Comprehensive Forcing Mechanism

Core Components and Initialization

The `CompForcingMechanism` class, implemented in C#, provides a graphical interface for simulating estuarine circulation using three solver types: 1D Simplified, 2D Shallow Water, and 3D Navier-Stokes. The simulation domain is an estuary with length $L = 10000\,\text{m}$, width $W = 2000\,\text{m}$, and depth $H = 10\,\text{m}$, discretized into grids of $n_x = 50$, $n_y = 20$, and $n_z = 10$ for 3D, or $n = 100$ for 1D. Key parameters include:

- River discharge: $Q_r = 0.1\,\text{m}^3/\text{s}$
- Tidal amplitude: $A_t = 1.0\,\text{m}$, period: $T_t = 43200\,\text{s}$
- Wind speed: $U_w = 5.0\,\text{m/s}$, direction: $\theta_w = 0^\circ$
- Salinity gradient: $\partial S/\partial x = 0.0035\,\text{PSU/m}$
- Wave height: $H_w = 0.5\,\text{m}$, period: $T_w = 10\,\text{s}$
- Storm surge amplitude: $A_s = 0.0\,\text{m}$
- Minimum depth for wetting/drying: $h_{\text{min}} = 0.01\,\text{m}$
- Gravitational acceleration: $g = 9.81\,\text{m/s}^2$
- Freshwater density: $\rho_0 = 1000\,\text{kg/m}^3$, ocean density: $\rho_{\text{ocean}} = 1025\,\text{kg/m}^3$
- Kinematic viscosity: $\nu = 10^{-6}\,\text{m}^2/\text{s}$
- Salinity diffusion coefficient: $\kappa = 10^{-4}\,\text{m}^2/\text{s}$
- Eddy viscosity: $\nu_e = 0.01\,\text{m}^2/\text{s}$
- Coriolis parameter: $f = 10^{-4}\,\text{s}^{-1}$
- Friction coefficient: $C_f = 0.0025$
- Atmospheric pressure gradient: $\partial P_a/\partial x = 0.0001\,\text{Pa/m}$
- Seasonal salinity amplitude: $S_a = 2.0\,\text{PSU}$

The class initializes a Windows Forms interface with input controls for parameters, a visualization panel, and an output console. It supports dynamic switching between solvers and optional wetting/drying algorithms.

Functioning Logic

The `CompForcingMechanism` class manages:

1. **User Interface**: Allows input of physical parameters and solver selection (1D, 2D, or 3D).
2. **Simulation Initialization**: Sets up grids and initial conditions based on solver type.

3. **Time Stepping**: Uses a Courant-Friedrichs-Lewy (CFL) condition with $C = 0.4$ to compute adaptive time steps Δt.
4. **Forcing Mechanisms**: Incorporates river discharge, tides, wind stress, waves, storm surge, baroclinic effects, Coriolis force, and friction.
5. **Numerical Solvers**: Updates velocity, salinity, water level, and temperature using explicit schemes with flux limiters.
6. **Visualization**: Plots velocity (blue), salinity (red), and water level (green) profiles.
7. **Stability Checks**: Detects numerical instabilities (NaN or Infinity) and stops the simulation if detected.

Solver Initialization

The `InitializeSimulation` method configures the solver based on the selected type:

- **3D Navier-Stokes**: Initializes 3D arrays for velocity components (u, v, w), pressure (p), salinity (S), and temperature (T). Spatial steps are $\Delta x = L/(n_x - 1)$, $\Delta y = W/(n_y - 1)$, $\Delta z = H/(n_z - 1)$. Initial conditions:

$$u_{i,j,k} = Q_r/(WH), \quad v_{i,j,k} = w_{i,j,k} = 0 \tag{1}$$
$$p_{i,j,k} = \rho_0 g\eta(1 - k\Delta z/H), \quad \eta = A_t \sin(2\pi x/L) + A_s e^{-x/L} \tag{2}$$
$$S_{i,j,k} = (\partial S/\partial x)x, \quad T_{i,j,k} = 20\,°\text{C} \tag{3}$$

