Variable Resolution

June 29, 2025

1 Functionalities

The `VariableResolutionGrid` class provides the following functionalities:

- **Grid Initialization**: Creates a 2D grid of size $N \times N$ with spatially variable resolution, adjusting Δx and Δy based on a coastal factor to refine resolution near coasts.
- **Spatial Step Retrieval**: Provides access to the spatial step arrays (Δx, Δy) for use in numerical computations in `Model.py`.
- **Coastal Refinement**: Simulates coastal regions by reducing spatial steps in a designated coastal zone, enhancing resolution where ocean-atmosphere interactions are more complex.
- **Logging and Error Handling**: Logs initialization and retrieval operations using the `logging` module, with exception handling to catch and log errors.

2 Simulation Logic

The simulation logic in `VariableResolution.py` centers around the `VariableResolutionG` class, which sets up and manages a variable resolution grid for the simulator.

2.1 Initialization

- **Purpose**: Initializes a 2D grid with spatially variable resolution, refining spatial steps near coastal regions.
- **Process**:
  - Initializes parameters: `grid_size` (N) and `coast_factor`, which determines the degree of refinement near coasts.
  - Creates 2D arrays Δx and Δy, initially set to 1.0 across the $N \times N$ grid.

- Defines a coastal region as the first $N/4$ grid points in both x and y directions.
- Reduces Δx and Δy by dividing by `coast_factor` in the coastal region to achieve finer resolution.
- Logs initialization details and handles exceptions.

2.2 Spatial Step Retrieval

- **Purpose**: Provides access to the spatial step arrays (Δx, Δy) for use in numerical computations.
- **Process**:
  - Returns the Δx and Δy arrays as a tuple.
  - Logs the retrieval operation and handles exceptions.

3 Algorithms

3.1 Initialization Algorithm

- **Input**: `grid_size` (N), `coast_factor`.
- **Steps**:
  1. Log initialization parameters: `grid_size`, `coast_factor`.
  2. Store `grid_size` and `coast_factor` as instance variables.
  3. Initialize Δx and Δy as $N \times N$ arrays filled with 1.0.
  4. Compute coastal width: coast_width $= \lfloor N/4 \rfloor$.
  5. For each grid point (i, j):
     - If $i <$ coast_width or $j <$ coast_width:
       - Set $\Delta x_{i,j} = \Delta x_{i,j}/$coast_factor.
       - Set $\Delta y_{i,j} = \Delta y_{i,j}/$coast_factor.
  6. Log completion of initialization.
  7. Handle exceptions and log errors if initialization fails.

3.2 Spatial Step Retrieval Algorithm (`get_spatial_steps`)

- **Input**: None.
- **Steps**:
  1. Log start of spatial step retrieval.
  2. Return the tuple $(\Delta x, \Delta y)$.
  3. Log errors if retrieval fails.