Advanced Parameters in HEC-RAS

USACE, Institute for Water Resources, Hydrologic Engineering Center

US Army Corps of Engineers ®

Overview

- Turbulence
  - Overview of Theory
  - Numerical Implementation
  - Example
  - Guidance and Best Practices
- Coriolis
  - Overview of Theory
  - Numerical Implementation
  - Guidance and Best Practices
- Matrix Solvers
  - Direct vs. Iterative Solvers
  - Iterative Solver Input Parameters
  - Example
  - Guidance and Best Practices

HEC-RAS Turbulence Modeling

- Turbulence is a type of fluid flow which is unsteady, irregular in space and time, dissipative, and diffusive
- Most natural flows are turbulent
- Dispersion is due to the non-uniform velocity distribution
- Turbulence and dispersion are represented with a diffusion term in the momentum equations
- Two aspects (both changed for Version 6.0)
  1. Formulation in momentum equations
  2. Eddy viscosity formulation (aka "turbulence model")

Momentum Equations: Diffusion Term

- Non-conservative Formulation
  - Only option in Versions 5.0.7 and earlier
  - Still in option in Version 6.0 and later

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}-f_c v=-g\frac{\partial z_s}{\partial x}-\frac{\tau_{b.x}}{\rho R}+\nu_t\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)$$

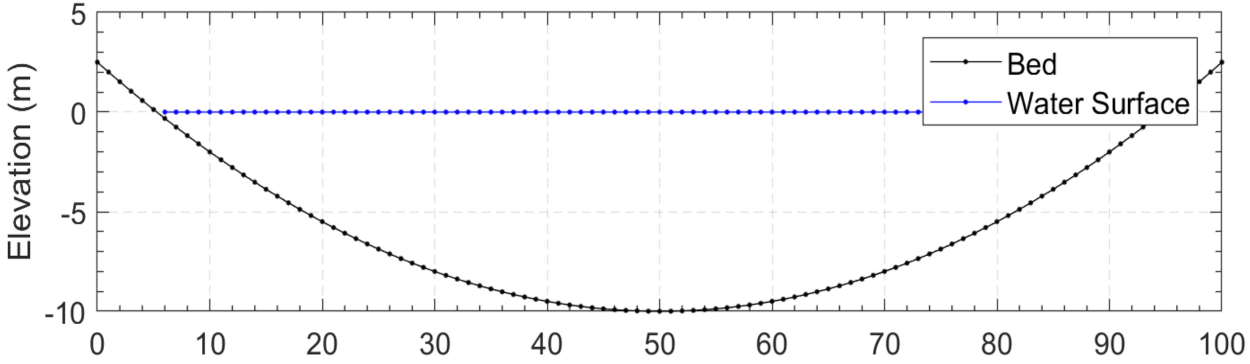
$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}+f_c u=-g\frac{\partial z_s}{\partial y}-\frac{\tau_{b.y}}{\rho R}+\nu_t\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right)$$

- Conservative Formulation
  - Version 6.0 and later

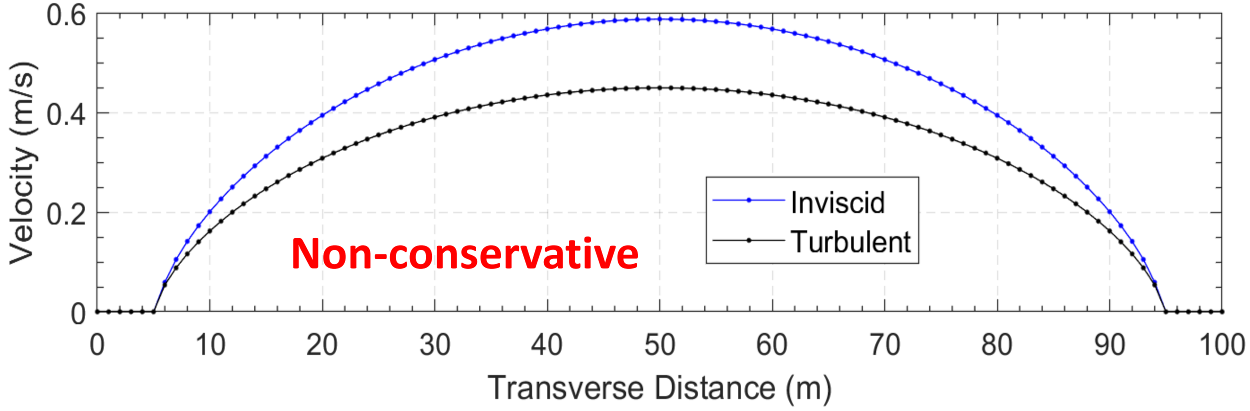
$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}-f_c v=-g\frac{\partial z_s}{\partial x}-\frac{\tau_{b.x}}{\rho R}+\frac{1}{h}\frac{\partial}{\partial x}\left(\nu_{t,xx}h\frac{\partial u}{\partial x}\right)+\frac{1}{h}\frac{\partial}{\partial y}\left(\nu_{t,yy}h\frac{\partial u}{\partial y}\right)$$

