

# Grid generation for SCHISM with SMS

Eli Ateljevich, DWR  
Joseph Zhang, VIMS

# The good news and bad news

## ❑ Bad news

- UG grid generation is often a laborious and iterative process
- Steepest part of the learning curve for UG models
- This is especially true for very large grids: a common source of frustration with SMS – build confidence gradually

## ❑ Good news

- SCHISM is not picky about grid quality (for triangles); at least won't blow up easily
- Once you master the G.G., the rest of the model setup is much more straightforward

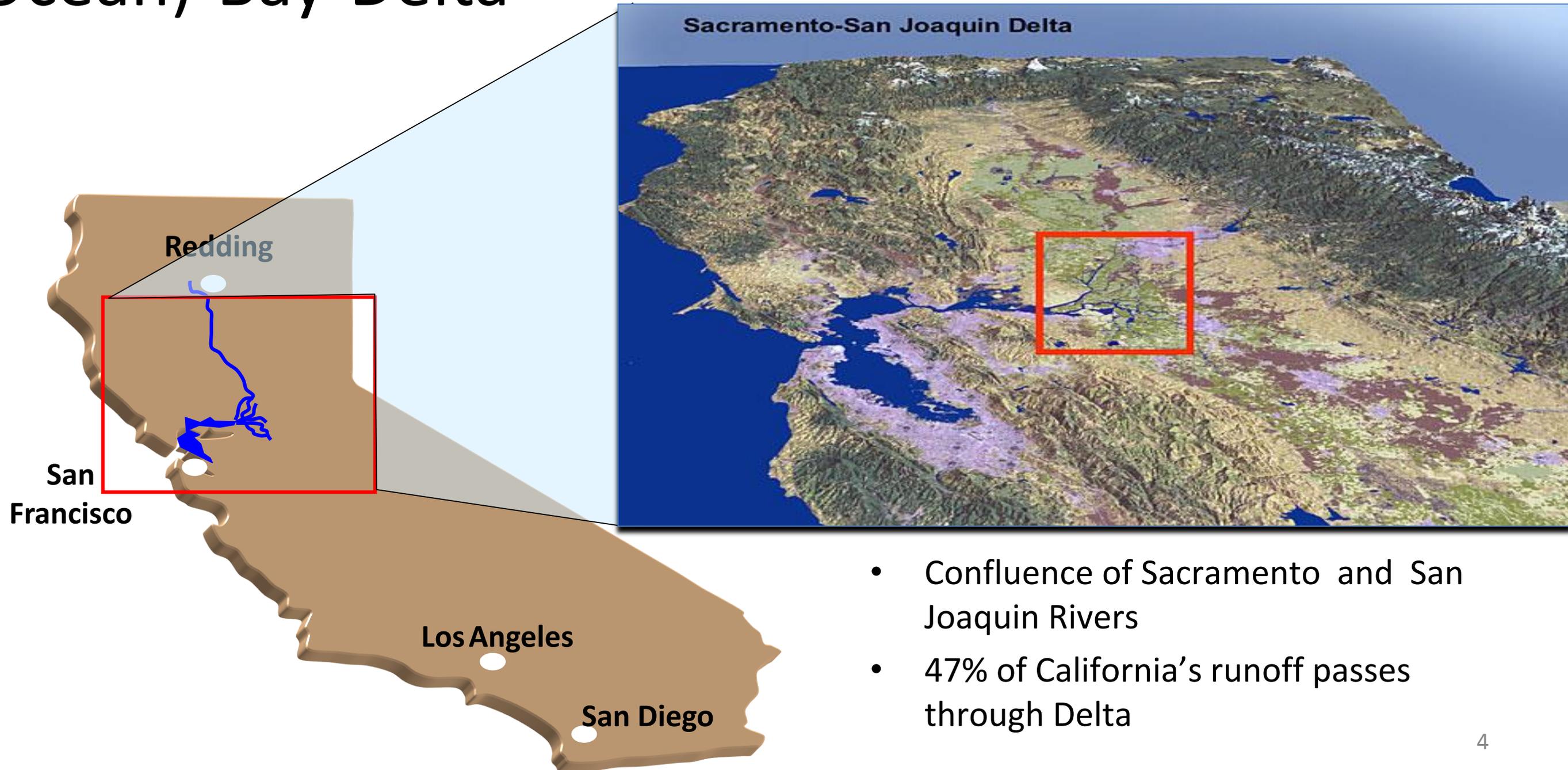
## ❑ A few ways to blow up the model

- Bad quality quads: split them to get pair of triangles
- Completely blocking the water flow, e.g., by under-resolving the main channel
- Dry open boundary: a few ways around this
- Momentum dissipation too low (parameter choice)

# Overview of The GG process

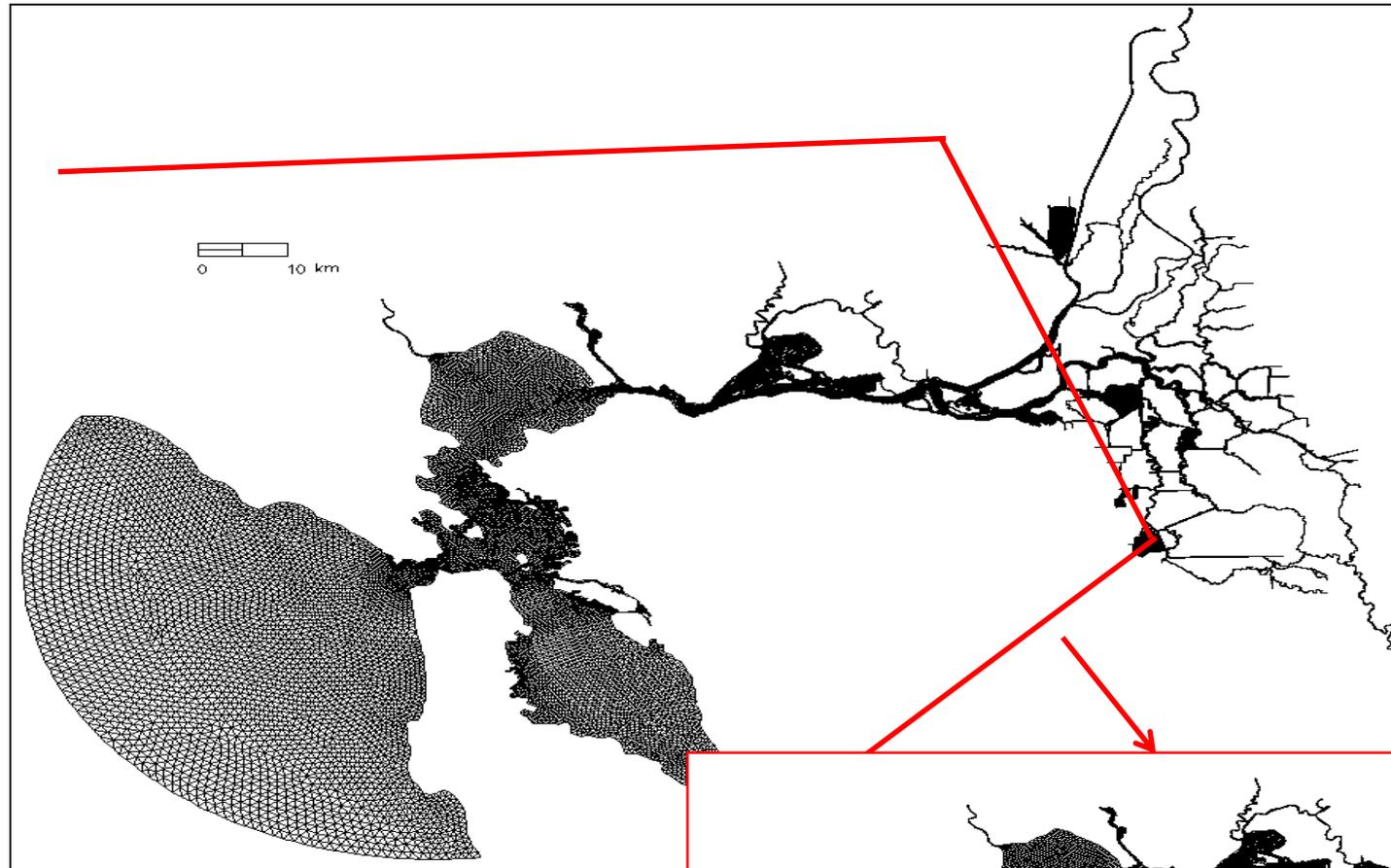
- Know your domain and target physics
- Understand SCHISM numerics
  - **Usually it's easy to get 'reasonable' results without much effort**
  - **You cannot take full advantage of SCHISM features unless you understand its physics and numerics**
- Elevation and contour data preparation
- Digitize or determine boundary (ideally as shape files from ArcGIS)
- Import and refine conceptual model
  - Identify critical features/contours
  - Identify resolution/density
  - Create polygons and patch/pave locations
  - Generate the mesh
  - Populate mesh elevations (depths): *no smoothing*
- Mesh quality and bathymetric metrics: for eddying regime
- Performance and accuracy metrics
- Clip mesh as necessary for subdomains of interest
- Concatenate meshes

# (Ocean)-Bay-Delta

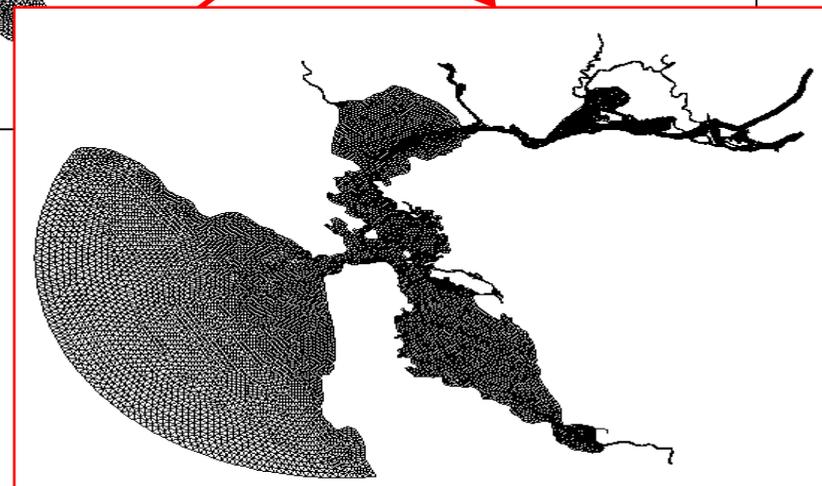


- Confluence of Sacramento and San Joaquin Rivers
- 47% of California's runoff passes through Delta

