Software for Estuarine Circulation Modelling

The Estuarine Circulation Modeling software is a desktop software written to simulate and analyze estuarine hydrodynamics, stratification, turbulence, sediment transport, tidal dynamics, and wave-current interactions. It integrates numerical solvers, turbulence models, and visualization tools to model complex estuarine processes. The software supports both structured and unstructured grids, multiple coordinate systems (sigma and z-level), and advanced numerical schemes like Total Variation Diminishing (TVD).

Functionalities

1. **Core Estuarine Model**:
   - Manages parameters: estuary length (1000 m), depth (10 m), tidal amplitude (1 m), tidal period (12 hours), salinity (river: 0 PSU, ocean: 35 PSU), temperature (river: 25°C, ocean: 20°C).
   - Supports hydrostatic and non-hydrostatic modes with a Reynolds-Averaged Navier-Stokes (RANS) solver.
   - Tracks salt wedge position and provides methods to retrieve salinity, temperature, and velocity profiles.
2. **Hydrodynamic Solver**:
   - Solves 2D shallow water dynamics on a 100×100 grid for water level and velocities.
   - Incorporates tidal forcing, wind stress, Coriolis force, and bottom friction.
   - Applies river (x=0) and ocean (x=1000 m) boundary conditions.
   - Handles wet/dry dynamics and wave-current interactions.
3. **2D Shallow Water Equations**:
   - Implements a 2D shallow water model with UI visualization of water level, velocity, and salinity.
   - Allows control over tidal amplitude, period, wind speed, wind direction, wave height, and grid size.
   - Supports wet/dry transitions and wave-enhanced friction.
4. **Total Variation Diminishing**:
   - Uses TVD scheme with Harten-Lax-van Leer (HLL) flux to advect salinity, temperature, turbulent kinetic energy (k), and dissipation rate (ϵ) or specific dissipation rate (ω).
   - Supports structured (200×100) and unstructured (triangular mesh, 40×25 nodes) grids with bathymetry (shallower near river, $x < 200$ m).

- Provides UI visualization: plan view, cross-section, contour, and quiver plots.
- Tracks fields: salinity, temperature, velocity, density, k, ϵ/ω, eddy viscosity, and diffusivity.

5. **Stratification**:
   - Models 1D vertical stratification of density, salinity, temperature, and passive scalars on a 100-point grid.
   - Computes gradient Richardson number and adjusts eddy viscosity using k-ϵ, k-ω, or constant turbulence models.
   - UI controls mixing coefficient, critical Richardson number, and river scalar concentration.
6. **Baroclinic Flow**:
   - Simulates buoyancy-driven flows due to density gradients.
   - Updates velocity fields using finite differences, incorporating buoyancy effects.
7. **Passive Scalar Transport**:
   - Solves 1D advection-diffusion for passive scalars (e.g., pollutants).
   - Integrates with baroclinic flow for consistent transport.
8. **Comprehensive Forcing**:
   - Combines tidal, wind, and wave forcing.
   - UI adjusts tidal amplitude, wind speed, wave height, and visualizes velocity and water level fields.
9. **Wave-Current Interaction**:
   - Computes Stokes drift velocities using linear wave theory.
   - Calculates wave-enhanced bottom friction via a simplified Grant-Madsen model.
   - Supports dynamic updates of wave height, direction, period, and depth.
10. **Wind Forcing**:
    - Computes wind drag coefficient and stress components (τ_x, τ_y).
    - Parameterizes drag coefficient based on wind speed and wave height.
11. **Wet & Dry Algorithm**:
    - Manages wet/dry transitions for tidal flats using a minimum depth threshold (D_{min}).
    - Sets velocities, water level, and salinity to zero in dry cells.
    - Ensures mass conservation via flux divergence corrections.