$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}+f_c u=-g\frac{\partial z_s}{\partial y}-\frac{\tau_{b.y}}{\rho R}+\frac{1}{h}\frac{\partial}{\partial x}\left(\nu_{t,xx}h\frac{\partial v}{\partial x}\right)+\frac{1}{h}\frac{\partial}{\partial y}\left(\nu_{t,yy}h\frac{\partial v}{\partial y}\right)$$

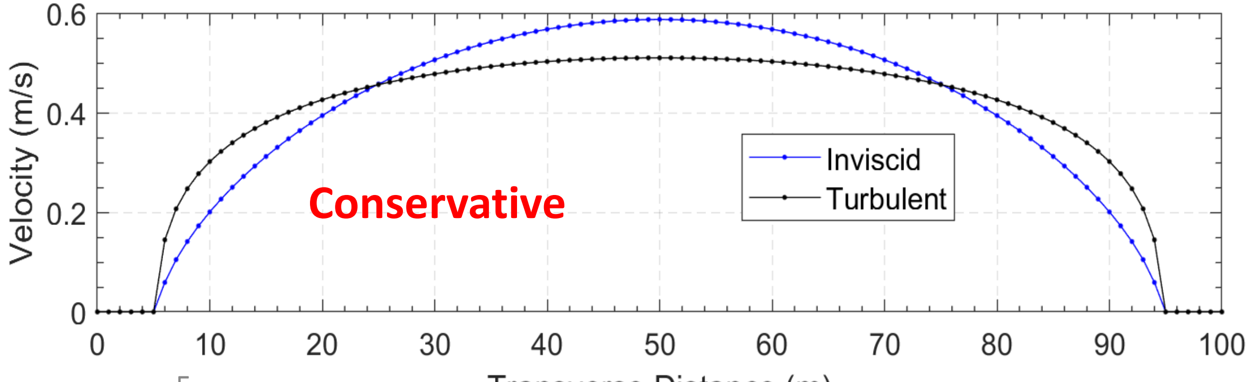
Mixing Term Formulation Comparison



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

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

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

  - 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

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

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

Turbulence Numerical Implementations

- Non-conservative mixing
  - Eulerian-Lagrangian Solver

Eulerian-Lagrangian

$$\frac{u_N^{n+1} - u_N^{*}}{\Delta t} = -g\frac{\partial z_s^{n+\theta}}{\partial N} + v_t^n\left(\Delta u_N^n\right)_X - c_f u_N^{n+1}$$

- Conservative mixing
  - Eulerian-Lagrangian Solver

Lagrangian

$$\frac{u_N^{n+1} - u_N^{*}}{\Delta t} = -g\frac{\partial z_s^{n+\theta}}{\partial N} + \left[\frac{1}{h^n}\nabla\cdot\left(v_t^n h^n \nabla u_N^n\right)\right]_X - c_f u_N^{n+1}$$

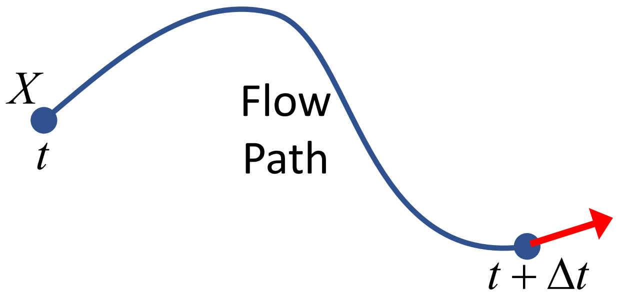
  - Eulerian Solver

Eulerian

$$\frac{u_N^{n+1} - u_N^{n}}{\Delta t} + \left(V\cdot\nabla\right)u_N^n - f u_T^n = -g\frac{\partial z_s^{n+\theta}}{\partial N} + \frac{1}{h^n}\nabla\cdot\left(v_t^n h^n \nabla u_N^n\right) - c_f u_N^{n+1}$$