# SCHISM Bay-Delta and Bay Subdomain

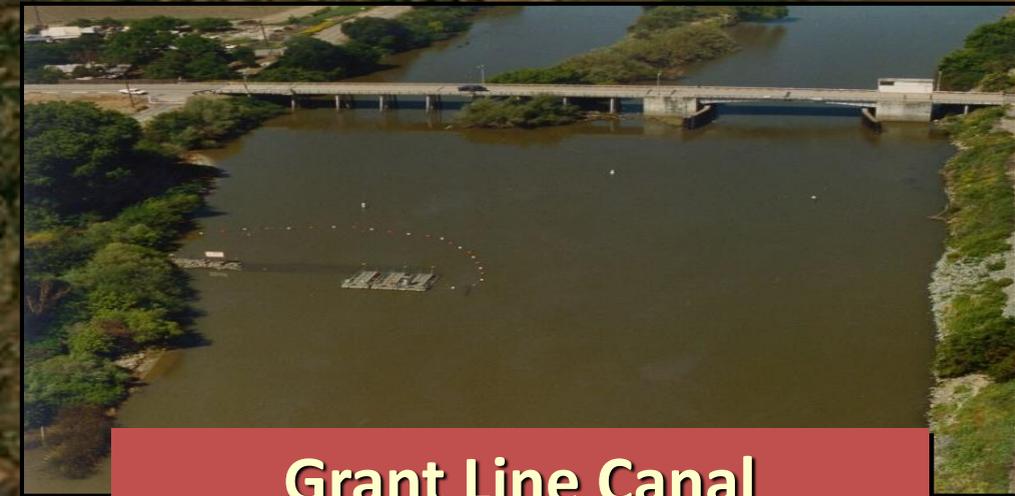


Bay-Delta SCHISM:  
~180,000 nodes  
~360,000 elements

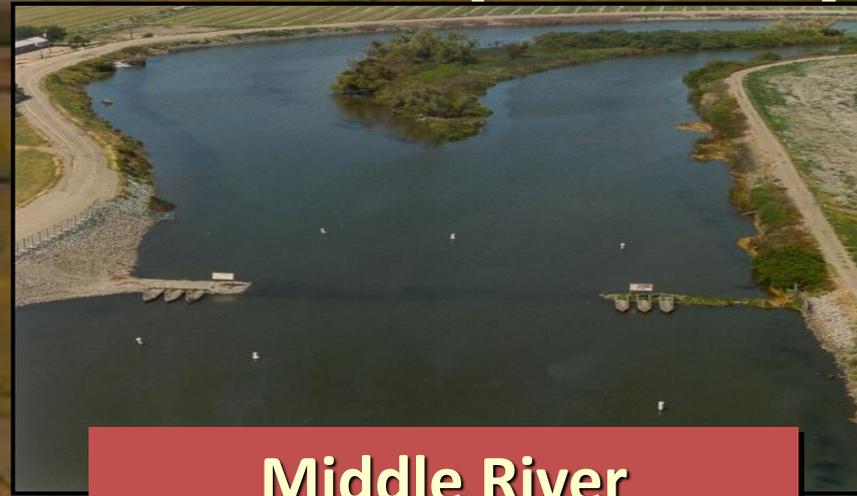


Bay SCHISM

# South Delta Temporary Barriers



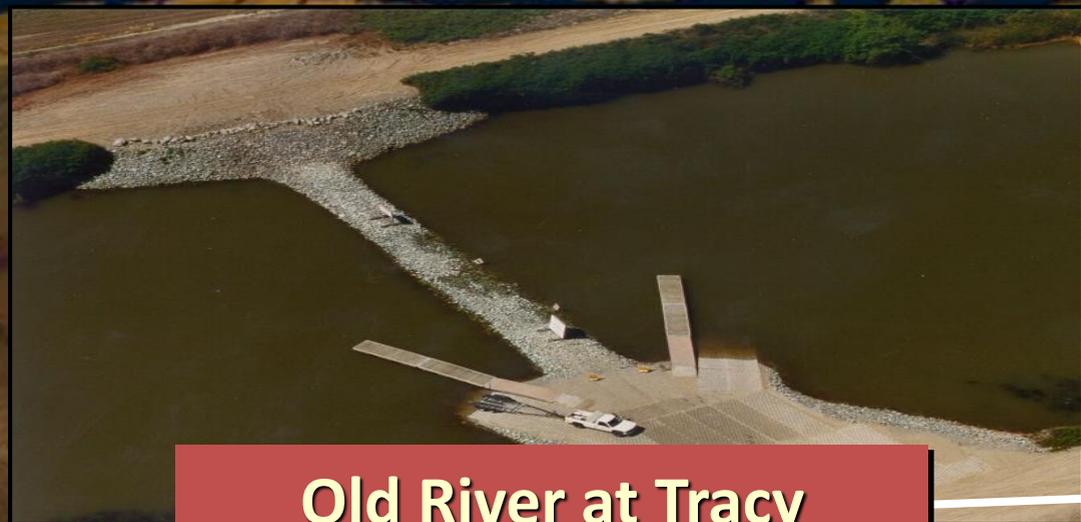
**Grant Line Canal**



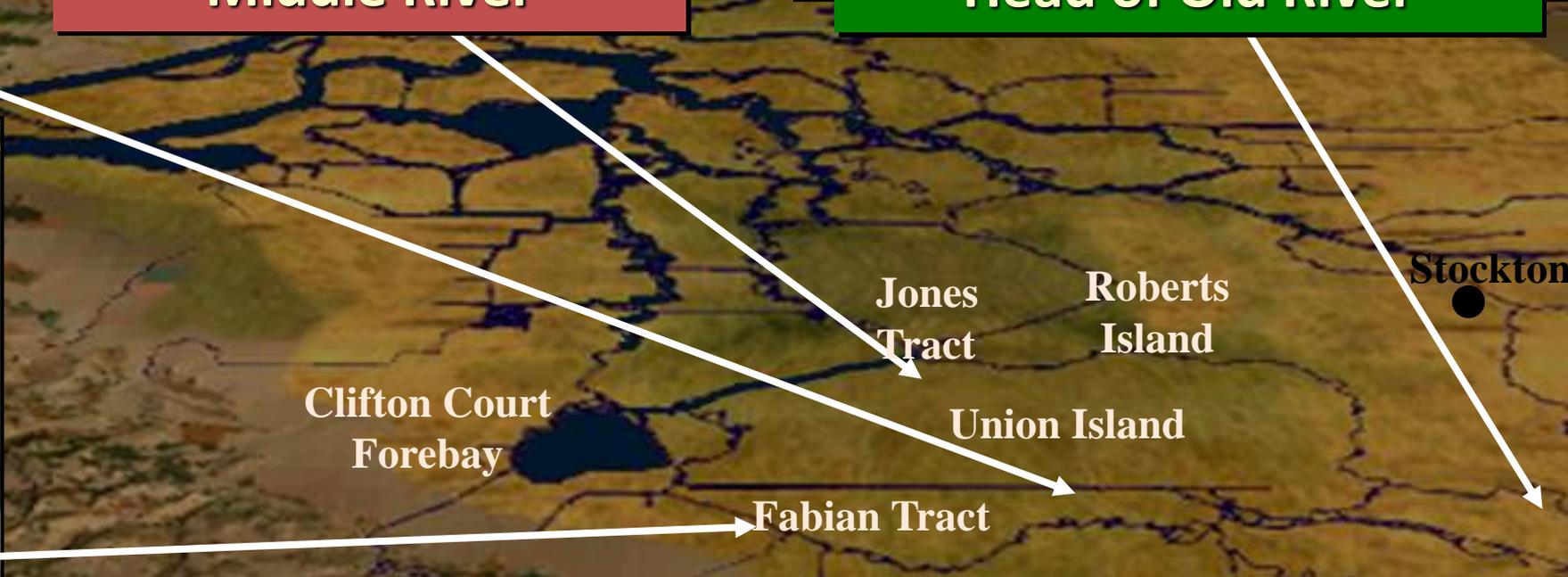
**Middle River**



**Head of Old River**



**Old River at Tracy**

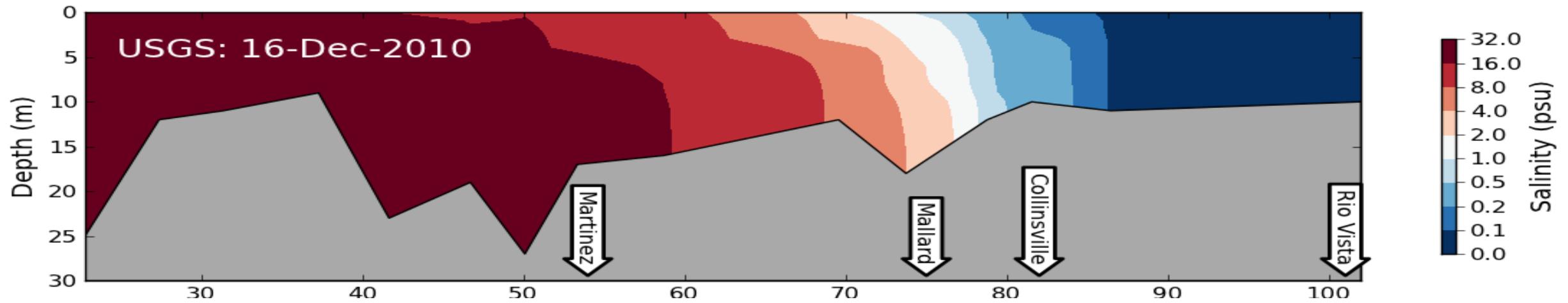
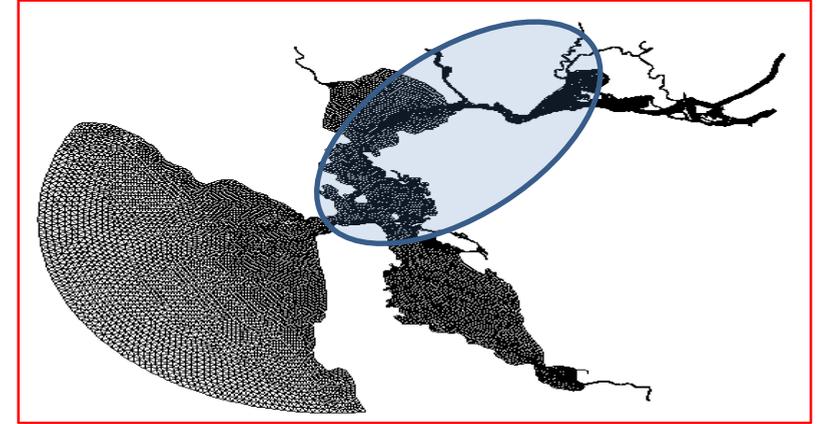


# Delta

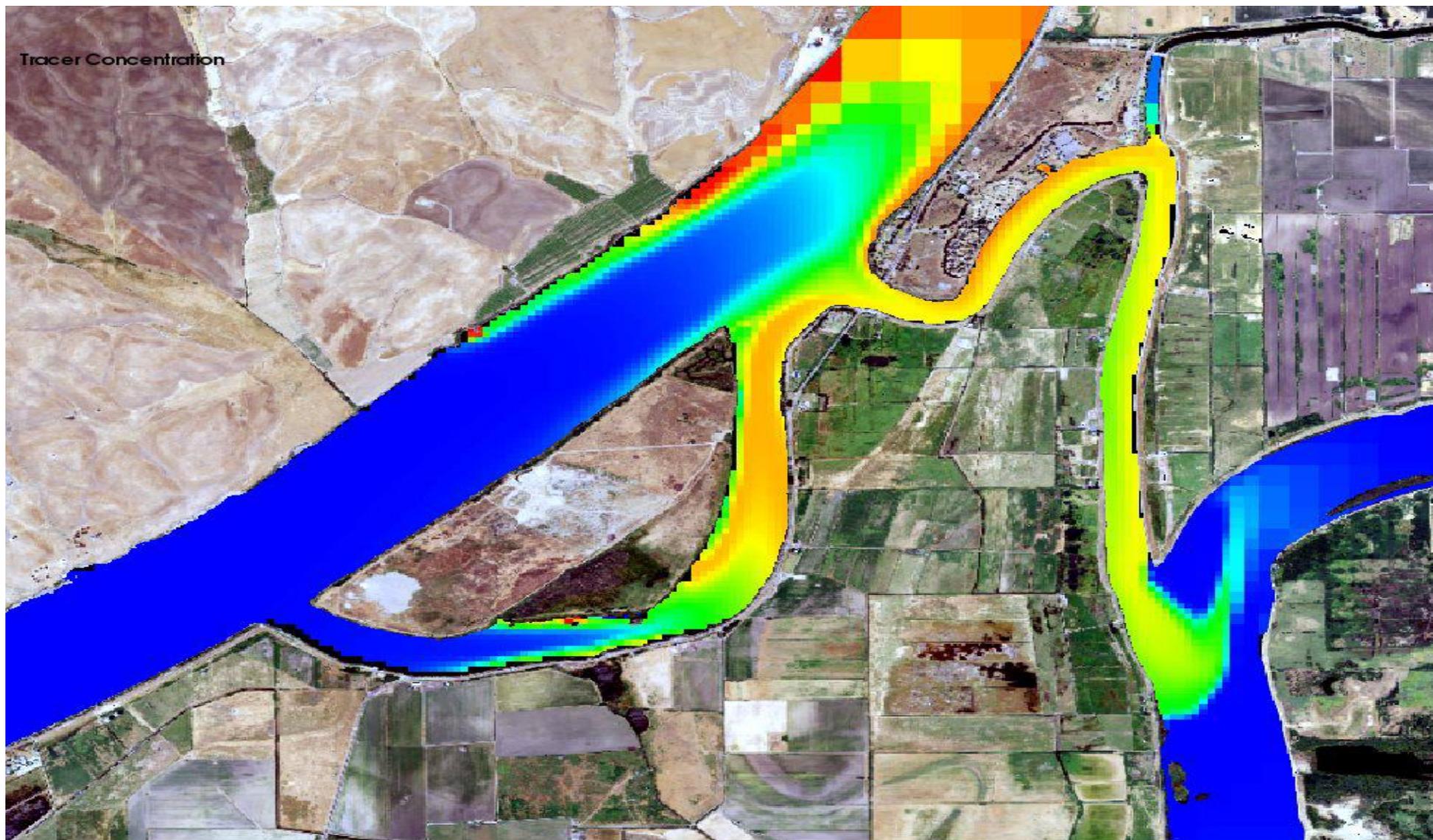
- Channelized, levees
- Levees and “islands”
- 3D challenges beyond estuary processes
- Extensive tidal influence



# Estuary Transport (3D)



# Shear Dispersion



From REALM,  
An adaptive  
mesh code 9

# Tools

- Recommend SMS using “conceptual maps”
  - Good advantage of terrain
  - Patch/pave comparison
  - Bugs, high level of manual work
- Triangle is a good free product
  - Driven mostly by exterior (2D shore)
- Janet: expensive, orthogonality not needed
- DistMesh: Beautiful introduction to density functions, curvature. Not all the good stuff is in the software.
- STOMEL (Holleman et al 2013): Flow-aligned mesh and numerical diffusion, but orthogonal
- GMESH, JIGSAW...

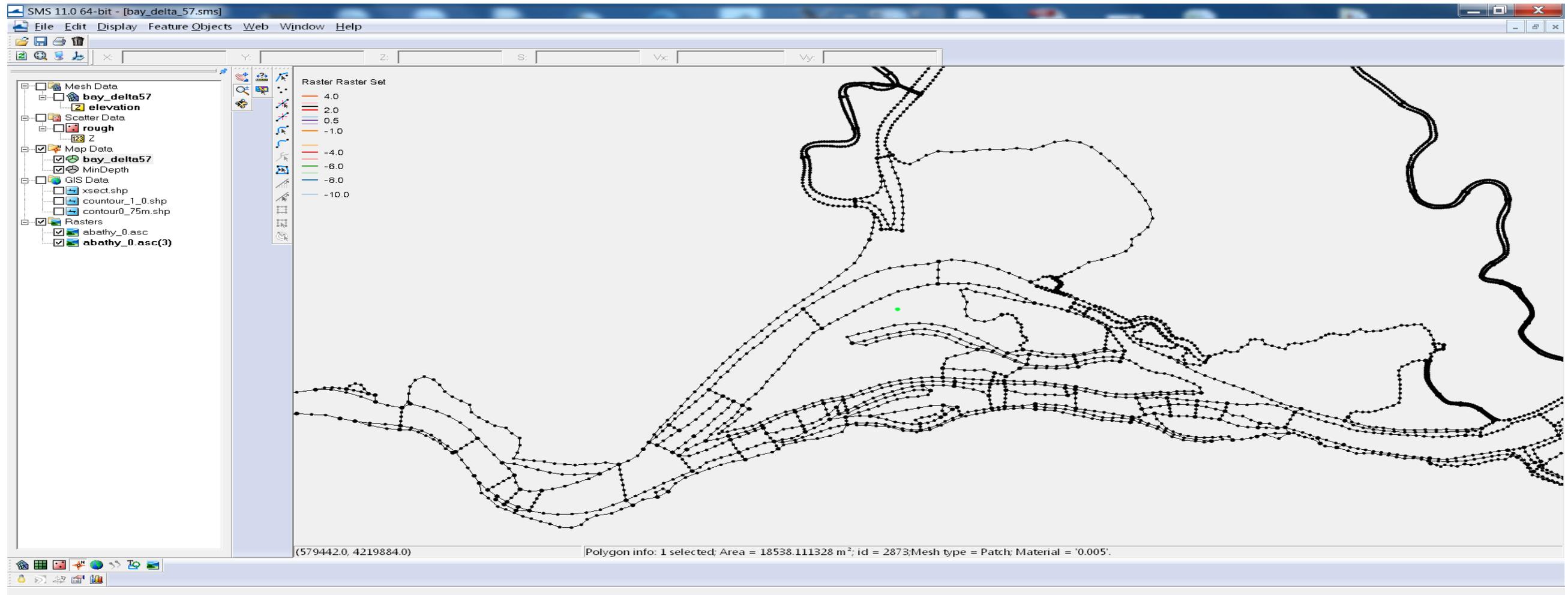
# SMS Resources

- SMS/Aquaveo website
- Todd Wood Masters Thesis:
  - Can be viewed from wiki ([www.schism.wiki](http://www.schism.wiki))
  - Very good step-by-step
  - Honest account of debugging such as “bisection” to find bad spots

# General rules

- Unlike explicit models, you'll find G.G. for SCHISM is more 'intuitive' and 'freer'
  - Implicit model allows you to focus on physics instead of numerics (CFL...)
  - You are freer to resolve important features without worrying about cost/instability
  - SCHISM is not picky about grid quality (*except for quads*); however, grid quality pays off for accuracy especially in some critical regions
  - G.G. for baroclinic applications requires more effort (often an iterative process): **establishing a good workflow is essential**
  - **Grid needs to be smooth in eddying and transitional regimes, in order to not distort eddy kinetics (Wang and Danilov 2016)**

