Cloud Microphysics

June 29, 2025

1 Functionalities

The `CloudMicroPhysicsWindow` class provides core functionalities for simulating and visualizing cloud microphysics processes:

- **Initialization**: Configures a window for analyzing cloud microphysics, simulating 100 cloud droplets with synthetic data for radii, concentrations, and liquid water content (LWC).
- **Simulation Control**: Supports starting, pausing, resuming, and stopping the simulation, with user-defined parameters (updraft velocity, supersaturation, droplet radius, concentration, aerosol properties, air temperature, wind speed, wind angle, autoconversion threshold, wind shear) and visualization style selection.
- **Visualization**: Displays three plots updated via animation (50–500 ms): a scatterplot or histogram (colored by LWC or droplet size), a histogram or vertical profile, and a time series of mean LWC, concentration, and rain rate.
- **Parameter Input**: Allows input of simulation parameters, visualization style selection (LWC-colored scatterplot, size-colored scatterplot, droplet size histogram, vertical profile), and animation speed adjustment, with validation for physical constraints.
- **Logging and Output**: Logs initialization, simulation control, and plot updates using the `logging` module, with a console displaying real-time simulation status and parameters.

2 Simulation Logic

The simulation logic in `CloudMicroPhysicsWindow` manages cloud microphysics processes and visualization for a droplet population.

2.1 Initialization

- **Purpose**: Sets up the analysis window with synthetic droplet data and visualization.
- **Process**:
  - Configures simulation for 100 droplets with time step $\Delta t = 0.1\,\mathrm{s}$, total time $100\,\mathrm{s}$.
  - Initializes synthetic data: droplet radii ($\sim \mathcal{N}(10, 1)\,\mu\mathrm{m}$), concentrations ($100\,\mathrm{cm}^{-3}$), LWC, supersaturation (0.5%), heights ($0\,\mathrm{m}$).
  - Sets default parameters: updraft velocity ($1.0\,\mathrm{m/s}$), wind speed ($10.0\,\mathrm{m/s}$), wind angle ($0°$), drag coefficient (0.001), air temperature ($293.15\,\mathrm{K}$), autoconversion threshold ($20.0\,\mu\mathrm{m}$), wind shear ($0.01\,\mathrm{s}^{-1}$), aerosol concentration ($100\,\mathrm{cm}^{-3}$), soluble fraction (0.5).
  - Creates a control panel for parameter inputs, visualization style selection, animation speed slider, status label, and console.
  - Initializes a matplotlib figure with three subplots: scatterplot/histogram, histogram/vertical profile, time series.
  - Sets up arrays for droplet properties and lists for time steps, LWC, concentration, and rain rate histories.
  - Configures a timer for animation updates and initializes the plot and console.
  - Logs initialization and handles exceptions via `QMessageBox` and console.

2.2 Simulation Control

- **Purpose**: Manages simulation start, pause, resume, and stop operations.
- **Process (Start)**:
  - Validates inputs: $w \geq 0$, $0 \leq S \leq 0.01$, $r > 0$, $C > 0$, aerosol concentration > 0, $0 \leq f \leq 1$, $200 \leq T \leq 350\,\mathrm{K}$, $U \geq 0$, $r_{\text{auto}} > 0$, $s \geq 0$.
  - Updates parameters, resets droplet radii ($\sim \mathcal{N}(\text{initial radius}, 0.1 \times \text{initial radius})$), concentrations, LWC, heights, and histories.
  - Clears console, logs parameters, and starts timer with user-defined animation speed.
  - Disables start button, enables pause/stop buttons, sets status to ”Running”.

- **Process (Pause)**:
  - Stops timer and animation, sets `is_paused` to True.
  - Disables pause button, enables resume/stop buttons, sets status to "Paused".
- **Process (Resume)**:
  - Restarts animation and timer, sets `is_paused` to False.
  - Enables pause/stop buttons, disables resume button, sets status to "Running".
- **Process (Stop)**:
  - Stops timer and animation, resets `is_paused` and `anim`.
  - Enables start button, disables pause/resume/stop buttons, sets status to "Stopped".
- **Logging**: Logs actions and exceptions to file and console, displaying errors via `QMessageBox`.

