Unstructured Mesh Generation for Storm Surge Modeling using OceanMesh2D

Keith J. Roberts
William J. Pringle
March, 25 2022

OceanMesh2D

MATLAB (original, many different capabilities and post-processing functions)

https://github.com/CHLNDDEV/OceanMesh2D

Note: OceanMesh2D does not require any special MATLAB toolboxes. OceanMesh2D is weakly compatible with Octave if MATLAB is unavailable. We are working on making it fully compatible.

Python (newer, less capabilities, but cleaner)

https://github.com/CHLNDDEV/OceanMesh

Many examples and tests are available on both versions

oceanmesh2d.slack.com

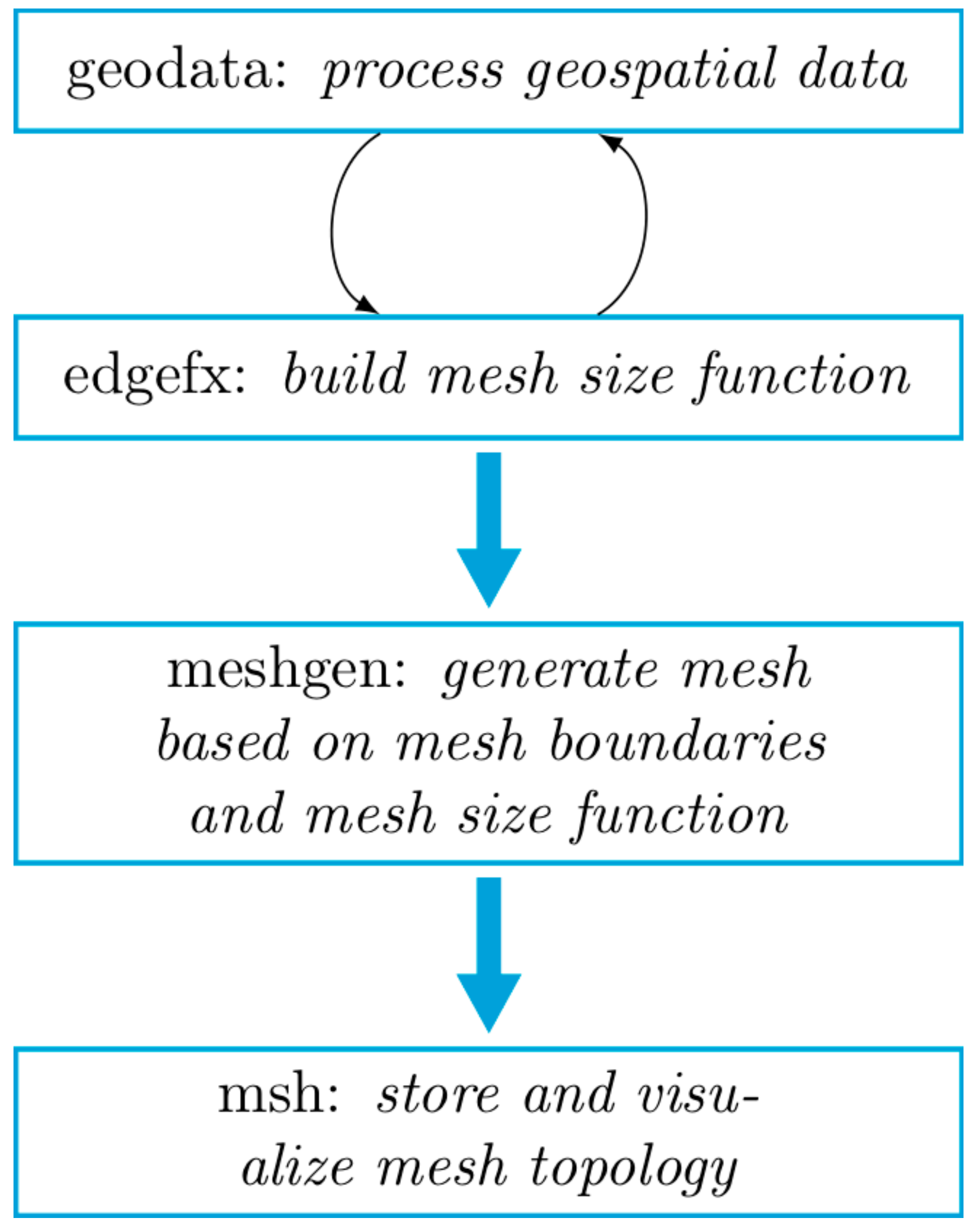
User-guide: Roberts, K. J., Pringle, W. J, 2018. OceanMesh2D: User guide - Precise distance-based two-dimensional automated mesh generation toolbox intended for coastal ocean/shallow water. https://doi.org/10.13140/RG.2.2.21840.61446/2.

Paper: Roberts, K. J., Pringle, W. J., & Westerink, J. J. (2019). OceanMesh2D 1.0: MATLAB-based software for two-dimensional unstructured mesh generation in coastal ocean modeling. *Geoscientific Model Development*, *12*, 1847–1868. https://doi.org/10.5194/gmd-12-1847-2019

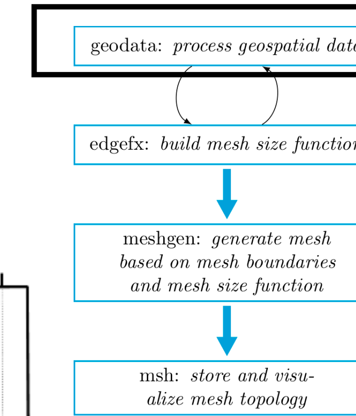
Automatic mesh generation: OceanMesh2D

An object orientated approach for coastal mesh development.

Process is scripted
and approximately reproducible



Geodata: shoreline is represented with signed-distance functions



Simplification of shoreline is not necessary beforehand

$$\Omega = S \cap bbox$$

$$d(\boldsymbol{x})_\Omega = \underbrace{s_\Omega(\boldsymbol{x})}_{\text{sign function}} \underbrace{\min_{\boldsymbol{y} \in \partial\Omega}(||\boldsymbol{x} - \boldsymbol{y}||)}_{\text{nearest distance}}$$

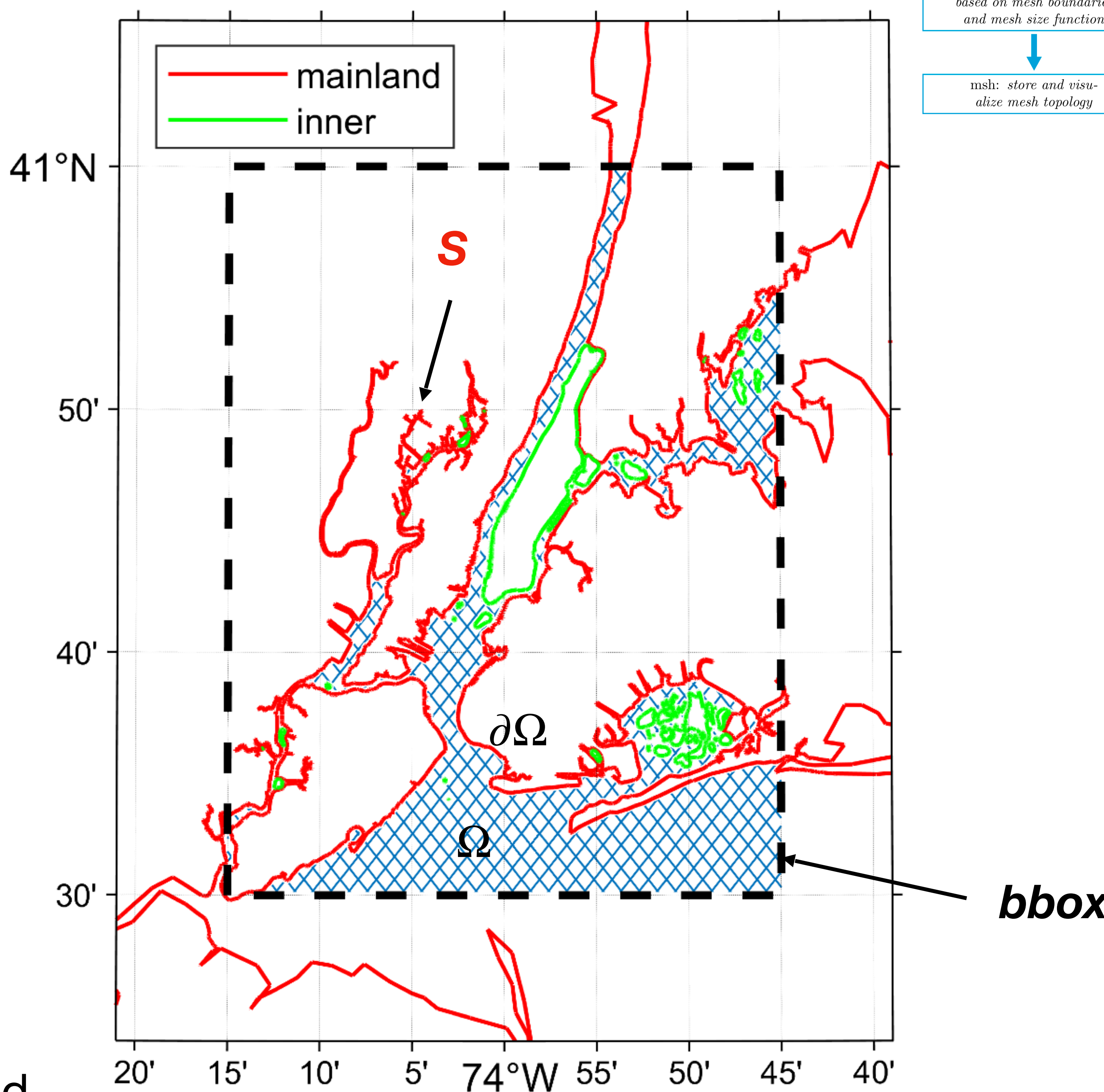
$$s(\boldsymbol{x})_\Omega := \begin{cases} -1, & \text{if } \boldsymbol{x} \in \Omega. \\ +1, & \text{if } \boldsymbol{x} \in \mathbb{R}^2 \setminus \Omega. \end{cases}$$

$$\Omega := \left\{ \boldsymbol{x} \in \mathbb{R}^2 : d(\boldsymbol{x})_\Omega \leq 0 \right.$$

$$\partial\Omega := \left\{ \boldsymbol{x} \in \mathbb{R}^2 : d(\boldsymbol{x})_\Omega = 0 \right.$$

*Shoreline are represented as piecewise linear segments at raw detail.