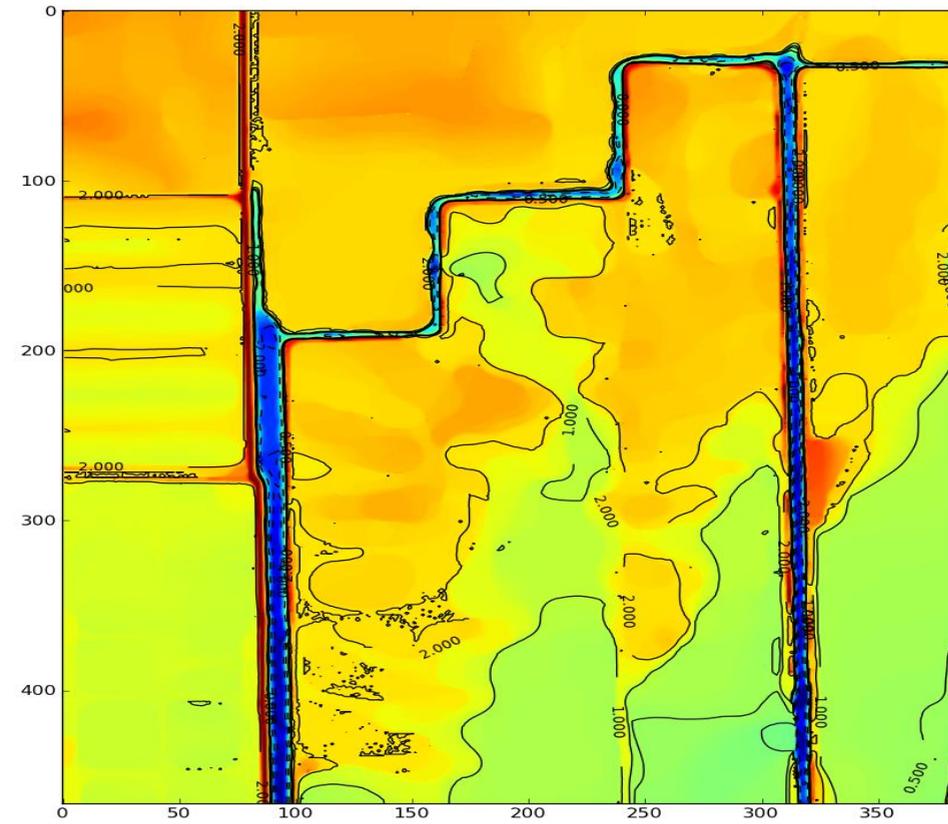
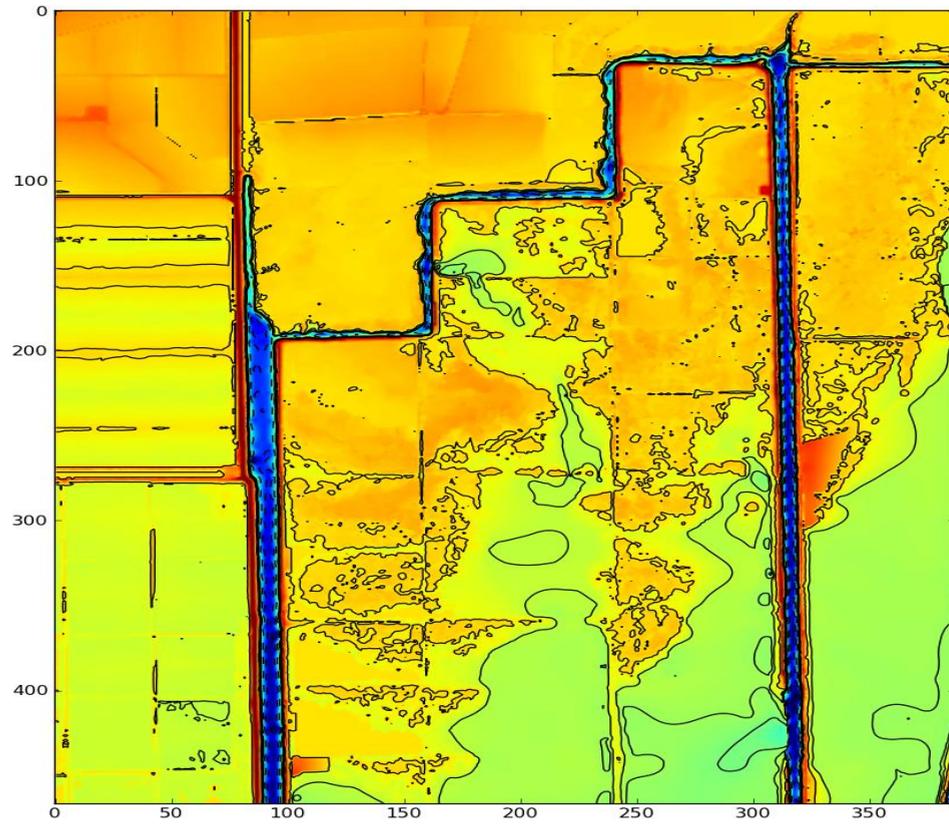
# SMS Conceptual Model



- Nodes, vertices, arcs and polygons

# DEM preparation: pre-filtering bathymetry

- Unresolved subgrid features may be eliminated
  - eliminates lunar landscape undulations on mudflats
  - level set or active contour methods (e.g. Malladi and Sethian): contours conservatively straighten
  - California DWR has python code `smooth_contour.py` for small regions
  - This is mainly to assist in the arc creation, less for actual interpolation of depths

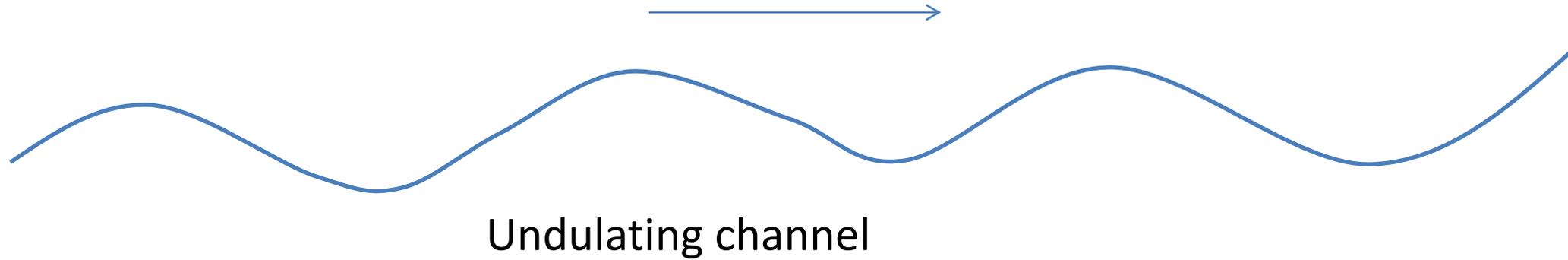


# Liberty Island Smoothing:

Malladi and Sethian  
Min-max curvature flow



# Wetting and Drying in the Frontal Direction is Costly (side shores are fine)



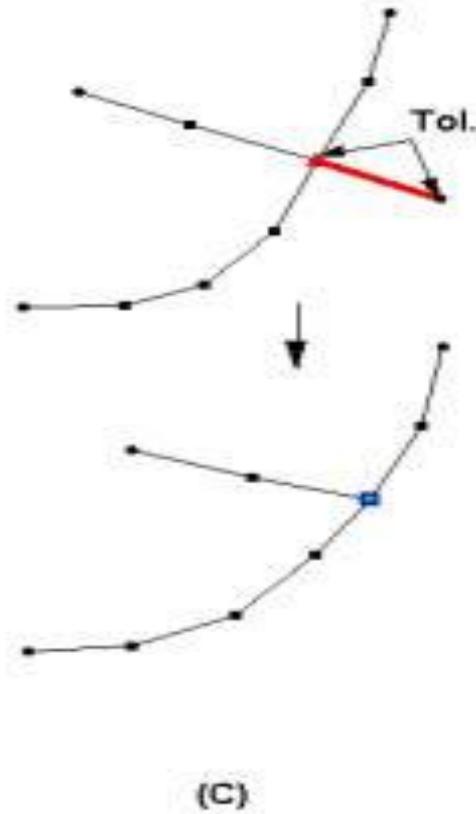
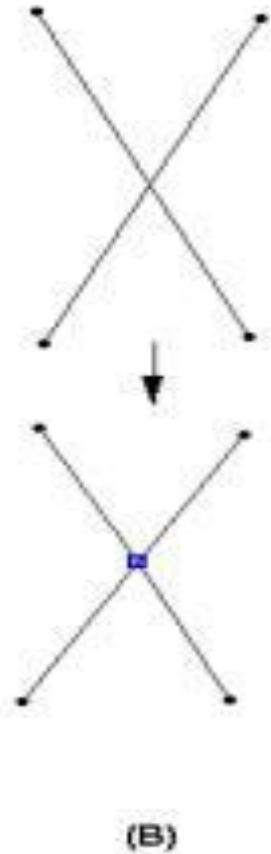
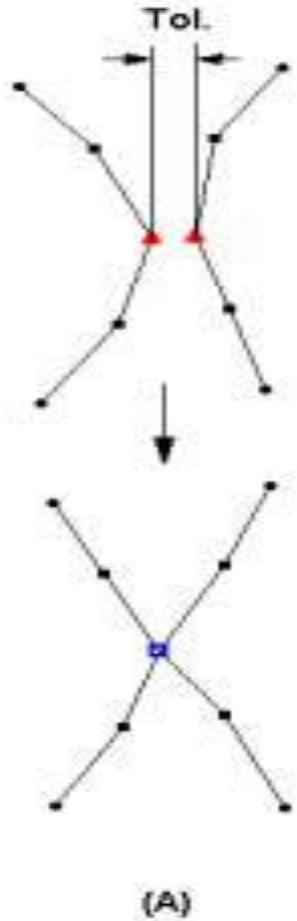
# Generating Contour Shapefiles

- ArcGIS allows graphical cleaning
- GDAL (gdal\_contour) is free and good for quick manipulation on-the-fly
- SMS sister product WMS has some contour tools
- All use bilinear interpolation
- Note that SMS < 11.1 has a georeferencing mistake for contours (cell- vs vertex-centered)
- Python shapefile library very useful for exporting arcs based on calculations in (x,y)

# Visualizing Contours

- SMS has very limited memory
- You can use a coarse tin or DEM for conceptual visualization or on small domains
- `clip_dem.py` allows tailored geographical information

