Introduction to 2D Hydraulics Equations

Alex Sánchez, Ph.D.

Senior Hydraulic Engineer
USACE, Institute for Water Resources, Hydrologic Engineering Center

US Army Corps of Engineers®

Hydraulic Modeling

Formal Framework:
- Mass Conservation
- Momentum Conservation
- Empirical Equations

Computational Concepts:
- Computational Mesh
- Numerical Methods
- Subgrid Bathymetry

HEC-RAS

Data:
- Topography, Maps, LIDAR
- Historical Information
- Modeler Experience

Engineering Problems:
- Project Planning & Design
- Dam and Levee Breaks
- Overland Flows, Floods

Outline

- Mass Conservation (Continuity)
- Momentum Conservation (Depth-Averaged)
  - Acceleration
  - Coriolis term
  - Hydrostatic pressure
  - Turbulent mixing
  - Friction
- Diffusion Wave Equation
- Numerical Methods

Mass Conservation

- Assuming a constant water density

$$\frac{\partial H}{\partial t} + \nabla \cdot (h\boldsymbol{V}) = q$$

- Integrating over a computational cell

$$\frac{\partial}{\partial t}\iiint_{\Omega} d\Omega + \iint_{S} (\boldsymbol{V} \cdot \boldsymbol{n}) dS = Q$$

- Finite-Volume Discretization

$$\frac{\Omega_i^{n+1} - \Omega_i^{n}}{\Delta t} + \sum_{k \in i} (\boldsymbol{V}_k \cdot \boldsymbol{n}_{ik}) A_k = Q$$

H : Water surface elevation

h : Water depth

q : Water souce/sink

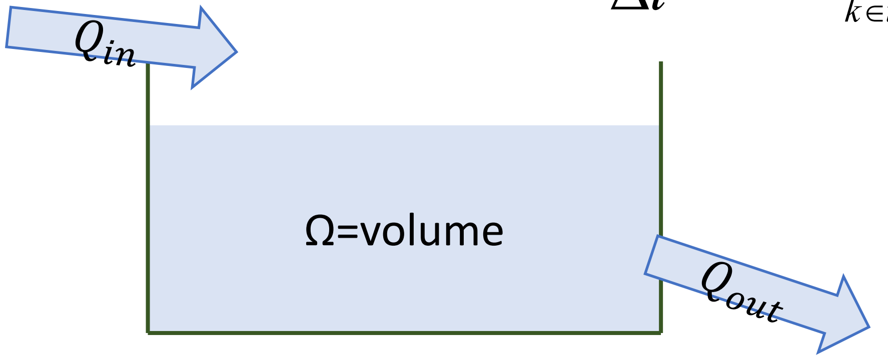
Ω_i : Cell water volume

A_k : Face area

$\boldsymbol{V}_k$: Face velocity

$\boldsymbol{n}_{ik}$: Outward face-normal unit vector

Δt : Time step



Change in volume in a system balances with flow through boundaries

Momentum Conservation

- Momentum Equation (non-conservative form)

$$\frac{\partial \boldsymbol{V}}{\partial t} + (\boldsymbol{V} \cdot \nabla)\boldsymbol{V} + f\boldsymbol{k} \times \boldsymbol{V} = -g\nabla H + \frac{1}{h}\nabla \cdot (\boldsymbol{\nu}_t h \nabla \boldsymbol{V}) - c_f \boldsymbol{V} + \frac{\boldsymbol{\tau}_s}{\rho h}$$

- From Newton's 2nd Law of motion
- Assumes constant water density, small vertical velocities, hydrostatic pressure, etc.
- Non-linear and a function of both velocity and water levels
- Continuity and Momentum Equations are the Shallow Water Equations or sometimes referred to as the "Full Momentum" equations in HEC-RAS

Acceleration and Total Derivative

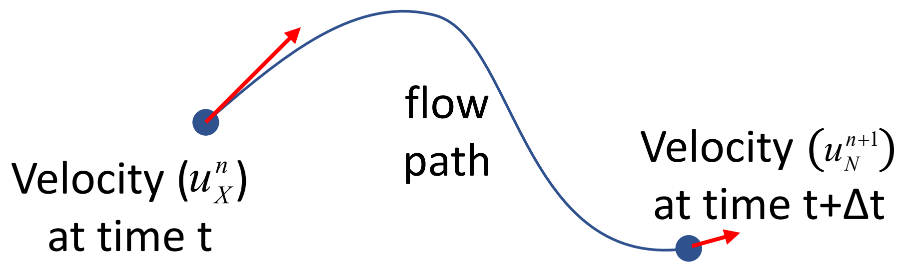
- **Eulerian:** Frame of reference is fixed in space and time

$$\frac{\partial \boldsymbol{V}}{\partial t} + (\boldsymbol{V} \cdot \nabla)\boldsymbol{V}$$

  - Easier to compute
  - Time-step restricted by Courant condition

- **Lagrangian:** Frame of reference moves with total derivative along flow path

$$\frac{\partial \boldsymbol{V}}{\partial t} + (\boldsymbol{V} \cdot \nabla)\boldsymbol{V} = \frac{D\boldsymbol{V}}{Dt} = \frac{\boldsymbol{V}^{n+1} - \boldsymbol{V}_X^n}{\Delta t}$$



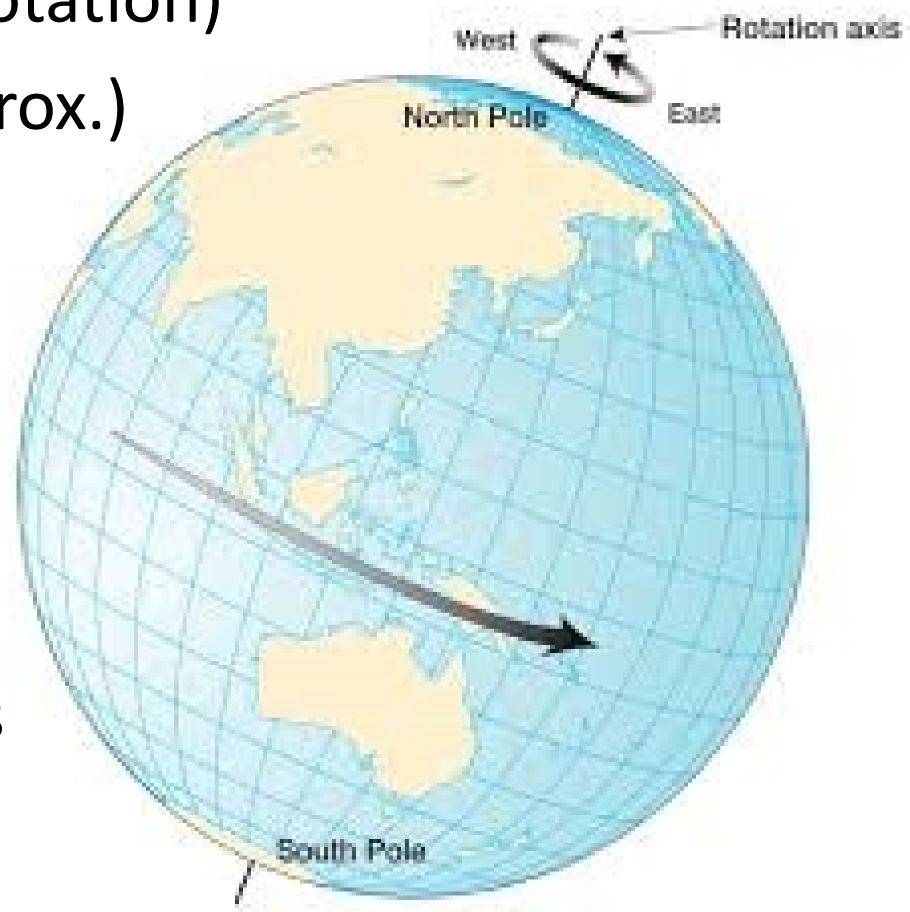
  - More expensive to compute
  - Allows larger time-steps

