Wet and Dry Algorithm

Core Components and Initialization

The `WetAndDryAlgo` class, implemented in C#, manages wetting and drying processes in a 2D estuarine circulation model, enabling simulation of areas that transition between inundated (wet) and exposed (dry) states. The class is initialized with the following parameters:

- Bathymetry: $h_b(i, j)$ (m, positive downward)
- Minimum depth threshold: D_{min} (m)
- Grid dimensions: n_x, n_y
- Estuary dimensions: length L, width W
- Spatial steps: $\Delta x = L/(n_x - 1)$, $\Delta y = W/(n_y - 1)$

The constructor initializes a boolean array `isWet` to track wet/dry status and calls `InitializeWetDry` with a zero water level to set initial conditions.

Functioning Logic

The class provides methods to handle wetting and drying:

1. `InitializeWetDry`: Sets the initial wet/dry status based on total water depth.
2. `ApplyWetDry`: Updates wet/dry status, enforces zero velocities and salinity in dry cells, adjusts fluxes at wet/dry interfaces, and applies mass conservation corrections.
3. `GetWetDryStatus`: Returns the current wet/dry status array.

Initialization

The `InitializeWetDry` method determines the wet/dry status for each grid cell:

$$\text{isWet}[i, j] = \eta_{i,j} + h_b(i, j) \geq D_{\text{min}} \tag{1}$$

where $\eta_{i,j}$ is the water level (positive upward) and $h_b(i, j)$ is the bathymetry (positive downward). A cell is wet if the total depth $\eta_{i,j} + h_b(i, j)$ exceeds the minimum depth threshold D_{min}.

Wet/Dry Application

The `ApplyWetDry` method updates the simulation state:

1. **Wet/Dry Status Update**: For each cell:

$$\text{isWet}[i, j] = \eta_{i,j} + h_b(i, j) \geq D_{\text{min}} \tag{2}$$

If a cell is dry ($\text{isWet}[i,j] = \text{false}$):

$$u_{i,j} = 0, \quad v_{i,j} = 0 \tag{3}$$

$$\eta_{i,j} = -h_b(i,j) \quad (\text{water level set to bed level}) \tag{4}$$

$$S_{i,j} = 0 \quad (\text{no salinity in dry cells}) \tag{5}$$

2. **Flux Adjustment**: At wet/dry interfaces, velocities are modified to prevent unphysical flow:
   - If cell $(i+1,j)$ is dry: $u_{i,j} = \min(u_{i,j}, 0)$ (prevent flow into dry cell).
   - If cell $(i-1,j)$ is dry: $u_{i-1,j} = \max(u_{i-1,j}, 0)$ (prevent flow from dry cell).
   - Similarly for v in the y-direction.
3. **Mass Conservation Correction**: For wet cells, water level is adjusted based on net flux:

$$\text{fluxIn} = \begin{cases} u_{i-1,j}(\eta_{i-1,j} + h_b(i-1,j))\Delta y \Delta t & \text{if isWet}[i-1,j] \\ v_{i,j-1}(\eta_{i,j-1} + h_b(i,j-1))\Delta x \Delta t & \text{if isWet}[i,j-1] \end{cases} \tag{6}$$

$$\text{fluxOut} = \begin{cases} u_{i,j}(\eta_{i,j} + h_b(i,j))\Delta y \Delta t & \text{if isWet}[i+1,j] \\ v_{i,j}(\eta_{i,j} + h_b(i,j))\Delta x \Delta t & \text{if isWet}[i,j+1] \end{cases} \tag{7}$$

$$\eta_{i,j} \leftarrow \eta_{i,j} + \frac{\text{fluxIn} - \text{fluxOut}}{\Delta x \Delta y} \tag{8}$$

$$\eta_{i,j} = \max(-h_b(i,j), \eta_{i,j}) \quad (\text{ensure no negative depth}) \tag{9}$$

Physical and Mathematical Models

The `WetAndDryAlgo` class employs the following models:

- **Wet/Dry Condition**:

$$\text{isWet}[i,j] = \eta_{i,j} + h_b(i,j) \geq D_{\text{min}} \tag{10}$$

- **Dry Cell Conditions**:

$$u_{i,j} = v_{i,j} = S_{i,j} = 0, \quad \eta_{i,j} = -h_b(i,j) \tag{11}$$

- **Flux Adjustment**:

$$u_{i,j} = \min(u_{i,j}, 0) \quad \text{if } \neg\text{isWet}[i+1,j] \tag{12}$$

$$u_{i-1,j} = \max(u_{i-1,j}, 0) \quad \text{if } \neg\text{isWet}[i-1,j] \tag{13}$$

$$v_{i,j} = \min(v_{i,j}, 0) \quad \text{if } \neg\text{isWet}[i,j+1] \tag{14}$$

$$v_{i,j-1} = \max(v_{i,j-1}, 0) \quad \text{if } \neg\text{isWet}[i,j-1] \tag{15}$$

- **Mass Conservation**:

$$\eta_{i,j} \leftarrow \eta_{i,j} + \frac{\sum \text{fluxIn} - \sum \text{fluxOut}}{\Delta x \Delta y}, \quad \eta_{i,j} \geq -h_b(i,j) \tag{16}$$

These models ensure physically consistent transitions between wet and dry states, prevent unphysical flows, and maintain mass conservation in the 2D shallow water model.