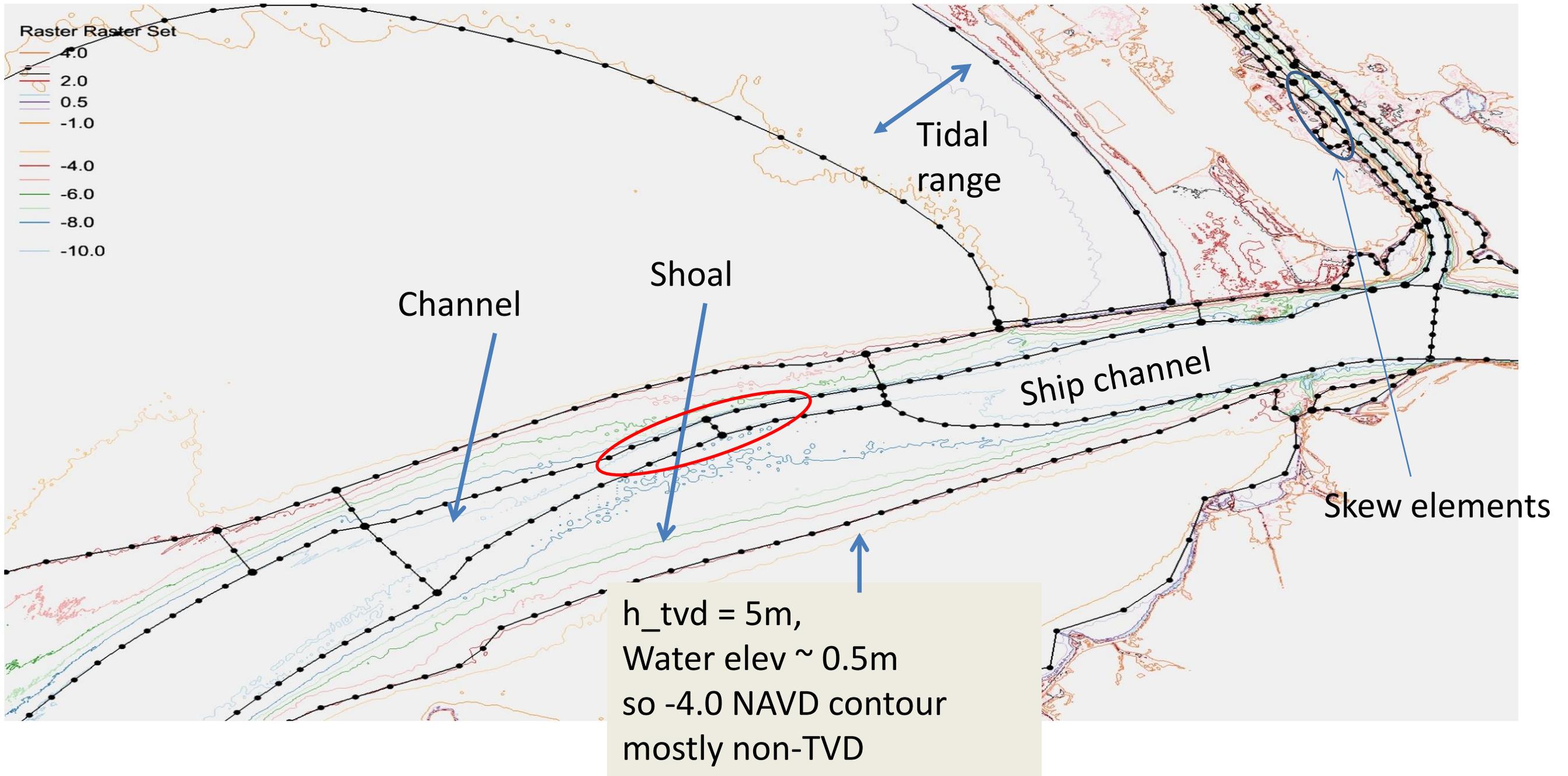
# Cleaning



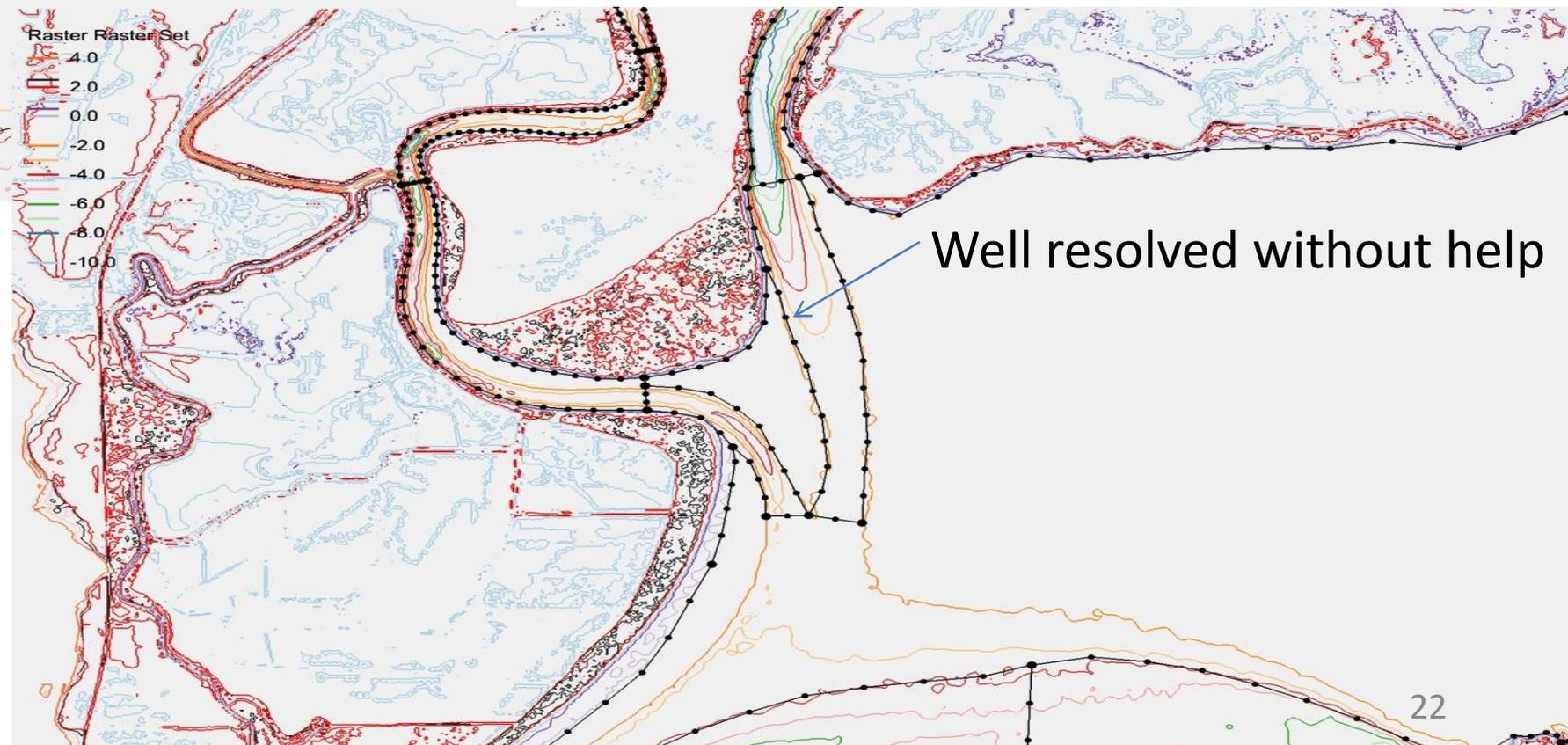
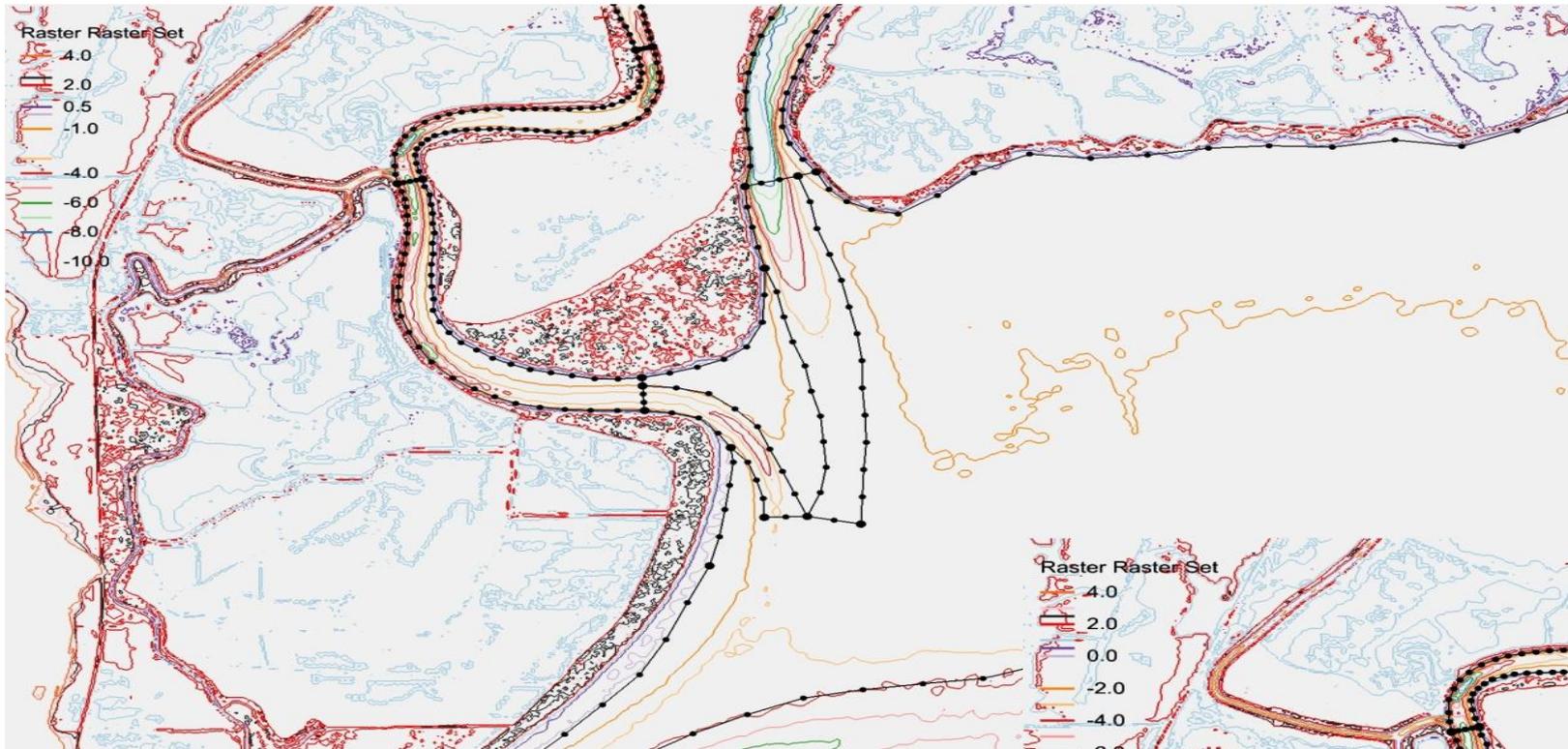
- Snapping vertices
- Intersecting arcs
- Dangling arcs
- Mixed feelings about them all

# Magic Contours

- Foot/top of slope, thalweg
- High-gradient zone in DEM for ‘features’
- Shoreline and probable “real” water levels
- Flooding and mudflats
- Key features (jetty, breakwater...)
- TVD (h\_tvd) and other algorithm switches
- Depths used in friction and other user threshold choices

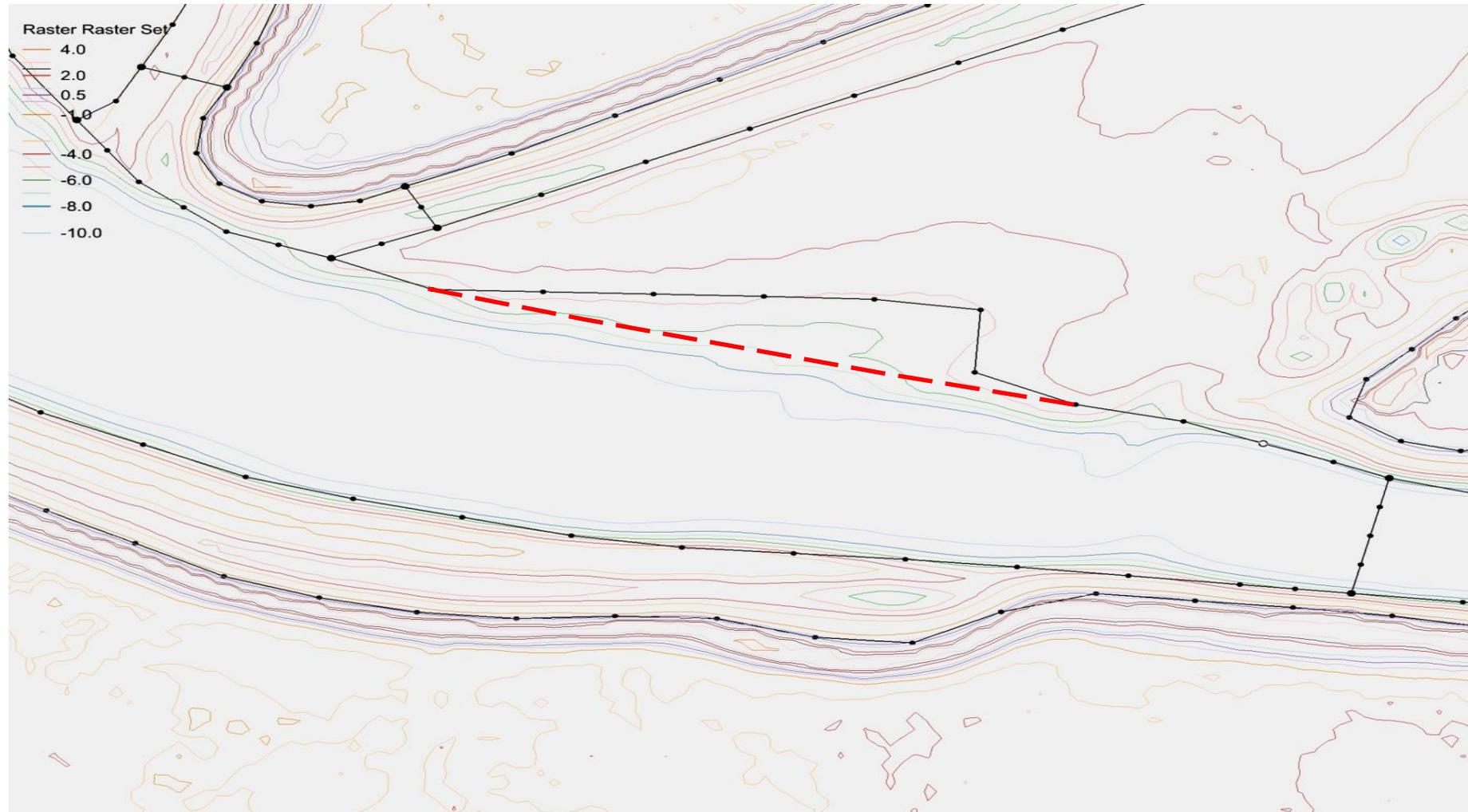


# Choose Connected Contours

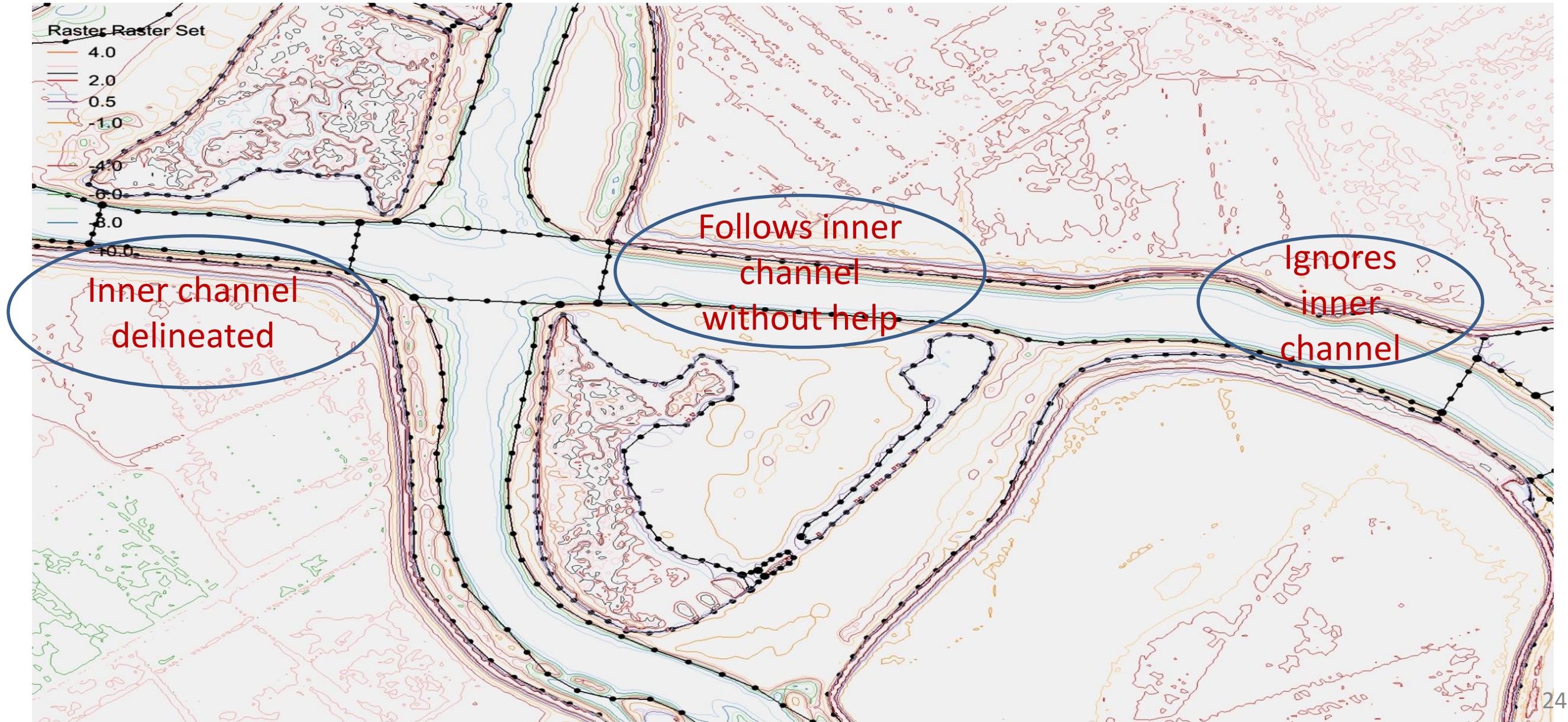


Changing from -2 to -3m  
increases connectivity  
No need to be religious about  
following contours exactly

# Choose Smoother Deep Contours



# Stick with the center



# Grid Density

- Need guidance from inverse CFL>0.4
  - Start from a **smallest** expected  $\Delta t$  (e.g., 100s)
  - Use CFL>0.4 to back calculate the **coarsest**  $\Delta x$  at a given depth
- Deeper regions generally coarser than shallows but not always (compare this with explicit models which are constrained by CFL)
  - Often channel needs to be resolved for tracer transport
- Curvature should be resolved:
  - 2D: bend in shorelines
  - 3D: changes in slope (**vertical grid plays a role**)
- Width of features (sills, beaches, channels)
  - Skeletonization algorithms (see Per-Olof Persson dissertation)

# Grid resolution Guide

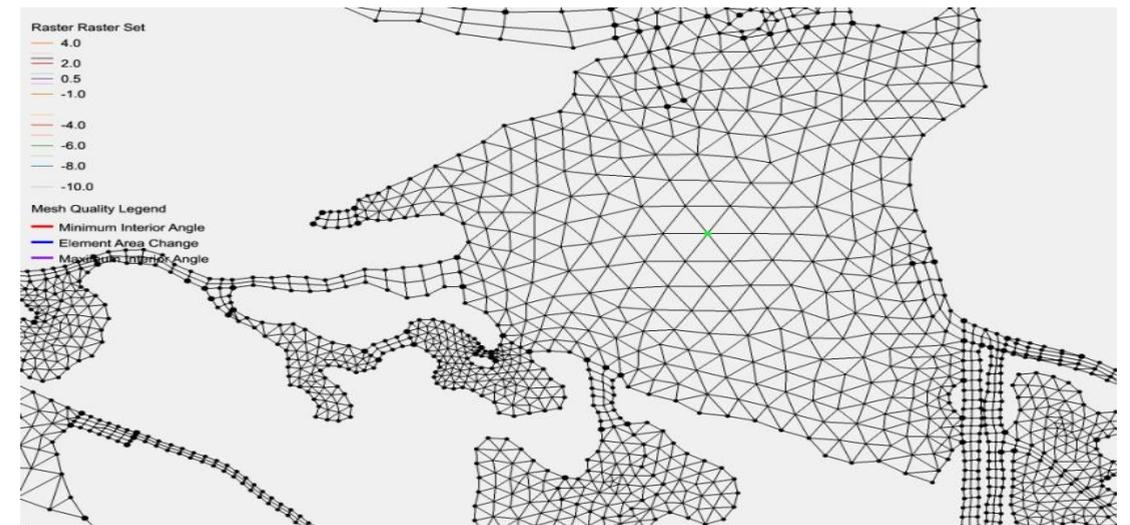
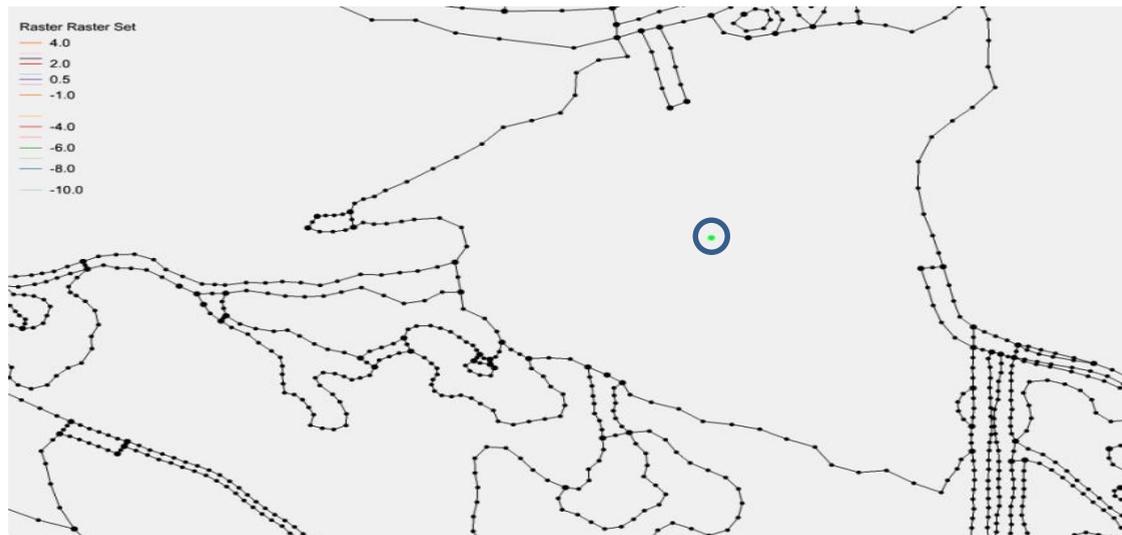
$\Delta t=100s$  (expected minimum\*)