2.3 Plot Update

- **Purpose**: Updates visualizations of droplet properties and their evolution.
- **Process**:
  - Checks if simulation should continue (`current_step` $<$ total_time$/\Delta t$).
  - Updates droplet properties via Köhler activation, condensation growth, evaporation, collision-coalescence, autoconversion, and wind shear effects.
  - Updates LWC, heights, supersaturation, and rain rate.
  - Computes wind stress for vector visualization.
  - Logs mean properties to console and updates status label.
  - Stores mean LWC, concentration, and rain rate for time series.
  - Updates plots: scatterplot/histogram (top-left), histogram/vertical profile (top-right), time series (bottom).
  - Increments simulation step and logs actions.

3 Physics and Mathematical Models

The `CloudMicroPhysicsWindow` class implements simplified models for cloud microphysics processes, simulating droplet activation, growth, and interactions.

3.1 Köhler Activation

- **Purpose**: Determines droplet activation based on aerosol properties.
- **Equations**:

$$\begin{aligned} A &= \frac{3.3 \times 10^{-5}}{T}, \\ B &= 4.3 \times 10^{-6} f, \\ r_c &= \sqrt{\frac{3B}{A}}, \end{aligned}$$

  where A is the curvature term ($\mu\mathrm{m}$), B is the solute term ($\mu\mathrm{m}^3$), T is air temperature (K), f is the soluble fraction, and r_c is the critical radius ($\mu\mathrm{m}$). Droplets with $r > r_c$ have concentration set to $0.5\times$aerosol concentration cm^{-3}.

- **Implementation**: `compute_kohler_activation` updates concentrations.

3.2 Condensation Growth

- **Purpose**: Computes droplet radius growth due to condensation.
- **Equations**:

$$\begin{aligned} e_s &= 611.2 \exp\left(\frac{L_v}{R_v}\left(\frac{1}{273.15} - \frac{1}{T}\right)\right), \\ G &= \left(\frac{\rho_w L_v}{e_s T} + 1\right)^{-1}, \\ \frac{dr}{dt} &= \frac{G S f_v}{\max(r, 10^{-6})}, \end{aligned}$$

  where e_s is saturation vapor pressure (Pa), $L_v = 2.5\times 10^6$ J/kg, $R_v = 461.5$ J/kg/K, $\rho_w = 1000\,\mathrm{kg/m^3}$, S is supersaturation (fraction), $f_v = 1 + 0.1w$ is the ventilation factor with w as updraft velocity ($\mathrm{m/s}$), and $\frac{dr}{dt}$ is clipped to $[-10^{-3}, 10^{-3}]$ $\mu\mathrm{m/s}$.

- **Implementation**: `compute_condensation_growth` calculates radius growth rate.

3.3 Collision-Coalescence

- **Purpose**: Simulates stochastic droplet growth via collisions.

- **Equations**:

$$
\begin{aligned}
P_{\text{coll}} &= 0.01 \left(\frac{r}{r_{\text{auto}}}\right)^2, \\
r_{\text{new}} &= 1.2r \quad \text{(if collision occurs)}, \\
C_{\text{new}} &= 0.8C,
\end{aligned}
$$

where P_{coll} is the collision probability (clipped to $[0, 0.1]$), r_{auto} is the autoconversion threshold (μm), and C is concentration (cm^{-3}).

- **Implementation**: `compute_collision_coalescence` updates radii and concentrations.

3.4 Autoconversion

- **Purpose**: Converts cloud water to rain based on droplet size.
- **Equations**:

$$
\begin{aligned}
a &= 0.001 \frac{\sum_{r_i > r_{\text{auto}}} (r_i - r_{\text{auto}})}{r_{\text{auto}}}, \\
R &= a \sum_{r_i > r_{\text{auto}}} \text{LWC}_i \frac{3600}{\rho_w}, \\
C_{\text{new}} &= C \exp(-a \Delta t),
\end{aligned}
$$

where a is the autoconversion rate (s^{-1}), R is the rain rate (mm/hr), LWC_i is liquid water content (g/m^3), and $\Delta t = 0.1\,\text{s}$.

- **Implementation**: `compute_autoconversion` calculates rain rate and updates concentrations.

3.5 Evaporation

- **Purpose**: Computes droplet radius reduction due to evaporation.
- **Equation**:

$$
\frac{dr}{dt} = \frac{0.1S}{\max(r, 10^{-6})} \quad \text{(if } S < 0\text{)},
$$

where $\frac{dr}{dt}$ is clipped to $[-10^{-3}, 0]\,\mu\text{m/s}$.