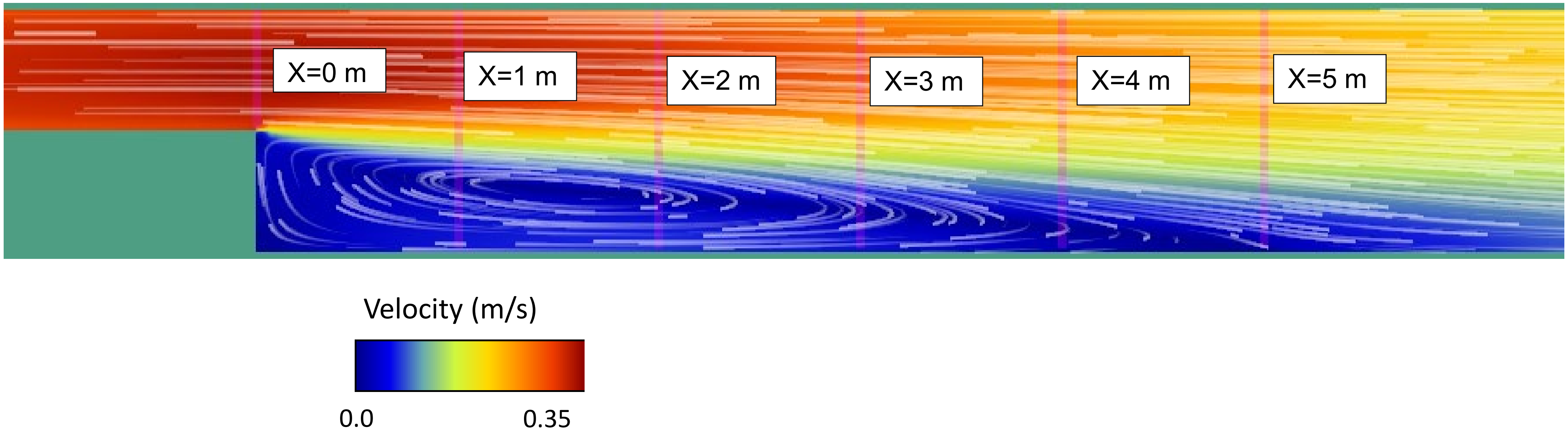
- All mixing terms explicit
- Approximate Stability Criteria for Eulerian Solver

$$\frac{v_t \Delta t}{\Delta x^2} \le \frac{1}{2}$$



Example: Sudden Expansion Lab Experiment

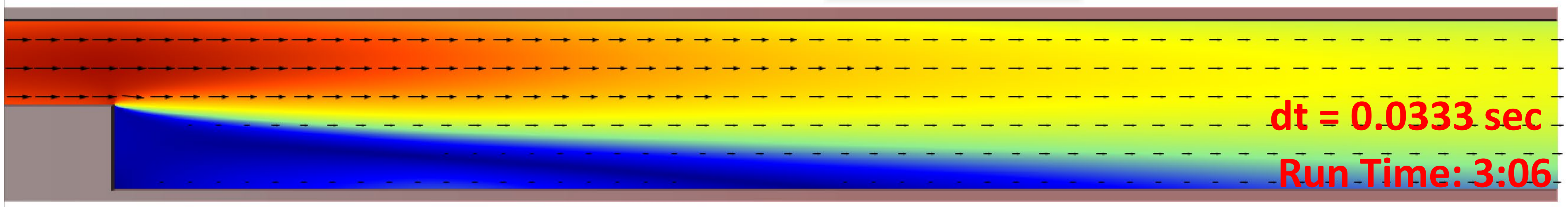
- Inflow: 0.039 m^3/s
- Downstream depth: 11 cm
- Slope: 0.0001
- Grid Resolution: 2.5 cm
- Time step: 0.02 and 0.0333 s
- Manning's n: 0.015 $s/m^{1/3}$