*Segments are classified by length and intersection with ***bbox***



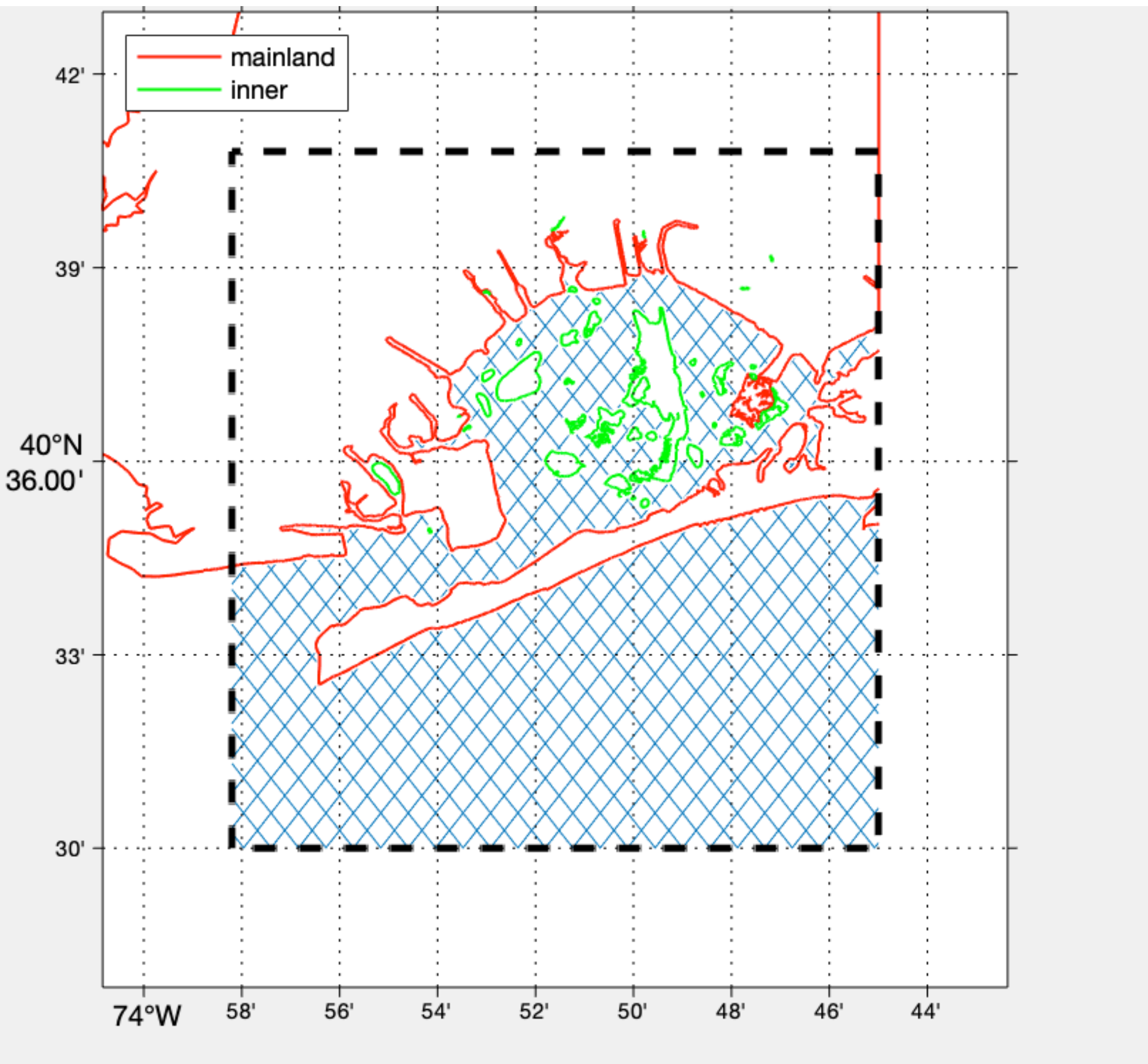
OceanMesh2D

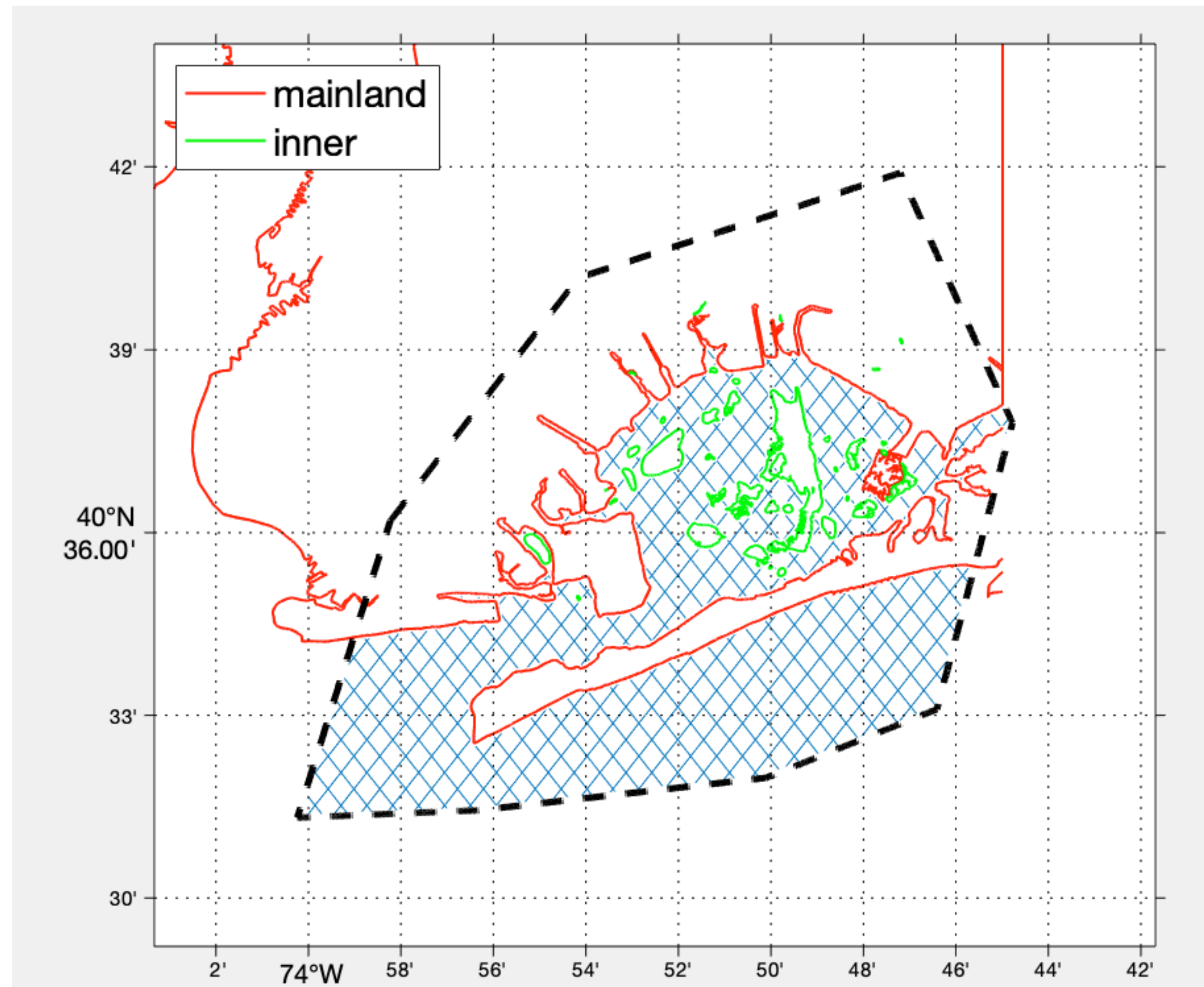
Geodata

Domains just have to be a polygon

bbox = [-73.9705   40.6032
-73.9028   40.6696
-73.7867   40.6983
-73.7463   40.6300
-73.7735   40.5517
-73.8351   40.5329
-73.9389   40.5240
-74.0040   40.5220
-73.9705   40.6032];

bbox = [-73.97 -73.75 % lon_min lon_max
40.5 40.68]; % lat_min lat_max

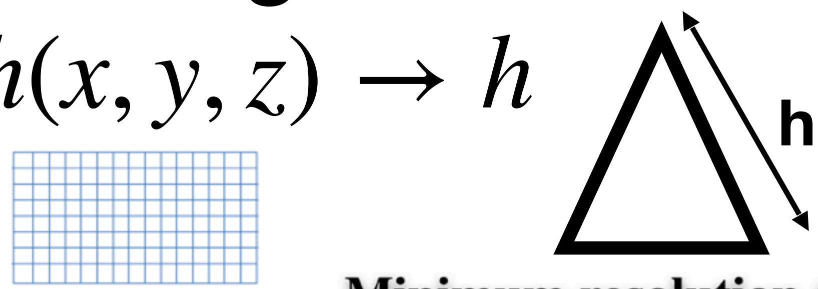




OceanMesh2D

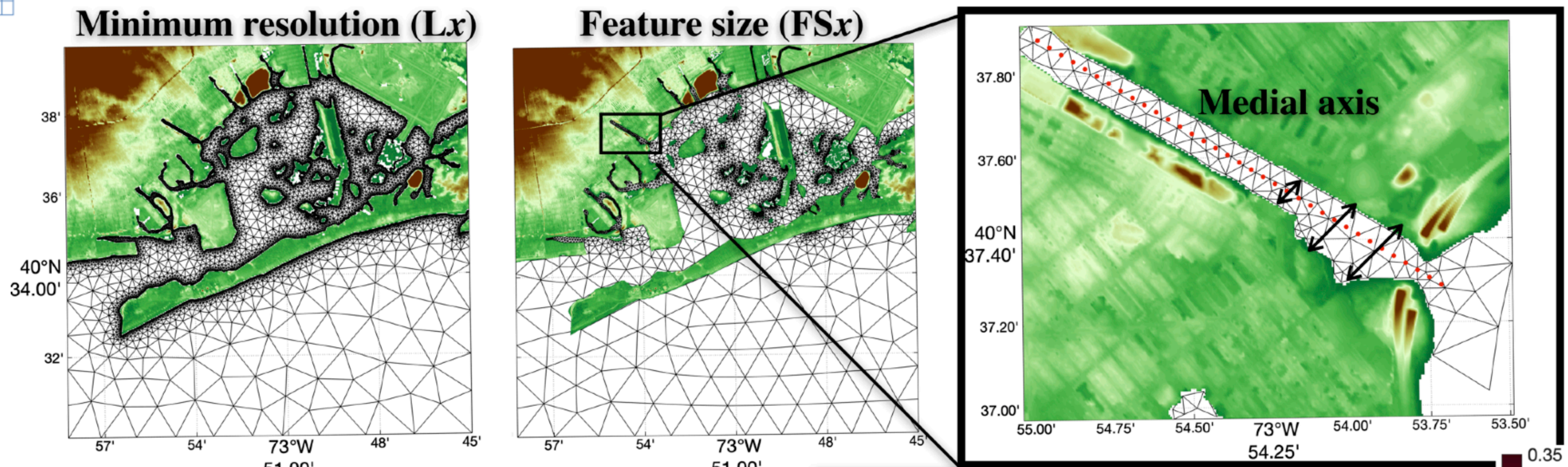
Edgefx: Mesh sizes are controlled via sizing functions

$$h(x, y, z) \rightarrow h$$



Combinations of mesh size functions are created

$$h = min\left[(h_{Lx} \text{or} h_{Fsx}), h_{Sx}, h_{Cx}\right] \text{with} g \leq G$$

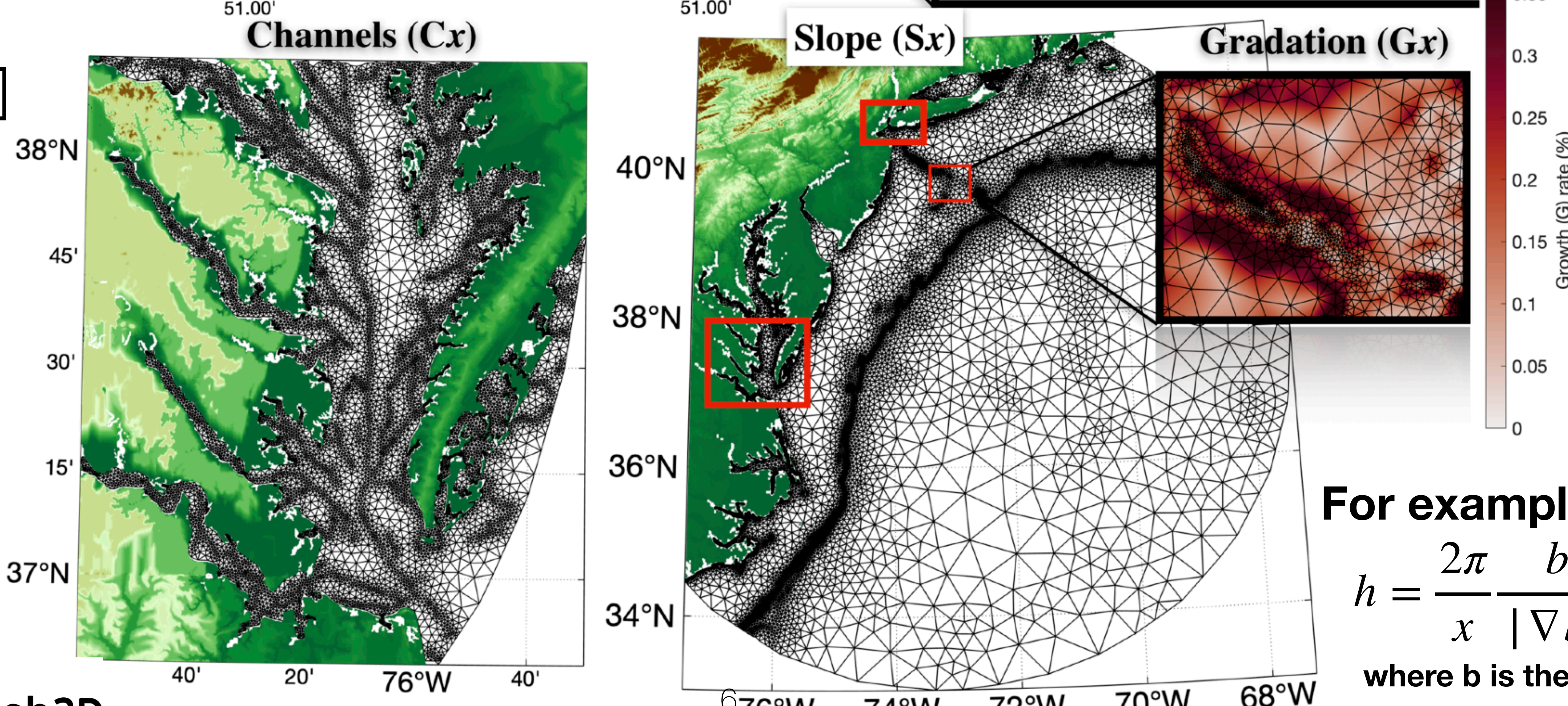


geodata: *process geospatial data*

edgefx: *build mesh size function*

meshgen: *generate mesh based on mesh boundaries and mesh size function*

msh: *store and visualize mesh topology*



For example,

$$h = \frac{2\pi}{x} \frac{b}{|\nabla b|}$$

where b is the depth

OceanMesh2D

Edgefx

Sizing fields can be built directly on DEM grids (and many at the same time)

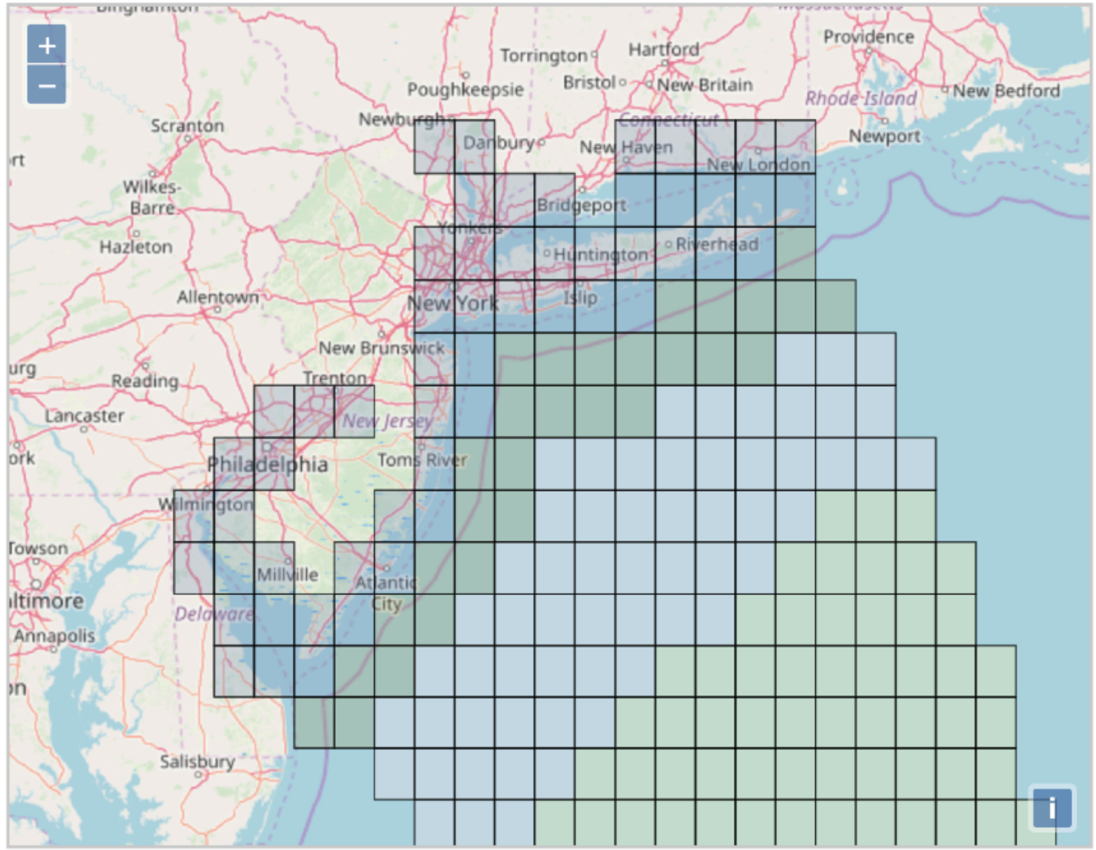
NCEI Hurricane Sandy Digital Elevation Models

NOAA's National Centers for Environmental Information (NCEI) is developing a suite of digital elevation models (DEMs) of the U.S. Atlantic Coast impacted by Hurricane Sandy in October 2012. These DEMs are the initial part of a planned framework for a seamless depiction of merged bathymetry and topography along U.S. coasts.

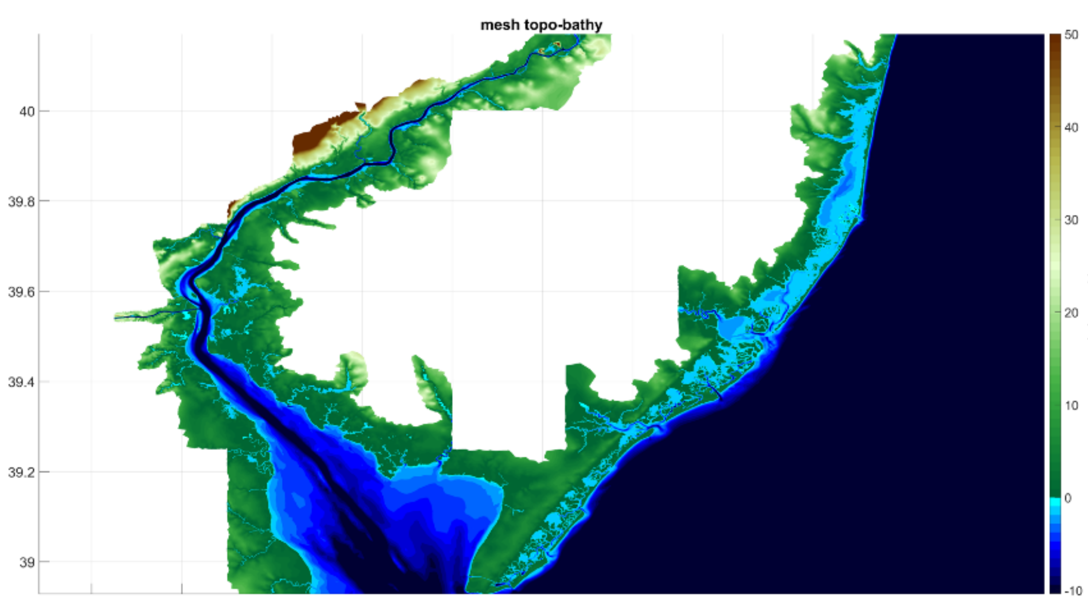
The DEMs telescope from the deep ocean floor to the coastal zone in 3, 1, 1/3, and 1/9 arc-second cell sizes. The 1/9 arc-second DEMs integrate both bathymetric and topographic data at the coast, while the offshore DEMs map bathymetry only.