$h$ (m)	$\Delta x_{\max}$ (m)
$\leq 1$	790
10	2500
50	5.5e3
100	7.9e3
500	1.7e4
1000	2.5e4
4000	5e4

- \* Remember to refine grid if for some reason you have to reduce  $\Delta t$
- \* Small patches of violation is fine especially near shoreline; avoid it in open area

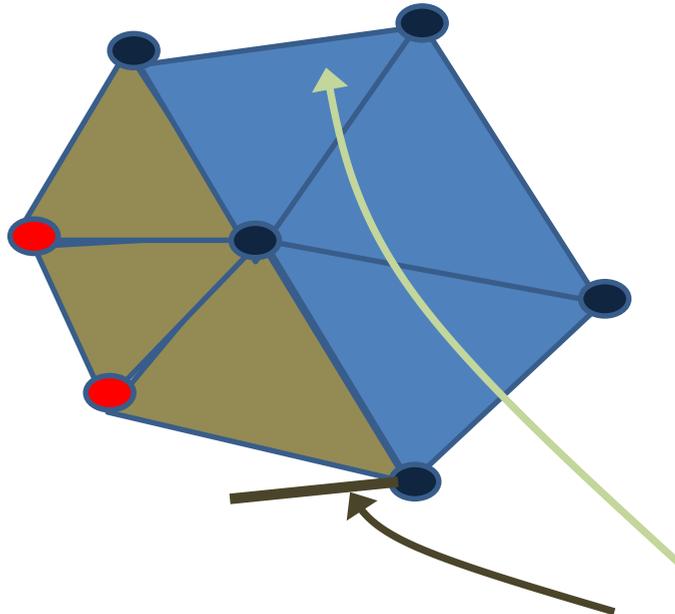
# Control Grid Density In SMS

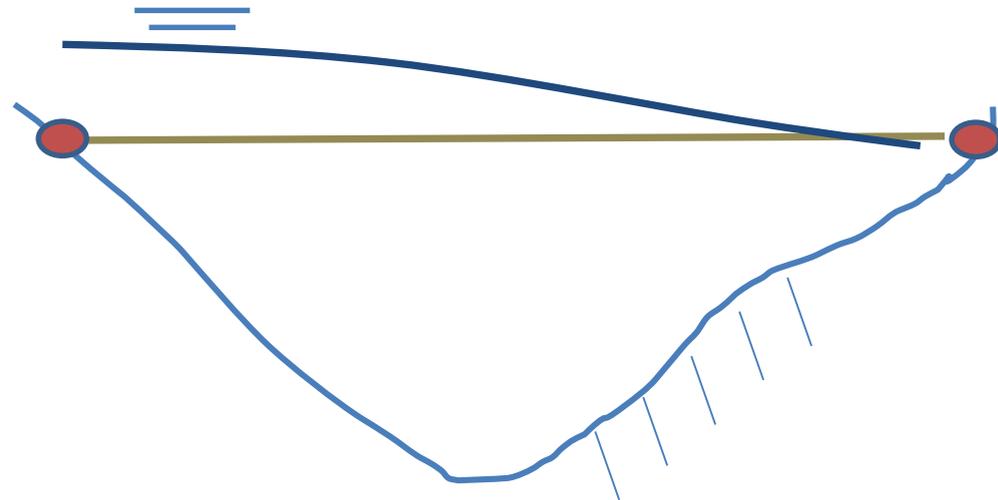
- Density-based paving
- Shore resolution propagates in
- Refinement nodes (node in map tool) can enforce local resolution (but may produce skew elements)



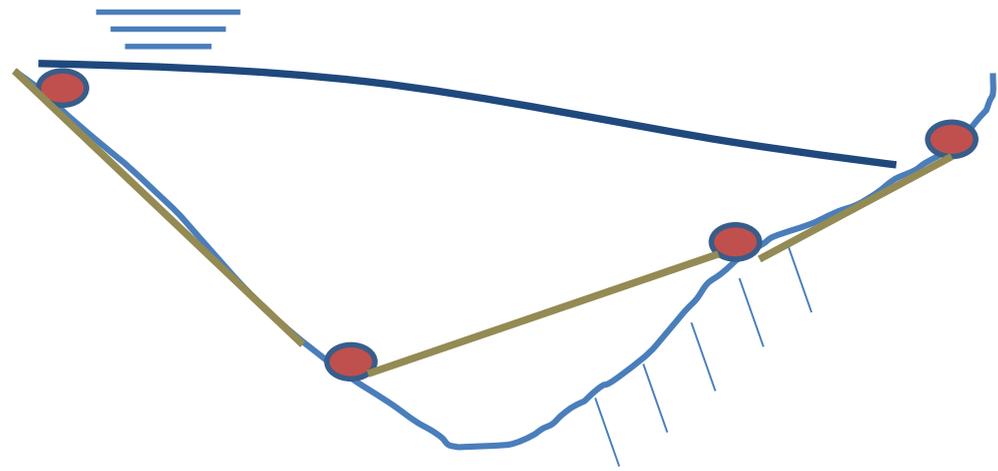
# Wetting and Drying

- Element-based designation
  - Based on node elevations
  - All 3 | 4 wet => wet element
  - No partial wet/dry
- Velocities calculated on edges of wet elements



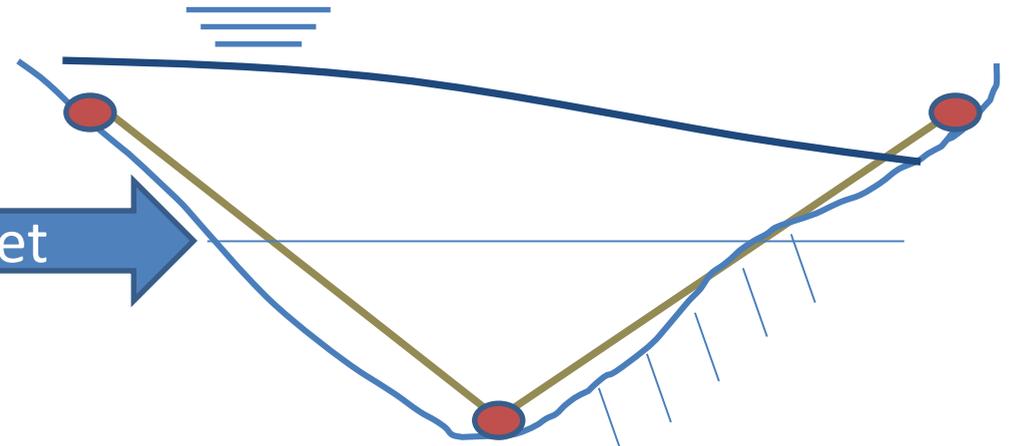


Clogged

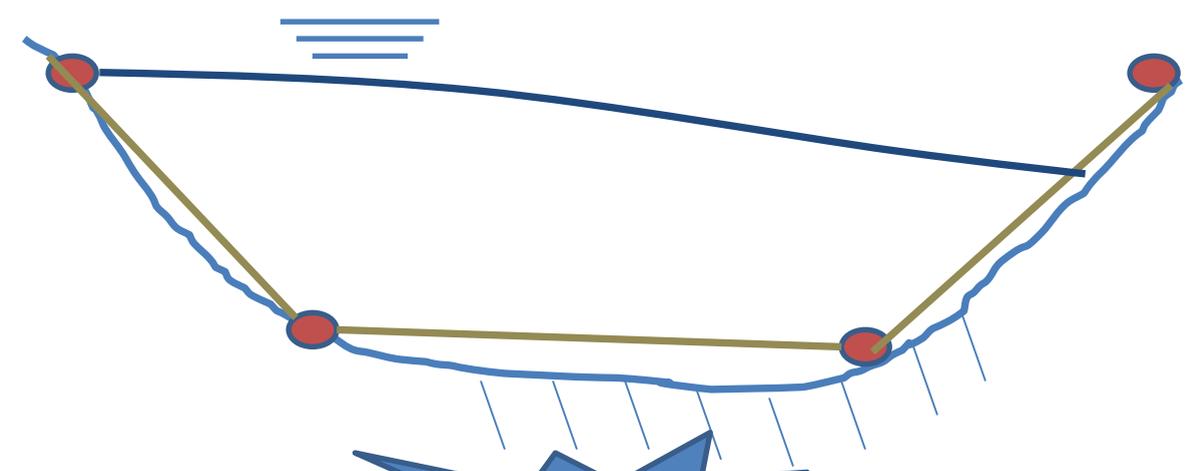


Mildly inaccurate: wrong contour?

Always wet



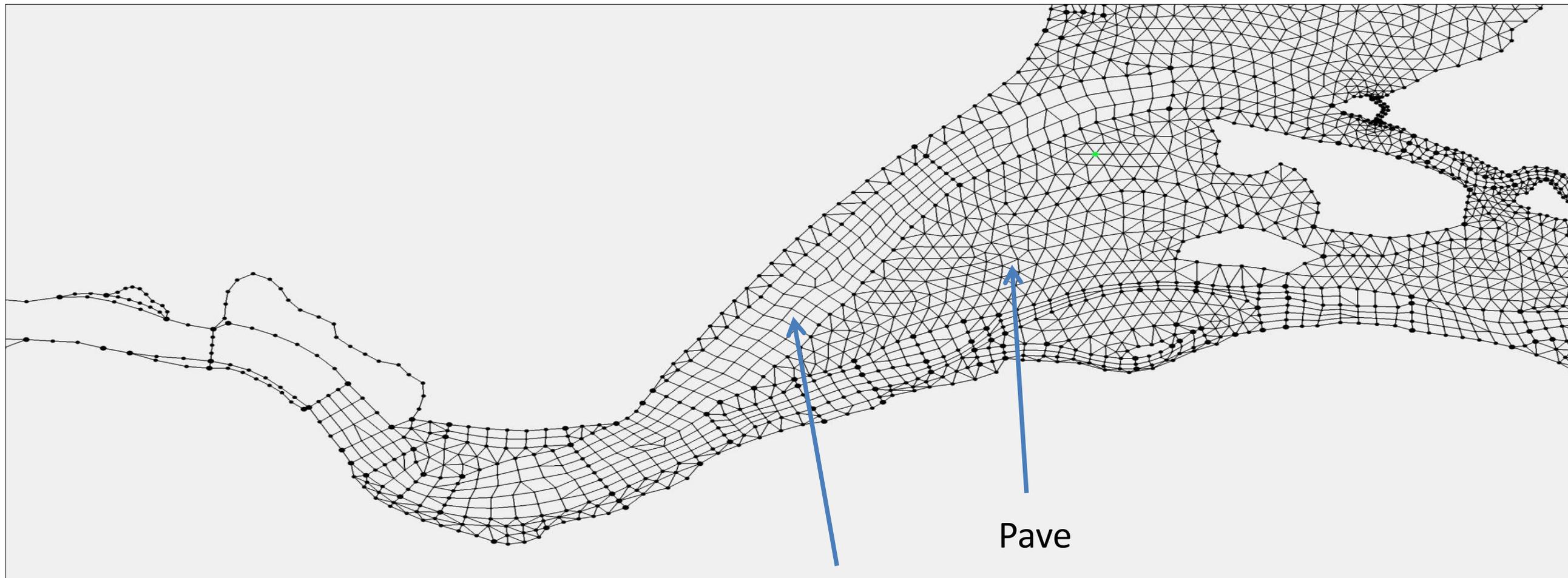
Clogged or narrow



**Robust!!**

- With SCHISM you can faithfully represent channels (and other features), thru good choices of horizontal and vertical grids
- In estuarine and near-shore applications, faithful representation of bathymetry and features is the right approach, as opposed to artificial manipulation of bathymetry

