Smoke Modeling

Daniel Zilberg

March 5, 2024

1 Problem Description

We are asked to model the movement of smoke particles in air after being "created" at a smokestack's aperture. The particles experience a buoyancy force due to the temperature difference between the air and the smoke particles, and the air is assumed to have a relative velocity to the smokestack's reference frame.

The buoyancy force at creation b_0 and time decay coefficient τ are provided, as is the air velocity w assumed to be in the xy plane. The simulation can also be provided with an array of existing particles and their kinematic properties.

The aperture diameter of the smokestack is provided and smoke particles are created at a given rate at random positions in the smokestack's aperture at random velocity with a given mean and standard deviation and are added to the existing smoke particles.

Given a time frame (t_i, t), the simulation will provide a new particle list consisting of the pre-existing and newly created particles with the updated kinematics.

2 Force Modeling

2.1 Drag Force

We model the kinematics of each smoke particle as a solid sphere in a viscus fluid, and therefore wish to use Stokes Law to model the drag force:

$$\vec{F}_d = -6\pi\eta A\vec{v} \equiv \gamma(\dot{\vec{x}} - \vec{w}),$$

2.1.1 Note on Brownian Motion

Because of the relatively small size and mass of smoke particles, Brownian motion could be considered for second order effects. A model can be the Langevin equation:

$$m\vec{a}_d = \vec{F}_d + \vec{f}(t)$$

Where $\vec{f}$ is sampled from a gaussian distribution of functions. Solved, the equation provides the Stokes drag solution as a mean value, and a standard deviation which depends on the temperature. Since we are asked not to assume further randomness, we assume that this order of magnitude should be neglected.

2.2 Buoyancy

We are given the buoyancy acceleration at creation b_c and a time decay constant τ. We assume exponential decay of the buoyancy force, so that for a particle created at time t_c, the buoyancy at time t_f will be

$$\vec{a}_b(t) = b_c e^{-\frac{t-t_c}{\tau}}\hat{z}$$

For the particles provided to the simulation, we can deduce the buoyancy from the acceleration in the z direction at time t_i:

$$b(t_i) = \ddot{z}(t_i) + \frac{\gamma}{m}\dot{z}(t_i) + g$$

So for the existing particles we set t_i as the creation time for the given particles and their buoyancy creation constant $b_c \equiv b(t_i)$. The buoyancy acceleration at time t_f is therefore:

$$\vec{a}_b(t) = b(t_i)e^{-\frac{t-t_i}{\tau}}\hat{z}$$

Finally, we shift the timeline between start to finish $t \leftarrow t - t_i$ for all ts in the simulation to ease the calculations and numerics.

3 Equations of Motion

The the equations of motion will now be:

$$\ddot{z} + \alpha\dot{z} = -g + a_b(t) \tag{1}$$
$$\ddot{x} + \alpha\dot{x} = \alpha w_x \tag{2}$$

Where $\alpha \equiv \frac{\gamma}{m}$ is the drag coefficient. Since the problem is symmetric in the xy plane, the equation and solution in the y coordinate will be the same as in the x coordinate up to initial conditions. Solving the equations with initial conditions for the creation moment we get:

$$z(t) = z(t_c) - \frac{\tau^2 b_c}{1-\alpha\tau}(e^{-\frac{t-t_c}{\tau}} - 1) + \tag{3}$$
$$+ \frac{1}{\alpha}(\dot{z}(t_c) + \frac{g}{\alpha} + \frac{\tau b_c}{1-\alpha\tau})(1 - e^{-\alpha(t-t_c)}) - \frac{g}{\alpha}(t - t_c) \tag{4}$$
$$x(t) = x(t_c) + \frac{1}{\alpha}(\dot{x}(t_c) - w_x)(1 - e^{-\alpha(t-t_c)}) + w_x(t - t_c) \tag{5}$$

The velocities:

$$\dot{z}(t) = \dot{z}(t_c)e^{-\alpha(t-t_c)} + \frac{\tau b_c}{1-\alpha\tau}(e^{-\alpha(t-t_c)} - e^{-\frac{t-t_c}{\tau}}) + \frac{g}{\alpha}(e^{-\alpha(t-t_c)} - 1) \tag{6}$$
$$\dot{x}(t) = w_x + (\dot{x}(t_c) - w_x)e^{-\alpha(t-t_c)} \tag{7}$$