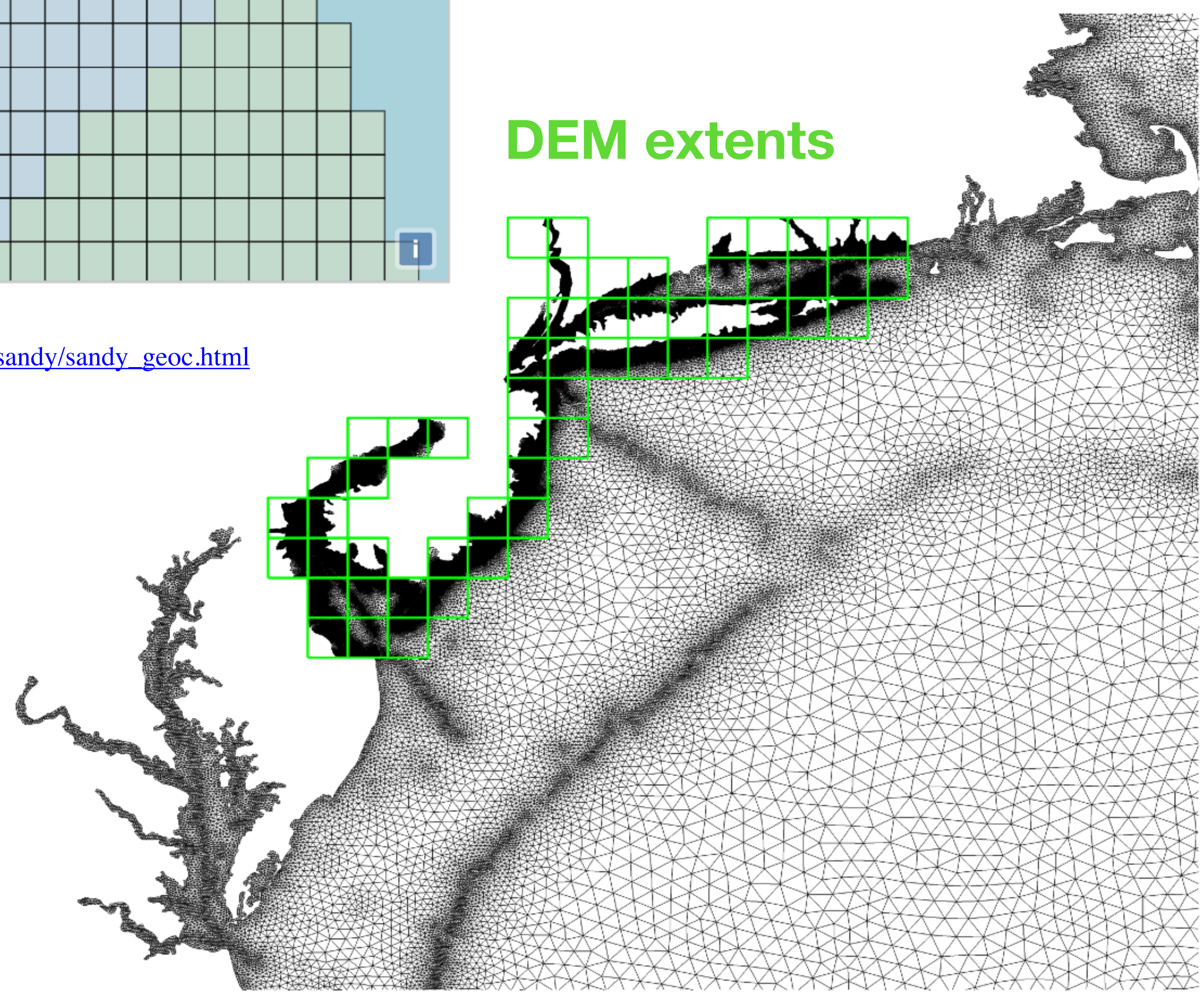
DEMs are tiled to enable targeted, rapid updates as new data become available.

Data Download:
Select region on map ➜
Access All Data
Access Metadata



https://www.ngdc.noaa.gov/mgg/inundation/sandy/sandy_geoc.html



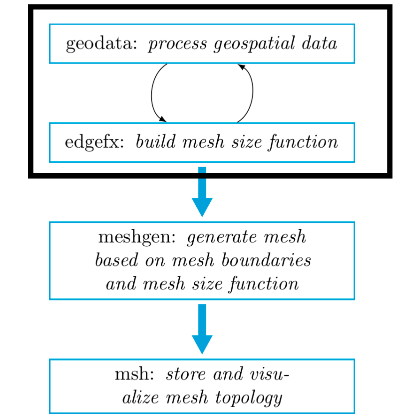


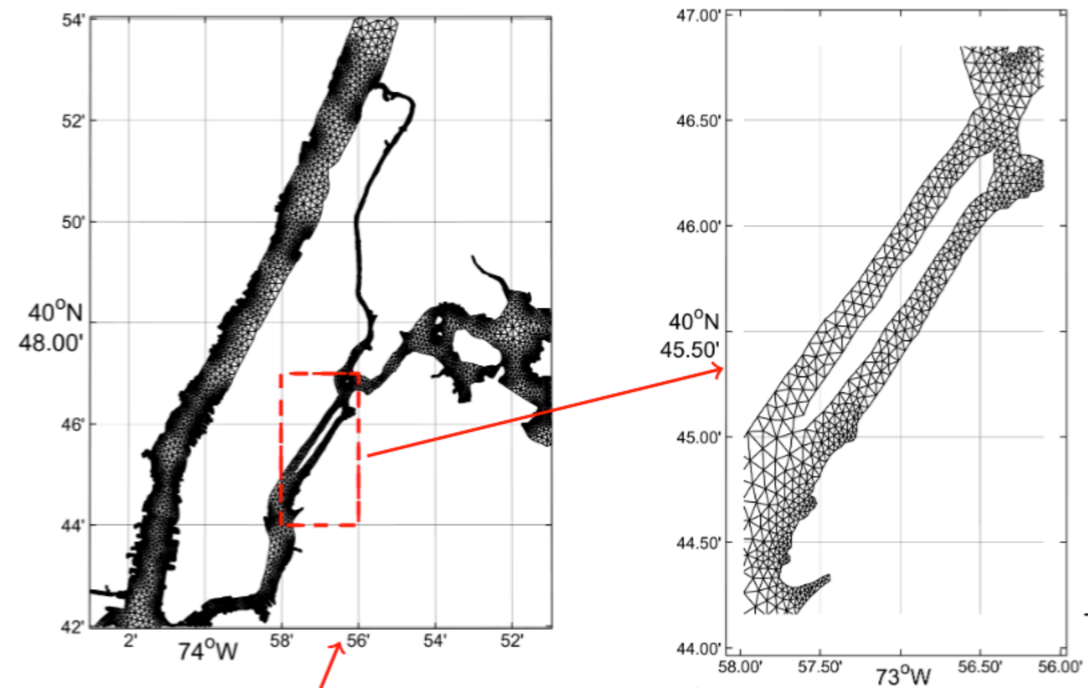
OceanMesh2D

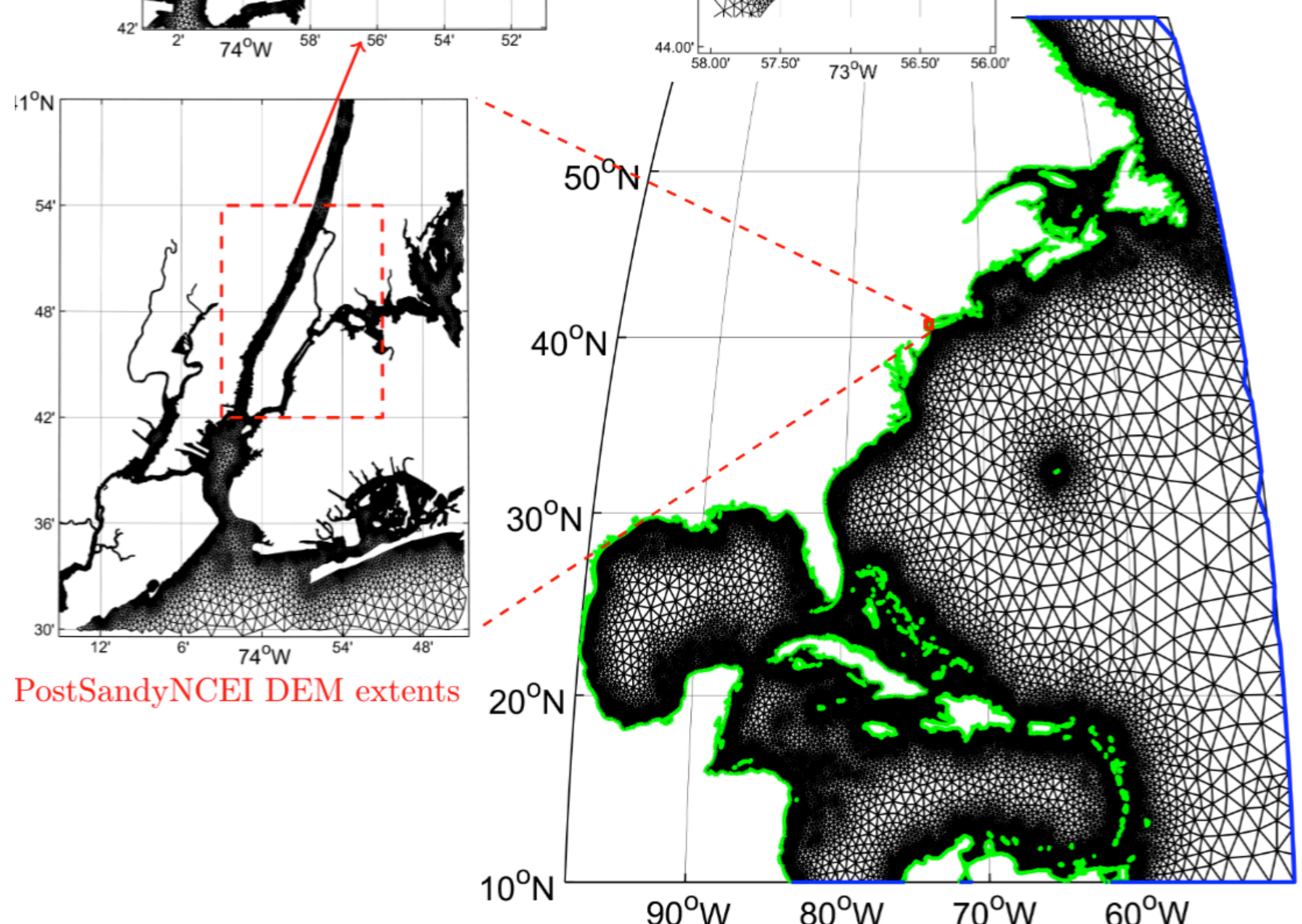
Multiscale meshing

Sizing fields can be nested with varying options to produce meshes with great element size variation.

```matlab
%% STEP 1: set mesh extents and set parameters for mesh.
%% The greater US East Coast and Gulf of Mexico region

bbox = [-71.6 42.7; -64 30; -80 24; -85 38; -71.6 42.7]; %polygon boubox
min_el    = 1e3;                    % minimum resolution in meters.
max_el    = 50e3;                   % maximum resolution in meters.
wl        = 30;             % 60 elements resolve M2 wavelength.
dt        = 0;              % Automatically set timestep based on nearshore res
grade     = 0.35;           % mesh grade in decimal percent.
R         = 3;                      % Number of elements to resolve feature.

%% STEP 2: specify geographical datasets and process the geographical data
%% to be used later with other OceanMesh classes...
dem       = 'SRTM15+V2.nc';
coastline = 'GSHHS_f_L1';
gdat1 = geodata('shp',coastline,'dem',dem,'h0',min_el,...
                'bbox',bbox);

%% STEP 3: create an edge function class
fh1 = edgefx('geodata',gdat1,...
             'fs',R,'wl',wl,'max_el',max_el,...
             'dt',dt,'g',grade);

%% Repeat STEPS 1-3 for a high resolution domain for High Res New York Part
min_el    = 30;             % minimum resolution in meters.
max_el    = 1e3;            % maximum resolution in meters.
max_el_ns = 240;            % maximum resolution nearshore.

coastline = 'PostSandyNCEI';
dem       = 'PostSandyNCEI.nc';

%polygon boubox
bbox2 = [-74.25 40.5; -73.75 40.55; -73.75 41; -74 41; -74.25 40.5];
gdat2 = geodata('shp',coastline,'dem',dem,'h0',min_el,'bbox',bbox2);

fh2 = edgefx('geodata',gdat2,'fs',R,'wl',wl,...
             'max_el',max_el,'max_el_ns',max_el_ns,...
             'dt',dt,'g',grade);

%% STEP 4: Pass your edgefx class object along with some meshing options
%% and build the mesh...
mshopts = meshgen('ef',{fh1 fh2},'bou',{gdat1 gdat2},...
                  'plot_on',1,'proj','lam');
```