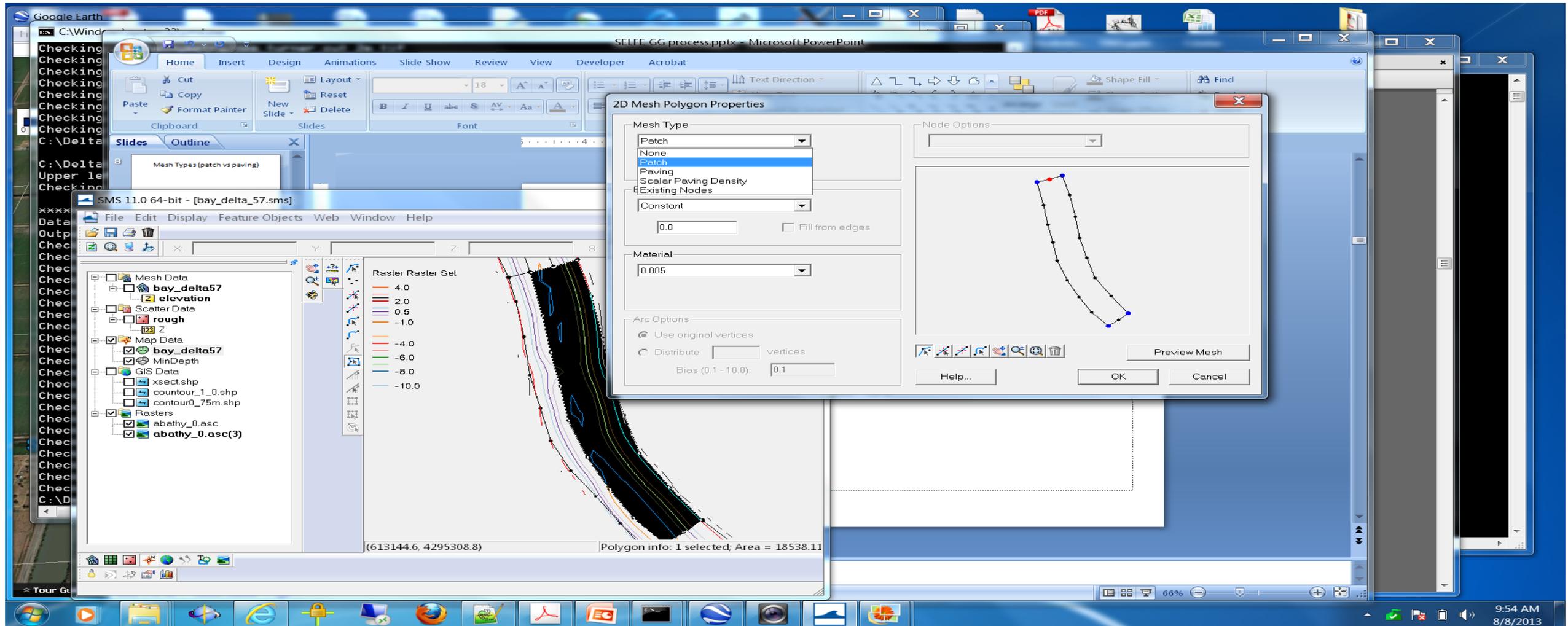
# Patching and Paving

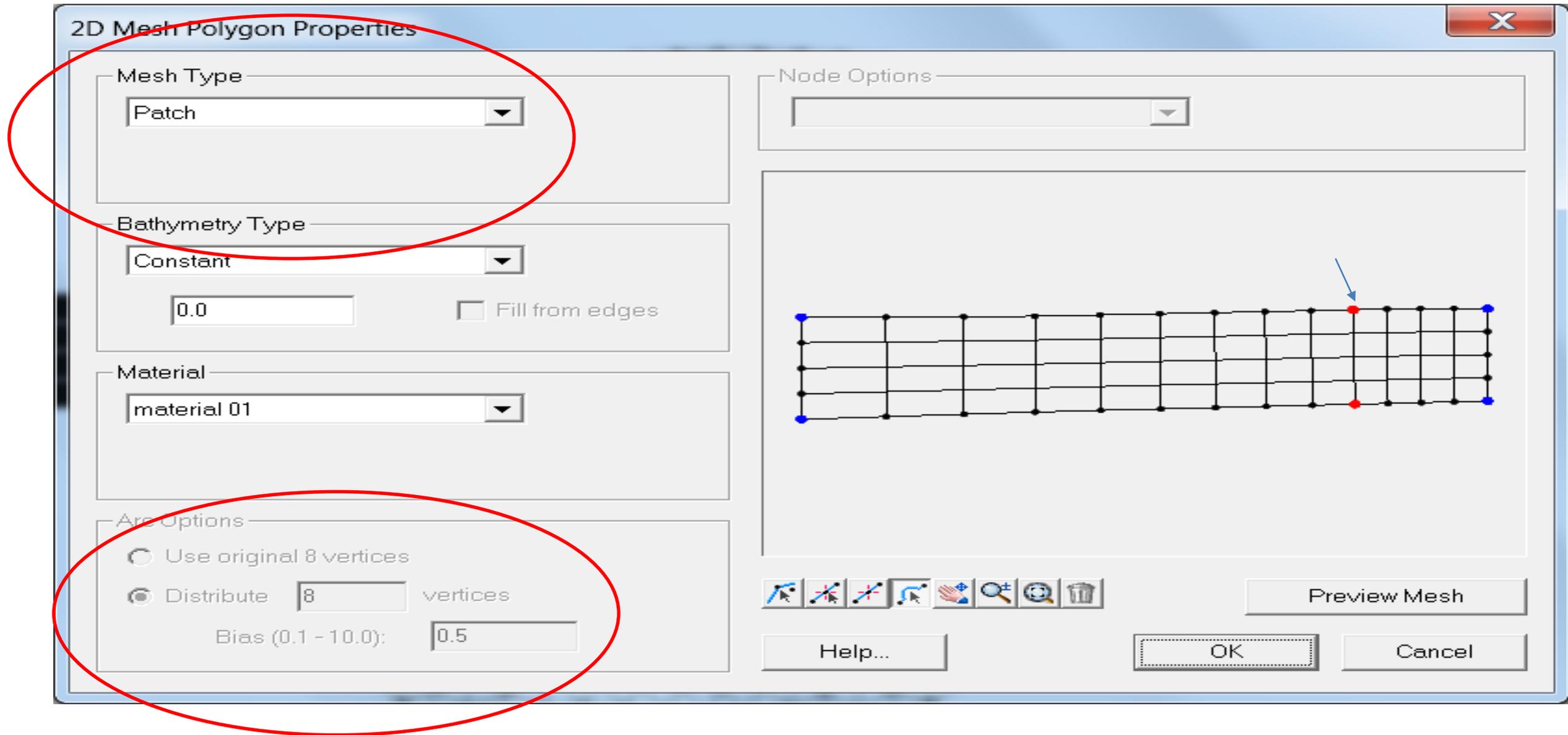


Patch

Pave

# Mesh Types (Patch vs Paving)



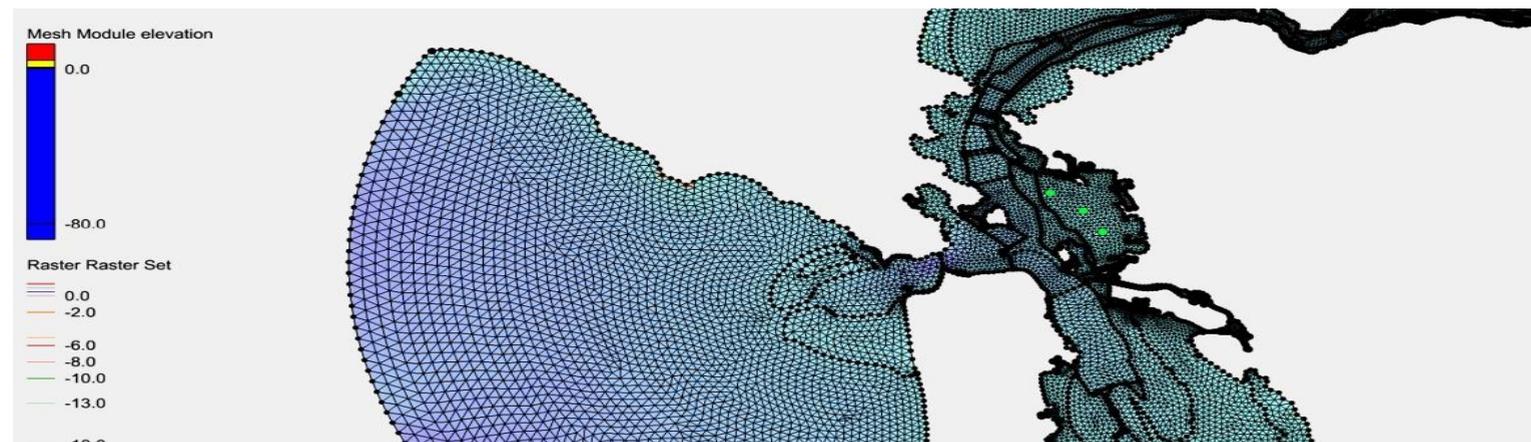


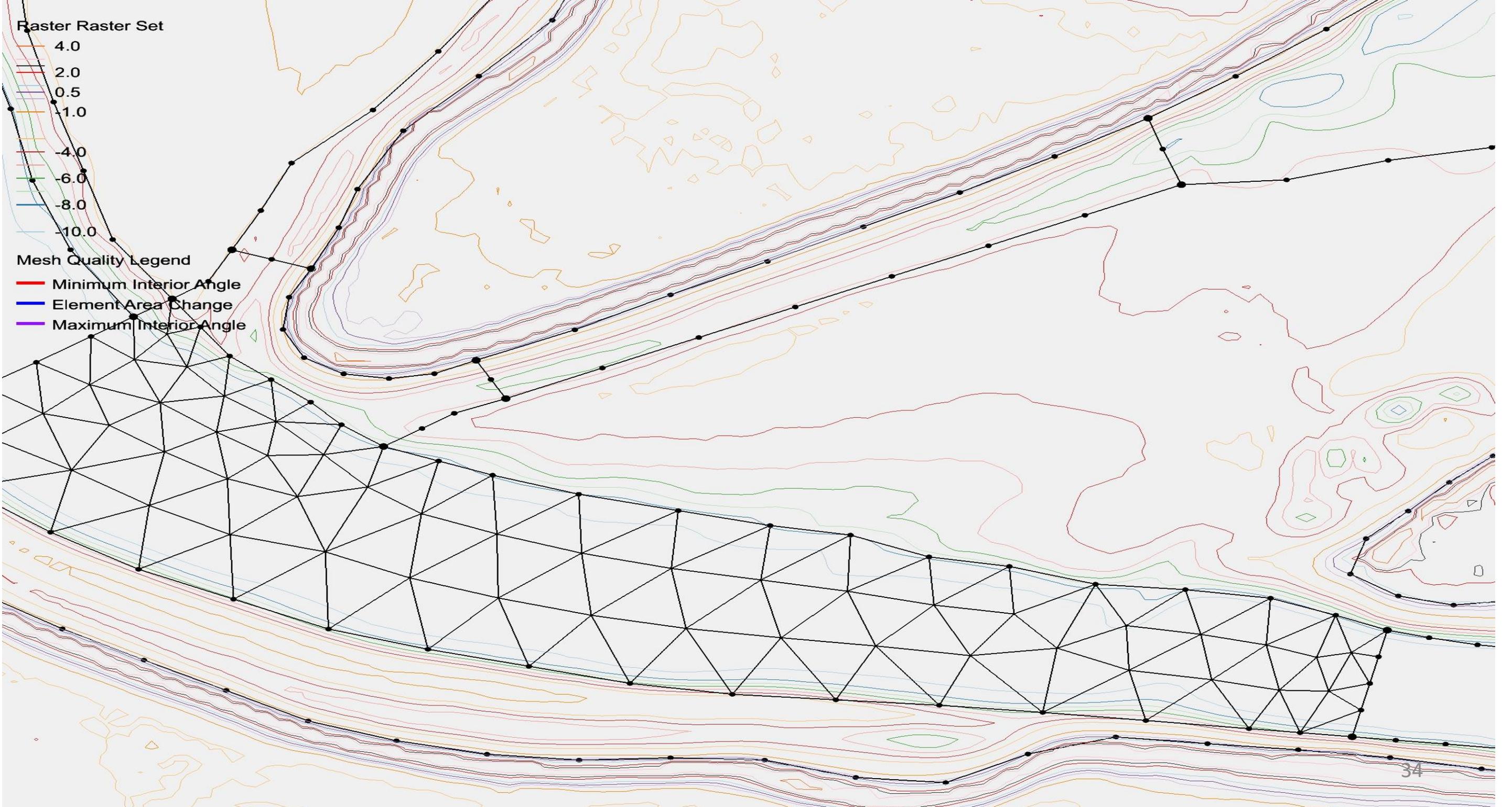
- Bias easier to “undo” here
- Make sure there are only 4 corner nodes (merge some if necessary)

# Patch vs Pave

Pave:

- Recommended for quality grids on large, well-resolved water bodies
- Works well when polygon is big and wide compared to arc nodes
- Most robust of all methods
- Resolution harder to control in the interior

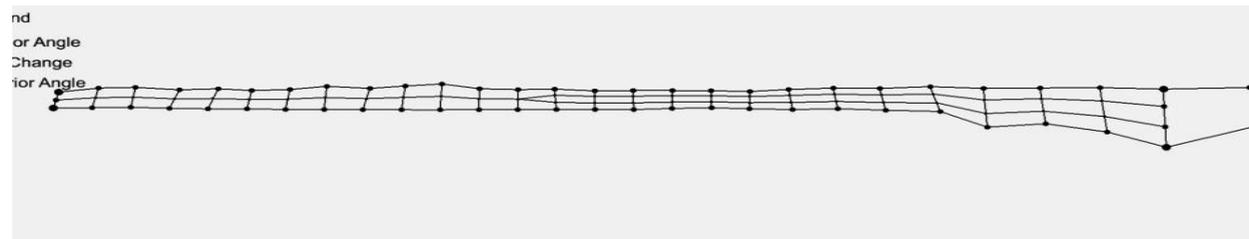




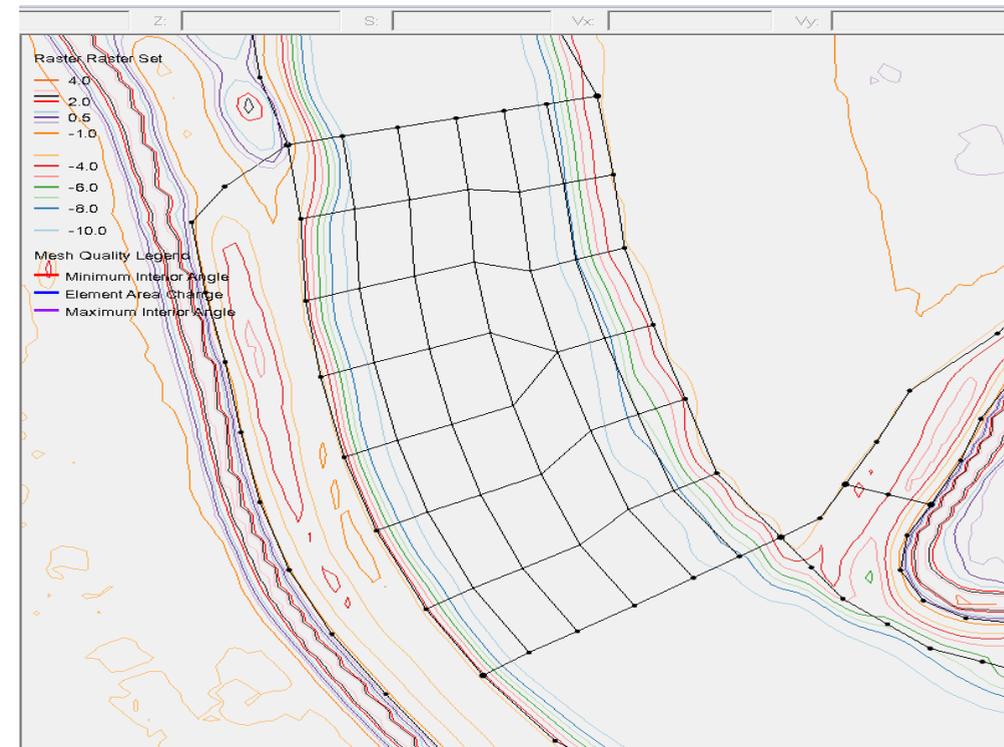
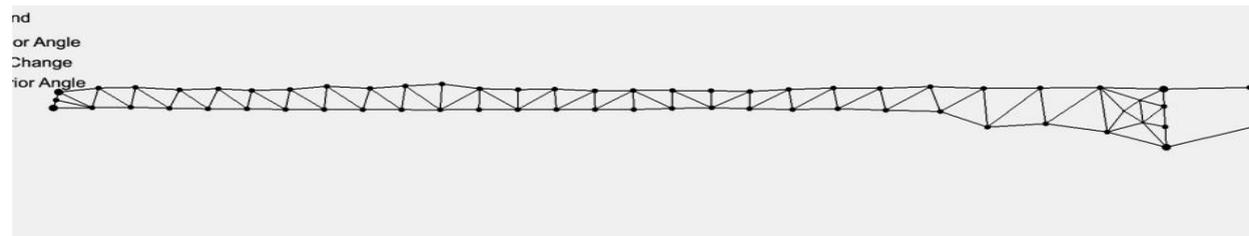
# (Adaptive) Coon's Patch

- Quasi-structured grid block
- Force mesh to be n-elements wide (try to match on opposite boundaries)
- Arbitrary anisotropy (3:1 or 4:1 common)
- Follow the channel!!! Don't round corners
- Easier to control cross-channel resolution

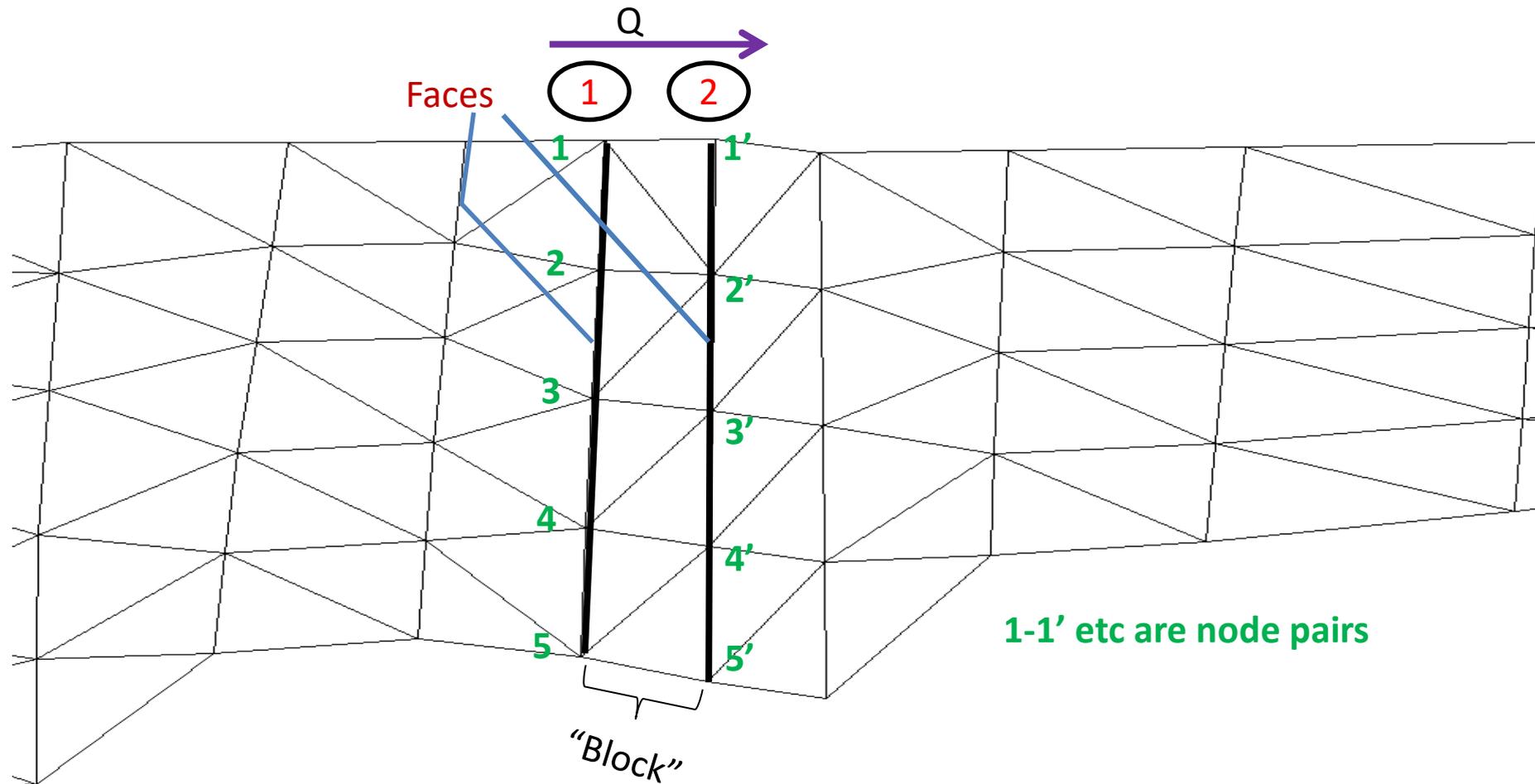
Patch (Adaptive Coon's Patch)