12. **Simpson-Hunter Mechanism**:
    - Models tidal straining and internal tide effects in a sigma-coordinate system.
    - Computes Stokes drift, vertical velocity, and turbulent kinetic energy (TKE) production.
13. **Asymmetric Tidal Mixing**:
    - Simulates asymmetric tidal mixing effects on salinity and velocity fields.
    - Uses a `Cell` class to manage local properties (salinity, velocity, TKE).
14. **Bifurcated Estuary Model**:
    - Models circulation in bifurcated channels, accounting for branching flow dynamics.
15. **Equations of State**:
    - Computes water density based on salinity and temperature.
16. **Vertical Discretization (`VerticalDiscretization`)**:
    - Supports sigma (terrain-following) and z-level (fixed layers) coordinate systems.
    - Computes metric terms (z_ξ, z_η) for grid transformations.
17. **Richardson Number and SSI Mixing**:
    - Computes gradient Richardson number and strain-induced mixing for turbulence closure.
    - Supports k-ϵ and k-ω models.
18. **Large Eddy Simulation**:
    - Implements LES with a Smagorinsky subgrid model.
    - UI controls Smagorinsky coefficient and visualizes velocity/vorticity fields.
19. **Lattice Boltzmann LES**:
    - Uses Lattice Boltzmann Method (LBM) with a D3Q19 lattice for LES.
    - UI adjusts grid size and relaxation time, visualizing vorticity contours.
20. **Spectral Analyzer**:
    - Performs spectral analysis (Welch's method) and EOF/POD analysis on variables like Richardson number, velocity, and salinity.
    - UI controls variable selection, window type (Hanning, Hamming, Blackman, Rectangular), and visualization (PSD, heatmaps, modes).
21. **Multi-Fraction Sediment Transport**:
    - Models bedload and suspended load for multiple grain sizes.

- UI adjusts sediment properties (grain size, settling velocity) and visualizes concentration profiles.

22. **Adaptive Mesh Refinement**:
    - Refines grid based on velocity gradients or water level changes.
    - Dynamically adjusts resolution to capture fine-scale features.
23. **Visualization Renderer**:
    - Renders 2D visualizations of water level, salt wedge, salinity, temperature, passive scalar, and velocity profiles.

Simulation Logic

1. **Grid and Field Initialization**:
    - Structured grids (e.g., 200×100 in `TotalVariationDiminishing`, 100×100 in `HydrodynamicSolver`) or unstructured triangular meshes (40×25 nodes).
    - Sigma or z-level coordinates account for bathymetry (shallower near river, $x < 200$ m).
    - Fields (salinity, temperature, velocity, density, TKE, ϵ/ω) initialized with profiles (e.g., salinity: 0 PSU at river to 35 PSU at ocean).
2. **Forcing and Boundary Conditions**:
    - Tidal forcing via sinusoidal water level variations.
    - Wind forcing (`WindForcing`) and wave-current interactions (`WaveCurrentInterac`
      add stresses.
    - River ($x = 0$, low salinity/temperature) and ocean ($x = 1000$ m, high salinity/temperature) boundaries.
3. **Numerical Solvers**:
    - `HydrodynamicSolver` and `ShallowWaterEq2D`: Semi-implicit finite differences for shallow water dynamics ($\Delta t = 0.1$ s).
    - `TotalVariationDiminishing`: TVD scheme with HLL flux for stable advection.
    - `LargeEddySim`: Smagorinsky subgrid model for LES.
    - `LatticeBoltzmannLES`: LBM with D3Q19 lattice for turbulence.
    - `PassiveScalarTransportEq` and `Stratification`: Finite differences for advection-diffusion.
4. **Turbulence Modeling**:
    - k-ϵ and k-ω models compute TKE and dissipation/specific dissipation.

- Eddy viscosity modulated by gradient Richardson number.

5. **Transport and Stratification**:
   - Scalars (e.g., pollutants) advected/diffused using velocity fields from `BaroclinicFlow` or `HydrodynamicSolver`.
   - Density gradients drive baroclinic flows in `BaroclinicFlow`.
6. **Wet/Dry Dynamics**:
   - `WetAndDryAlgo` updates cell status based on depth threshold (D_{min}).
   - Adjusts fluxes to prevent flow into dry cells and ensures mass conservation.
7. **Visualization and Analysis**:
   - `VisualizationRenderer` plots plan views, cross-sections, contours, and quiver plots.
   - `SpectralAnalyzer` computes PSD and EOF/POD modes.
8. **Dynamic Updates**:
   - `AdaptiveMeshRef` refines grids based on gradients.
   - User inputs (e.g., tidal period, wind speed) updated via UI or programmatic methods.