- **Implementation**: `compute_evaporation` calculates radius reduction for negative supersaturation.

3.6 Wind Shear Effect

- **Purpose**: Perturbs droplet concentrations due to wind shear.

- **Equation**:

$$C_{\text{new}} = C\left(1 + 0.1\frac{sr}{10}\right),$$

where s is wind shear (s^{-1}), and the factor is clipped to $[0.5, 1.5]$.

- **Implementation**: `compute_wind_shear_effect` updates concentrations.

3.7 Supersaturation Update

- **Purpose**: Updates supersaturation based on cooling and condensation.
- **Equations**:

$$\begin{aligned} e_s &= 611.2\exp\left(\frac{L_v}{R_v}\left(\frac{1}{273.15} - \frac{1}{T}\right)\right), \\ \frac{\partial T}{\partial t} &= -0.01w, \\ \frac{\partial S}{\partial t} &= -\frac{0.1\sum \text{LWC}_i}{\rho_w} + \frac{L_v}{c_p T}\frac{\partial T}{\partial t}, \end{aligned}$$

where $\frac{\partial T}{\partial t}$ is the cooling rate ($\mathrm{K/s}$), $c_p = 1004\,\mathrm{J/kg/K}$, and S is clipped to $[-0.01, 0.01]$.

- **Implementation**: `update_supersaturation` updates air temperature and supersaturation.

3.8 Wind Stress

- **Purpose**: Computes wind stress for vector visualization.
- **Equations**:

$$\begin{aligned} \theta &= \theta_0 + 0.1t, \\ \tau_x &= \rho_{\text{air}} C_d U^2 \cos(\theta), \\ \tau_y &= \rho_{\text{air}} C_d U^2 \sin(\theta), \end{aligned}$$

where θ_0 is the initial wind angle (radians), U is wind speed ($\mathrm{m/s}$), $\rho_{\text{air}} = 1.225\,\mathrm{kg/m^3}$, and $C_d = 0.001$.

- **Implementation**: `update_plot` calculates τ_x, τ_y for quiver plots.

4 Algorithms

4.1 Initialization Algorithm

- **Input**: None (uses synthetic data).
- **Steps**:

1. Log initialization to file and console.
2. Set parameters: 100 droplets, $\Delta t = 0.1\,\mathrm{s}$, total time $100\,\mathrm{s}$.
3. Initialize synthetic data: $r \sim \mathcal{N}(10, 1)\,\mu\mathrm{m}$, $C = 100\,\mathrm{cm}^{-3}$, LWC $= \frac{4}{3}\pi(r \times 10^{-6})^3 \times 10^6 C\,\mathrm{g/m}^3$, $S = 0.005$, $z = 0\,\mathrm{m}$.
4. Set defaults: $w = 1.0\,\mathrm{m/s}$, $U = 10.0\,\mathrm{m/s}$, $\theta = 0°$, $C_d = 0.001$, $T = 293.15\,\mathrm{K}$, $r_{\text{auto}} = 20.0\,\mu\mathrm{m}$, $s = 0.01\,\mathrm{s}^{-1}$, aerosol concentration $100\,\mathrm{cm}^{-3}$, soluble fraction 0.5.
5. Set window title to "Enhanced Cloud Microphysics Analysis" and size to 1000×700.
6. Create control panel with inputs, visualization selector ("Color by LWC", "Color by Droplet Size", "Droplet Size Histogram", "Vertical Profile"), speed slider (50–500 ms), status label, console.
7. Create start, pause, resume, stop buttons (pause/resume/stop initially disabled).
8. Initialize matplotlib figure with three subplots: scatterplot/histogram, histogram/vertical profile, time series.
9. Initialize arrays for droplet properties and lists for histories.
10. Set up timer, initialize plot with `update_plot(0)`, log to console.
11. Log completion or errors, displaying critical errors via `QMessageBox` and console.

