	0.15	0.6	0.50
Jun	0.15	0.5	0.50
Jul	0.15	0.5	0.50
Aug	0.15	0.5	0.50
Sep	0.18	0.6	0.50
Oct	0.25	0.8	0.30
Nov	0.35	1.0	0.30
Dec	0.35	1.5	0.30

> | **How to customize these values:** - **If your site is in a rural/agricultural area:** Lower roughness (0.05--0.15 m), albedo varies by crop/season, Bowen ratio varies by irrigation. - **If your site is in a dense urban area:** Higher roughness (0.5--1.5 m), higher Bowen ratio (dry impervious surfaces), lower albedo (dark pavement). - **If there's winter snow cover:** Increase albedo to 0.50--0.70 during snowy months. - **When in doubt:** Use the suburban values above. They're conservative and widely used in regulatory work.

In the script each parameter is a list of twelve numbers, January first. To change a month, edit its number in the list.

Site Location

`station=surface` tells METPREP to use the surface station you described for Stage 1, with its latitude and longitude.

Add Stage 3 to your script:

```
stage3 = AERMETStage3(
    station=surface,
    #        Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov   Dec
```

```
    albedo=[0.35, 0.35, 0.25, 0.18, 0.15, 0.15, 0.15, 0.15, 0.18, 0.25, 0.35, 0.35],
    bowen=[1.5, 1.5, 1.0, 0.8, 0.6, 0.5, 0.5, 0.5, 0.6, 0.8, 1.0, 1.5],
    roughness=[0.30, 0.30, 0.30, 0.30, 0.50, 0.50, 0.50, 0.50, 0.50, 0.30, 0.30, 0.30],
    methods=[("REFLEVEL", "SUBNWS")],
    start_date="2020/01/01",
    end_date="2020/12/31",
    surface_file="aermod.sfc",
    profile_file="aermod.pfl",
).with_inputs_from(stage1)
```

`methods=[("REFLEVEL", "SUBNWS")]` tells AERMET that the wind data come from a National Weather Service station, which current AERMET requires to be stated. `.with_inputs_from(stage1)` points METPREP at the files Stage 1 writes.

Step 4: Write the Two Input Files

End your script with the lines that write the two files:

```
Path("aermet_stage1.inp").write_text(stage1.to_aermet_input())
Path("aermet_stage3.inp").write_text(stage3.to_aermet_input())
print("Wrote aermet_stage1.inp and aermet_stage3.inp")
```

Run it in a terminal, from the folder that holds the script:

```
python make_aermet_inputs.py
```

You now have two AERMET input files to run:

File	Stage	What It Does
`aermet_stage1.inp`	Extract & QA/QC	Reads raw data, checks quality
`aermet_stage3.inp`	METPREP	Merges surface + upper air, computes AERMOD-ready parameters

Open them in a text editor and find the keywords listed in Step 2 and your monthly surface parameters (`SITE_CHAR` lines) in the Stage 3 file.

Step 5: Run AERMET

AERMET itself must be run separately (it's a different executable from AERMOD). First, verify that `aermet` is installed:

```
which aermet        # Mac/Linux — should print a path like ~/bin/aermet
where aermet        # Windows — should print a path like C:\AERMOD\aermet.exe
```

If nothing is printed, go back to Section 3.2 to install it (Option A for Mac/Linux, Option B for Windows).

Then, in a terminal, navigate to the directory containing your input files and run each stage:

```
# Stage 1
aermet aermet_stage1.inp

# METPREP (merge + boundary layer)
aermet aermet_stage3.inp
```

AERMET reads the input file named after the command (not from < redirection). METPREP reads the QAOUT files Stage 1 writes, so the two must be run **in order**. When METPREP completes, you'll have your `aermod.sfc` and `aermod.pfl` files — the meteorological inputs AERMOD needs.

AERMET exits without an error code even when it fails. Check that the screen output ends with `AERMET FINISHED SUCCESSFULLY` and read the messages file each deck names (`MESSAGES`) for lines with an `E` code such as `E05`. From Python, `pyaermod.aermet_runner.run_aermet_pipeline` runs both decks and reports each run's `success` and first error for you.