Algorithms

1. **Initialization**:
   - Initialize grids: structured (200×100 or 100×100) or unstructured (40×25 nodes).
   - Set bathymetry: shallower near river ($x < 200\,\text{m}$).
   - Initialize fields: salinity ($S(x) = 35 \cdot x/1000$), temperature ($T(x) = 25 - 5 \cdot x/1000$), velocity, TKE, ϵ/ω.
2. **Hydrodynamic Solver**:
   - Solve continuity: $\frac{\partial \eta}{\partial t} + \frac{\partial (Hu)}{\partial x} = 0$.
   - Solve momentum: $\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = -g\frac{\partial \eta}{\partial x} + \frac{1}{H}\frac{\partial (H\tau)}{\partial x} - \frac{C_f|u|u}{H} + F$.
   - Apply boundaries: river ($u = Q_r/(BH)$), ocean ($\eta = A\,\text{sin}(\omega t)$).
   - Use semi-implicit finite differences, $\Delta t = 0.1\,\text{s}$.
3. **TVD Advection**:

- Compute HLL flux: $F_{\text{HLL}} = \begin{cases} F_L, & s_L \geq 0 \\ F_R, & s_R \leq 0 \\ \frac{s_R F_L - s_L F_R + s_L s_R (U_R - U_L)}{s_R - s_L}, & \text{otherwise} \end{cases}$, where $s_L = \min(u_L - c_L, u_R - c_R, 0)$, $s_R = \max(u_L + c_L, u_R + c_R, 0)$.
- Apply Superbee limiter: $\psi(r) = \max(0, \min(2r, (1+r)/2, 2))$, $r = \frac{C_i - C_{i-1}}{C_{i+1} - C_i}$.
- Update salinity, temperature, k, ϵ/ω.

4. **Turbulence Modeling**:
   - k-ϵ model:
     - $\frac{\partial k}{\partial t} + u \cdot \nabla k = \nabla \cdot \left[\left(\nu + \frac{\nu_t}{\sigma_k} \right) \nabla k \right] + P_k - \epsilon$.
     - $\frac{\partial \epsilon}{\partial t} + u \cdot \nabla \epsilon = \nabla \cdot \left[\left(\nu + \frac{\nu_t}{\sigma_\epsilon} \right) \nabla \epsilon \right] + C_{1\epsilon} \frac{\epsilon}{k} P_k - C_{2\epsilon} \frac{\epsilon^2}{k}$.
     - $\nu_t = C_\mu \frac{k^2}{\epsilon}$, $P_k = \nu_t \left(\frac{\partial u}{\partial x} \right)^2$.
   - k-ω model:
     - $\frac{\partial k}{\partial t} + u \cdot \nabla k = \nabla \cdot \left[\left(\nu + \frac{\nu_t}{\sigma_k} \right) \nabla k \right] + P_k - \beta^* k \omega$.
     - $\frac{\partial \omega}{\partial t} + u \cdot \nabla \omega = \nabla \cdot \left[\left(\nu + \frac{\nu_t}{\sigma_\omega} \right) \nabla \omega \right] + \alpha_\omega \frac{\omega}{k} P_k - \beta \omega^2$.
     - $\nu_t = \frac{k}{\omega}$.
   - Richardson number: $Ri = \frac{N^2}{S^2}$, $N^2 = -\frac{g}{\rho_0} \frac{\partial \rho}{\partial z}$, $S^2 = \left(\frac{\partial u}{\partial z} \right)^2 + \left(\frac{\partial v}{\partial z} \right)^2$.
   - Adjust ν_t: $\nu_t = \frac{\nu_0}{1 + 10Ri}$ (k-ϵ) or $\nu_t = \nu_0 \frac{1 + 0.5 \min(Ri, Ri_{\text{crit}})}{1 + Ri}$ (k-ω).
5. **Wet/Dry Algorithm**:
   - Check depth: cell is wet if $H = \eta + h \geq D_{\text{min}}$, else dry.
   - Set dry cell values: $u = v = \eta = S = 0$.
   - Limit fluxes: $u_{i,j} = \min(u_{i,j}, 0)$ if eastern neighbor is dry, etc.
   - Update water level: $\Delta \eta = \frac{\text{Flux}_{\text{in}} - \text{Flux}_{\text{out}}}{\Delta x \Delta y}$.
6. **Spectral Analysis**:
   - Welch's method for PSD:
     - Window: Hanning ($w_i = 0.5(1 - \cos(2\pi i/(N-1)))$), Hamming, Blackman, or Rectangular.
     - Segment: $\text{segmentData}_i = \text{data}_{\text{start}+i} \cdot w_i$.
     - PSD: $\text{PSD}_i = \frac{1}{\text{windowPower} \cdot \text{samplingRate}} \cdot |\text{FFT}(\text{segmentData})_i|^2 / \text{numSegments}$.
     - Frequencies: $f_i = \frac{i}{\text{samplingRate} \cdot \text{segmentLength}}$.
   - EOF/POD via SVD:
     - Data matrix: $\text{dataMatrix}_{s,t} = \text{data}_t[s, 0] - \text{mean}_t$.