4.2 Start Simulation Algorithm (`start_simulation`)

- **Input**: None (uses input fields).
- **Steps**:
  1. Log start to file and console.
  2. Validate inputs: $w \geq 0$, $0 \leq S \leq 0.01$, $r > 0$, $C > 0$, aerosol concentration > 0, $0 \leq f \leq 1$, $200 \leq T \leq 350\,\mathrm{K}$, $U \geq 0$, $r_{\text{auto}} > 0$, $s \geq 0$.
  3. Update parameters, reset: $r \sim \mathcal{N}(\text{initial radius}, 0.1 \times \text{initial radius})\,\mu\mathrm{m}$, $C = \text{initial concentration}\,\mathrm{cm}^{-3}$, LWC, $z = 0\,\mathrm{m}$, histories.
  4. Clear console, log parameters (updraft, supersaturation, radius, concentration, aerosol).
  5. Start timer with user-defined animation speed.

6. Disable start button, enable pause/stop buttons, set status to ”Running”.
7. Log completion or errors, displaying warnings or critical errors via `QMessageBox` and console.

4.3 Pause Simulation Algorithm (`pause_simulation`)

- **Input**: None.
- **Steps**:
  1. Log pause to file and console.
  2. Stop timer and animation, set `is_paused` to True.
  3. Disable pause button, enable resume/stop buttons, set status to ”Paused”.
  4. Log completion or errors, displaying critical errors via `QMessageBox` and console.

4.4 Resume Simulation Algorithm (`resume_simulation`)

- **Input**: None.
- **Steps**:
  1. Log resume to file and console.
  2. If `is_paused`, restart animation and timer with current speed.
  3. Set `is_paused` to False, enable pause/stop buttons, disable resume button, set status to ”Running”.
  4. Log completion or errors, displaying critical errors via `QMessageBox` and console.

4.5 Stop Simulation Algorithm (`stop_simulation`)

- **Input**: None.
- **Steps**:
  1. Log stop to file and console.
  2. Stop timer and animation, set `anim` to None and `is_paused` to False.
  3. Enable start button, disable pause/resume/stop buttons, set status to ”Stopped”.

4. Log completion or errors, displaying critical errors via `QMessageBox` and console.

4.6 Plot Update Algorithm (update_plot)

- **Input**: Frame number (tracks via `current_step`).
- **Steps**:
  1. Log update start at `current_step` to file and console.
  2. If `current_step` $\geq$ total_time/Δt, call `stop_simulation`.
  3. Update droplet properties:
     – Köhler activation: $C = 0.5 \times$ aerosol concentration if $r > r_c$.
     – Condensation growth: $\frac{dr}{dt} = \frac{GSf_v}{\max(r, 10^{-6})}$.
     – Evaporation: $\frac{dr}{dt} = \frac{0.1S}{\max(r, 10^{-6})}(S < 0)$.
     – Collision-coalescence: $r \to 1.2r$, $C \to 0.8C$ if $P_{\text{coll}} >$ random.
     – Autoconversion: $R = a \sum \text{LWC}_i \frac{3600}{\rho_w}$, $C \to C \exp(-a\Delta t)$.
     – Wind shear: $C \to C\left(1 + 0.1\frac{sr}{10}\right)$.
  4. Update LWC: LWC $= \frac{4}{3}\pi(r \times 10^{-6})^3 \times 10^6 C \, \text{g/m}^3$.
  5. Update heights: $z \to z + w\Delta t$.
  6. Update supersaturation: $\frac{\partial S}{\partial t} = -\frac{0.1 \sum \text{LWC}_i}{\rho_w} + \frac{L_v}{c_p T}\frac{\partial T}{\partial t}$.
  7. Compute wind stress: $\tau_x = \rho_{\text{air}} C_d U^2 \cos(\theta)$, $\tau_y = \rho_{\text{air}} C_d U^2 \sin(\theta)$.
  8. Update status label and console with mean radius (μm), LWC (g/m^3), concentration (cm^{-3}), rain rate (mm/hr), supersaturation (%), height (m).
  9. Store mean LWC, concentration, rain rate for time series.
  10. Update plots:
      – Top-left: Scatterplot (r vs. C, colored by LWC or r) or histogram (r, bins=20, range=[0,100]).
      – Top-right: Histogram (r) or vertical profile (r vs. z, colored by LWC).
      – Bottom: Time series of mean LWC, concentration, rain rate.

11. Add wind stress vectors to scatterplot/vertical profile.
12. Increment `current_step`.
13. Log completion or errors, displaying warnings or critical errors via `QMessageBox` and console.