Test Virtual Sources

Steve Stage

February 1, 2024

Revised February 7, 2024

# Purpose

Test that routines which compute virtual sources are the inverses of routines that compute sigmas.

Test that sigmas are correctly computed for releases having non-zero source sigmas.

By definition, the virtual source s the position a point source would need to be upwind of the actual source in order to have sigma=sigma0 at x=0.

Note that the virtual sources will generally be different for sig\_x, sig\_y, and sig\_z.

# Files

|  |  |
| --- | --- |
| File | What it tests |
| test\_virtual\_sources\_01.csv | Computation of virtual sources |
| test\_virtual\_sources\_02.csv | Computation of virtual sources |
| test\_calc\_sigmas.csv | Computation of sigmas for cases having non-zero source sigmas |

# How Tests were Made

Files were created by Steve’s programs plume.py and test\_plume.py.

## test\_virtual\_sources\_01 and -02

For each stability class and wind speed combination:

For x in xx (a set of downwind distances):

Calculate sig\_x, sig\_y, and sig\_z. at x for a point source

Use these values as source sigmas.

Compute the virtual sources, xv\_x, xv\_y, and xv\_z.

Compute the differences between the distances used in computing the source sigmas and the virtual.

These differences are small (less than 1.0E-4).

test\_virtual\_sources\_01.csv uses the same distances for sig\_x, sig\_y, and sig\_z.

test\_virtual\_sources\_02.csv uses different distances for sig\_x, sig\_y, and sig\_z.

## test\_calc\_sigmas.csv

For specified combinations of stability class and wind speed:

For specified combinations of sig\_x0, sig\_y0, and sig\_z0:

Compute the virtual sources, xv\_x, xv\_y, and xv\_z.

For x in xx (a set of downwind distances):

Calculate sig\_x, sig\_y, and sig\_z. at x

**Sample from test\_virtual\_sources\_01.csv**



**Sample from test\_virtual\_sources\_02.csv**



**Sample from test\_calc\_sigmas.csv**



# How to Test FastPlume

## test\_virtual\_sources\_01 and -02

**Inputs:**

* Use istab and speed for stability and wind speed.
* Use sig\_x0, sig\_y0, and sig\_z0 for source sigmas.

**Expected results:**

* Calculate the virtual source locations, xv\_x, xv\_y, and xv\_z.
* These should match the values in the files to within a tolerance of 1.0E-4.

**Notes:**

Correct behavior is:

xv\_x = 0.001 m when sigx\_0 <= 0.001 m

xv\_y = 0.001 m when sigy\_0 <= 0.001 m

xv\_z = 0.001 m when sigz\_0 <= 0.001 m

xv\_x = 100000 m when sigx\_0 >= max(sigx from hpac\_dispersion\_coefs.csv)

xv\_y = 100000 m when sigy\_0 >= max(sigy from hpac\_dispersion\_coefs.csv)

xv\_z = 100000 m when sigz\_0 >= max(sigz from hpac\_dispersion\_coefs.csv)

## test\_calc\_sigmas.csv

**Inputs:**

* Use istab and speed for stability and wind speed.
* Use sig\_x0, sig\_y0, and sig\_z0 for source sigmas.
* Use x for downwind distances.

**Expected results:**

* Calculate sig\_x, sig\_y and sig\_z for each row.
* These should match the values in the files to within a tolerance of 1.0E-2.

**Notes:**

Correct behavior for point sources is sig\_x = sig\_y = sig\_z = 0.001 m when x < 0.001 m

Correct behavior for points with source sigmas greater than 0 is:

sig\_x = sig\_x0

sig\_y = sig\_y0

sig\_z = sig\_z0

when x < 0.001 m

Correct behavior when x >= 100,000 m is

sig\_x = sig\_x(x=100000)

sig\_y = sig\_y(x=100000)

sig\_z = sig\_z(x=100000)