  - Power iteration: iterate $u_i = \sum_j \text{dataMatrix}_{i,j} v_j$, normalize u; $v_j = \sum_i \text{dataMatrix}_{i,j} u_i$, normalize v.
  - Singular value: $\sigma = \sqrt{\sum_i (\text{Av}_i)^2}$.
  - Explained variance: $\text{variance}_m = \frac{\sigma_m^2}{\sum_i \sigma_i^2} \cdot 100$.

7. **Sediment Transport**:
   - Update concentrations: $\frac{\partial C_f}{\partial t} = -u\frac{\partial C_f}{\partial x} + D_f \frac{\partial^2 C_f}{\partial x^2} - w_{s,f} C_f$.
   - Compute bedload flux: $q_b = 8(\theta - \theta_c)^{1.5}\sqrt{\left(\frac{\rho_s}{\rho_0} - 1\right) g d^3}$ if $\theta > \theta_c$, else 0.

8. **Adaptive Mesh Refinement**:
   - Compute gradients: $G_{i,j} = \sqrt{\sum(\frac{\partial \phi}{\partial x})^2 + (\frac{\partial \phi}{\partial z})^2}$ for $\phi = S, T, u$.
   - Flag top 10% gradient cells.
   - Initialize 2×2 subgrids with bilinear interpolation.

Physical and Mathematical Models

Shallow Water Dynamics

- Continuity:

$$\frac{\partial \eta}{\partial t} + \frac{\partial (Hu)}{\partial x} = 0 \tag{1}$$

- Momentum:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = -g\frac{\partial \eta}{\partial x} + \frac{1}{H}\frac{\partial (H\tau)}{\partial x} - \frac{C_f |u| u}{H} + F \tag{2}$$

  where η is water surface elevation, $H = h + \eta$, h is bathymetry, u is velocity, $\tau = \nu_{\text{eff}} \frac{\partial u}{\partial x}$, $C_f = 0.002 - 0.005$, F includes Coriolis, wind, and baroclinic terms.

Baroclinic Flow

- Density:

$$\rho = \rho_0[1 - \alpha(T - T_0) + \beta_S(S - S_0)] \tag{3}$$

  where $\rho_0 = 1000\,\text{kg/m}^3$, $\alpha = 2 \times 10^{-4}\,/°\text{C}$, $\beta_S = 8 \times 10^{-4}\,/\text{PSU}$, $T_0 = 20°\text{C}$, $S_0 = 35\,\text{PSU}$.