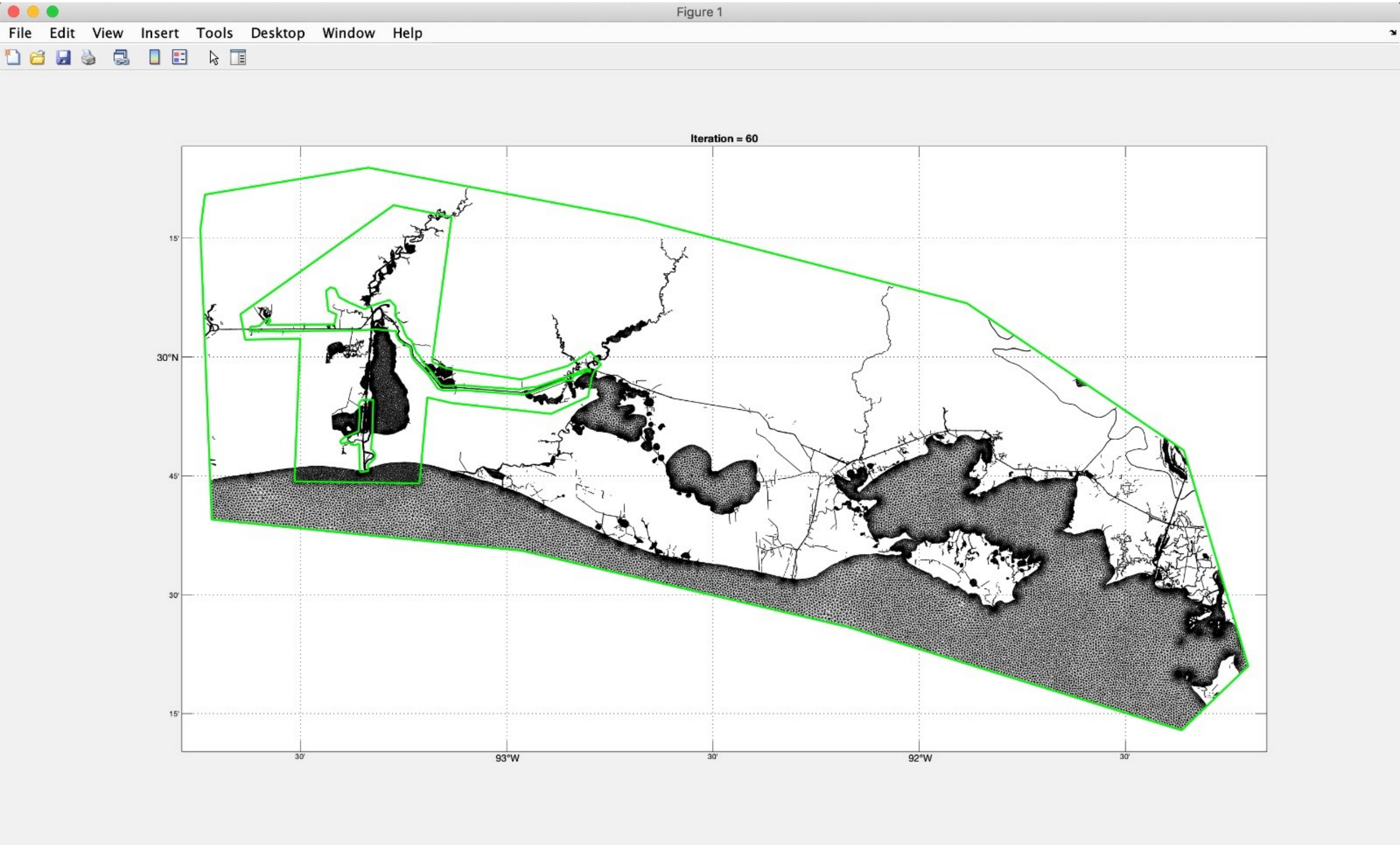




SRTM15+V2

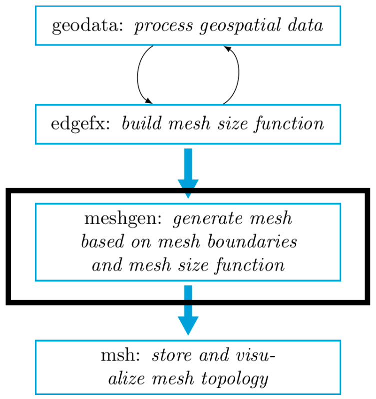
PostSandyNCEI DEM extents

Multiscale meshing



ceanMesh2D

Meshgen: generation via modified *DistMesh2D* algorithm



- P.-O. Persson, G. Strang, **A Simple Mesh Generator in MATLAB**.

  SIAM Review, Volume 46 (2), pp. 329-345, June 2004 (PDF)

$p(0) = p_0$ ⟶ ***Delaunay triangulation*** ***Move points based on F*** ⟶ $F(p) = 0$

$\mathbf{p} = [xy]$

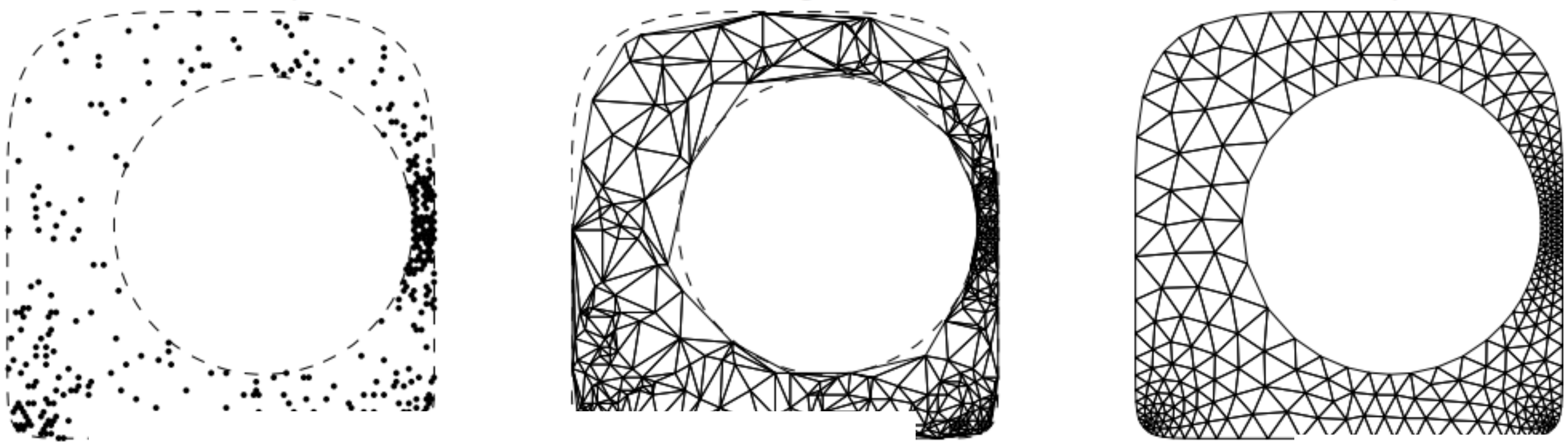
$\mathbf{F}(\mathbf{p}) = [\mathbf{F}_x(\mathbf{p})\mathbf{F}_y(\mathbf{p})]$

$$\frac{dp}{dt} = \mathbf{F}(\mathbf{p}), t \geq 0$$

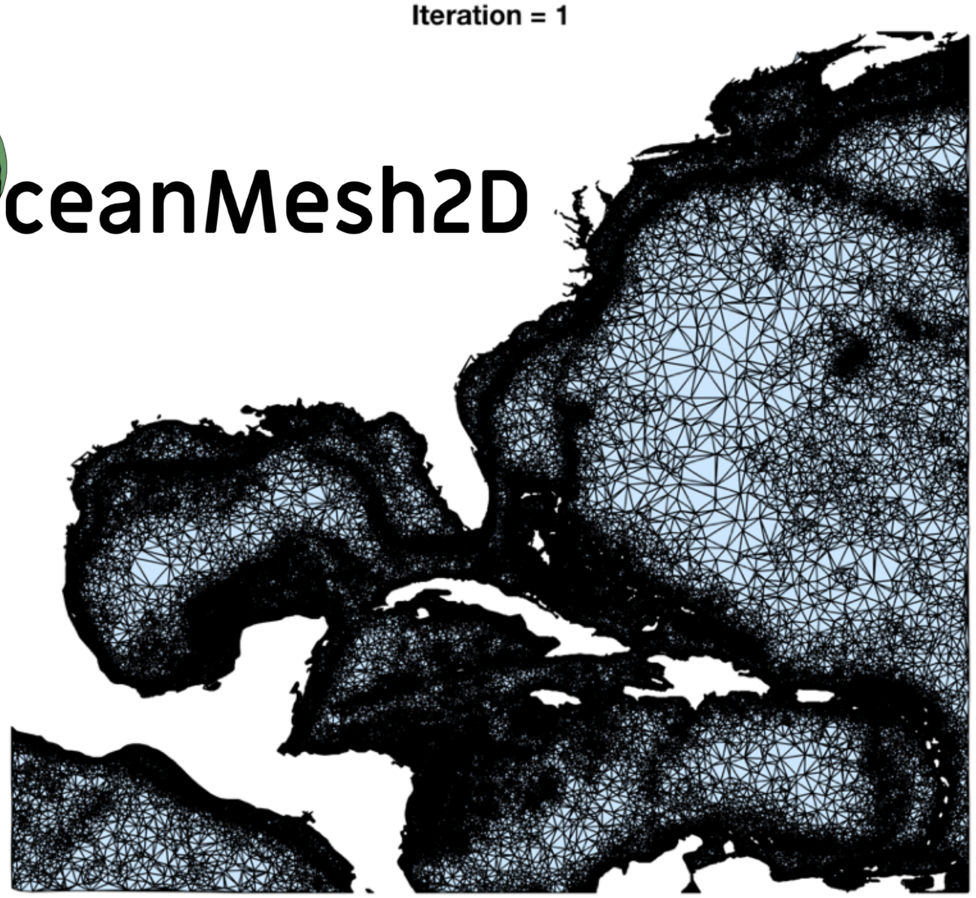
$$\mathbf{p_{n+1}} = \mathbf{p_n} + \Delta t \mathbf{F}(\mathbf{p_n})$$

****Points that exit are reprojected back into meshing domain**

1−2: Distribute points | 3: Triangulate | 4−7: Force equilibrium



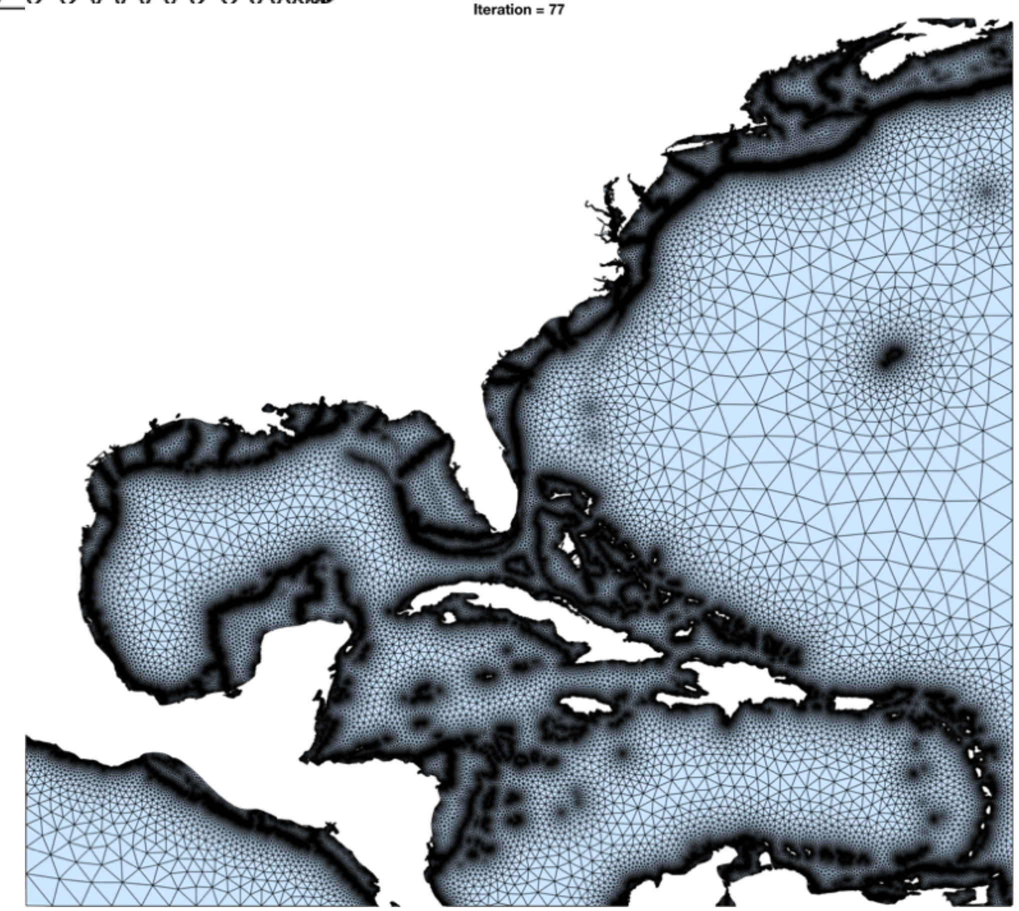
Iteration = 1



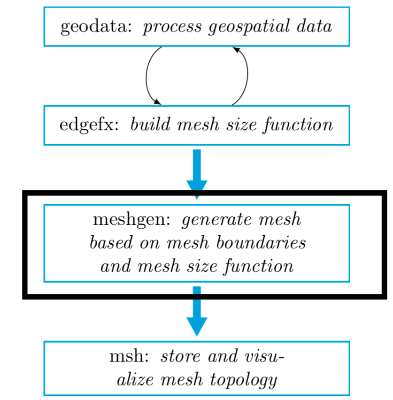
OceanMesh2D

$\mathcal{O}(minutes)$ ⟶

Iteration = 77



Meshgen



User passes the edgefx and geodata class instances

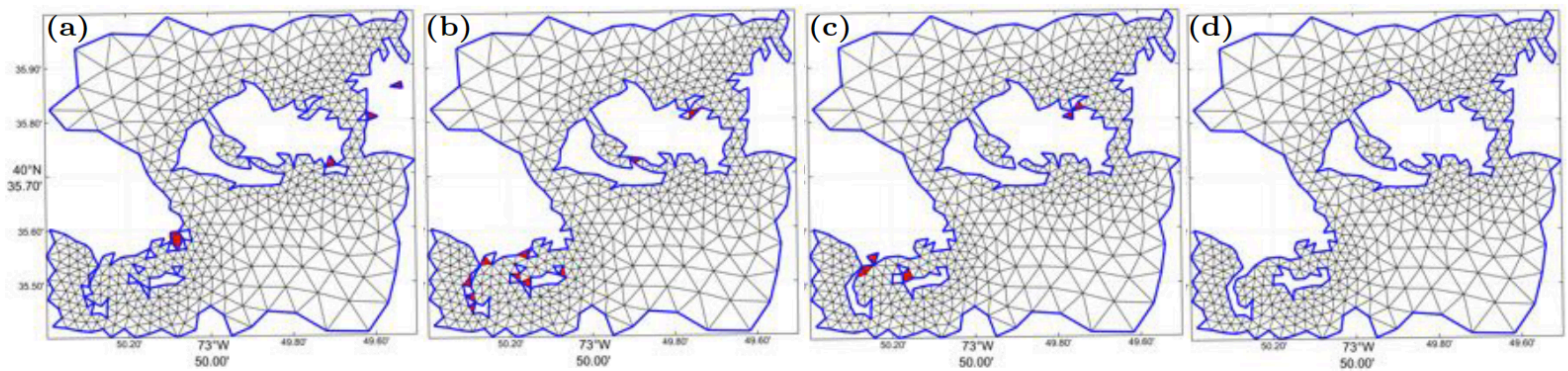
```
%% STEP 4: Pass your edgefx class object along with some meshing options and
% build the mesh...
mshopts = meshgen('ef',fh,'bou',gdat,'plot_on',1,'proj','lambert');
% now build the mesh with your options and the edge function.
mshopts = mshopts.build;
%% STEP 5: Plot it and write a triangulation fort.14 compliant file to disk.
m = mshopts.grd;
```

Boundary simplification

Cleaning algorithm leads to a valid mesh boundary for our finite element methods.