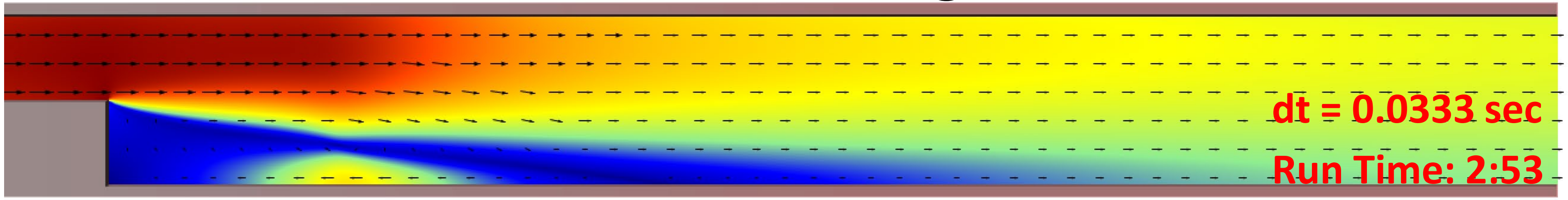
Results: Sudden Expansion Lab Experiment

ELM Solver

Non-conservative Mixing

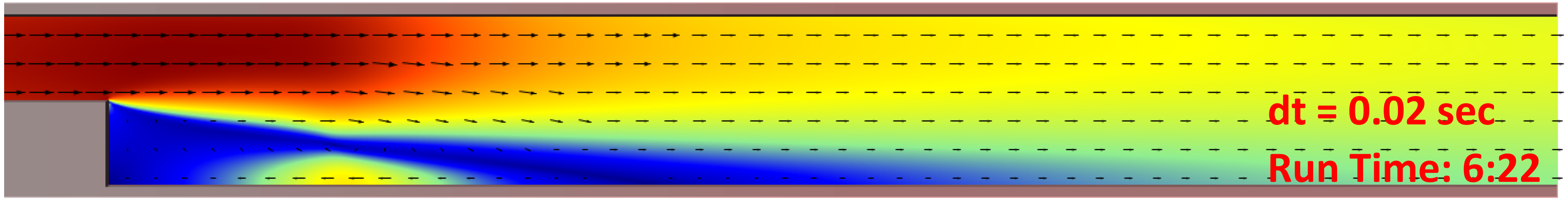


Conservative Mixing



EM Solver

Conservative Mixing



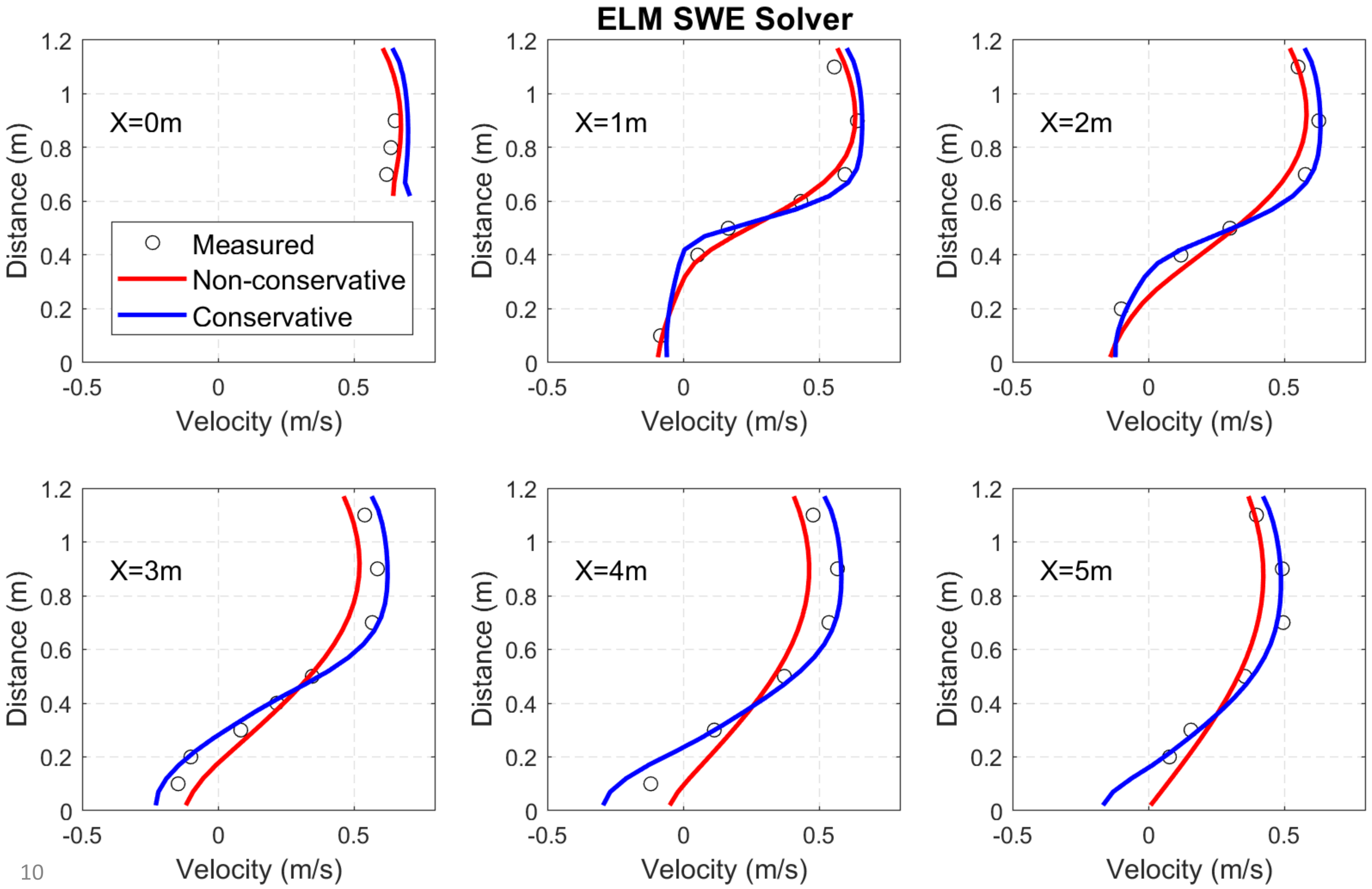
$$D = 1.4$$

$$D_L = 0.6$$

$$D_T = 0.3$$

$$C_s = 0.1$$

Results: Sudden Expansion Lab Experiment



Turbulence Coefficients

- Best to calibrate coefficients with velocity measurements
- Non-conservative formulation generally requires larger values (2x) compared to the conservative formulation
- If coefficients are too large, the non-conservative formulation can produce excess dissipation and increase water levels
- When calibrating, start on the low end of the range

Mixing Intensity	Geometry and Surface	D_L	D_T