Coriolis Acceleration

- Effect of rotating frame of reference (earth's rotation)
- Constant for the each 2D domain (f-plane approx.)

$$f = 2\,\omega \sin\varphi$$

  - ω: sidereal angular velocity of the Earth
  - φ: latitude. Positive for northern hemisphere. Negative for southern hemisphere
- Coriolis acceleration disabled by default to save computational time
- Negligible for most river and flood simulations
- When to enable Coriolis term?
  - Large domains
  - Higher latitudes



Hydrostatic Pressure

- Assumes vertical water accelerations are small compared to gravity
- Total pressure is

$$P = Patm + \rho g(H - z)$$

  - P_{atm}: atmospheric pressure (assumed to be constant)
  - ρ : constant water density
  - g: gravity acceleration constant
  - H: water surface elevation
  - z: vertical coordinate
- Pressure gradient

$$\frac{\partial P}{\partial x} = \rho g \frac{\partial H}{\partial x}$$

Diffusion of Momentum

- Non-conservative Formulation
  - Only option in Version 5.0.7 and earlier,
  - Optional in Version 6.0

$$\frac{D\boldsymbol{V}}{Dt} = -g\nabla H + \boxed{\nu_t \Delta \boldsymbol{V}} - c_f \boldsymbol{V}$$

- Conservative Formulation
  - Default in Version 6.0
  - Only option for Eulerian SWE solver

$$\frac{D\boldsymbol{V}}{Dt} = -g\nabla H + \boxed{\frac{1}{h}\nabla \cdot \left(\boldsymbol{\nu}_t h \nabla \boldsymbol{V}\right)} - c_f \boldsymbol{V}$$

$\Delta = \nabla^2$: Laplacian

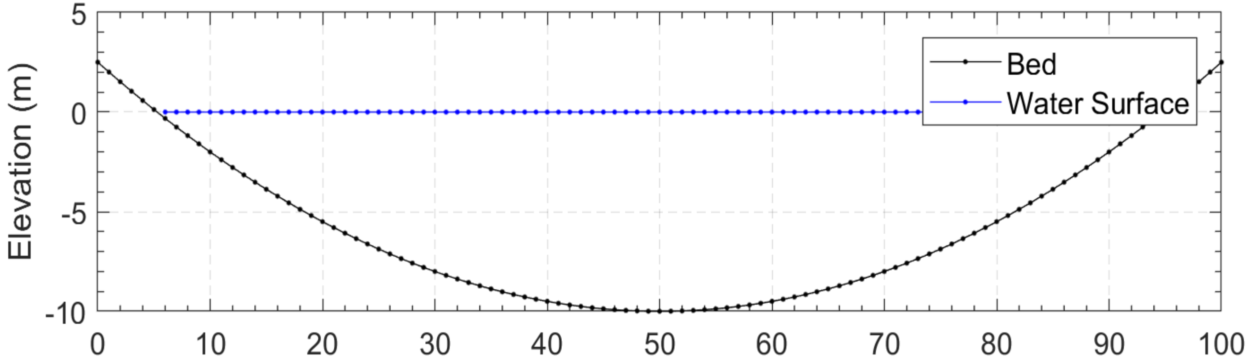
u_N : Face-normal velocity

ν_t : Turbulent eddy viscosity

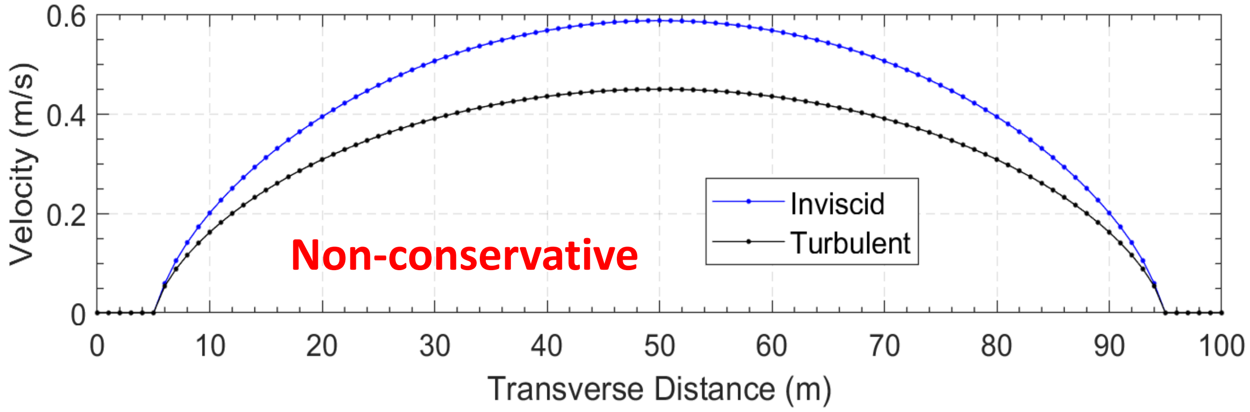
h : Water depth

c_f : Non-linear friction coefficient

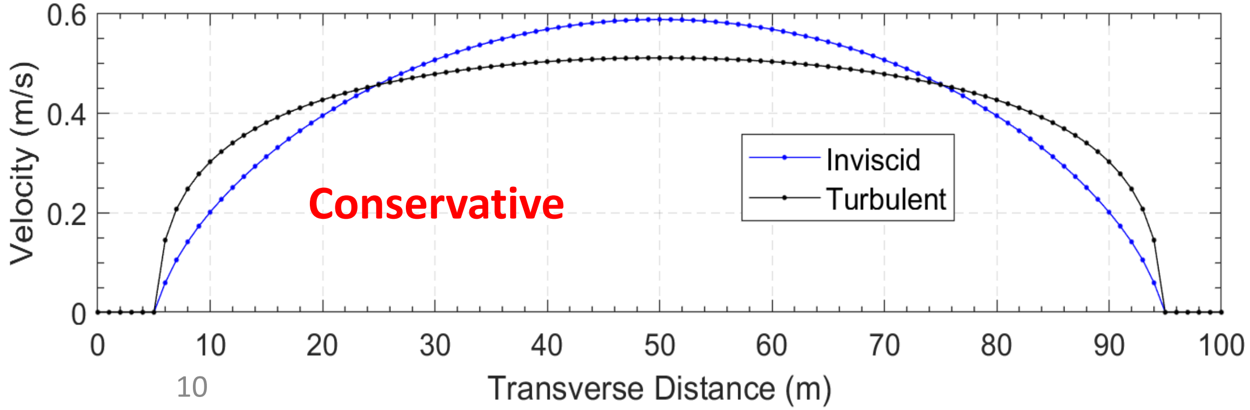
Mixing Term Formulation Comparison



Bathymetry and water level



Produces a net dissipation



Decreases velocities in middle of channel but increases velocities near banks

Eddy Viscosity: Turbulence Model

- Old: Parabolic $\nu_t = D u_* h$
  - Versions 5.0.7 and earlier
  - Isotropic (same in all directions)
  - 1 parameter: mixing coefficient D
- New: Parabolic-Smagorisnky

$$\boldsymbol{\nu}_t = \boldsymbol{D} u_* h + (C_s \Delta)^2 |\bar{S}|$$

$$|\bar{S}| = \sqrt{2\left(\frac{\partial u}{\partial x}\right)^2 + 2\left(\frac{\partial v}{\partial y}\right)^2 + \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)^2} \qquad \boldsymbol{D} = \begin{bmatrix} D_{xx} & 0 \\ 0 & D_{yy} \end{bmatrix}$$

$$D_{xx} = D_L \cos^2\theta + D_{,T} \sin^2\theta$$
$$D_{yy} = D_L \sin^2\theta + D_T \cos^2\theta$$