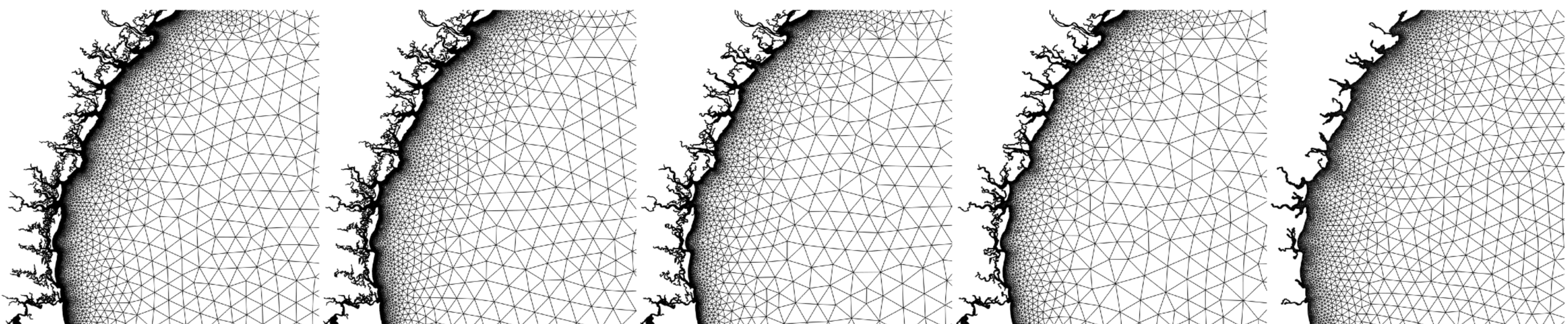
Repeat until no more geometric violations

Delete exterior disjoint → Delete interior disjoint → Delete exterior disjoint → Cleaned



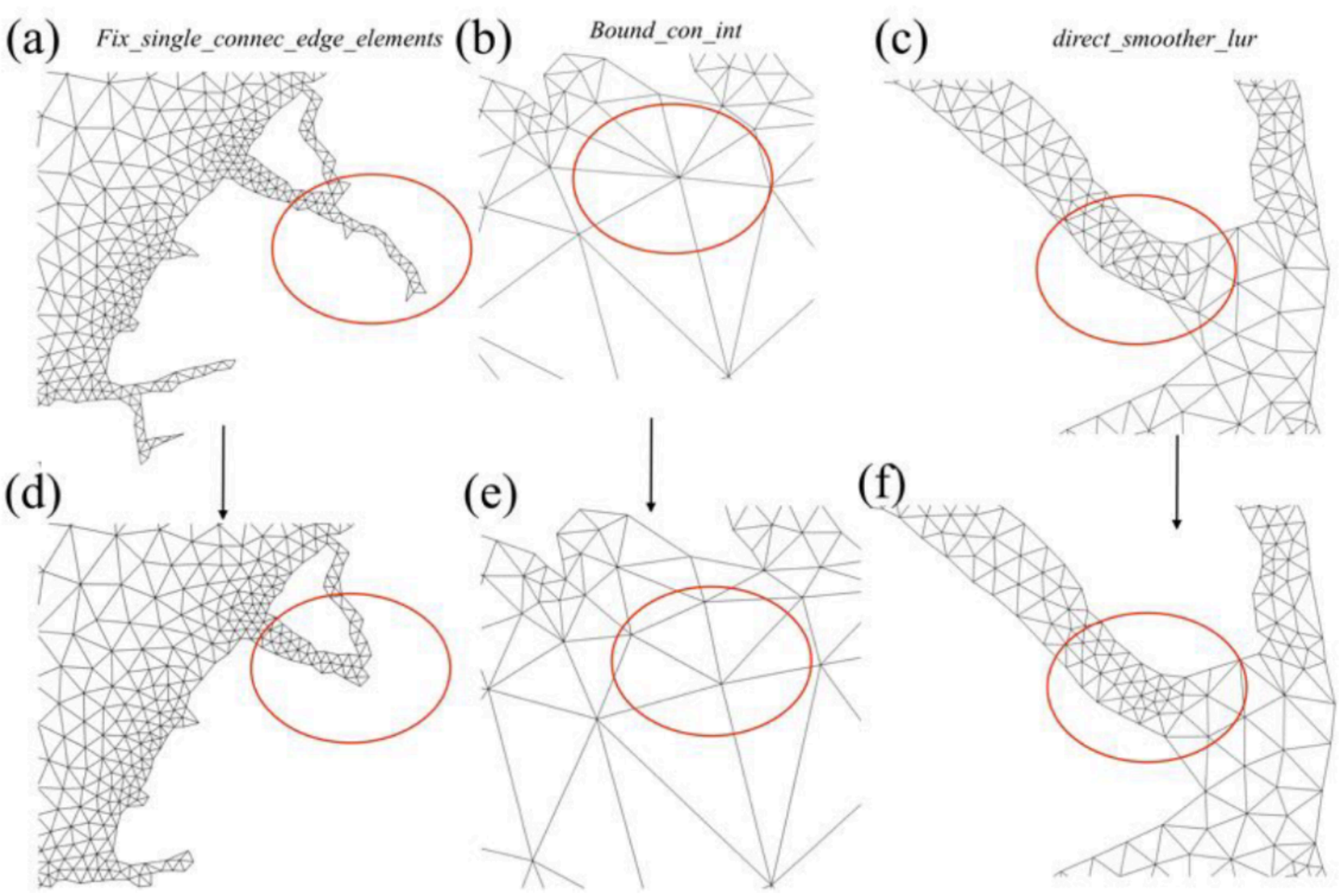
Number of boundary vertices = number of boundary segments

Finer shoreline resolution → Coarser shoreline resolution



Mesh improvement/boundary simplification

```
>> help m.clean
--- help for msh/clean ---

  [obj,qual] = clean(obj,varargin)
  obj - msh object
  varargin - optional base cleaning type followed by optional
  name-value pairs as listed below:

  base cleaning types: 'medium' (or 'default'), 'passive', 'aggressive'

  optional name-value pairs
  'db'  - boundary element cutoff quality (0 - 1)
  'ds'  - perform direct smoother? (0 or 1)
  'con' - upper bound on connectivity (6-19)
  'djc' - dj_cutoff (0 - 1 [area portion] or > 1 [km^2])
  'sc_maxit' - max iterations for deletion of singly connected
          elements ( >= 0, if set to 0 operation not performed)
  'mqa' - allowable minimum element quality (0 - 1); setting
          this too value high may prevent convergence
  'nscreen' - print info to screen? (default = 1)
  'pfix' - fixed points to keep (default empty)
  'proj' -to project or not (default = 1)

>> |
```

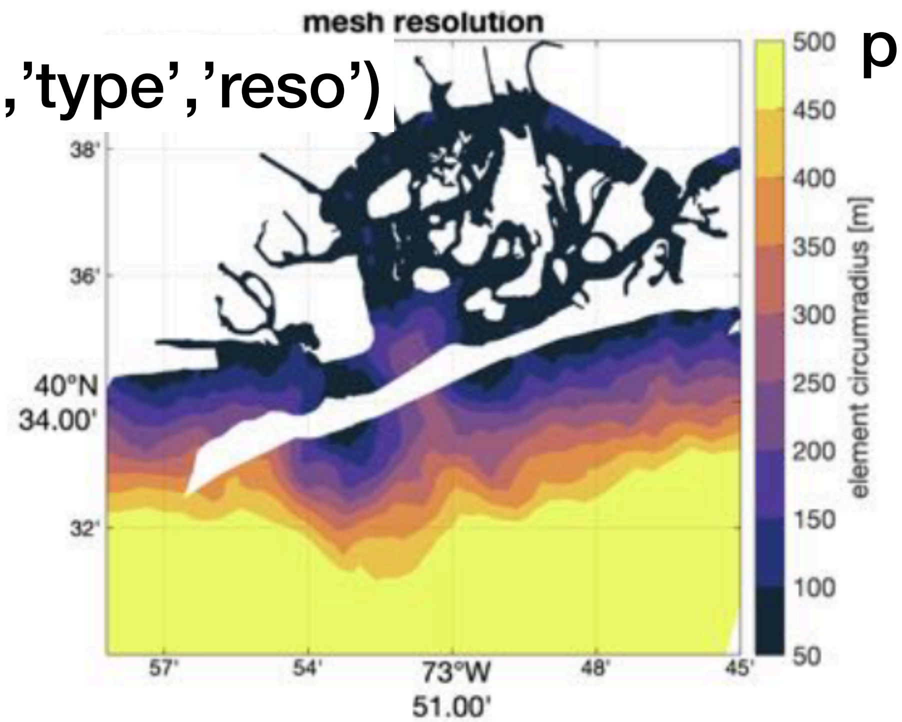


OceanMesh2D

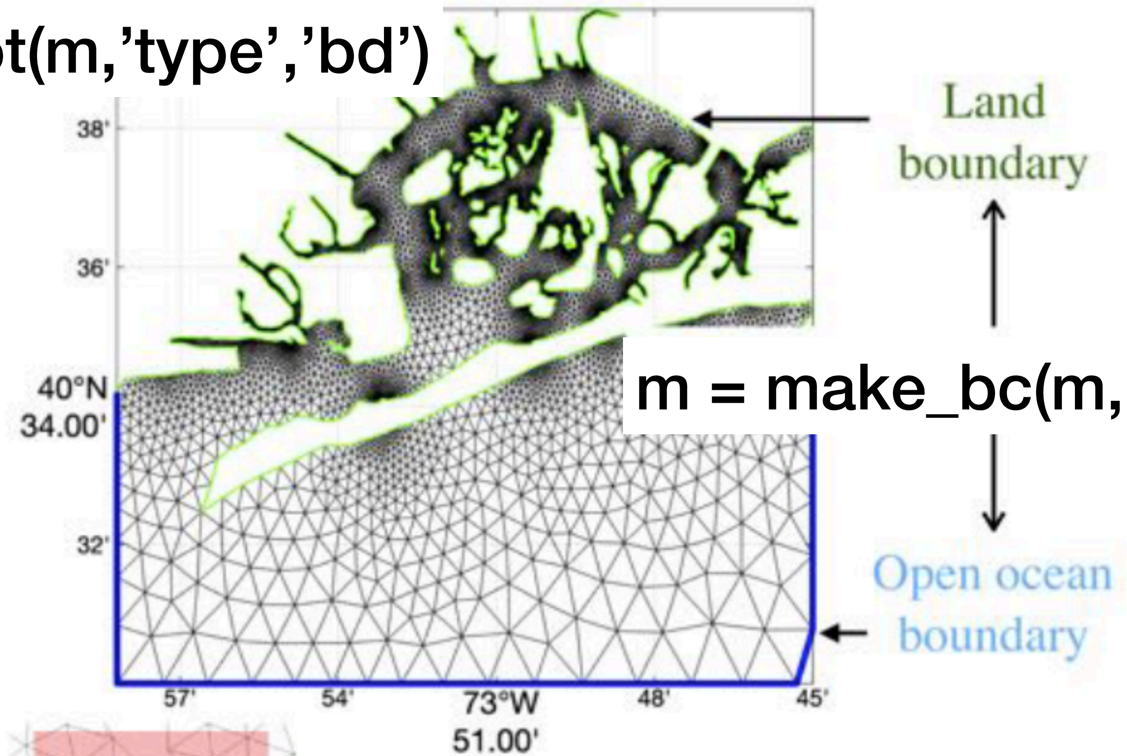
Msh class

OceanMesh2D

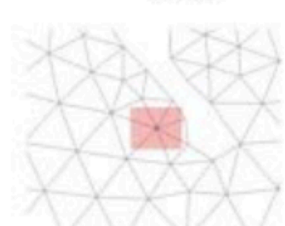
plot(m,'type','reso')

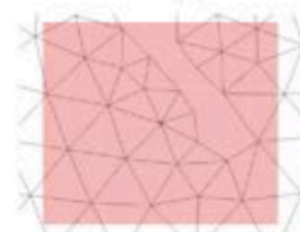


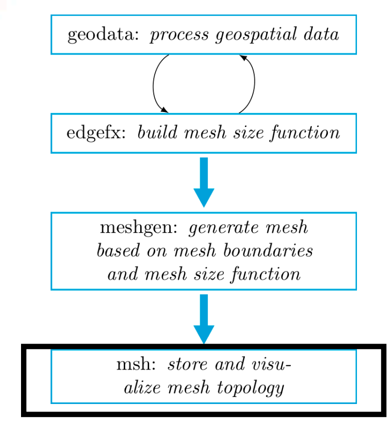
plot(m,'type','bd')



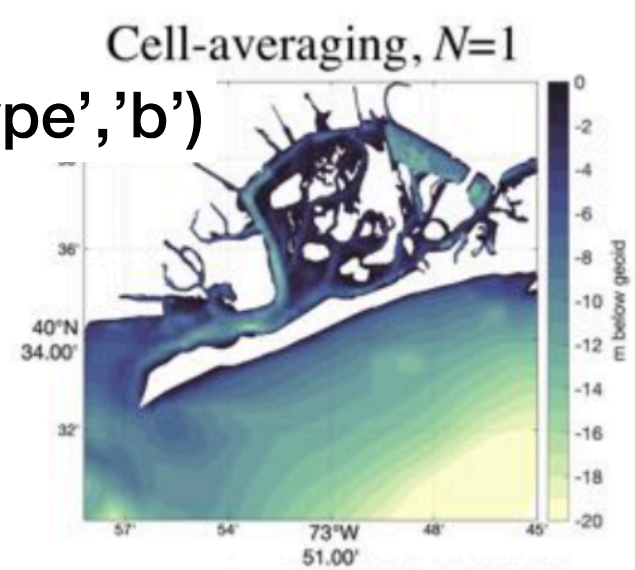
m = make_bc(m,'auto',gdat)

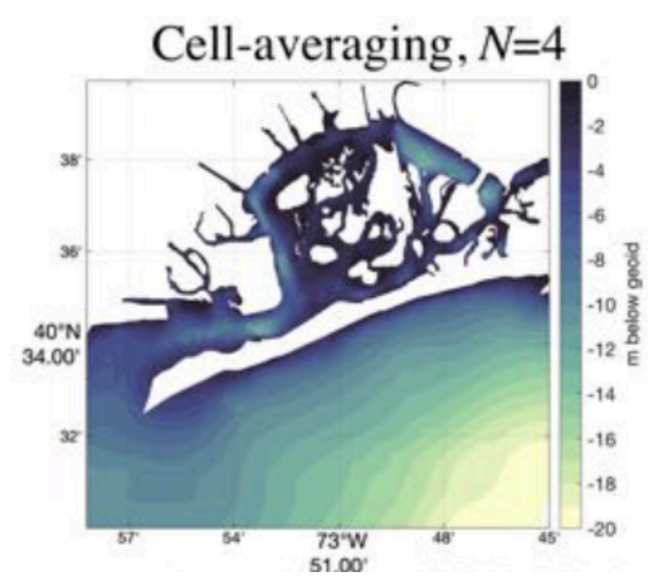


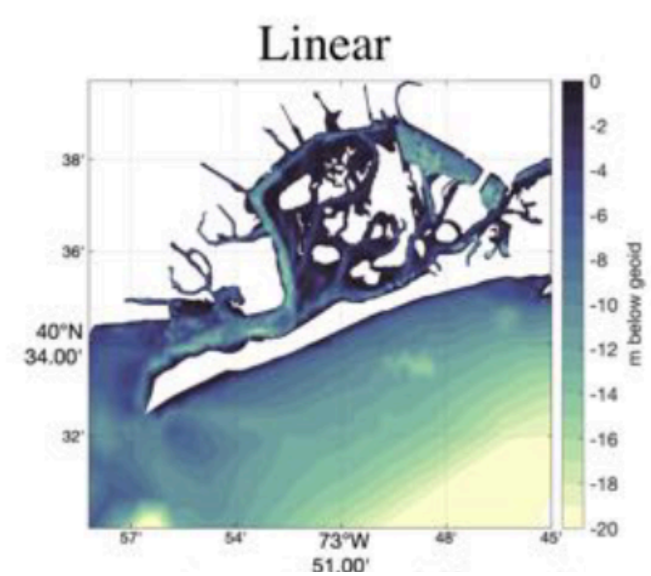




plot(m,'type','b')







m = interp(m,dem,'interp','CA','N',1)
[default]