Weak	Straight channel Smooth Surface	0.1 to 0.3	0.04 to 0.1
Moderate	Gentle meanders Moderately irregular	0.3 to 1	0.1 to 0.3
Strong	Strong meanders Rough surface	1 to 3	0.3 to 1

Smagorinsky Coefficient: 0.05 to 0.2

Discussion and Conclusions

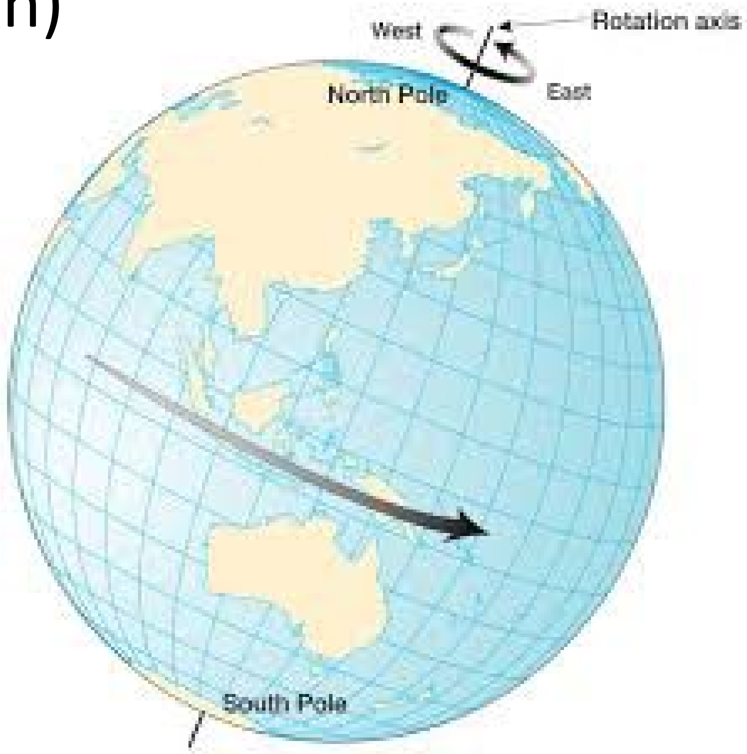
- Non-conservative formulation
  - Does not conservative momentum
  - Will generally lead to a net dissipation of momentum
  - Still an option in Version 6.0 for backwards compatibility
- Conservative formulation
  - Conserves momentum
  - Recommended approach for both EM and ELM solvers
  - Generally requires smaller mixing coefficients than the non-conservative formulation
  - EM Solver has a time step restriction
  - ELM solver does not have a time step restriction
  - EM and ELM solvers should produce similar results for most cases