  - Default method in Version 6.0
  - Non-Isotropic (not the same in all directions)
  - 3 parameters: D_L, D_T, and C_s

u_* : Shear velocity

h : Water depth

D : Mixing coefficient

D_L : Longitudinal mixing coefficient

D_T : Transverse mixing coefficient

C_s : Smagorinsky coefficient

Bottom Friction

- Resisting force due to relative motion of fluid against the bed
- Non-linear friction coefficient

$$c_f = \frac{n^2 g |\boldsymbol{V}|}{R^{4/3}}$$

- n :Manning coefficient
- g: gravity acceleration constant
- $|\boldsymbol{V}|$: velocity magnitude
- R: hydraulic radius

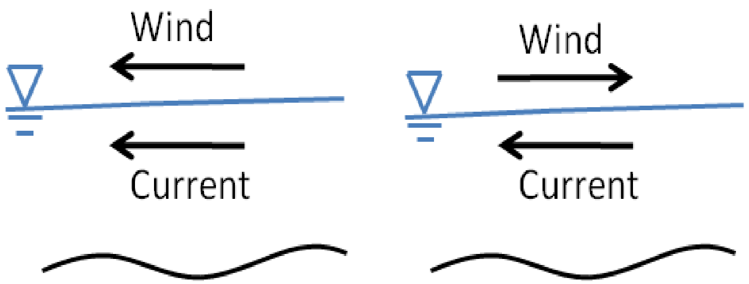
Wind Stress

- Surface Stress is given by

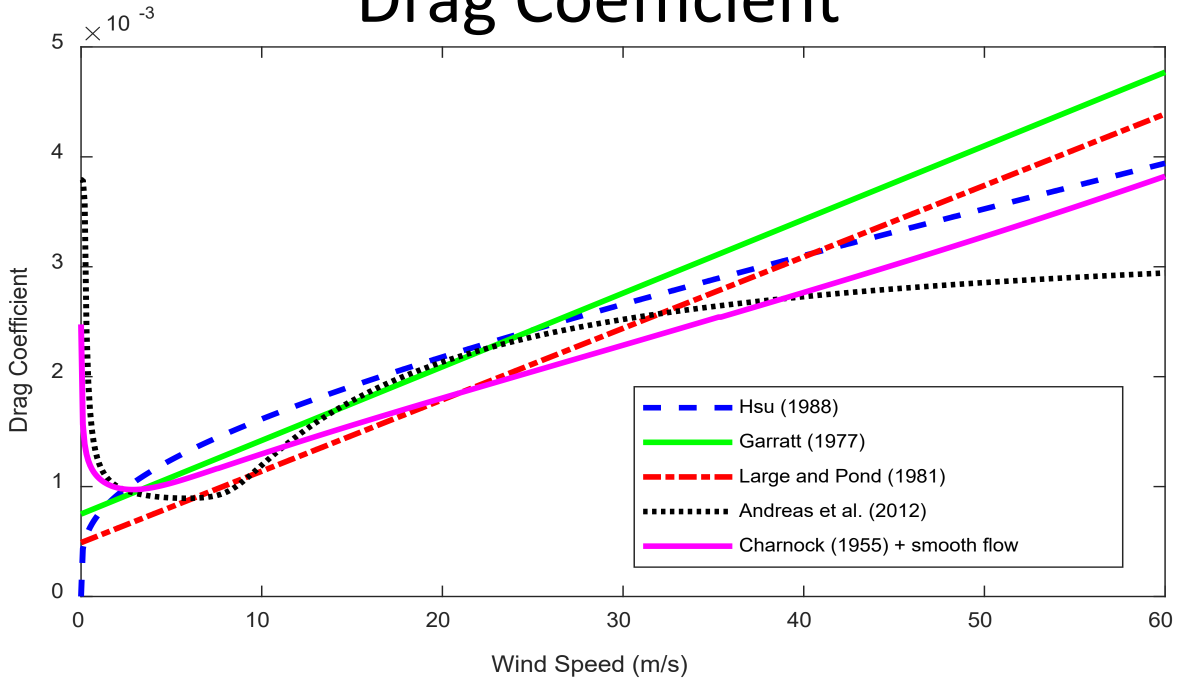
$$\boldsymbol{\tau}_s = \rho_a C_D |\boldsymbol{W}_{10}| \boldsymbol{W}_{10}$$

- Wind Reference Frame

$$\boldsymbol{W}_{10} = \begin{cases} \boldsymbol{W}_{10}^E - \boldsymbol{V} & \text{for Lagrangian} \\ \boldsymbol{W}_{10}^E & \text{for Eulerian} \end{cases}$$



Drag Coefficient



Diffusion-Wave Approximation

- Ignoring the acceleration, Coriolis, mixing, and wind-forcing terms, the momentum equation reduces to

$$\frac{n^2}{R^{4/3}}\left|\boldsymbol{V}\right|\boldsymbol{V} = -\nabla H$$

- Dividing both sides by the square of its norm

$$\boldsymbol{V} = -\frac{R^{2/3}}{n}\frac{\nabla H}{\left|\nabla H\right|^{1/2}}$$

- Inserting the above equation into the Continuity Equation leads to the Diffusion-Wave Equation (DWE)