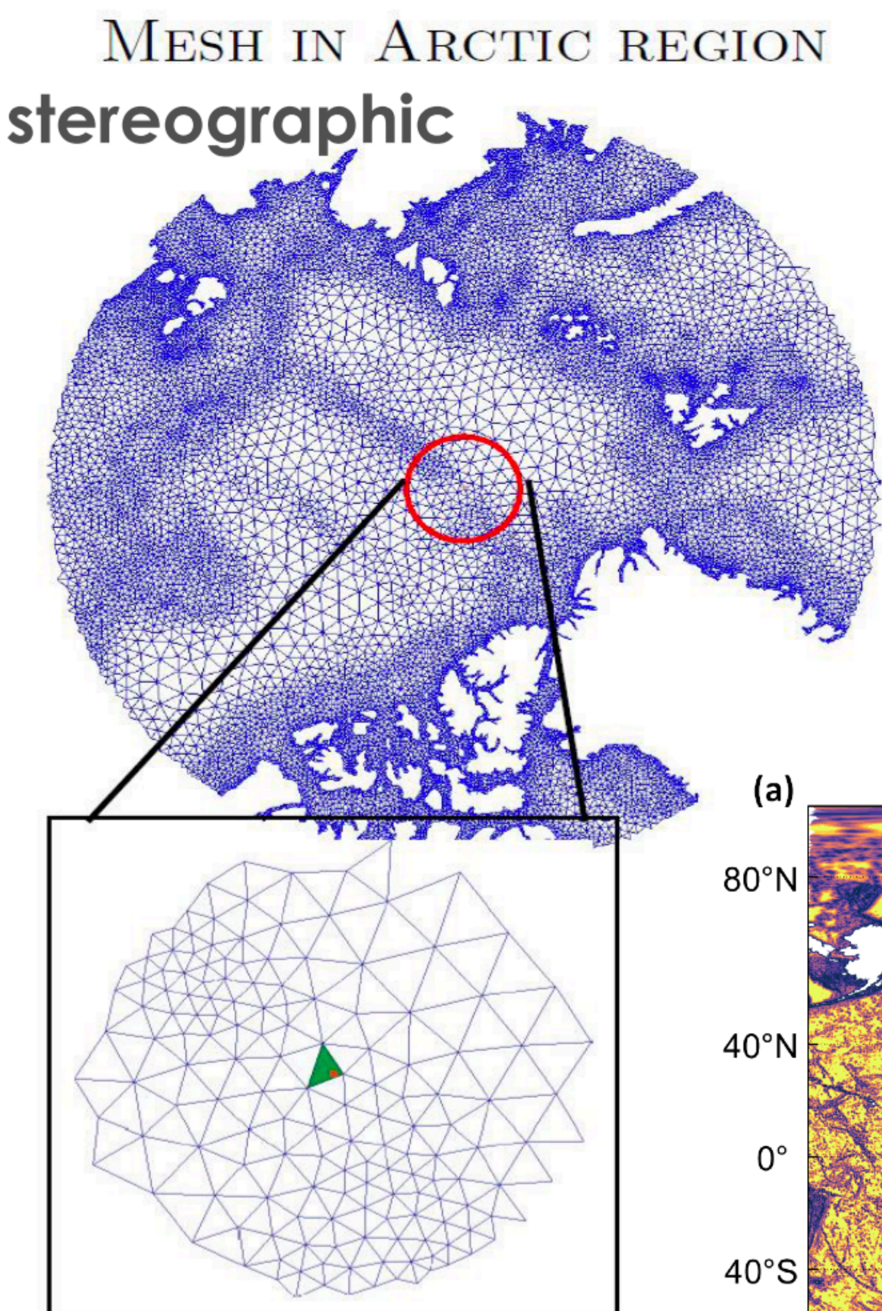
m = interp(m,dem,'interp','CA','N',4)

m = interp(m,dem,'interp','linear')

m = interp(m,dem,varargin)

Global meshing

Meshes can be built in a stereographic projection centered at the North Pole



https://github.com/CHLNDDEV/OceanMesh2D/

Branch: dev ▾ OceanMesh2D / Example_7_Global.m

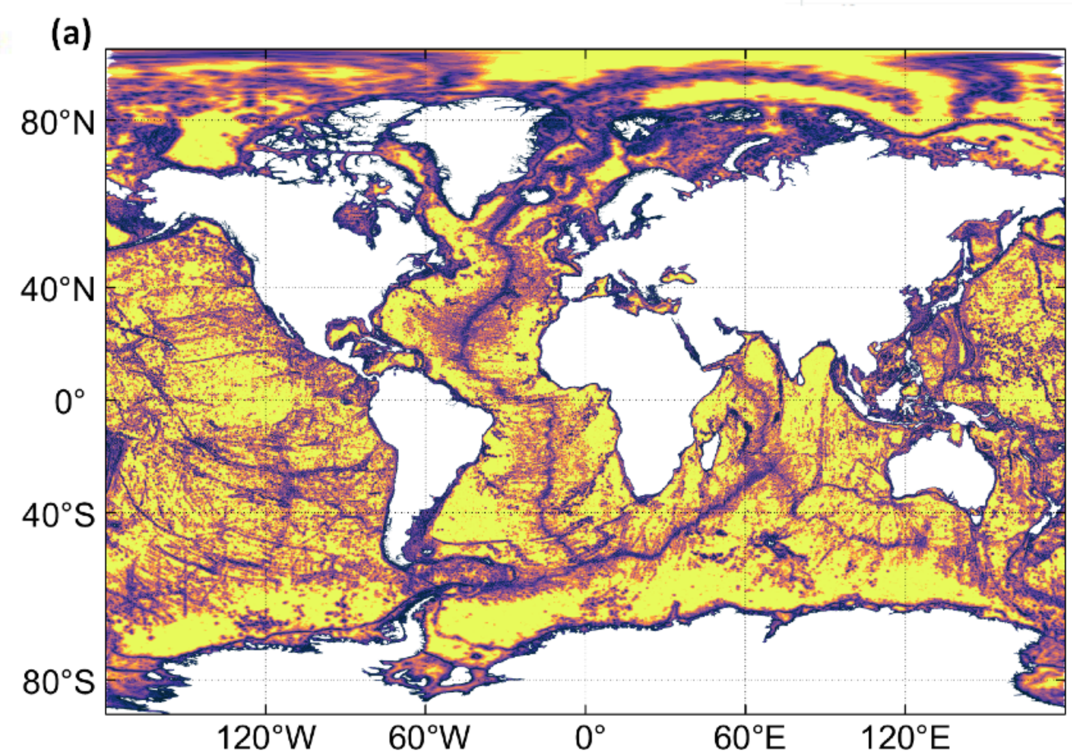
WPringle Update Example_7_Global.m

1 contributor

44 lines (37 sloc) | 1.65 KB

```
% Example_7_Global: Make a global mesh
clearvars; clc;
addpath(genpath('utilities/'));
addpath(genpath('datasets/'));
addpath(genpath('m_map/'));

%% STEP 1: set mesh extents and set parameters for mesh.
%% The globe
bbox      = [-180 180; -88  90]; % lon min lon max; lat min lat max
min_el    = 4e3;                % minimum resolution in meters.
max_el    = 20e3;                  % maximum resolution in meters.
wl        = 30;              % 30 elements resolve M2 wavelength.
dt        = 0;               % Only reduces res away from coast
grade     = 0.25;            % mesh grade in decimal percent.
R         = 3;                     % Number of elements to resolve feature.
slp       = 10;              % slope of 10

outname = 'Global_4km_20km';
```

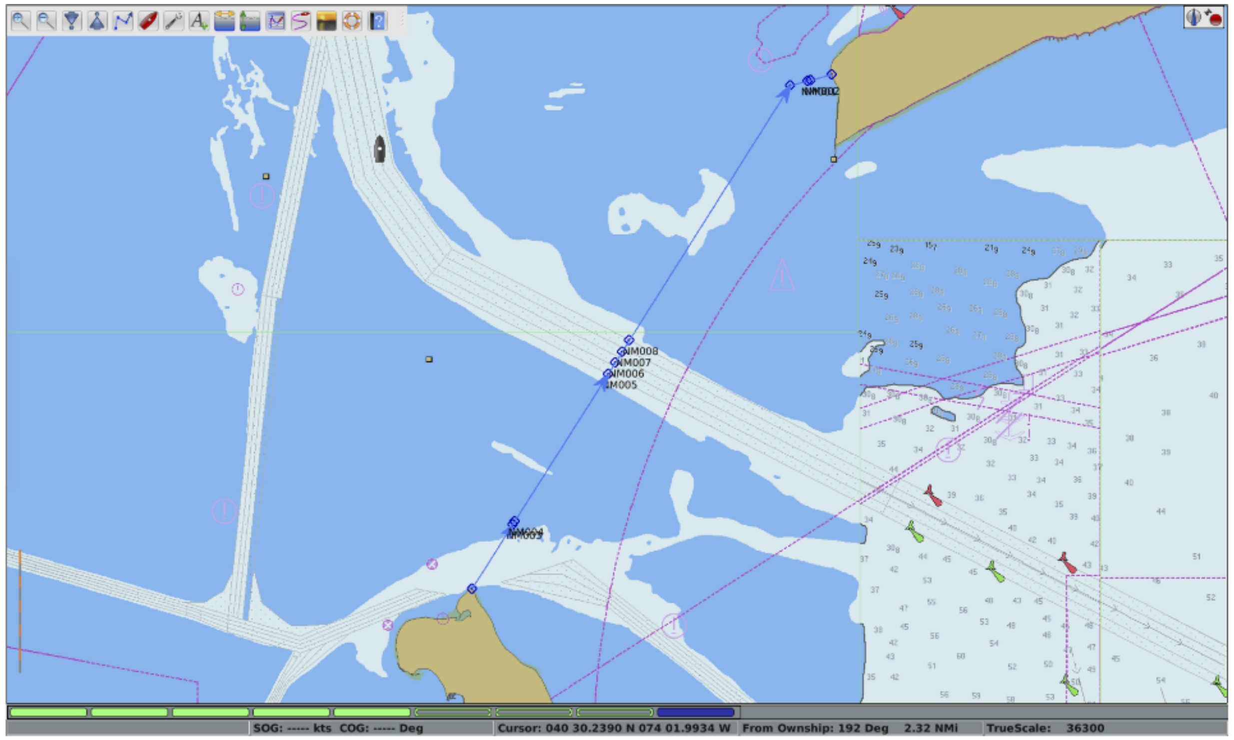
10

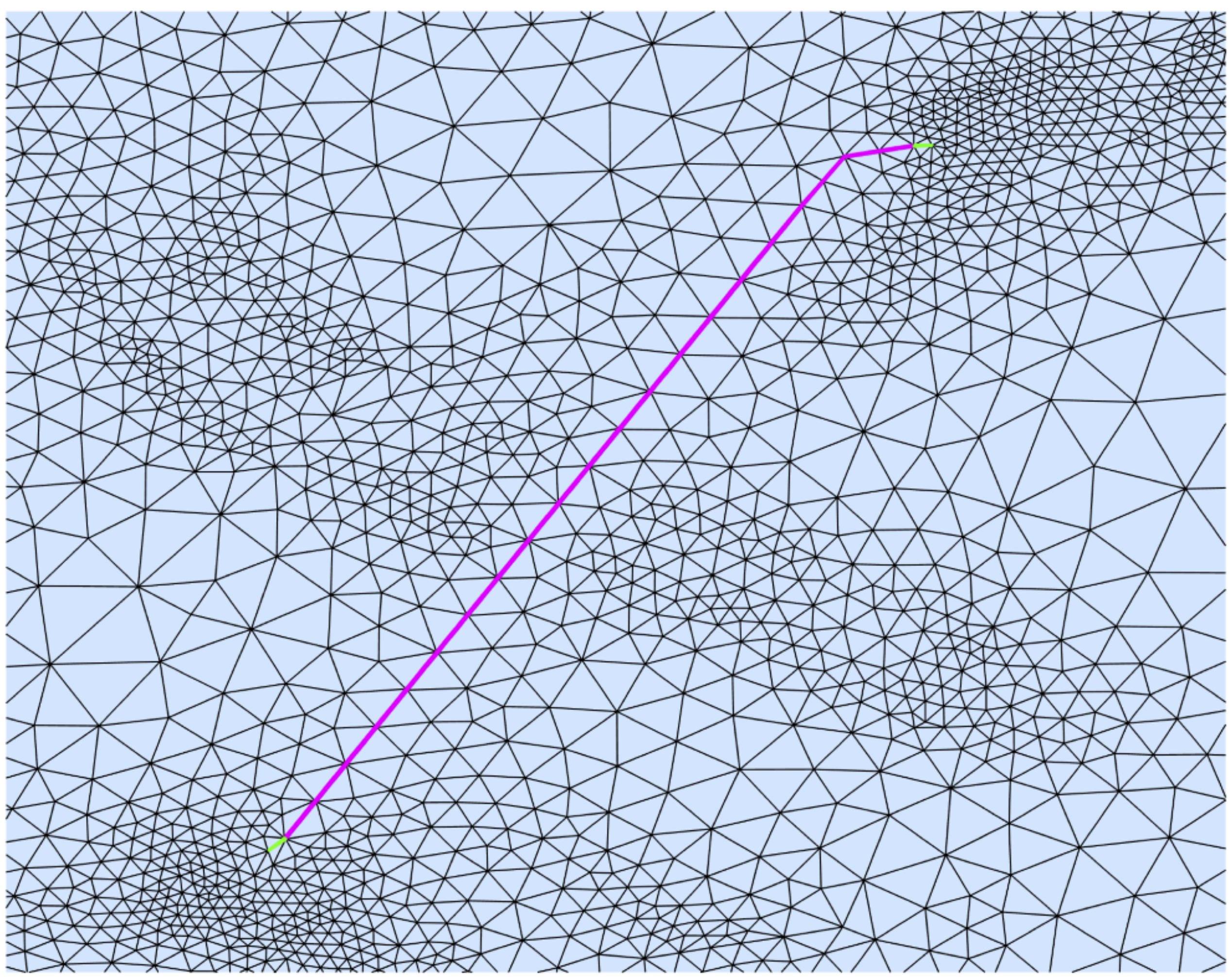


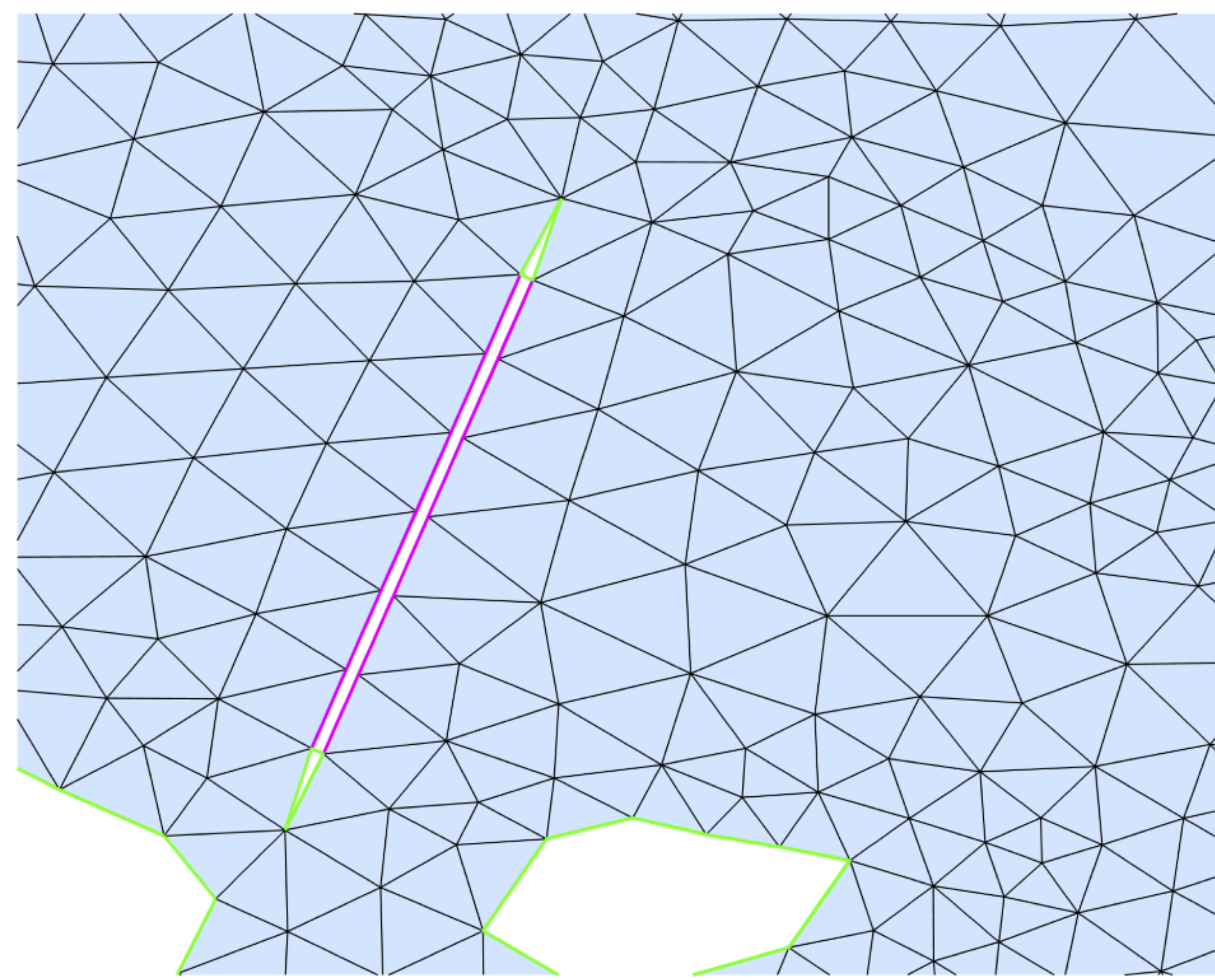
Pringle, W. J., Wirasaet, D., Roberts, K. J., & Westerink, J. (2021). Global Storm Tide Modeling with ADCIRC v55: Unstructured Mesh Design and Performance. *Geoscientific Model Development*, *14*(2), 1125–1145. https://doi.org/10.5194/gmd-14-1125-2021

