Air-Sea Interaction

June 29, 2025

1 Functionalities

The `AirSeaInteractionWindow` class provides the following functionalities:

- **Initialization**: Sets up a window for air-sea interaction analysis, using synthetic data for ocean and atmosphere fields (temperatures, velocities, CO2 concentrations, salinity, moisture).
- **Simulation Control**: Allows users to start, pause, resume, and stop a simulation, updating parameters (drag coefficient, wind speed, wind angle, ocean current speed, evaporation rate, precipitation rate, CO2 transfer coefficient, plot coordinates) and selecting variables (wind stress, heat flux, freshwater flux, CO2 flux).
- **Visualization**: Displays a Hovmöller diagram, a heatmap with wind stress vectors and contour lines, and a time series of flux values at a selected point, updated in real-time via animation with adjustable speed.
- **Parameter Input**: Enables users to input simulation parameters, select variables, and adjust animation speed, with validation to ensure physical constraints.
- **Logging and Error Handling**: Logs initialization, simulation control, and plot updates using the `logging` module, with exception handling to display errors via message boxes.

2 Simulation Logic

The simulation logic in `AirSeaInteractionWindow` centers around managing air-sea interaction analysis and visualization with synthetic data.

2.1 Initialization

- **Purpose**: Initializes the analysis window with synthetic data and visualization setup.

- **Process**:
  - Sets up a 100×100 grid with $\Delta x = \Delta y = 1000$ m, $\Delta t = 0.1$ s, and total time 100 s.
  - Initializes synthetic fields: ocean/atmosphere temperatures, ocean velocities, salinity, moisture, and CO2 concentrations using sinusoidal/cosinusoidal patterns.
  - Sets default parameters: drag coefficient (0.001), wind speed (10.0 m/s), wind angle ($0°$), ocean current speed (0.1 m/s), evaporation/precipitation rates (10^{-6} kg/m^2/s), CO2 transfer coefficient (10^{-5} m/s).
  - Sets up a window with a control panel for inputting parameters, selecting variables, and adjusting animation speed (50–500 ms).
  - Initializes a matplotlib figure with three subplots: Hovmöller diagram, heatmap with vectors, and time series.
  - Initializes arrays for wind stress, heat flux, freshwater flux, and CO2 flux, and lists for time steps and time series data.
  - Sets up a timer for animation updates and initializes the plot.
  - Logs initialization details and handles exceptions, displaying errors via `QMessageBox`.

2.2 Simulation Control

- **Purpose**: Manages the start, pause, resume, and stop of the air-sea interaction simulation.
- **Process (Start)**:
  - Validates inputs: drag coefficient (> 0), wind speed (≥ 0), ocean current speed (≥ 0), evaporation rate (≥ 0), precipitation rate (≥ 0), CO2 transfer coefficient (> 0), plot coordinates (within grid).
  - Updates simulation parameters and resets synthetic fields with updated ocean current speed and time-varying patterns.
  - Resets simulation step, time steps, flux arrays, and time series data.
  - Starts a timer with user-defined animation speed.
  - Disables start button and enables pause/stop buttons.
- **Process (Pause)**:
  - Stops the timer and animation, setting `is_paused` to True.

  - Disables pause button and enables resume/stop buttons.

- **Process (Resume)**:
  - Restarts the animation and timer with current animation speed.
  - Sets `is_paused` to False, enables pause/stop buttons, and disables resume button.

- **Process (Stop)**:
  - Stops the timer and animation, resets `is_paused`.
  - Enables start button and disables pause/resume/stop buttons.

- **Logging**: Logs actions and handles exceptions, displaying errors via `QMessageBox`.

2.3 Plot Update

- **Purpose**: Updates the visualization of selected fluxes and their temporal/spatial evolution.

- **Process**:
  - Checks if the simulation should continue (current step $<$ total_time$/\Delta t$).
  - Updates synthetic fields with time-varying sinusoidal patterns.
  - Computes wind stress, heat flux, freshwater flux, and CO2 flux.
  - Stores data for Hovmöller (mid-y slice) and time series (at user-defined coordinates).
  - Updates plots: Hovmöller diagram, heatmap with wind stress vectors and contours, and time series of all fluxes at the selected point.
  - Increments the simulation step and logs actions.

3 Physics and Mathematical Models

The `AirSeaInteractionWindow` class implements models for air-sea interaction fluxes, incorporating momentum, heat, freshwater, and CO2 exchanges.

3.1 Sea Surface Roughness

- **Purpose**: Computes an adjusted drag coefficient based on sea surface roughness.

- **Equations**:

$$
\begin{aligned}
u_* &= \sqrt{\frac{\rho_{\text{air}} C_d U^2}{\rho_{\text{water}}}}, \\
z_0 &= \alpha \frac{u_*^2 + 0.1 U_{\text{ocean}}^2}{g}, \\
C_d^{\text{adj}} &= C_d \left(1 + 0.1 \ln\left(\max(z_0, 10^{-6})\right)\right),
\end{aligned}
$$

  where u_* is the friction velocity (m/s), C_d is the drag coefficient, U is wind speed (m/s), $\rho_{\text{air}} = 1.225\,\text{kg/m}^3$, $\rho_{\text{water}} = 1025\,\text{kg/m}^3$, $U_{\text{ocean}} = \sqrt{u_{\text{ocean}}^2 + v_{\text{ocean}}^2}$ (m/s), $\alpha = 0.018$, $g = 9.81\,\text{m/s}^2$, and C_d^{adj} is clipped to $[10^{-4}, 10^{-2}]$.