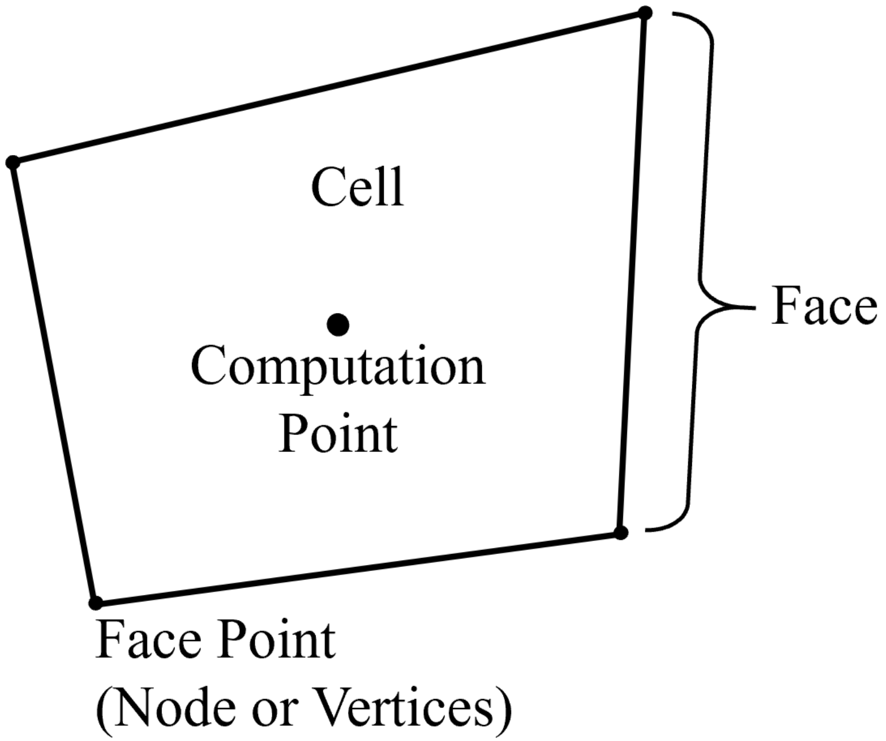
$$\frac{\partial H}{\partial t} = \nabla\cdot\left(\beta\nabla H\right) + q \qquad \beta = \frac{R^{2/3}h}{n\left|\nabla H\right|^{1/2}}$$

SWE vs. DWE

- Use SWE for:
  - Flows with dynamic changes in acceleration
  - Studies with important wave effects, tidal flows
  - Detail solution of flows around obstacles, bridges or bends
  - Simulations influenced by Coriolis, mixing, or wind
  - To obtain high-resolution and detailed flows
- Use DWE for:
  - Flow is mainly driven by gravity and friction
  - Fluid acceleration is monotonic and smooth, no waves
  - To compute approximate global estimates such as flood extent
  - To assess approximate effects of dam breaks
  - To assess interior areas due to levee breeches
  - For quick estimations or preliminary runs

Computational Mesh

- Mesh/grid can be unstructured
- Polygonal cells of up to 8 sides
- Cells must be concave
- Multiple 2D mesh can be run together or independently.
- Grid Notation
  - Cells, Faces, Face Points (i.e. nodes or vertices), Computational Points, etc.



Numerical Methods

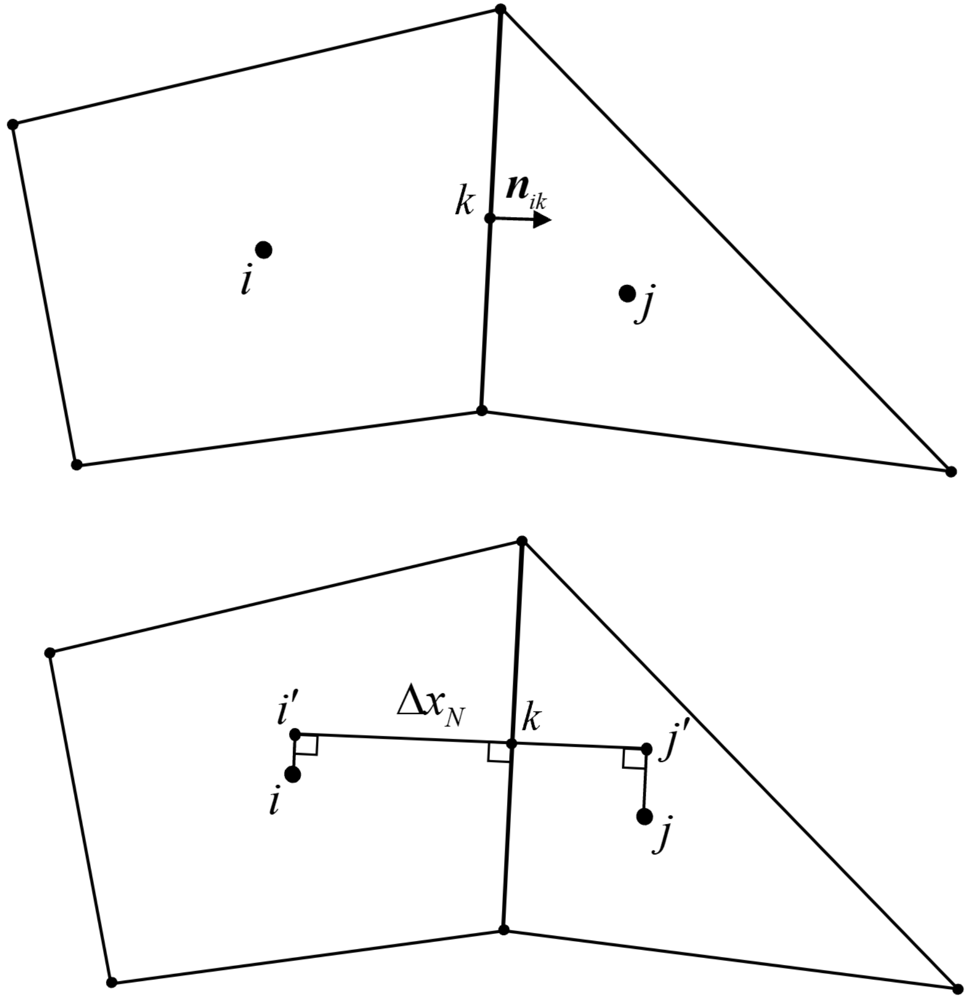
- Both DWE and SWE solvers are **Semi-implicit**
- Terms treated as:
  - Explicit: acceleration and diffusion terms
  - Semi-implicit: friction, flow divergence terms, and water level gradient
  - Fully-Implicit: water level gradient term
- By treating the “fast” pressure gradient term implicitly, the time step limitation based on the wave celerity can be removed
- Both DWE and SWE use **Finite-Difference** and **Finite-Volume** Methods
- Time integration: **Finite-Difference**
- Continuity Equation: **Finite-Volume**
- Momentum Equation: **Finite-Difference** (no control volume)

Face Water Surface Gradient

- Face-Normal Gradient

$$\nabla H \cdot \boldsymbol{n}_{ik} = \frac{\partial H}{\partial N} \approx \frac{H_j - H_i}{\Delta x_N}$$

  - Uses **Cell Centroids** and **NOT** the **Computation Points**
  - Future versions may include non-orthogonal
  - Compact two-point stencil is computationally efficient and robust



Momentum Conservation

- Momentum conservation is directionally invariant
- Only "face-normal" component is needed at faces so

$$\frac{\partial u_N}{\partial t} + (\boldsymbol{V} \cdot \nabla) u_N - f u_T = -g \frac{\partial H}{\partial N} + \frac{1}{h} \nabla \cdot (\boldsymbol{v}_t h \nabla u_N) - c_f u_N + \frac{\tau_{s,N}}{\rho h}$$

where u_N is the velocity in the N direction

Face-Tangential Velocity

- Tangential velocities are computed on left and right of face with a Least-squares Formulation

$$S_R = \sum_{k \in R}^{3} \left(\boldsymbol{V}_R \cdot \boldsymbol{n}_k - (u_N)_k \right)^2 \qquad S_L = \sum_{k \in L}^{3} \left(\boldsymbol{V}_L \cdot \boldsymbol{n}_k - (u_N)_k \right)^2$$