- Baroclinic pressure gradient:

$$\frac{\partial p_b}{\partial x} = -g \int_{\eta}^{-h} \frac{\partial \rho}{\partial x}\, dz \tag{4}$$

Passive Scalar Transport

- Advection-diffusion:

$$\frac{\partial C}{\partial t} + u\frac{\partial C}{\partial x} = \frac{\partial}{\partial x}\left(K\frac{\partial C}{\partial x}\right) \tag{5}$$

where $K = K_m + \frac{\nu_t}{\mathrm{Sc}_t}$, $K_m \approx 10^{-9}\,\mathrm{m^2/s}$, $\mathrm{Sc}_t \approx 0.7$.

Turbulence Models

- k-ϵ model:

$$\frac{\partial k}{\partial t} + u\cdot\nabla k = \nabla\cdot\left[\left(\nu + \frac{\nu_t}{\sigma_k}\right)\nabla k\right] + P_k - \epsilon \tag{6}$$

$$\frac{\partial \epsilon}{\partial t} + u\cdot\nabla \epsilon = \nabla\cdot\left[\left(\nu + \frac{\nu_t}{\sigma_\epsilon}\right)\nabla \epsilon\right] + C_{1\epsilon}\frac{\epsilon}{k}P_k - C_{2\epsilon}\frac{\epsilon^2}{k} \tag{7}$$

where $\nu_t = C_\mu\frac{k^2}{\epsilon}$, $P_k = \nu_t\left(\frac{\partial u}{\partial x}\right)^2$, $C_\mu = 0.09$, $\sigma_k = 1.0$, $\sigma_\epsilon = 1.3$, $C_{1\epsilon} = 1.44$, $C_{2\epsilon} = 1.92$.

- k-ω model:

$$\frac{\partial k}{\partial t} + u\cdot\nabla k = \nabla\cdot\left[\left(\nu + \frac{\nu_t}{\sigma_k}\right)\nabla k\right] + P_k - \beta^* k\omega \tag{8}$$

$$\frac{\partial \omega}{\partial t} + u\cdot\nabla \omega = \nabla\cdot\left[\left(\nu + \frac{\nu_t}{\sigma_\omega}\right)\nabla \omega\right] + \alpha_\omega\frac{\omega}{k}P_k - \beta\omega^2 \tag{9}$$

where $\nu_t = \frac{k}{\omega}$.

Wave-Current Interaction

- Stokes drift:

$$u_s = \frac{a^2\omega k\cosh(2kz)}{2\sinh^2(kd)} \tag{10}$$

where $L = \sqrt{gd}\cdot T$, $k = \frac{2\pi}{L}$, $\omega = \frac{2\pi}{T}$.

- Wave-enhanced friction:

$$C_d = C_{d0}(1 + \beta U_b), \quad U_b = \frac{a\omega}{\sinh(kd)} \tag{11}$$

where $\beta = 0.2$, $C_{d0} = 0.0025$, $C_d \le 0.01$.

Wind Forcing

- Drag coefficient:

$$C_d = (0.75 + 0.067U_{10} + 0.1H_s)\times 10^{-3}, \quad 0.001 \le C_d \le 0.003 \tag{12}$$

- Wind stress:

$$\tau_x = \rho_{\mathrm{air}}C_d U_{10}^2\cos(\theta), \quad \tau_y = \rho_{\mathrm{air}}C_d U_{10}^2\sin(\theta) \tag{13}$$

where $\rho_{\mathrm{air}} = 1.225\,\mathrm{kg/m^3}$.

Wet/Dry Algorithm

- Depth criterion: wet if $H = \eta + h \geq D_{\text{min}}$, else dry.
- Flux limiting: $u_{i,j} = \text{min}(u_{i,j}, 0)$ if eastern neighbor is dry, etc.
- Mass conservation:

$$\Delta\eta = \frac{\text{Flux}_{\text{in}} - \text{Flux}_{\text{out}}}{\Delta x \Delta y}, \quad \text{Flux}_u = u(\eta + h)\Delta y \Delta t \tag{14}$$

Simpson-Hunter Mechanism

- Stokes drift:

$$u_s = a\omega e^{-2kz} \sin(\theta) \tag{15}$$

- Internal tide:

$$N^2 = -\frac{g}{\rho_0}\frac{\partial \rho}{\partial z}, \quad w_{\text{tide}} = A \sin(\theta)\sqrt{N^2} \cos\left(\frac{2\pi z}{H}\right) \tag{16}$$