OceanMesh2D

Meshing Barriers





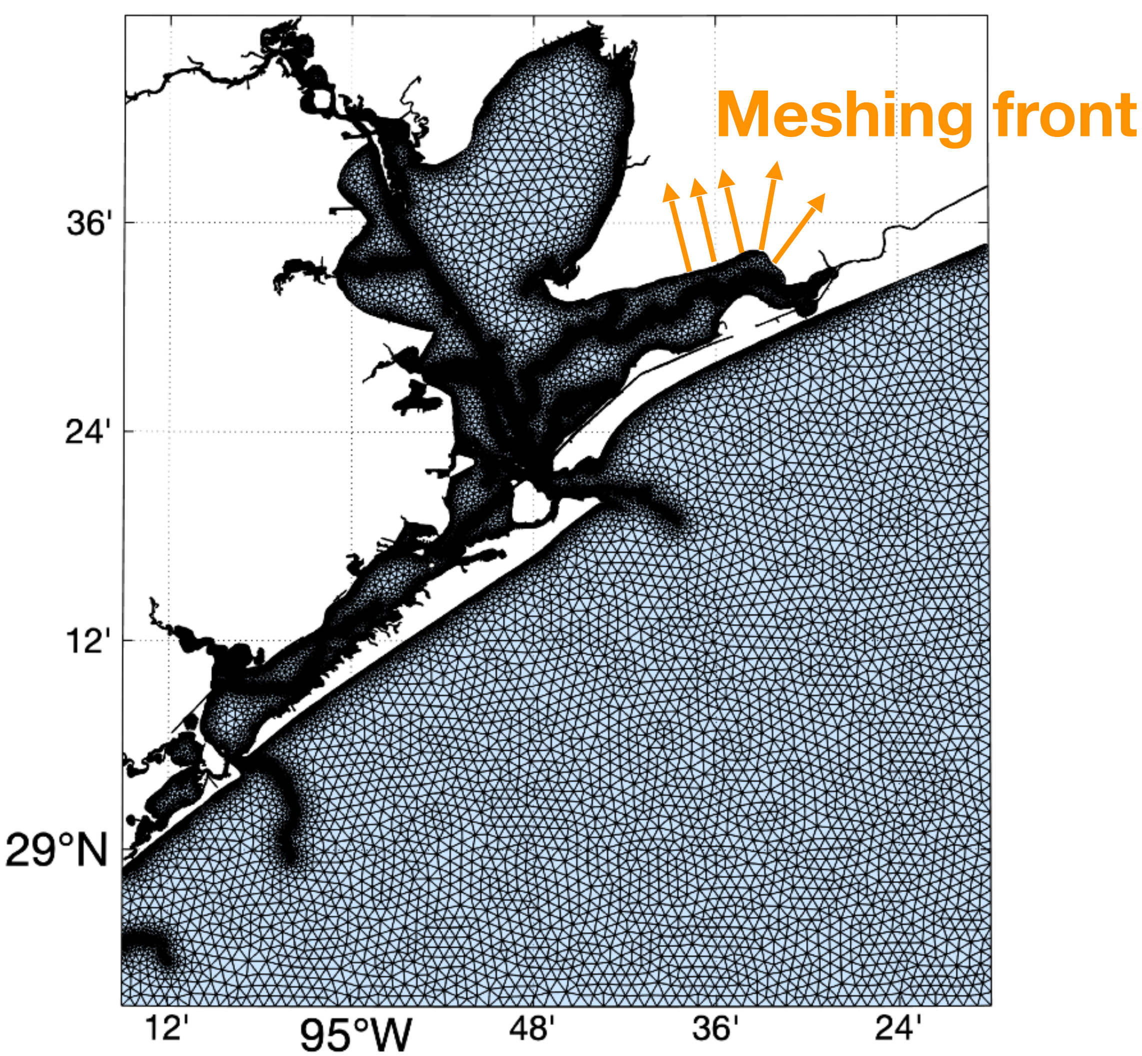


Floodplain meshing Strategies

Del. refine (using Mesh2D)

1. Build oceanside of domain

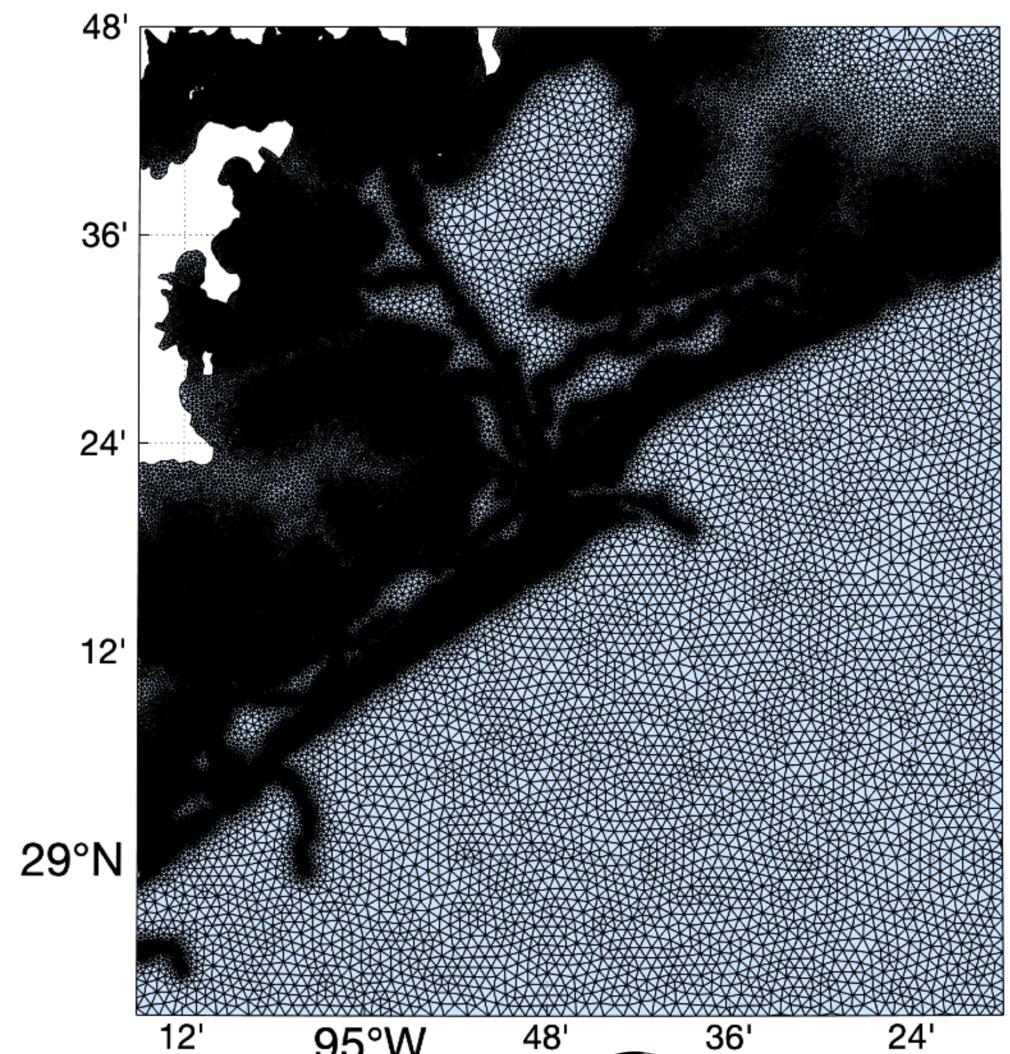
2. Build floodplain mesh propagating meshing front from oceanside boundary



Internal point and edge constraints

1. Build oceanside of domain

2. Build up to floodplain contour while constraining boundary of oceanside mesh

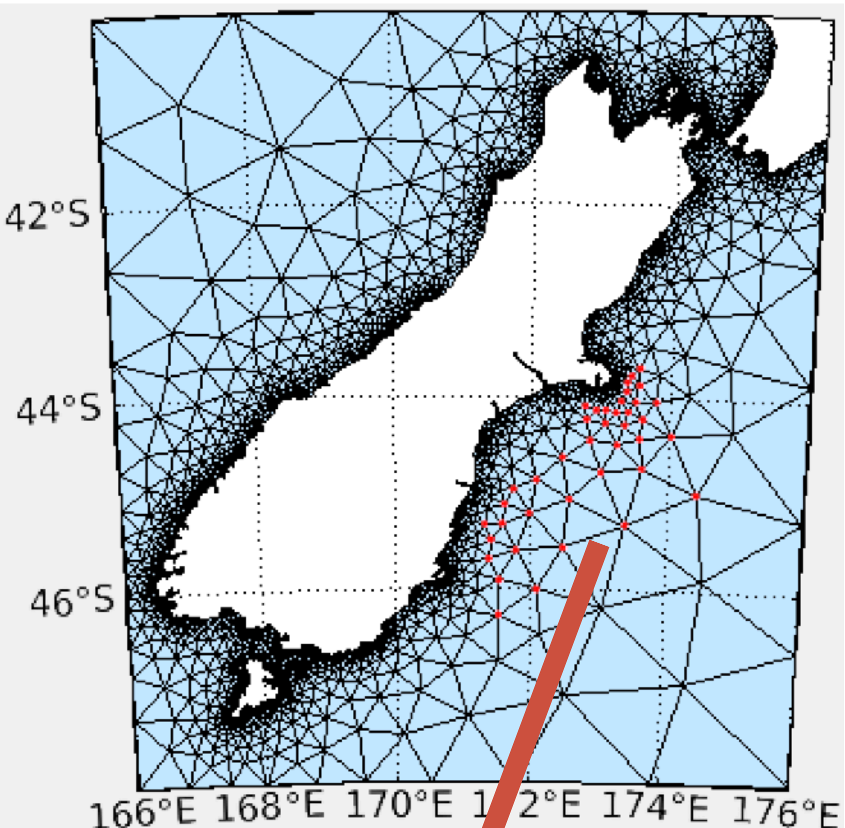


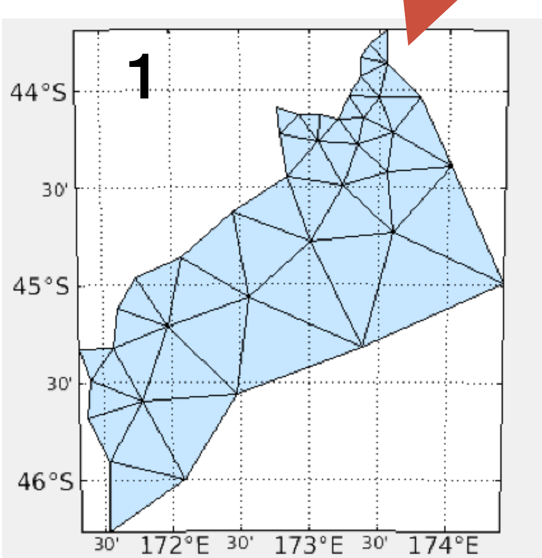
OceanMesh2D

OceanMesh2D

Merging Meshes 2

Original mesh



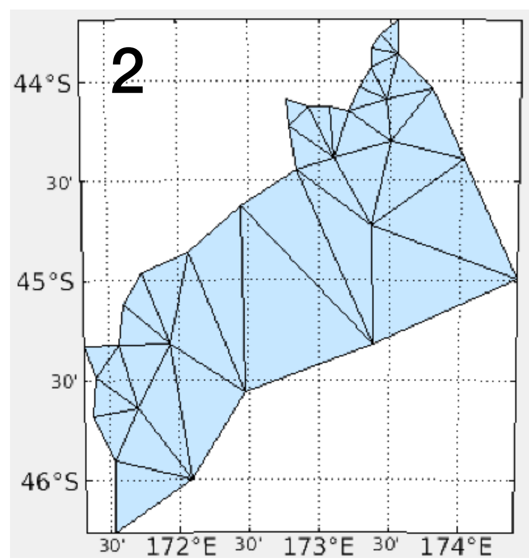


Match Merge

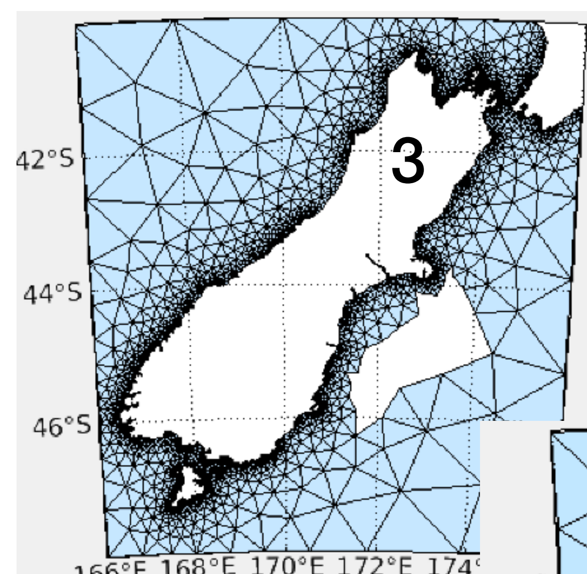
Merge meshes where the
boundary vertices perfectly align/match

m_merge = plus(m_fine,m_coarse,'match')

Example_11_Remeshing_Patches



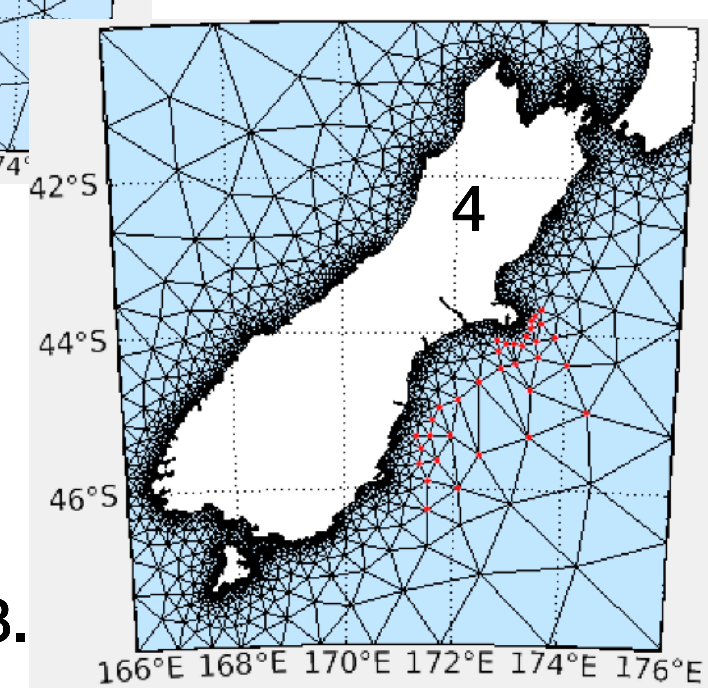
+



1. Extract out a portion of the mesh,
2. Remesh that portion using Delaunay front triangulation based on mesh boundary and edgefx
3. Extract that portion from the original mesh
4. Boundaries match so just match merge 2. & 3.