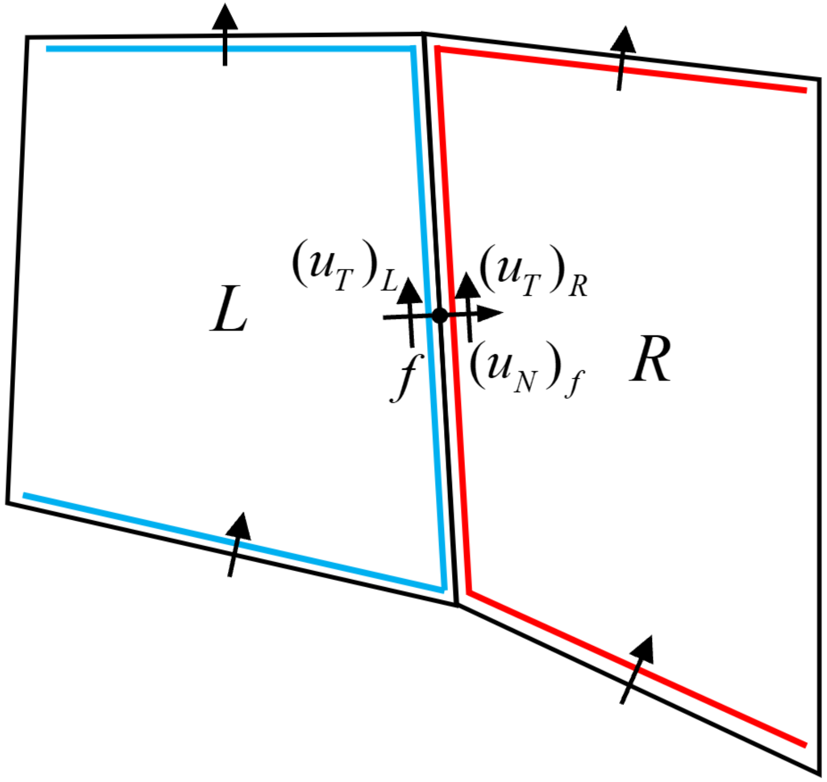
- Of the left and right reconstructed velocities, only the tangential component is used, because the normal component is known

$$(u_T)_R = \boldsymbol{V}_R \cdot \boldsymbol{t}_f \qquad (u_T)_L = \boldsymbol{V}_L \cdot \boldsymbol{t}_f$$

- Average face-tangential velocity computed as

$$(u_T)_f = \frac{(u_T)_R + (u_T)_L}{2}$$

Double-C Stencil



Discretization

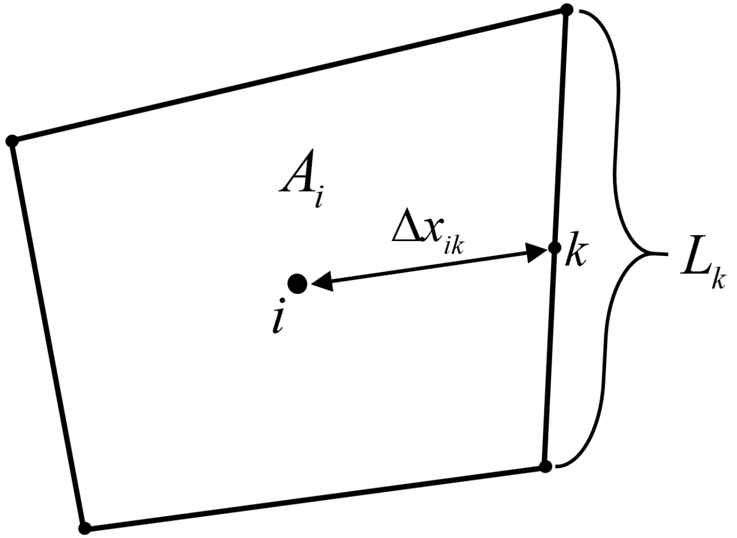
- Cell Velocity Gradient (x-direction)
  - Gauss' Divergence Theorem

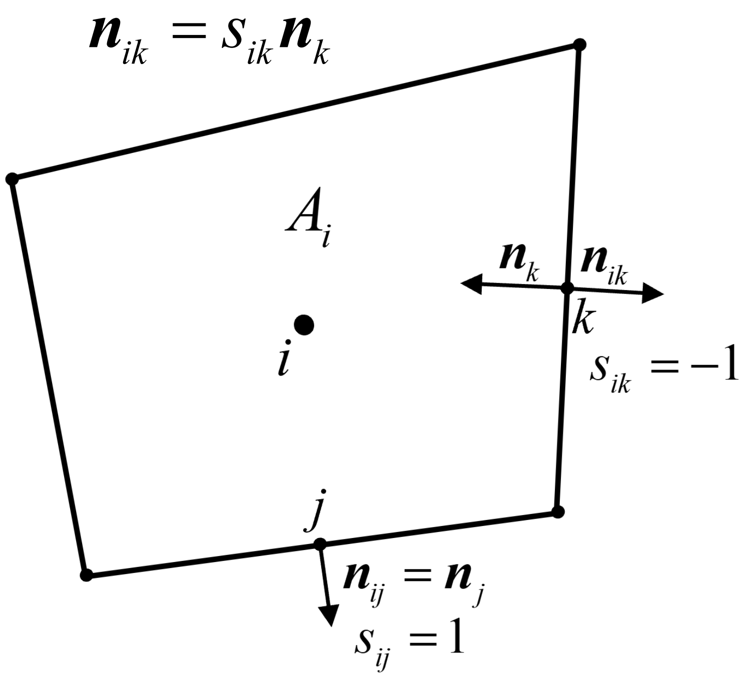
$$\nabla u_i = \frac{1}{A_i}\int_A \nabla u \mathrm{d}A = \frac{1}{A_i}\oint_L u\boldsymbol{n}\mathrm{d}L = \frac{1}{A_i}\sum_{k\in i} u_k \boldsymbol{n}_{ik} L_k$$

  - Needed tor turbulence modeling
- Cell Velocity
  - Perot's Method

$$\boldsymbol{V}_i = \frac{1}{A_i}\sum_{k\in i} \Delta x_{ik} L_k \boldsymbol{n}_k (u_N)_k$$

  - Needed for the conservative form of the mixing term and for Eulerian advection





Discretization: Laplacian

- Node Laplacian

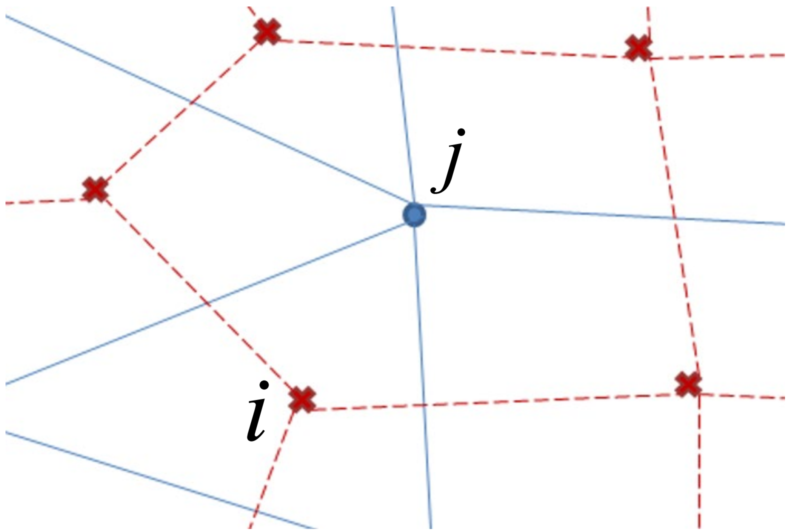
$$\left(\nabla^2 \boldsymbol{V}\right)_j = \left[\nabla \cdot \left(\nabla \boldsymbol{V}\right)\right]_j \approx \sum_i d_i \left(\nabla \boldsymbol{V}\right)_i$$