- **2D Shallow Water**: Instantiates `ShallowWaterEq2D` with bathymetry $h_{i,j} = H(1 - x/L)$ if wetting/drying is enabled, using `WetAndDryAlgo`.
- **1D Simplified**: Initializes arrays for velocity (u), salinity (S), and water level (η) with $\Delta x = L/(n - 1)$:

$$u_i = Q_r/H, \quad S_i = (\partial S/\partial x)i\Delta x, \quad \eta_i = A_t \sin(2\pi x/L) \tag{4}$$

CFL Time Step

The `ComputeCFLTimeStep` method ensures numerical stability:

- **3D**: $\Delta t = C \min\left(\frac{\Delta x}{|u|_{\max}}, \frac{\Delta y}{|v|_{\max}}, \frac{\Delta z}{|w|_{\max}}, \frac{\Delta x^2}{2(\nu+\nu_e)}, \frac{\Delta y^2}{2(\nu+\nu_e)}, \frac{\Delta z^2}{2(\nu+\nu_e)}\right)$
- **2D**: Delegates to `ShallowWaterEq2D.ComputeCFLTimeStep`.
- **1D**: $\Delta t = C \min\left(\frac{\Delta x}{|u|_{\max}}, \frac{\Delta x^2}{2(\nu+\nu_e)}\right)$

The time step is constrained to $\Delta t \geq 0.01$ s.

3D Navier-Stokes Solver

The `UpdateSimulation` method for 3D solves:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z} = -\frac{1}{\rho_0}\frac{\partial p}{\partial x} + (\nu + \nu_e)\nabla^2 u + fv - \frac{g}{\rho_0}\frac{\partial \rho}{\partial x} + F_u \tag{5}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z} = -\frac{1}{\rho_0}\frac{\partial p}{\partial y} + (\nu + \nu_e)\nabla^2 v - fu + F_v \tag{6}$$

$$\frac{\partial w}{\partial t} + u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z} = -\frac{1}{\rho_0}\frac{\partial p}{\partial z} + (\nu + \nu_e)\nabla^2 w - g \tag{7}$$

$$\frac{\partial S}{\partial t} + u\frac{\partial S}{\partial x} + v\frac{\partial S}{\partial y} + w\frac{\partial S}{\partial z} = \kappa\nabla^2 S + 0.1\left|\frac{\partial u}{\partial x}\right| S \tag{8}$$

$$\frac{\partial T}{\partial t} + u\frac{\partial T}{\partial x} + v\frac{\partial T}{\partial y} + w\frac{\partial T}{\partial z} = \kappa\nabla^2 T \tag{9}$$

where $\rho = \rho_0 + S(\rho_{\text{ocean}} - \rho_0)/35 - 0.2(T - 20)$, and forcing terms include:

- Wind stress: $\tau_x = 0.001\rho_0 U_w^2 \cos(\theta_w \pi/180)$, $\tau_y = 0.001\rho_0 U_w^2 \sin(\theta_w \pi/180)$
- Wave velocity: $u_w = 0.5 H_w k\sqrt{gH}$, $k = 2\pi/(L/10)$
- Quadratic drag: $-C_f|u|u/H$
- Atmospheric pressure gradient: $\partial P_a/\partial x/\rho_0$

Advection uses upwind schemes with a minmod flux limiter:

$$\phi_{\text{lim}} = \max(0, \min(1, r)), \quad r = \frac{\phi - \phi_{\text{upwind}}}{\phi_{\text{downwind}} - \phi + 10^{-10}} \tag{10}$$

Pressure is corrected iteratively to enforce incompressibility:

$$\nabla^2 p = \nabla\cdot\mathbf{u}, \quad p_{i,j,k} = \frac{p_{i+1,j,k} + p_{i-1,j,k} + p_{i,j+1,k} + p_{i,j-1,k} + p_{i,j,k+1} + p_{i,j,k-1} - \Delta x^2(\nabla\cdot\mathbf{u})}{6} \tag{11}$$

Boundary conditions:

$$x = 0: \quad u = Q_r/(WH), \quad v = w = 0, \quad S = 0, \quad T = 20 \tag{12}$$

$$x = L: \quad u = A_t(2\pi/T_t)\cos(2\pi t/T_t), \quad v = w = 0, \quad p = \rho_0 g(A_s e^{-t/86400} + 0.05 A_t A_s \sin(2\pi t/T_t)), \quad S \tag{13}$$

2D Shallow Water Solver

The `ShallowWaterEq2D` class updates velocity (u, v), water level (η), and salinity (S) using shallow water equations, with optional wetting/drying via `WetAndDryAlgo`. Forcing includes wind, tides, waves, and storm surge.