Coriolis Acceleration

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

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

  - 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



Coriolis Numerical Implementations

- Eulerian-Lagrangian Solver
  - Fractional Step Method (Semi-implicit)

$$\begin{pmatrix} 1 & \theta\Delta t f_c \\ \theta\Delta t f_c & 1 \end{pmatrix}\begin{pmatrix} u^* \\ v^* \end{pmatrix} = \begin{pmatrix} u_X^n + (1-\theta)\Delta t f_c v_X^n \\ v_X^n + (1-\theta)\Delta t f_c u_X^n \end{pmatrix} \quad \boldsymbol{V}^* = \begin{pmatrix} u^* \\ v^* \end{pmatrix}$$

$$\frac{u_N^{n+1} - u_N^*}{\Delta t} = -g\frac{\partial z_s^{n+\theta}}{\partial N} - c_f u_N^{n+1} \qquad u_N^* = \boldsymbol{n}_f \cdot \boldsymbol{V}^*$$

f_c : Coriolis Parameter

θ : Implicit weighting factor

Δt : Time step interval

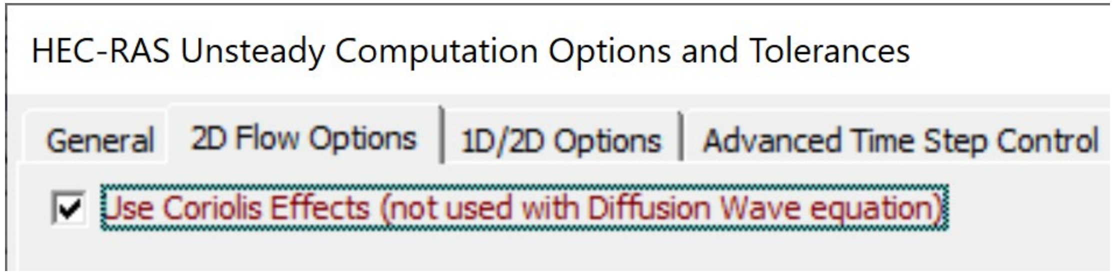
u_X^n : Backtracking velocity

- Eulerian Solver
  - Explicit

$$\frac{u_N^{n+1} - u_N^n}{\Delta t} + (V \cdot \nabla) u_N^n - f_c u_T^n = -g\frac{\partial z_s^{n+\theta}}{\partial N} - c_f u_N^{n+1}$$

Coriolis Discussion and Conclusions

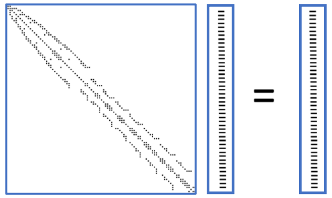
- Only for Shallow Water Equations (Not Diffusion-Wave Equation)
- Relatively inexpensive
- Does not impact stability or introduce time step limitations
- When in doubt, turn it on
- Requires one input parameter: average latitude for each 2D area
- Also requires checking a box



Matrix Solvers: Introduction

- Linear System of Equations

$$Ax = b$$



- Direct Solvers
  - Examples: Gaussian elimination, LU, Cholesky, and QR decompositions
  - Can be “black boxes”
  - Usually have few input parameters
  - High-accuracy
  - Can fail or be very slow for large matrices
  - Can be slow for unsteady or non-linear systems
- Iterative Solvers
  - Examples: GS, SOR, CG, GMRES
  - Require more options and input parameters
  - Usually require preconditioners, matrix balancing, ordering, etc.
  - Less accurate
  - Good for large problems
  - Good for unsteady or non-linear systems
  - Improper use can lead to instability problems or solution divergence

HEC-RAS Direct Solver

- PARDISO

  High-performance, robust, memory efficient and easy-to-use solver for symmetric and non-symmetric linear systems.

  - Version in Intel Math Kernel Library
  - Parallel on PC's
  - Can be used as a "black box"
    - Very little parameters
    - No need matrix balancing, ordering, etc.