$$\left(\nabla \boldsymbol{V}\right)_i = \sum_k c_k \boldsymbol{V}_k$$

i : Cells

j : Nodes

k : Faces



- Used only by non-conservative turbulence

Backtracking

1. Interpolate node velocities from faces
2. Set starting location and time as $A = f,\ T_R = \Delta t$
3. From starting location and velocity, find location B
4. Compute time to location B: $T_B = (\boldsymbol{x}_A - \boldsymbol{x}_B)\boldsymbol{V}_A^{-1}$
5. Interpolate velocity at location B: $\boldsymbol{V}_B = w_{n1}\boldsymbol{V}_{n1} + w_{n2}\boldsymbol{V}_{n2}$

if $T_B > T_R$

6. Set $A = B,\ T_R = T_R - T_B$, and go to step 3

else

7. Find location X as $\boldsymbol{x}_X = \boldsymbol{x}_f - T_R\boldsymbol{V}_A$
8. Interpolate velocity vector at X

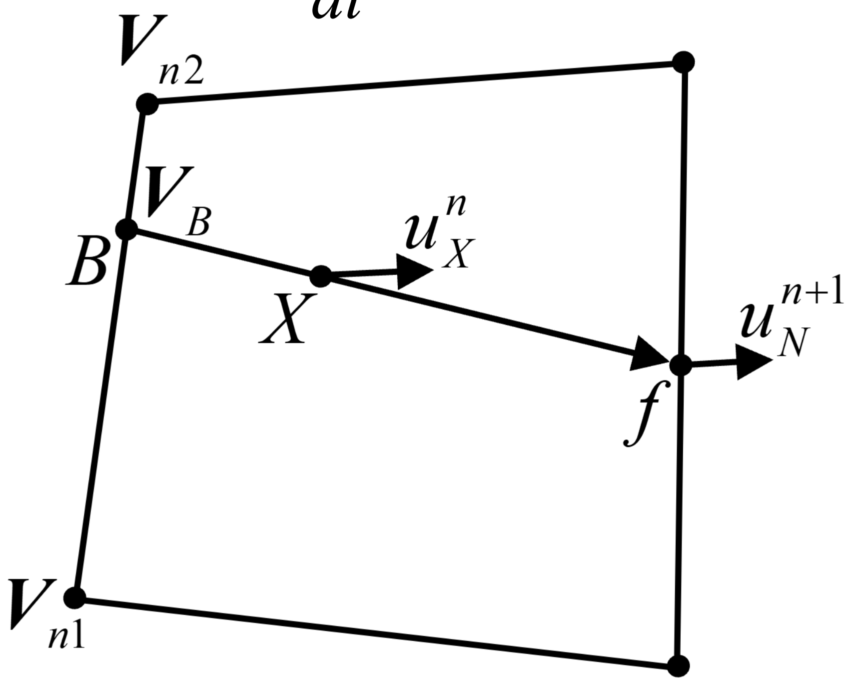
$$\boldsymbol{V}_X = T_B^{-1}\left[T_R\boldsymbol{V}_B + \left(T_B - T_R\right)\boldsymbol{V}_A\right]$$

9. Compute advective velocity

$$u_X = \boldsymbol{n}_f \cdot \boldsymbol{V}_X$$

$$\frac{\partial u_N}{\partial t} + \left(\boldsymbol{V}\cdot\nabla\right)u_N = \frac{Du_N}{Dt} \approx \frac{u_N^{n+1} - u_X^n}{\Delta t}$$

$$\frac{d\boldsymbol{x}_P}{dt} = \boldsymbol{V}(\boldsymbol{x}, t)$$



Fractional Step Method (ELM only)

- Coriolis Term approximated as

$$f\boldsymbol{k}\times\boldsymbol{V}\approx\begin{pmatrix} f\left[(1-\theta)fv_X^n+\theta v^{n+1}\right] \\ -f\left[(1-\theta)fu_X^n+\theta u^{n+1}\right]\end{pmatrix}$$

where

f : Coriolis Parameter

θ : Implicit weighting factor

$\boldsymbol{k}$: Unit vector in the vertical direction

$\boldsymbol{V}=(u,v)^T$: Velocity at face

$\boldsymbol{V}_X=(u_X,v_X)^T$: Velocity at face at location X

- First (Coriolis) Step

$$\begin{pmatrix} 1 & \theta\Delta tf \\ \theta\Delta tf & 1\end{pmatrix}\begin{pmatrix} u^* \\ v^*\end{pmatrix}=\begin{pmatrix} u_X^n+(1-\theta)\Delta tfv_X^n \\ v_X^n+(1-\theta)\Delta tfu_X^n\end{pmatrix} \qquad \boldsymbol{V}^*=\begin{pmatrix} u^* \\ v^*\end{pmatrix}$$

- Second Step includes all other terms

Eulerian-Lagrangian Momentum Equation

- Semi-discrete form (2nd Fractional Step)

$$\frac{u_N^{n+1} - u_N^*}{\Delta t} = -g\frac{\partial H^{n+\theta}}{\partial N} + \left[\frac{1}{h}\nabla\cdot\left(\nu_t h^n \nabla u_N\right)\right]_X^n - c_f u_N^{n+1} + \frac{\tau_{s,N}}{\rho h_f^n}$$

where

$$H^{n+\theta} = (1-\theta)H^n + \theta H^{n+1}$$

$$u_N^* = \boldsymbol{V}^* \cdot \boldsymbol{n}_f$$

- Velocity $\boldsymbol{V}^*$ includes Coriolis
- Mixing term is interpolated at backtracking location X and based on previous time step velocity field
- Friction term is semi-implicit

Eulerian Momentum Equation

- Semi-discrete form

$$\frac{u_N^{n+1} - u_N^n}{\Delta t} + \left(V^n \cdot \nabla\right) u_N^n - f u_T^n = -g \frac{\partial H^{n+\theta}}{\partial N} + \left[\frac{1}{h} \nabla \cdot \left(\nu_t h \nabla u_N\right)\right]_f^n - c_f u_N^{n+1} + \frac{\tau_{s,N}}{\rho h_f^n}$$

where

$$H^{n+\theta} = (1-\theta) H^n + \theta H^{n+1} \qquad \overline{h}_f = \alpha_f^L h_L + \alpha_f^R h_R$$

- Coriolis term computed at face f and is explicit
- No fractional step method like ELM solver
- Mixing term is computed at face f and is explicit
- Friction and pressure gradient terms are semi-implicit