Pave (advancing front)

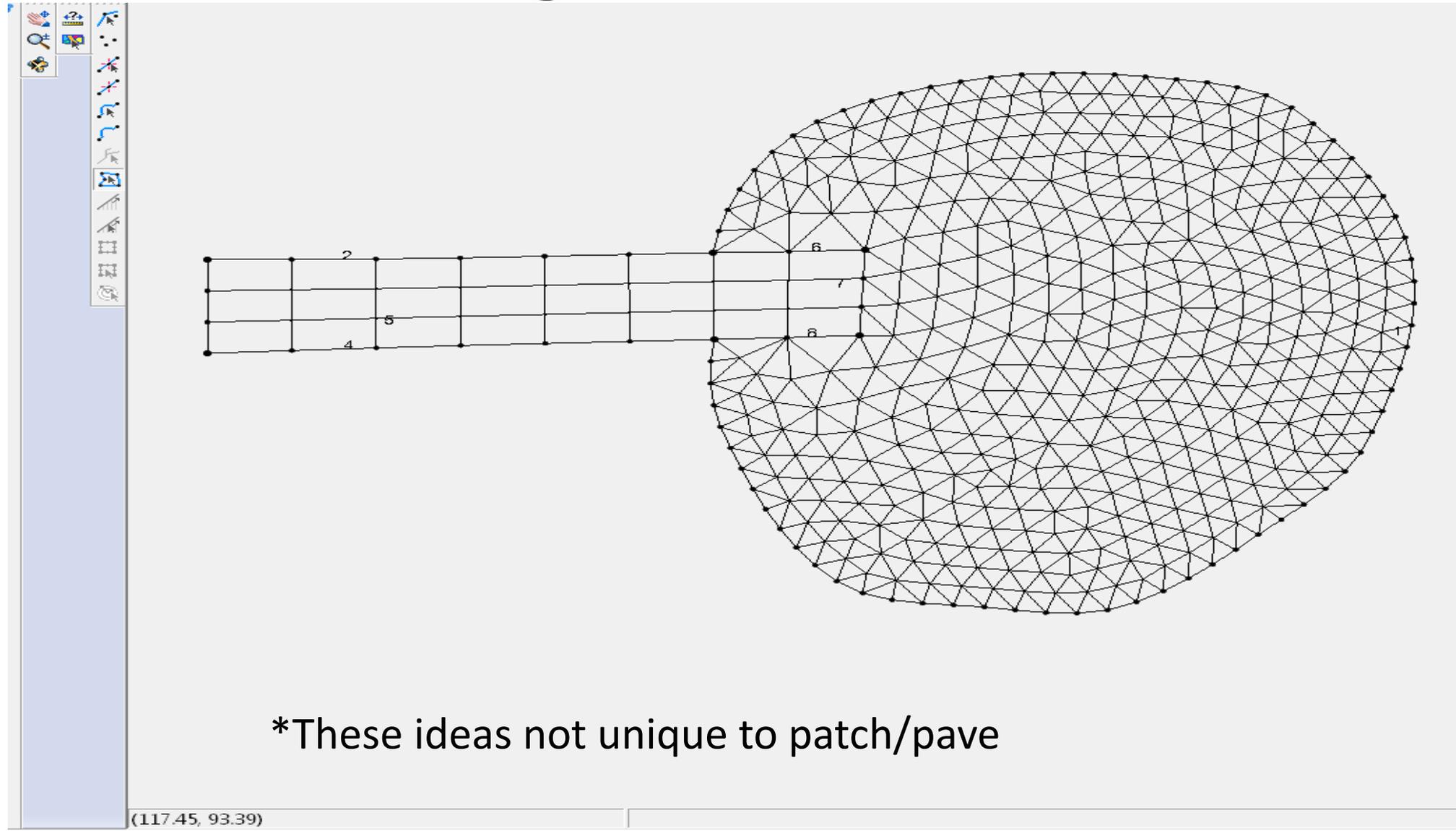


# Hydraulic Structures



Patches are natural choices here...

# Dovetailing Patch and Pave\*



\*These ideas not unique to patch/pave

\*Note the change in effective resolution...

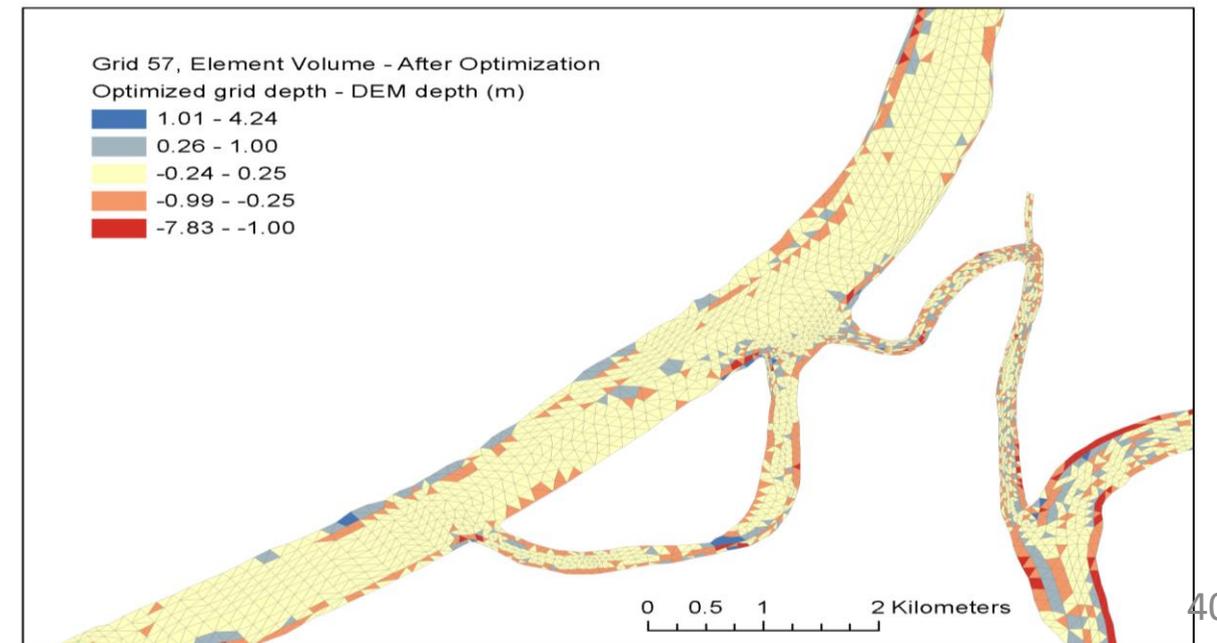
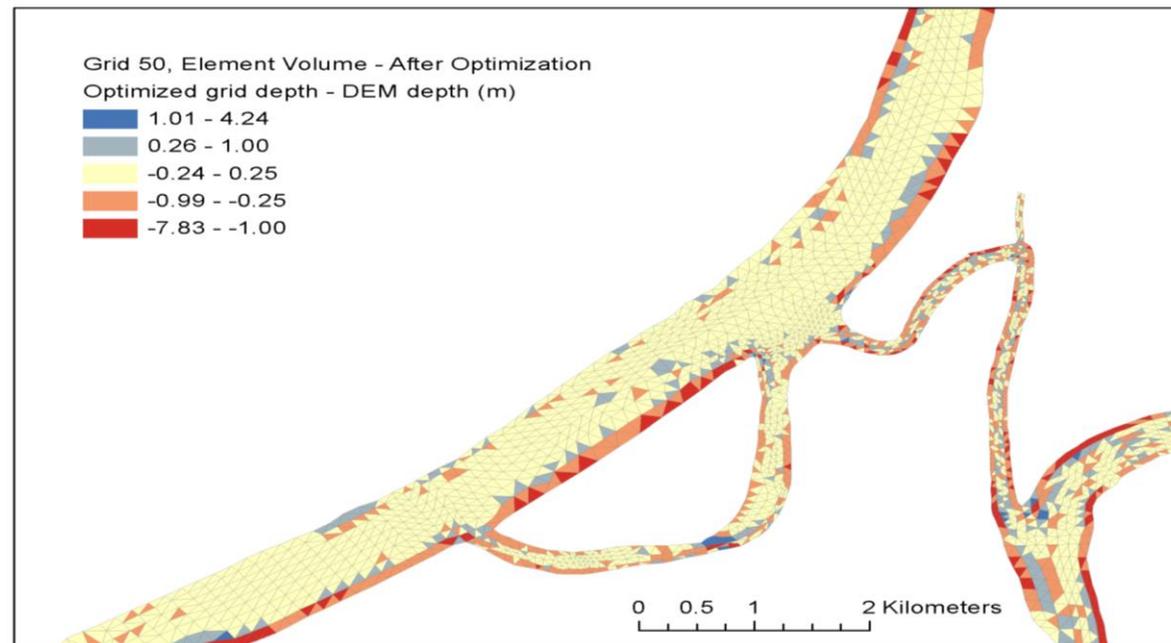
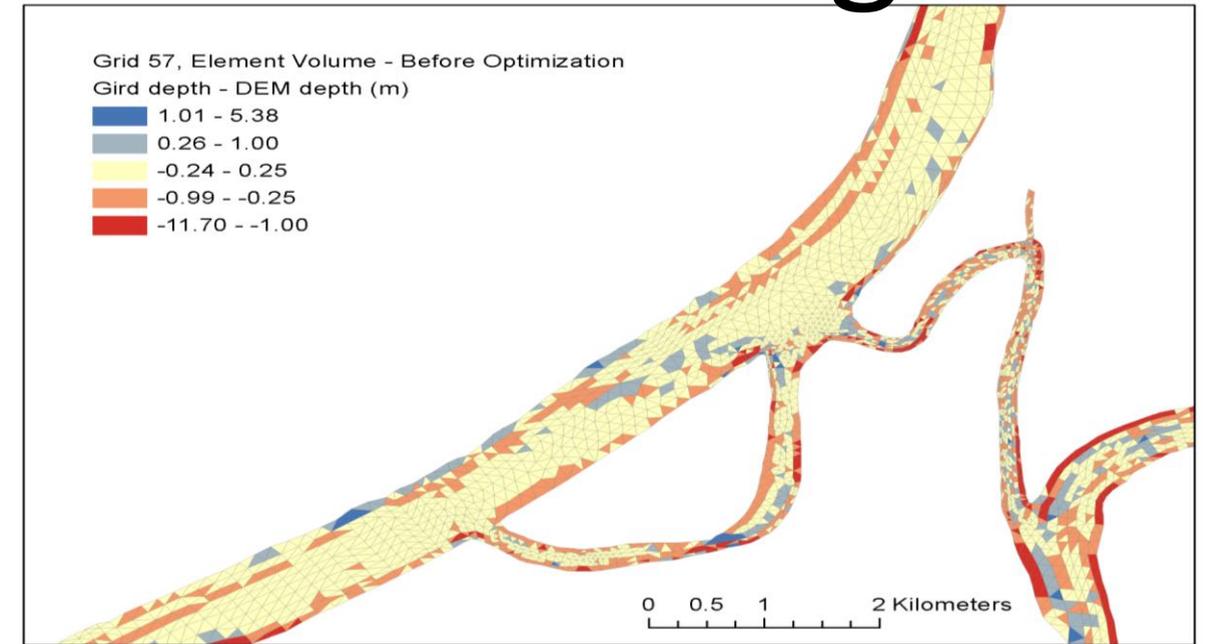
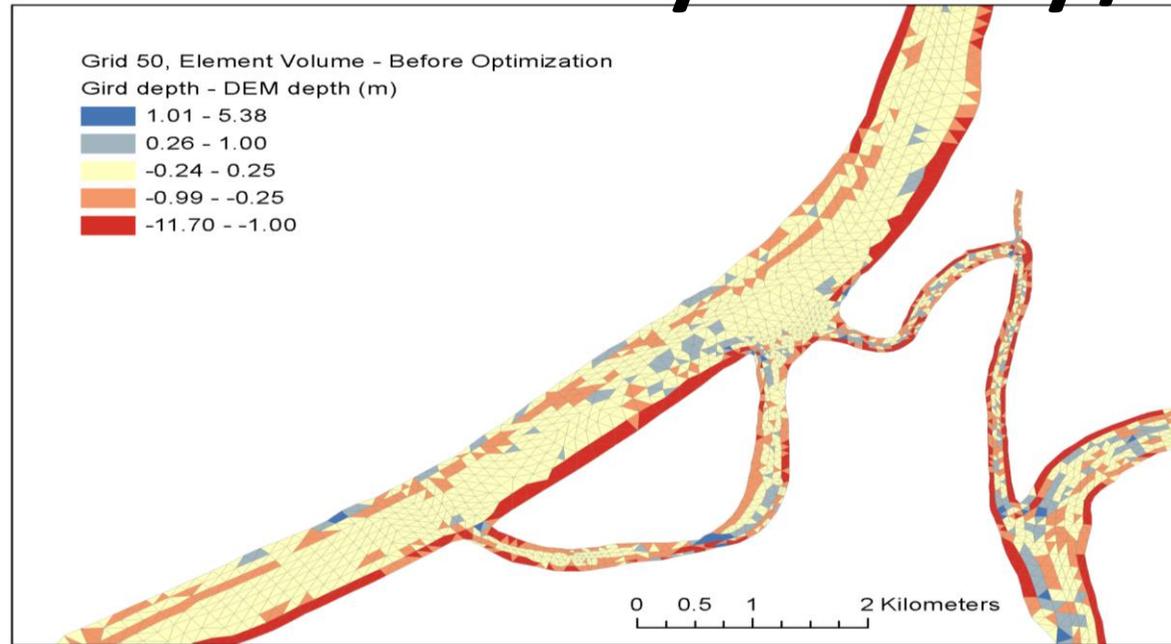
# Populating Elevations