HEC-RAS Iterative Solvers

- SOR: Successive Over-Relaxation
  - Relaxation factor ($0 < \omega < 2$)
  - Asynchronous (ASOR) parallel implementation
  - Extremely simple
  - May take many iterations to converge but each iteration is very inexpensive
- FGMRES-SOR: Flexible Generalized Minimal RESidual
  - "Flexible" variant of GMRES which allows preconditioner to vary from iteration to iteration
  - SOR as preconditioner

Iterative Solver Input Parameters

- Convergence Tolerance
  - Determines the overall accuracy
- Minimum Iterations
  - Increases accuracy, avoids solution drift, and allows the solver to stabilize
- Maximum Iterations
  - Avoids stalling and too many iterations caused by a small convergence tolerance
- Restart Iteration (Only FGMRES-SOR)
  - Reduces run time and memory requirements
- Relaxation Factor
  - Used for both SOR solver and preconditioner
- SOR Preconditioner Iterations (Only FGMRES-SOR)
  - In-lieu of checking convergence which would be slower

Parameter	Range
Convergence Tolerance	0.001 – 0.000001
Minimum Iterations	3 – 6
Maximum Iterations	5 – 30
Restart Iteration (FGMRES Only)	8 – 12
Relaxation Factor	1.1 – 1.5
SOR Preconditioner Iterations (FGMRES Only)	5 – 20

Iterative Solvers: Stopping Criteria

- Error Estimate

$$E^m = \frac{\left\| \boldsymbol{D}^{-1}\left(\boldsymbol{A}\boldsymbol{x}^m - \boldsymbol{b}\right)\right\|_2}{\sqrt{N}}$$