Discretization: Eulerian Advection

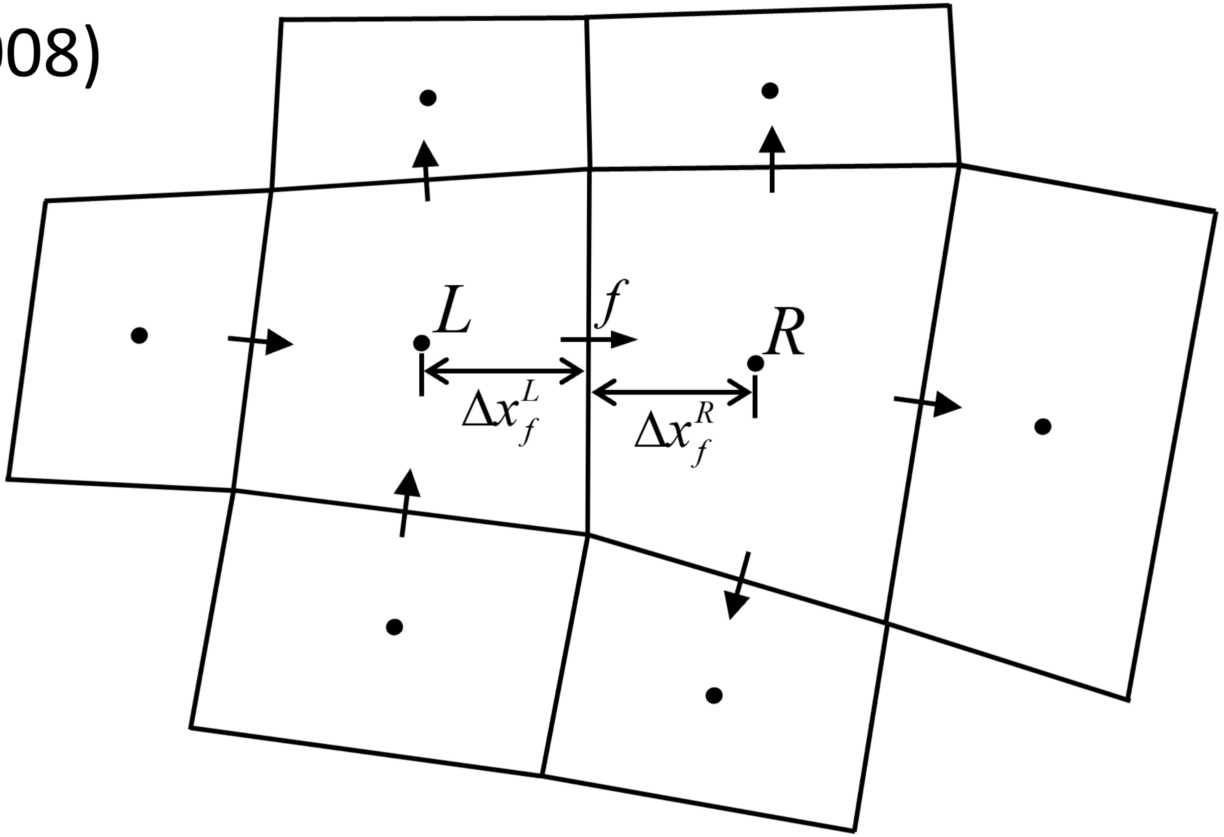
- Approach from Kramer and Stelling (2008)

$$(\boldsymbol{V}\cdot\nabla)u_N \approx \frac{\alpha_f^L}{\overline{h}_f A_L}\sum_{k\in L} s_{Lk}Q_k\left[\boldsymbol{V}_k^u\cdot\boldsymbol{n}_f-(u_N)_f\right]$$

$$+\frac{\alpha_f^R}{\overline{h}_f A_R}\sum_{k\in R} s_{Rk}Q_k\left[\boldsymbol{V}_k^u\cdot\boldsymbol{n}_f-(u_N)_f\right]$$

$$\alpha_f^L=\frac{\Delta x_f^L}{\Delta x_f^L+\Delta x_f^R} \qquad \overline{h}_f=\alpha_f^L h_L+\alpha_f^R h_R$$

$$\alpha_f^R=1-\alpha_f^L$$



- Courant-Freidrichs-Lewy (CFL) Condition

$$C=\frac{U\Delta t}{\Delta x}\le 1$$

Discretization: Mixing Term

- Non-Conservative Form

$$\boldsymbol{v}_t \nabla^2 u_N \approx \boldsymbol{v}_{t,f}^n \left(\nabla^2 \boldsymbol{V} \right)_X^n \cdot \boldsymbol{n}_f$$

- Conservative Form

$$\frac{1}{h} \nabla \cdot \left(\boldsymbol{v}_t h \nabla u_N \right) \approx \frac{\alpha_f^L}{\bar{h}_f A_L} \sum_{k \in L} A_k v_{t,k} \frac{\boldsymbol{n}_f \cdot \left(\boldsymbol{V}_j - \boldsymbol{V}_L \right)}{\Delta x_{L,j}} + \frac{\alpha_f^R}{\bar{h}_f A_R} \sum_{k \in R} A_k v_{t,k} \frac{\boldsymbol{n}_f \cdot \left(\boldsymbol{V}_j - \boldsymbol{V}_R \right)}{\Delta x_{R,j}}$$

- Discretization same for both ELM and EM solvers
- Approximate Stability Criteria for EM solver $\dfrac{v_t \Delta t}{\Delta x^2} \le \dfrac{1}{2}$
- ELM interpolates term to location X

$$\left[\frac{1}{h} \nabla \cdot \left(\boldsymbol{v}_t h \nabla u_N \right) \right]_X^n$$

Eulerian-Lagrangian vs. Eulerian SWE Solvers

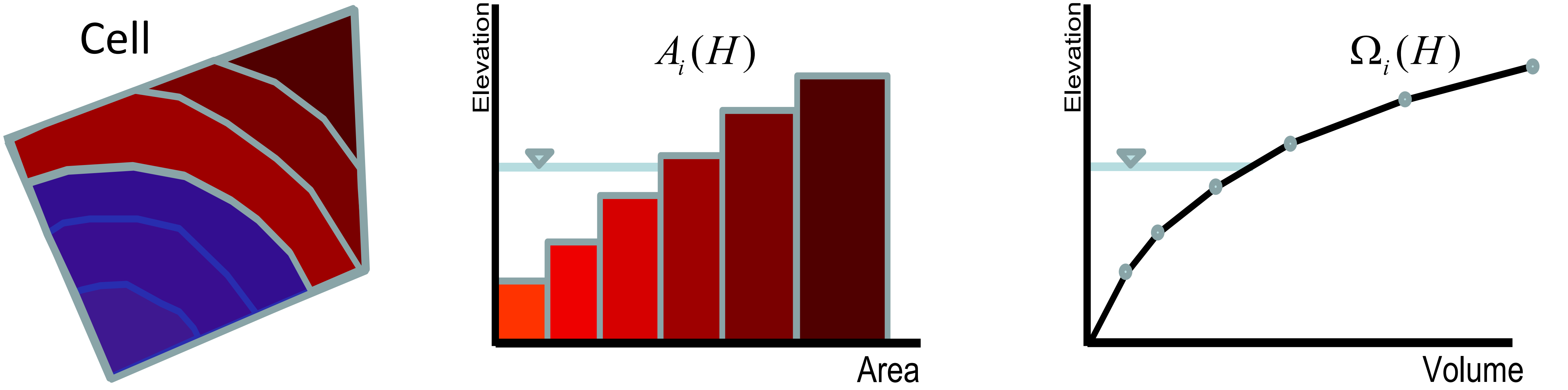
- ELM-SWE
  - Only solver available in V5.0.7 and earlier
  - Default in V6.0
  - Not limited by Courant condition
  - Excellent stability
  - Can have momentum conservation problems around shocks or where the flow changes rapidly
- EM-SWE
  - New to V6.0 as an option
  - Limited to Courant less than 1.0
  - Good Stability
  - Improved momentum conservation for all flow conditions