- Big problem again ... SMS doesn't hold the data gracefully
- Have heard of people doing it in stages
- Can use `stacked_dem_fill.py`, which:
  - Populates with correct georeference
  - Matches contours
  - Uses prioritized DEMs
  - Gives some warnings
- We have FORTRAN scripts for loading very large TIN (UG) DEMs
- We always use linear interpolation (consistent with SCHISM's shape function), ***without any smoothing***
  - Some you should really visualize the bathymetry immediately after G.G.; correct mistakes (e.g. blocked channels) immediately

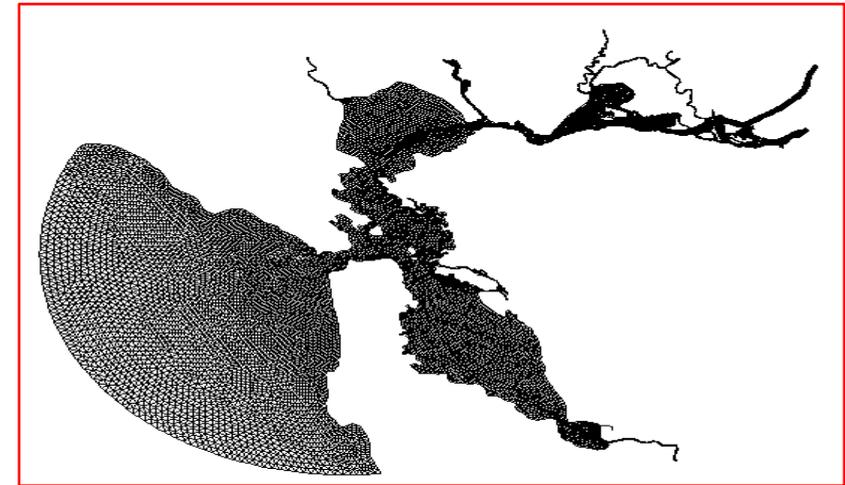
# Metrics

- Volumetric error (actually, average height)
  - Patterns can be systematic
- Face/edge aperture or area
- Storage area
- Skew: SCHISM *very* tolerant, but good to look (if you use xmgredit5, try skewness of 14)
- Need to use quality quads (xmgredit5: angle ratio >0.5)
- Area change: SCHISM very tolerant, but good to look
- Slope: PGE issue usually not a problem (vgrid plays a role also)
- Local error: Richardson extrapolation
  - Painful without tools
- TVD<sup>2</sup> and sub-cycling measures (fort.17; dtbe outputs): less of a problem than explicit TVD
- Eddying regime requires more uniform and higher-quality grid to avoid distortion of eddy dynamics
- CFL (>0.4)
  - Xmgredit5 can easily check CFL for you
  - Dimensionless wave number may also need to be looked at

# Bathymetry/Volume Matching



# Automatic regrid/subgrid

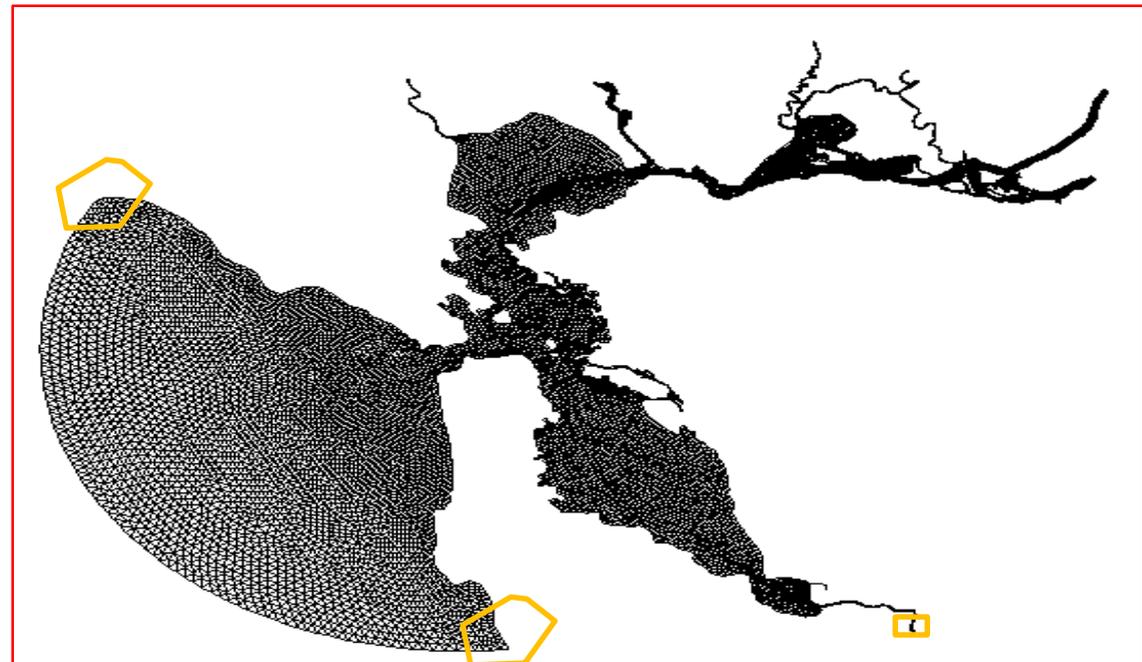


...it's generally easier (and quicker) to regrid the whole grid

# Common Ad Hoc Fixup Steps

- Deepening of boundary depths to prevent open boundary drying
- There is a fancier approach to preserve the original depth using bed deformation module or point sources

In gredit, create a region and  
`depth=max(depth,3)`



# Conclusions

- SMS is an adequate tool for non-orthogonal grids
- Good grid generation resolves flow/bathy features faithfully
  - Some finer scales ignored if not resolved
  - Implicit FE model provides you with more freedom for resolution
- Shape functions and wet-dry rules; understand the physics and numerics!
- You can't work on this stuff unless you can fix-test-fix-test
  - So front-end on node-based inputs (see BayDeltaSCHISM)
- **IMPORTANT: always start from an estimate of smallest  $\Delta t$  you anticipate for the application and use  $CFL > 0.4$  to back-calculate the coarsest resolution at each depth**
  - If you somehow decide to reduce  $\Delta t$  later, you may have to refine the grid
  - As a rough guide,  $\Delta t = 100-450s$  for b-tropic,  $100-200s$  for b-clinic applications (however, smaller  $\Delta t$  might be needed for some extreme cases)

# **A more complex river case**

# Hangzhou Bay + Qiantang River



# Qiantang River

Scatter Module elevation

— 2.0

— 1.0

— 0.0

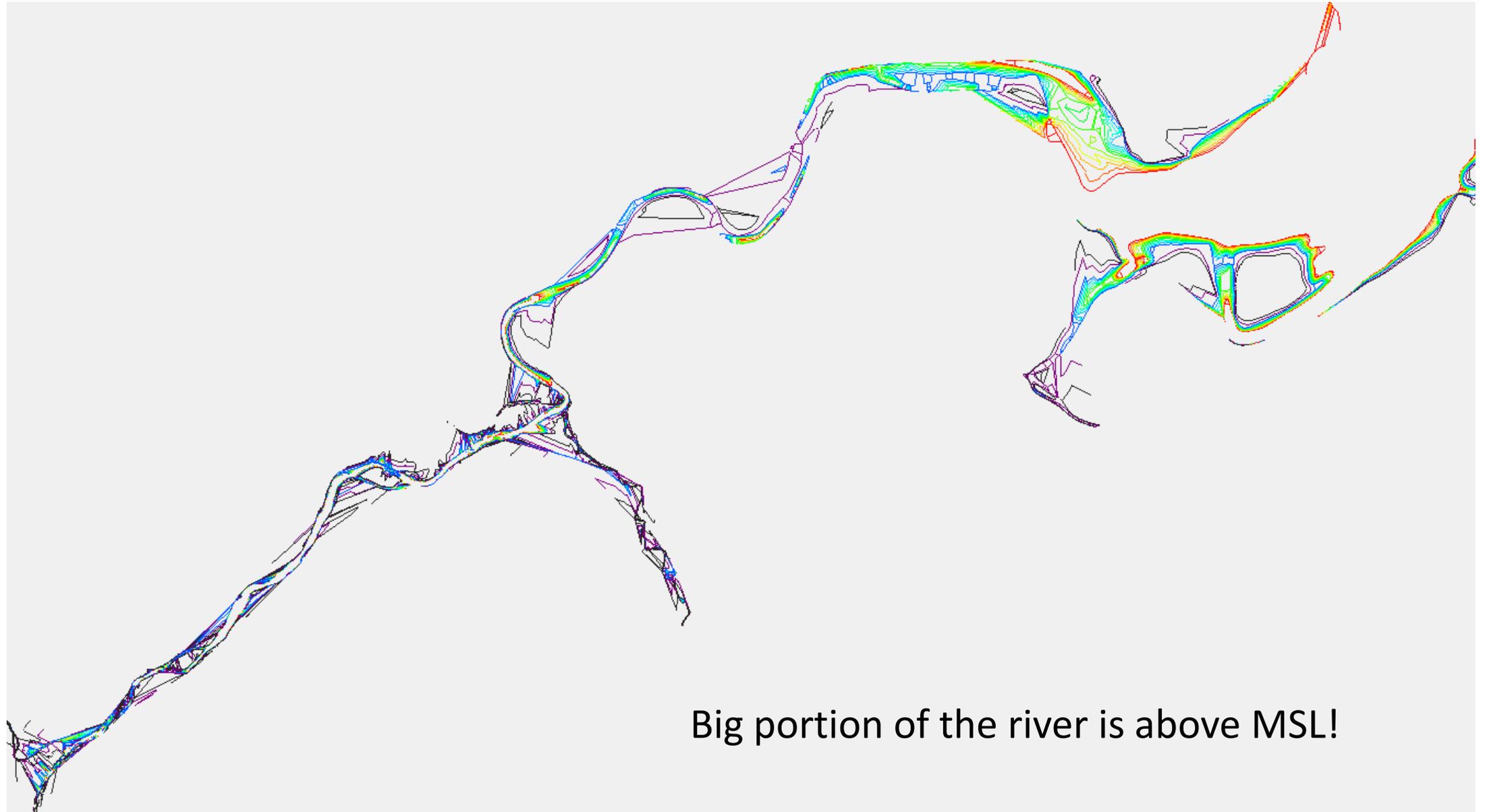
— -0.4

— -0.8

— -1.2

— -1.6

— -2.0

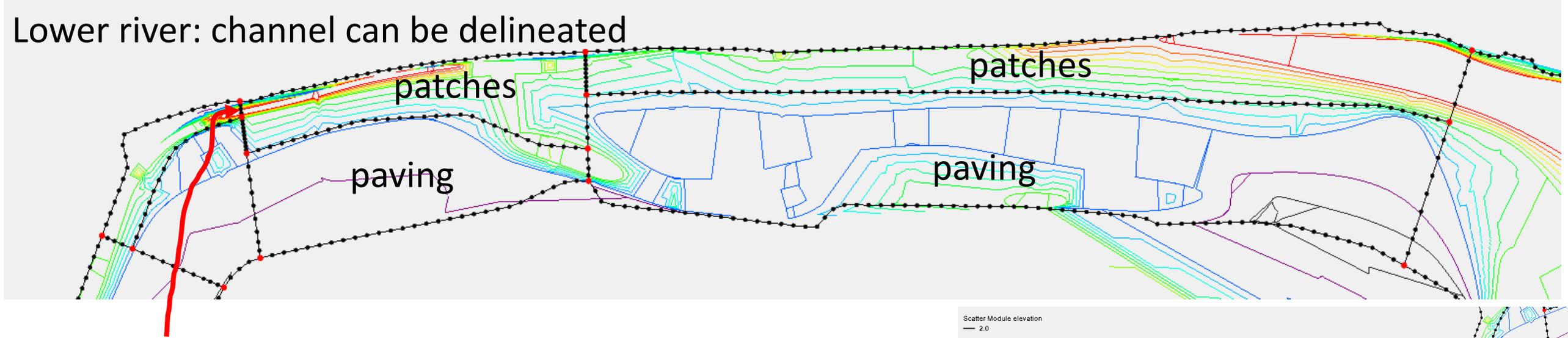


# Gridding strategy

- From observation we know that the channel rarely dries up, as the surface elevation is 1-2m above the bottom
- In the model setup, we can specify a non-zero i.c. for elev; e.g. we know from observation that the elevation in the upper most stretch of river is  $\sim 3\text{m}$  above MSL, so we can use  $\eta=3.2\text{m}$  everywhere as i.c. (including the ocean part)
- Then we gradually ramp down the elev toward  $\eta=0$  at the ocean boundary
  - System response is barotropic and so fast
- During gridding, capture the 'below MSL' portion of the channel in the lower part of the river first
  - Be game in resolving narrow channel!
- In the upper stretches where the bottom is mostly above MSL, it's up to you whether to skip the details of bathymetry during gridding
- At the river inflow boundary, you may dredge a small region to let water come in. The model will set up the surface slope automatically

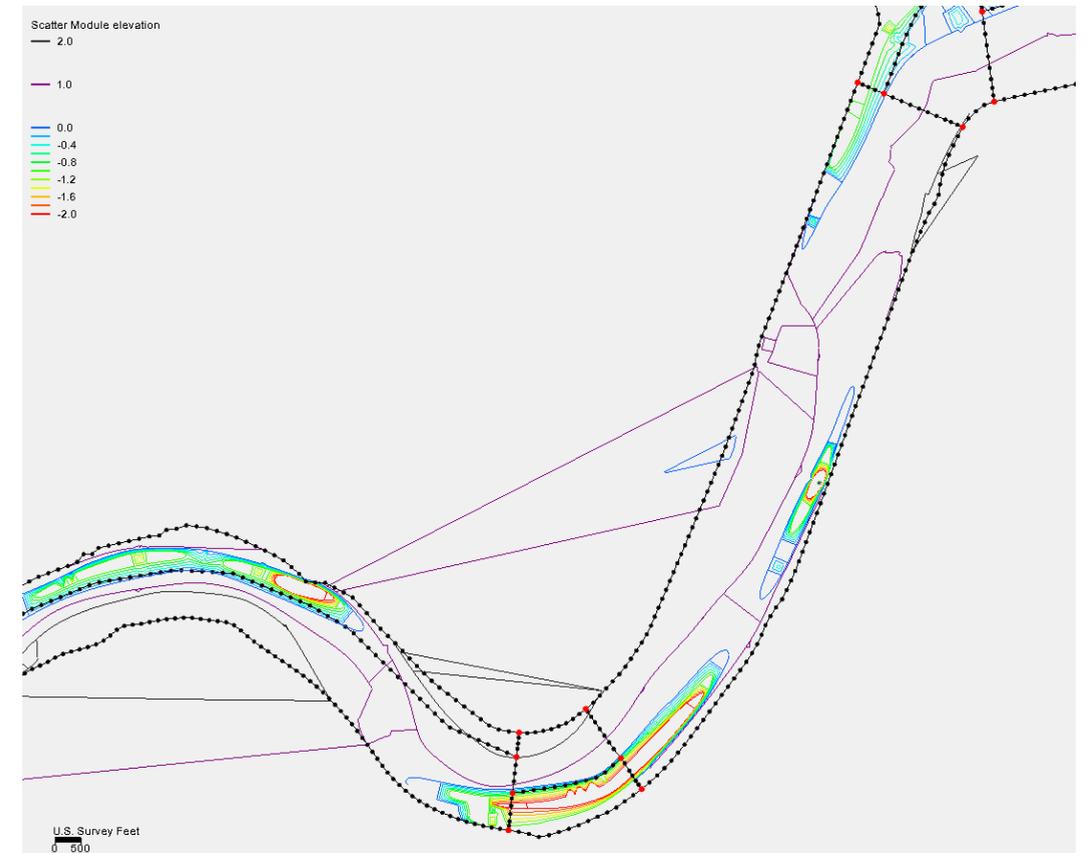
# Gridding strategy

Lower river: channel can be delineated



**Don't be timid in resolving the narrow channel!**  
(Actually the model may misbehave without this)

Upper river: your choice



# River inflow boundary