1D Simplified Solver

The 1D solver updates velocity (u), water level (η), and salinity (S):

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = -g\frac{\partial \eta}{\partial x} - \frac{g}{\rho_0}\frac{\partial \rho}{\partial x} + \frac{\tau_x}{\rho_0 H} + u_w + 0.05 A_t A_s \sin(2\pi t/T_t) - C_f|u|u/H + fu + \frac{\partial P_a/\partial x}{\rho_0} \tag{14}$$

$$\frac{\partial \eta}{\partial t} + \frac{\partial}{\partial x}[(H+\eta)u] = A_t \sin(2\pi(t/T_t - x/\sqrt{gH})) - \epsilon\eta \tag{15}$$

$$\frac{\partial S}{\partial t} + u\frac{\partial S}{\partial x} = (\kappa + \nu_e)\frac{\partial^2 S}{\partial x^2} + 0.1\left|\frac{\partial u}{\partial x}\right| S \tag{16}$$

Boundary conditions:

$$x = 0: \quad u = Q_r/H, \quad S = 0 \tag{17}$$

$$x = L: \quad u = A_t(2\pi/T_t)\cos(2\pi t/T_t), \quad \eta = A_t \sin(2\pi t/T_t) + A_s e^{-t/86400}, \quad S = 35 + S_a \sin(2\pi t/(365 \cdot \tag{18}$$

Advection uses upwind schemes with minmod flux limiter, and salinity is smoothed near boundaries.

Visualization

The `visualizationPanel_Paint` method plots:

- Velocity (blue), salinity (red), and water level (green) as 1D profiles along the estuary length.
- For 3D, variables are averaged over y and z.
- For 2D, wet/dry cells are shown as light blue (wet) or tan (dry) backgrounds.

Values are scaled to the panel dimensions, with checks for NaN or Infinity.

Physical and Mathematical Models

- **3D Navier-Stokes**:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} = -\frac{1}{\rho_0}\nabla p + (\nu + \nu_e)\nabla^2\mathbf{u} + \mathbf{f}\times\mathbf{u} - \frac{g}{\rho_0}\nabla\rho + \mathbf{F} \tag{19}$$

$$\nabla\cdot\mathbf{u} = 0, \quad \frac{\partial S}{\partial t} + \mathbf{u}\cdot\nabla S = \kappa\nabla^2 S + 0.1\left|\frac{\partial u}{\partial x}\right| S, \quad \frac{\partial T}{\partial t} + \mathbf{u}\cdot\nabla T = \kappa\nabla^2 T \tag{20}$$

- **2D Shallow Water**: Handled by `ShallowWaterEq2D` with wetting/drying.
- **1D Simplified**:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = -g\frac{\partial \eta}{\partial x} - \frac{g}{\rho_0}\frac{\partial \rho}{\partial x} + F_{\text{total}}, \quad \frac{\partial \eta}{\partial t} + \frac{\partial}{\partial x}[(H+\eta)u] = F_\eta, \quad \frac{\partial S}{\partial t} + u\frac{\partial S}{\partial x} = (\kappa+\nu_e)\frac{\partial^2 S}{\partial x^2} + \tag{21}$$

- **Forcing Terms**: Wind stress, wave velocity, tide-surge interaction, quadratic drag, Coriolis, atmospheric pressure gradient.
- **Numerical Methods**: Explicit time stepping, upwind advection with min-mod limiter, iterative pressure correction (3D), and boundary smoothing.

The model integrates multiple physical processes, ensuring robust simulation of estuarine dynamics with user-configurable parameters and real-time visualization.