Strength/Feature/Capability	ELM-SWE	EM-SWE
Larger Time Step	X	
Best Stability	X	
Courant Stability Criteria		X
Diffusion Stability Criteria		X
Computational Speed	X	
Wet/dry > 1 cell per time step	X	
Best Momentum Conservation		X
Non-Conservative Mixing		X
Conservative Mixing	X	X
Wind	X	X

Subgrid Modeling

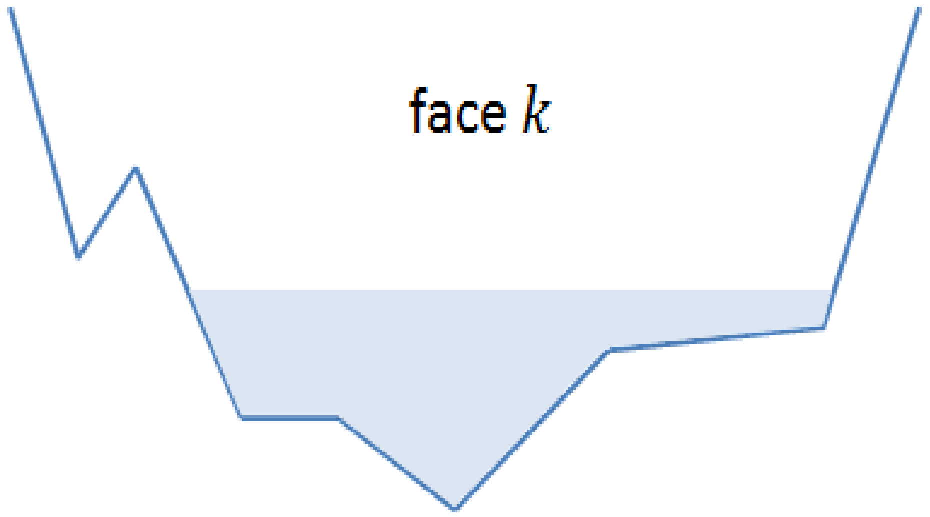
- Problem
  - Water levels usually vary much more smoothly than the terrain
  - Unfeasible to resolve every detail of the terrain with the computational mesh
- Approach
  - Utilize a grid resolution sufficient to resolve the hydraulics
  - Capture the details of the subgrid terrain through hydraulic properties tables

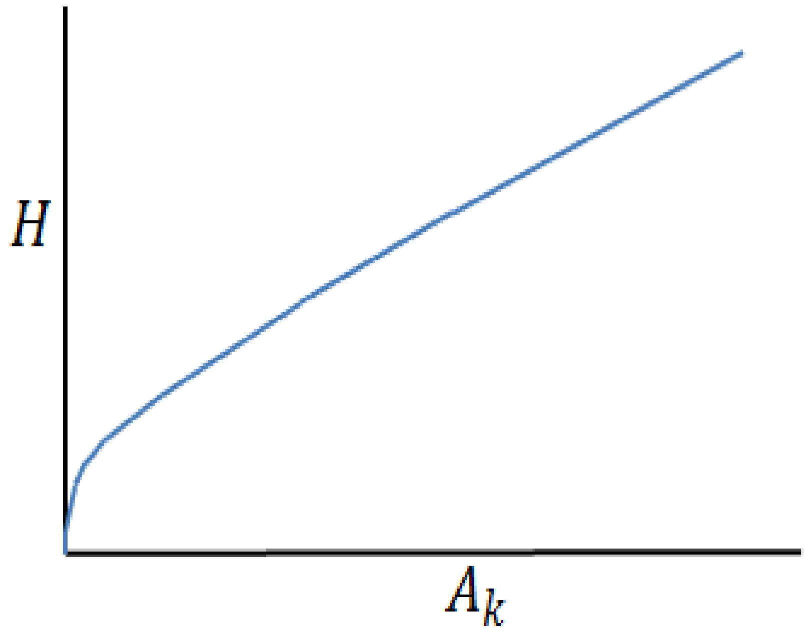
Subgrid Bathymetry: Cells



Subgrid Bathymetry: Faces

- Faces treated similar to cells
- Hydraulic property tables computed
  - Wetted length
  - Wetted Perimeter
  - Area





Benefits of Subgrid Bathymetry

Solution Procedure

For both DWE and SWE:

- The face-normal momentum equation is inserted into the continuity equation to obtain a non-linear and implicit equation for water levels
- For example, in the case of DWE

$$\frac{\Omega_i^{n+1} - \Omega_i^n}{\Delta t} - \sum_{k \in i} \alpha_k \left(\frac{H_j^{n+\theta} - H_i^{n+\theta}}{\Delta x_N} \right) = Q_i \qquad \Omega_i = \Omega_i(H_i)$$

- Implicit weighting

$$H^{n+\theta} = (1-\theta)H^n + \theta H^{n+1}$$

- Once the water levels are computed, the new time step face-normal velocities are computed

Solution Procedure

- System of equations

$$\boldsymbol{\Omega} + \boldsymbol{\Psi}\boldsymbol{H} = \boldsymbol{b}$$

- Algorithm
  1. Compute Right-Hand-Side $\boldsymbol{b}$
     - Contains explicit terms: advection, diffusion, wind, etc.
  2. Outer Loop (Assembly and Updates)
     - Update linearized terms and variables including coefficient matrix $\boldsymbol{\Psi}$
  3. Inner Loop (Newton Iterations)

$$\boldsymbol{H}^{m+1} = \boldsymbol{H}^{m} - \left[\boldsymbol{\Psi} + \boldsymbol{A}^{m}\right]^{-1}\left(\boldsymbol{\Omega}^{m} + \boldsymbol{\Psi}\boldsymbol{H}^{m} - \boldsymbol{b}\right)$$

$\boldsymbol{H}$: Water level

$\boldsymbol{\Omega}$: Water volume

ψ : Coefficient matrix

$\boldsymbol{b}$: Right-hand-side

m : Iteration index

$\boldsymbol{A}$: Diagonal matrix of cell wet surface areas

Boundary Conditions

- **Stage Hydrograph**. Upstream or downstream
- **Flow Hydrograph**. Upstream or downstream. Local conveyance and velocities computed automatically.
- **Normal Depth BC.** At downstream boundaries.
- **Rating Curve BC.**
- **Wind.** Only for shallow-water equations.
- **Precipitation, evapotranspiration, and infiltration.** Included as sources and sinks in the continuity equation.
- 1D reaches and 2D areas can be connected
- Multiple 2D areas can be connected to each other
- 2D areas can be connected to 1D lateral structures such as levees to simulate levee breaches

Computational Implementation

- Multiple 2D areas can be computed independently and simultaneously
- All solvers are can be run on multiple cores
- 2D solvers and parameters can be selected independently for each 2D area
- A partial grid solution keeps track of active portion of mesh and only computes the solution for active portion significantly reducing computational times.

Thank You!

HEC-RAS Website:
https://www.hec.usace.army.mil/software/hec-ras/

Online Documentation:
https://www.hec.usace.army.mil/confluence/rasdocs

US Army Corps of Engineers ®