> | **Troubleshooting common errors:** - *"File not found"*: Check that the data file paths in Stage 1 are correct and the files exist in the working directory. - *"Missing data exceeds threshold"*: Your raw data has too many gaps. AERMET flags this in the Stage 1 output. You may need to use a different station or a year with more complete records. - *"Upper air sounding missing"*: The upper air station may not have data for all dates. A few missing soundings are normal; AERMET interpolates.

Step 6: Use the Processed Met Data in AERMOD

Once you have `aermod.sfc` and `aermod.pfl`:

1. Launch the GUI (`pyaermod-app`) and go to the **Meteorology** step.
2. Under **Met files**, enter the full paths to your new `aermod.sfc` and `aermod.pfl` files in **Surface file** and **Profile file**. The step then says what the surface file holds.
3. Under **Stations**, enter **Surface station ID** `13874`, **Upper air station ID** `72215` and **Data start year** `2020`, the stations and year you processed.
4. Continue with your AERMOD run (Tutorial 3).

Understanding What AERMET Produced

The `.sfc` file contains one line per hour with columns including:

- Year, month, day, hour
- Sensible heat flux (W/m2) — energy driving vertical mixing
- Friction velocity (m/s) — a measure of wind turbulence near the ground
- Monin-Obukhov length (m) — characterizes atmospheric stability (negative = unstable, positive = stable)
- Mixing height (m) — the depth of the turbulent boundary layer
- Wind speed, direction, and temperature

The `.pfl` file contains vertical wind and temperature profiles for each hour — the information AERMOD uses to model how the plume disperses at different heights.

You don't need to read these files directly — AERMOD reads them automatically — but understanding what's in them helps you interpret your results. For example, if your maximum concentrations all occur during nighttime hours, it's likely because stable conditions (high Monin-Obukhov length, low mixing height) trapped the plume near the ground.

Checkpoint

At this point you should understand:

✓ Why AERMOD needs preprocessed meteorological data (not raw observations)

✓ The two AERMET stages: Stage 1 (extract and QA) and METPREP (merge and boundary layer)

✓ That surface characteristics (albedo, Bowen ratio, roughness) vary by

land use and season

- ✓ How to generate the two AERMET input files with PyAERMOD
- ✓ That AERMET must be run in order (Stage 1, then METPREP, which pyaermod calls Stage 3) to produce `.sfc` and `.pfl` files

---

9. What's Next?

Now that you've completed all five tutorials, here are some directions to explore:

Advanced Tutorials: Houston Refinery Modeling (Tutorials 6--8)

The **Houston Refinery Modeling Assignments** extend this guide with three advanced tutorials that take you through a realistic industrial modeling project:

- **Tutorial 6** — Process meteorological data with AERMET for a specific Gulf Coast location (Houston Ship Channel), with site-appropriate surface parameters for a coastal/industrial area.
- **Tutorial 7** — Use AERMAP to process digital elevation data and assign terrain elevations to all receptors and sources, even in a relatively flat area like Houston.
- **Tutorial 8** — Build a complete AERMOD model for a simplified 150,000 barrel-per-day petroleum refinery with 10 emission sources (point, area, and volume), multi-scale receptors, source groups, regulatory compliance analysis, and sensitivity studies. Includes a technical report assignment.

These tutorials build directly on the skills from Tutorials 1--5 and include assignment deliverables and a grading rubric suitable for a university course.

Multiple Sources and Source Groups

Add several stacks and area sources to the same project and see how their plumes overlap. Use **source groups** (on the Sources step, each source's **Source groups** field under **Advanced**, one group name per line) to separate contributions from different parts of a facility — for example, "STACKS" vs. "FUGITIVE" — and compare their relative impact.

Terrain Effects

Change **Terrain** on the Project step from Flat to Elevated. Elevated terrain can significantly increase concentrations on hilltops that are close to the effective plume height. This requires terrain elevation data processed by AERMAP.

Background Concentrations

Real-world air quality is never zero — there's always some background pollution from regional sources, traffic, and natural sources. AERMOD can add a background concentration to its predictions. The GUI cannot enter one yet; set it from Python (see Notebook 07) and open the saved project in the GUI.