- **Implementation**: `compute_sea_surface_roughness` calculates C_d^{adj} for use in momentum and heat flux computations.

3.2 Momentum Flux (Wind Stress)

- **Purpose**: Computes wind stress and its components based on relative wind-ocean velocity.

- **Equations**:

$$
\begin{aligned}
u_{\text{rel}} &= U \cos(\theta) - u_{\text{ocean}}, \\
v_{\text{rel}} &= U \sin(\theta) - v_{\text{ocean}}, \\
s_{\text{rel}} &= \sqrt{u_{\text{rel}}^2 + v_{\text{rel}}^2}, \\
\tau &= \rho_{\text{air}} C_d^{\text{adj}} s_{\text{rel}}^2, \\
\tau_u &= \tau \frac{u_{\text{rel}}}{\max(s_{\text{rel}}, 10^{-6})}, \\
\tau_v &= \tau \frac{v_{\text{rel}}}{\max(s_{\text{rel}}, 10^{-6})},
\end{aligned}
$$

  where θ is the wind angle (radians), s_{rel} is clipped to $[0, 10^3]$, and τ, τ_u, τ_v are clipped to $[-10^5, 10^5]\,\text{N/m}^2$.

- **Implementation**: `compute_momentum_flux` calculates τ, τ_u, and τ_v for visualization and vector plotting.

3.3 Heat Flux

- **Purpose**: Computes total heat flux (sensible and latent).

- **Equations**:

$$
\begin{aligned}
k_{\text{sensible}} &= 0.01 \frac{C_d^{\text{adj}}}{C_d}, \\
Q_{\text{sensible}} &= \rho_{\text{air}} C_p^{\text{air}} k_{\text{sensible}} (T_a - T_o), \\
Q_{\text{latent}} &= \rho_{\text{air}} L_v E \text{sgn}(T_a - T_o), \\
Q_{\text{total}} &= Q_{\text{sensible}} + Q_{\text{latent}},
\end{aligned}
$$

  where $C_p^{\text{air}} = 1005$ J/kg/K, $L_v = 2.5 \times 10^6$ J/kg, E is the evaporation rate (kg/m^2/s), $T_a - T_o$ is clipped to $[-100, 100]$ K, and Q_{total} is clipped to $[-10^6, 10^6]$ W/m^2.

- **Implementation**: `compute_heat_flux` calculates Q_{total} using adjusted drag coefficient and user-input evaporation rate.

3.4 Freshwater Flux

- **Purpose**: Computes freshwater flux and salinity tendency.

- **Equations**:

$$
\begin{aligned}
P &= P_0 \left(1 + 0.5 \frac{q}{0.01}\right), \\
E &= E_0 \left(1 + 0.1 \frac{S}{35.0}\right), \\
F &= P - E, \\
\frac{\partial S}{\partial t} &= -\frac{SF}{\rho_{\text{water}} h},
\end{aligned}
$$

  where P_0 is the precipitation rate (kg/m^2/s), $q/0.01$ is clipped to $[0, 2]$, $S/35.0$ is clipped to $[0.8, 1.2]$, $h = 1000$ m, and F, $\frac{\partial S}{\partial t}$ are clipped to $[-10^{-3}, 10^{-3}]$ kg/m^2/s.

- **Implementation**: `compute_freshwater_flux` calculates F and $\frac{\partial S}{\partial t}$ using user-input rates and synthetic salinity/moisture.

3.5 CO2 Flux

4 Algorithms

4.1 Initialization Algorithm

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

- **Steps**:

  1. Log initialization.

  2. Set grid (100×100), $\Delta x = \Delta y = 1000$ m, $\Delta t = 0.1$ s, total time 100 s.

3. Initialize synthetic fields: $T_o = 290 + 10\sin(2\pi x/L_x)$, $T_a = 295 + 5\cos(2\pi y/L_y)$, $u_{\text{ocean}} = 0.1\cos(2\pi x/L_x)$, $v_{\text{ocean}} = 0.1\sin(2\pi y/L_y)$, $S = 35$ psu, $q = 0.01$ kg/kg, $C_{\text{ocean}} = 400$ ppm, $C_{\text{atm}} = 410$ ppm.

4. Set default parameters: $C_d = 0.001$, $U = 10.0$ m/s, $\theta = 0°$, $U_{\text{ocean}} = 0.1$ m/s, $E_0 = P_0 = 10^{-6}$ kg/m^2/s, $k_0 = 10^{-5}$ m/s.

5. Set window title to "Enhanced Air-Sea Interaction Analysis" and size to 1000×700.

6. Create a control panel with inputs for parameters, variable selector, and animation speed slider (50–500 ms).

7. Create start, pause, resume, and stop buttons, with pause/resume/stop initially disabled.

8. Initialize a matplotlib figure with three subplots: Hovmöller, heatmap, and time series.

9. Initialize arrays for fluxes and lists for time steps and time series data.

10. Set up a timer and call `update_plot(0)`.

11. Log completion or errors, displaying critical errors via `QMessageBox`.

4.2 Start Simulation Algorithm (`start_simulation`)

4.3 Pause Simulation Algorithm (`pause_simulation`)

4.4 Resume Simulation Algorithm (`resume_simulation`)

4.5 Stop Simulation Algorithm (`stop_simulation`)

4.6 Plot Update Algorithm (`update_plot`)