- Tidal straining:

$$\frac{\partial S}{\partial t} = -Cu\frac{\partial S}{\partial x} \tag{17}$$

Asymmetric Tidal Mixing

- Navier-Stokes with Boussinesq:

$$\frac{\partial u}{\partial t} + u \cdot \nabla u = -\frac{\nabla p}{\rho_0} + \nu \nabla^2 u + \frac{g(\rho - \rho_0)}{\rho_0} + f \times u - \frac{\tau_b}{\rho_0} \tag{18}$$

- Tidal asymmetry: flood factor 1.2, ebb factor 0.8.

Vertical Discretization

- Sigma coordinates: $z = \sigma H$, $\sigma \in [0, 1]$.
- Z-level: $\Delta z = H/(N_z - 1)$.
- Metric terms:

$$z_\xi \approx \frac{z_{i+1,j} - z_{i-1,j}}{2\Delta\xi}, \quad z_\eta \approx \frac{z_{i,j+1} - z_{i,j-1}}{2\Delta\eta} \tag{19}$$

Richardson Number and SSI Mixing

- Richardson number:

$$Ri = \frac{N^2}{S^2}, \quad N^2 = -\frac{g}{\rho_0}\frac{\partial \rho}{\partial z}, \quad S^2 = \left(\frac{\partial u}{\partial z}\right)^2 + \left(\frac{\partial v}{\partial z}\right)^2 \tag{20}$$

- Viscosity adjustment:

$$\nu_t = \frac{\nu_0}{1 + 10Ri} \quad (\text{k-}\epsilon), \quad \nu_t = \nu_0 \frac{1 + 0.5\,\text{min}(Ri, 0.25)}{1 + Ri} \quad (\text{k-}\omega) \tag{21}$$

- TKE production:

$$P = \nu_t \left[\left(\frac{\partial u}{\partial z} \right)^2 + \left(\frac{\partial v}{\partial z} \right)^2 \right] \tag{22}$$

Lattice Boltzmann LES

- LBM:

$$f_i(x + c_i \Delta t, t + \Delta t) = f_i(x, t) + \frac{f_i^{\text{eq}}(x, t) - f_i(x, t)}{\tau} \tag{23}$$

- Smagorinsky:

$$\nu_t = (C_s \Delta)^2 \sqrt{2 S_{ij} S_{ij}}, \quad S_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \tag{24}$$

Spectral Analysis

- Synthetic data:
  - Richardson number: Value $= 0.5 + 0.3 \sin\left(\frac{2\pi t}{43200}\right) + 0.2 \sin\left(\frac{2\pi t}{17 \cdot 3600}\right) +$ noise.
  - Velocity: Value $= 0.1 + 0.05 \sin\left(\frac{2\pi t}{43200}\right) + 0.02 \sin\left(\frac{2\pi t}{21600}\right) +$ noise.
  - Salinity: $S_{x,y} = \left(30 + 2 \sin\left(\frac{2\pi t}{44712}\right)\right) (1 - 0.1(x + y)) +$ noise.
- Welch's PSD:

$$\text{PSD}_i = \frac{1}{\text{windowPower} \cdot \text{samplingRate}} \cdot \frac{|\text{FFT}(\text{segmentData})_i|^2}{\text{numSegments}} \tag{25}$$

- EOF/POD via SVD:

$$\text{dataMatrix}_{s,t} = \text{data}_t[s, 0] - \text{mean}_t \tag{26}$$

Sediment Transport

- Suspended load:

$$\frac{\partial C_f}{\partial t} = -u \frac{\partial C_f}{\partial x} + D_f \frac{\partial^2 C_f}{\partial x^2} - w_{s,f} C_f \tag{27}$$

- Bedload:

$$q_b = \begin{cases} 8 \left(\theta - \theta_c\right)^{1.5} \sqrt{\left(\frac{\rho_s}{\rho_0} - 1\right) g d^3}, & \theta > \theta_c \\ 0, & \text{otherwise} \end{cases} \tag{28}$$