The Python API

Once you're comfortable with the GUI concepts, try the Jupyter notebooks in `examples/notebooks/` to do the same work with Python code. The Python API gives you more flexibility for parameter sweeps, batch runs, and custom analysis.

Useful References

- AERMOD User's Guide (EPA) — The official 300+ page reference for all AERMOD options.
- AERMOD Model Formulation Document (EPA) — The mathematical basis behind the model.
- AERMET User's Guide (EPA) — Detailed documentation of all AERMET options and data formats.

- 40 CFR Part 51, Appendix W — EPA's Guideline on Air Quality Models (the regulatory framework for when and how to use AERMOD).

---

10. Glossary

Term	Definition
AERMOD	AMS/EPA Regulatory Model — the U.S. EPA's preferred near-field dispersion model
AERMET	AERMOD's meteorological preprocessor; converts raw weather data into `.sfc` and `.pfl` files
AERMAP	AERMOD's terrain preprocessor; extracts elevation data for receptors and sources
Averaging period	The time window over which concentrations are averaged (1-hour, 24-hour, annual, etc.)
Base elevation	The ground-level elevation at the base of a source or receptor (meters above sea level)
Buoyancy rise	The upward motion of a hot plume due to its lower density relative to ambient air
Cartesian grid	A rectangular array of evenly spaced receptor points
Concentration	The amount of pollutant in a given volume of air, typically in ug/m3
Discrete receptor	A single receptor placed at a specific location of interest
Effective stack height	Physical stack height + plume rise; the height at which the plume levels off
Emission rate	The mass of pollutant released per unit time (g/s for point sources, g/s/m2 for area sources)
Exceedance	When a predicted concentration is higher than the applicable air quality standard
Exit velocity	Speed of exhaust gas leaving the stack (m/s)
Gaussian plume	A mathematical model where pollutant concentration follows a bell curve in both horizontal and vertical cross-sections
Momentum rise	The upward motion of a plume due to the velocity of the exhaust gas
NAAQS	National Ambient Air Quality Standards — EPA's health-based concentration limits
Pathway	One of AERMOD's five input sections: CO (Control), SO (Source), RE (Receptor), ME (Meteorology), OU (Output)
Plume rise	The total height gain of exhaust gas above the physical stack top, from both buoyancy and momentum
Polar grid	A set of receptor points arranged in concentric rings at regular angular intervals around a center point
POSTFILE	An optional AERMOD output file containing concentration values at every receptor for every hour
Receptor	A location where AERMOD calculates ground-level pollutant concentration

Source group	A named subset of emission sources whose combined impact is reported separately
Stability	A measure of atmospheric turbulence: unstable (strong mixing), neutral, or stable (weak mixing)
Steady-state	An assumption that conditions (wind, stability) are constant during each simulation hour
Albedo	The fraction of incoming sunlight reflected by the ground surface (0 = absorbs all, 1 = reflects all). Fresh snow ~0.7, dark pavement ~0.1
Bowen ratio	Ratio of sensible heat flux to latent heat flux at the surface. High values (dry/urban), low values (moist/vegetated)
Friction velocity	A measure of wind-driven turbulence near the ground surface (m/s). Higher values mean more mechanical mixing
Fugitive emissions	Pollutants released from diffuse, ground-level sources (piles, lots, open areas) rather than through defined stacks
Initial side lengths (Xinit, Yinit)	AERMOD defines a rectangular area source by its southwest corner and the full length of its x and y sides (not half-widths); a rotation turns it clockwise about that corner
ISHD	Integrated Surface Hourly Data — NOAA's standard format for recent U.S. surface weather observations
Mixing height	The depth of the atmospheric boundary layer (meters). Pollutants released below this height mix vertically within it
Monin-Obukhov length	A parameter characterizing atmospheric stability. Negative = unstable (good mixing), positive = stable (poor mixing)
Radiosonde	An instrument carried aloft by a weather balloon to measure vertical profiles of temperature, humidity, and wind
Roughness length	Aerodynamic parameter describing how rough the ground surface is to wind flow. Open water ~0.001 m, city center ~1.0 m
Surface characteristics	Albedo, Bowen ratio, and roughness length — the three land-surface parameters AERMET needs for each month
UTM	Universal Transverse Mercator — a coordinate system that measures position in meters (easting, northing) within numbered zones