The Accelerations:

$$\ddot{z}(t) = -\alpha\dot{z}(t_c)e^{-\alpha(t-t_c)} + \frac{\tau b_c}{1-\alpha\tau}(-\alpha e^{-\alpha(t-t_c)} + \frac{1}{\tau}e^{-\frac{t-t_c}{\tau}}) - ge^{-\alpha(t-t_c)} \tag{8}$$
$$\ddot{x}(t) = -\alpha(\dot{x}(t_c) - w_x)e^{-\alpha(t-t_c)} \tag{9}$$

Notice that the ratio between the to dissipation rates provides an irregular point in the expressions, but a closer examination shows that it is just a point discontinuity that can be hard coded away. In the simulation we didn't write that case for better code readability, but it can be easily taken into consideration.

4 Simulation Specifics

We work according to the provided instructions, and model the way in which new particles are created by calculating the number of particles to be created and randomly sample points in time and in the smokestack's aperture.

The "particles" struct is implemented as a numpy array with dimesions [N,3,3] where dimension 0 are the individual particles, dimension 1 represents position, velocity and acceleration, and dimension 2 represents the three spatial directions. In addition, "test_smoke.ipynb" contains a 3d animation that is "smoke()" is run for several time steps with approximated values for drag and buoyancy constant, calculated from characteristic values of smoke particles found on the internet.

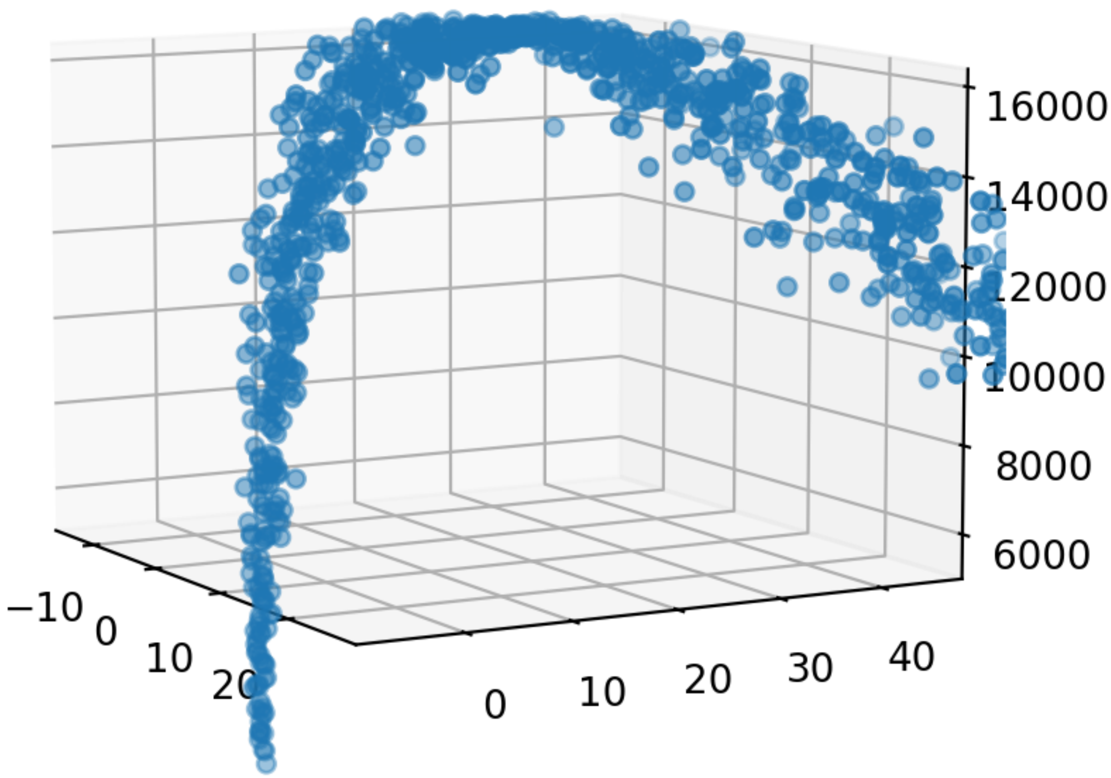

Figure 1: Smoke simulation results