$\boldsymbol{D}$: Diagonal of $\boldsymbol{A}$

$\boldsymbol{A}$: Coefficient matrix

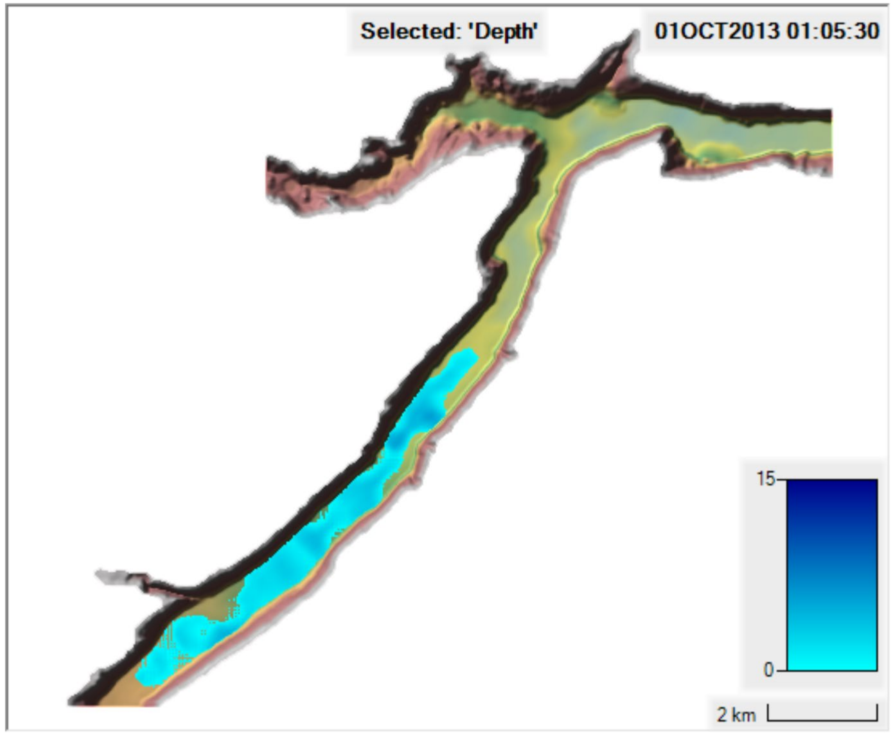
$\boldsymbol{x}$: Solution

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

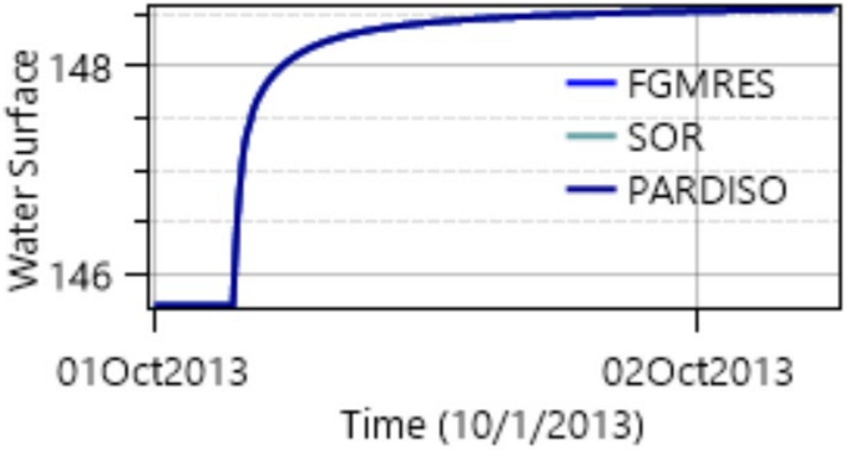
N : Number of rows

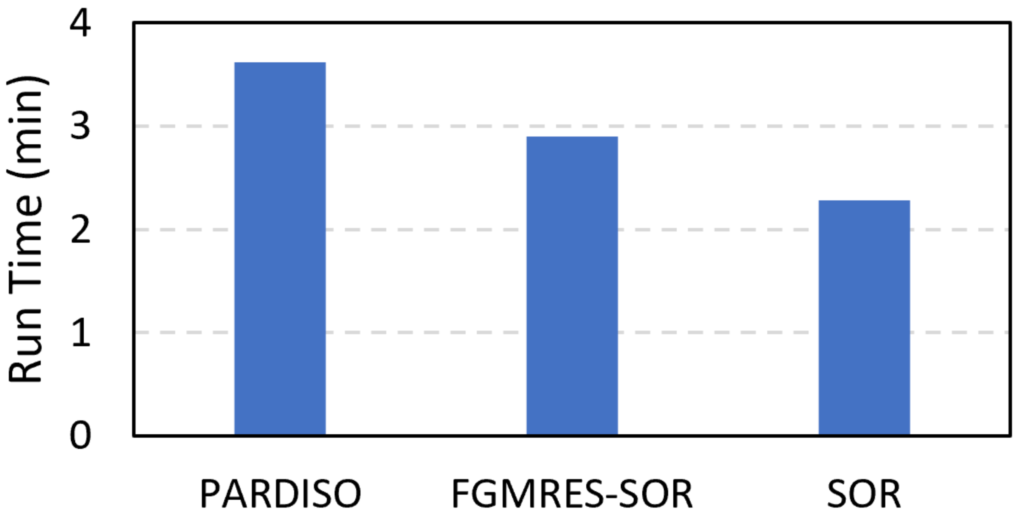
Iterative Solver Status	Criteria	Description
Iterating	$N_{\min} < m < N_{\max}$ and $E^m > T_C$ and $\frac{E^{m-1} - E^m}{E^1 - E^m} > T_S$ and $E^m < E^1$	Iterative solver is converging and will continue to iterate. N_{min}: Minimum number of iterations N_{max}: Maximum number of iterations T_C : Convergence tolerance T_S = $0.1T_C$: Stalling tolerance
Converged	$E^m \le T_C$	Convergence criteria met. Solution accepted.
Stalled	$\frac{E^{m-1} - E^m}{E^1 - E^m} \le T_S$	Convergence rate has decreased to an insignificant level without satisfying the converged criteria. The current solution is accepted and the iteration loop is exited.
Max Iterations	$m = N_{\max}$	Maximum number of iterations reached without reaching the converged criteria. The current solution is accepted and the iteration loop is exited.
Divergent	$E^m > E^1$	Iterative solution is divergent. Either the normalized residuals are getting larger, or a Not a Number (NaN) has been detected.

Example: EA Test 5 (Dam Failure)



Setting	FGMRES-SOR	SOR
Convergence Tolerance	0.0001	0.0001
Minimum Iterations	3	5
Maximum Iterations	20	30
Restart Iteration	10	
Relaxation Factor	1.3	1.3
SOR Preconditioner Iterations	10	





Matrix Solver Best Practices

- Start with PARDISO
- Ensure model is stable and not going to max iterations every time step or reporting large water surface or volume errors
- Try FGMRES and SOR solvers
- Start with conservative parameters
- Compare with PARDISO
- Adjust parameters to optimize run time
- Iteration parameters left empty or set to zero are assigned a default value based on mesh size

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 ®