=

New mesh



OceanMesh2D Merging Meshes 1

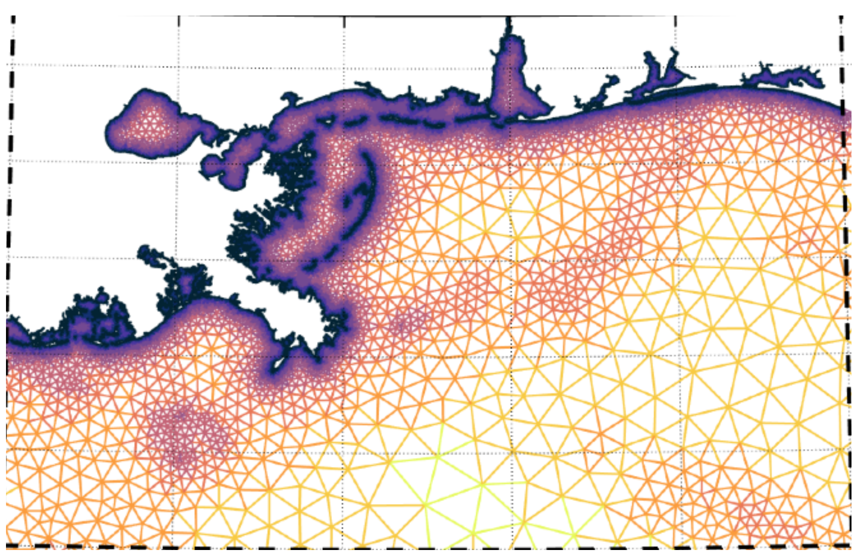
Arbitrary Merge

Merge two overlapping and
non-matching meshes together

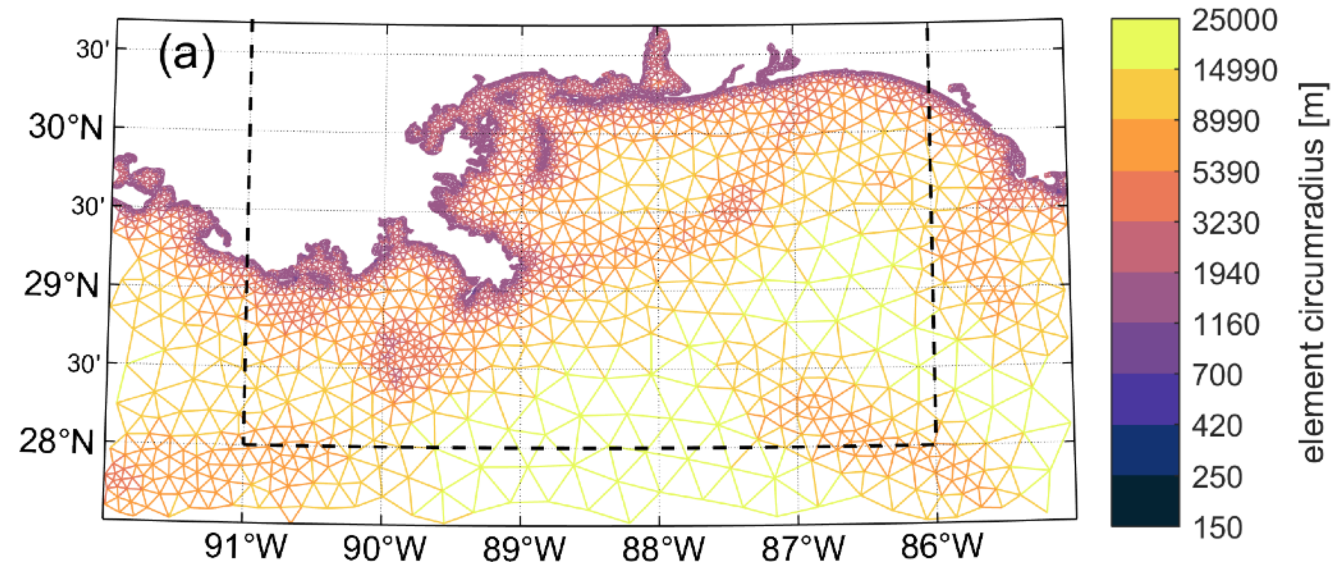
m_merge = plus(m_fine,m_coarse,’arb’)

[m_merge = m_fine + m_coarse]

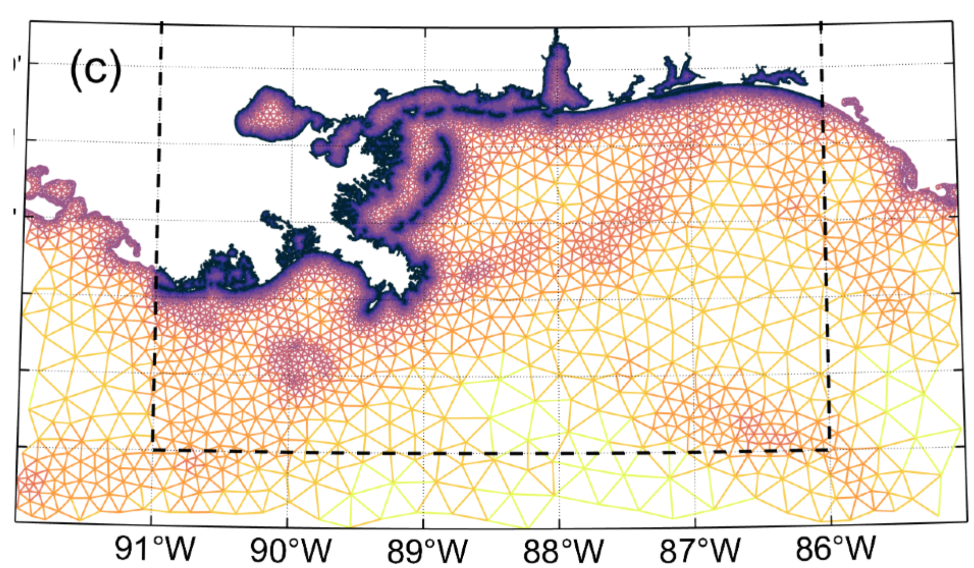
First entry takes priority!



+



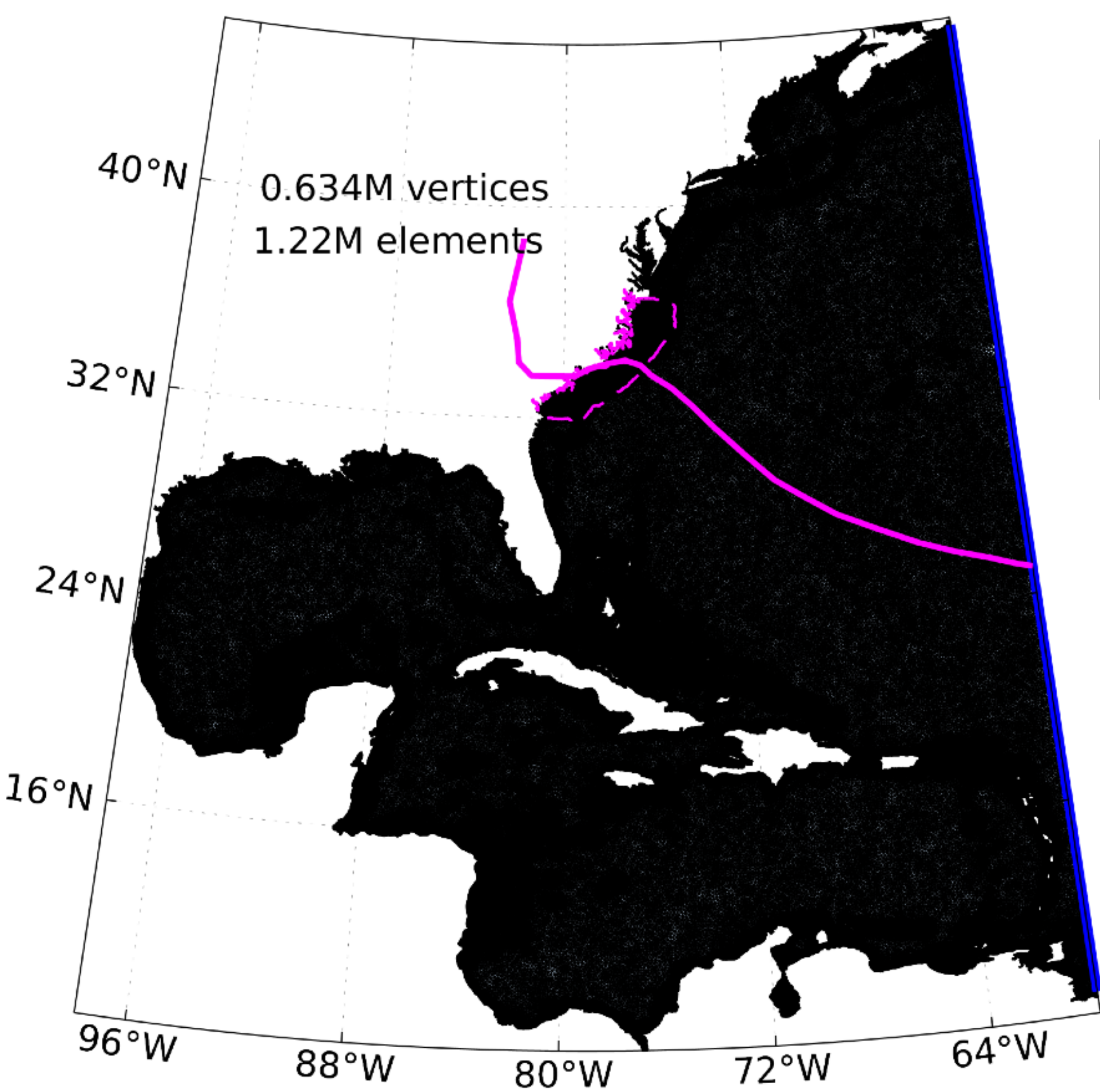
=

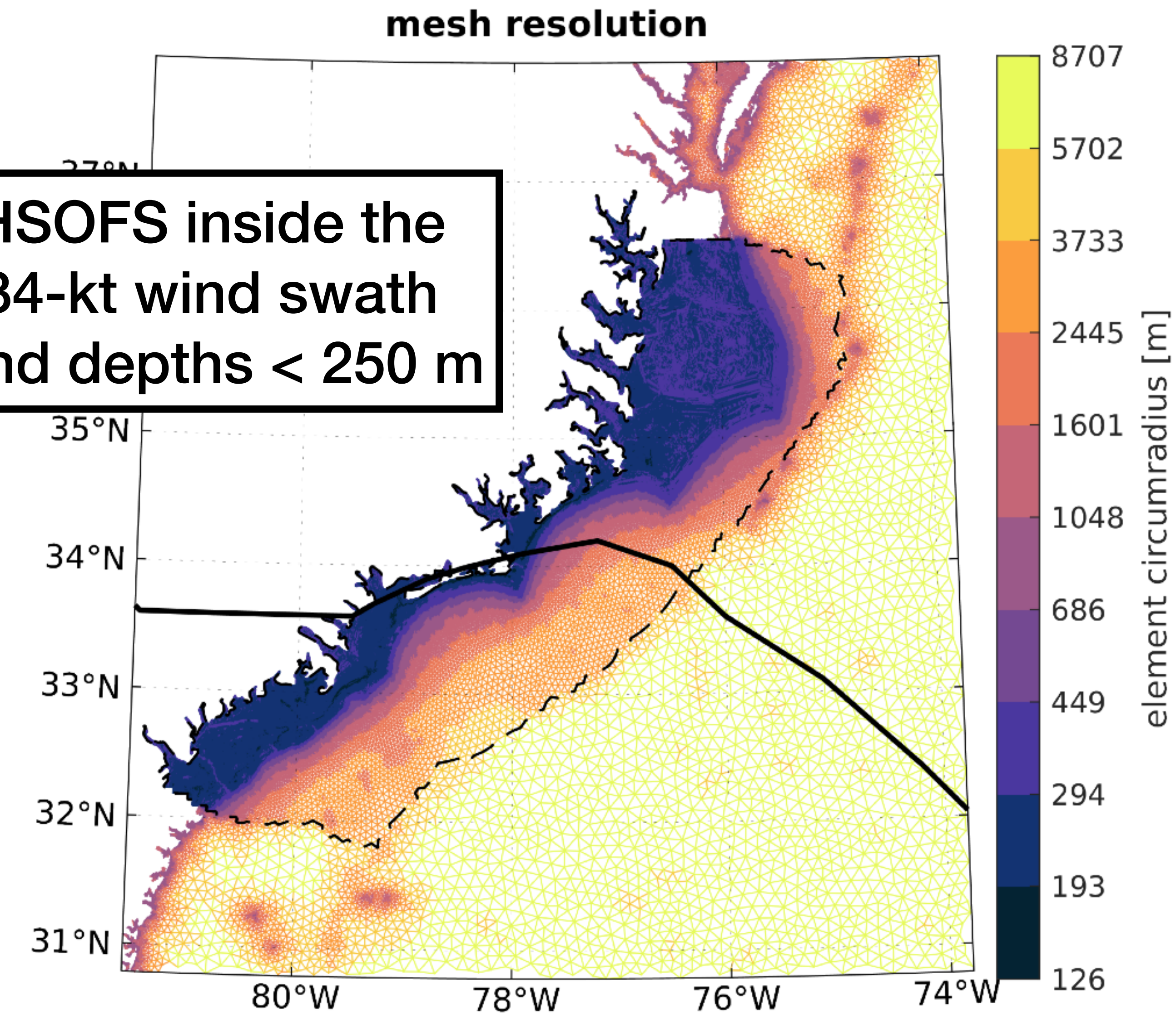


1. Get the mesh boundaries of both,
2. Perform ‘intersection’ polygon operation
3. Delete all coarse mesh vertices inside black box,
4. Concatenate all fine mesh vertices,
5. Retriangulate and clean

Subsetting and Merging based on Hurricane Track

Example for Atlantic HSOFS mesh and Hurricane Florence



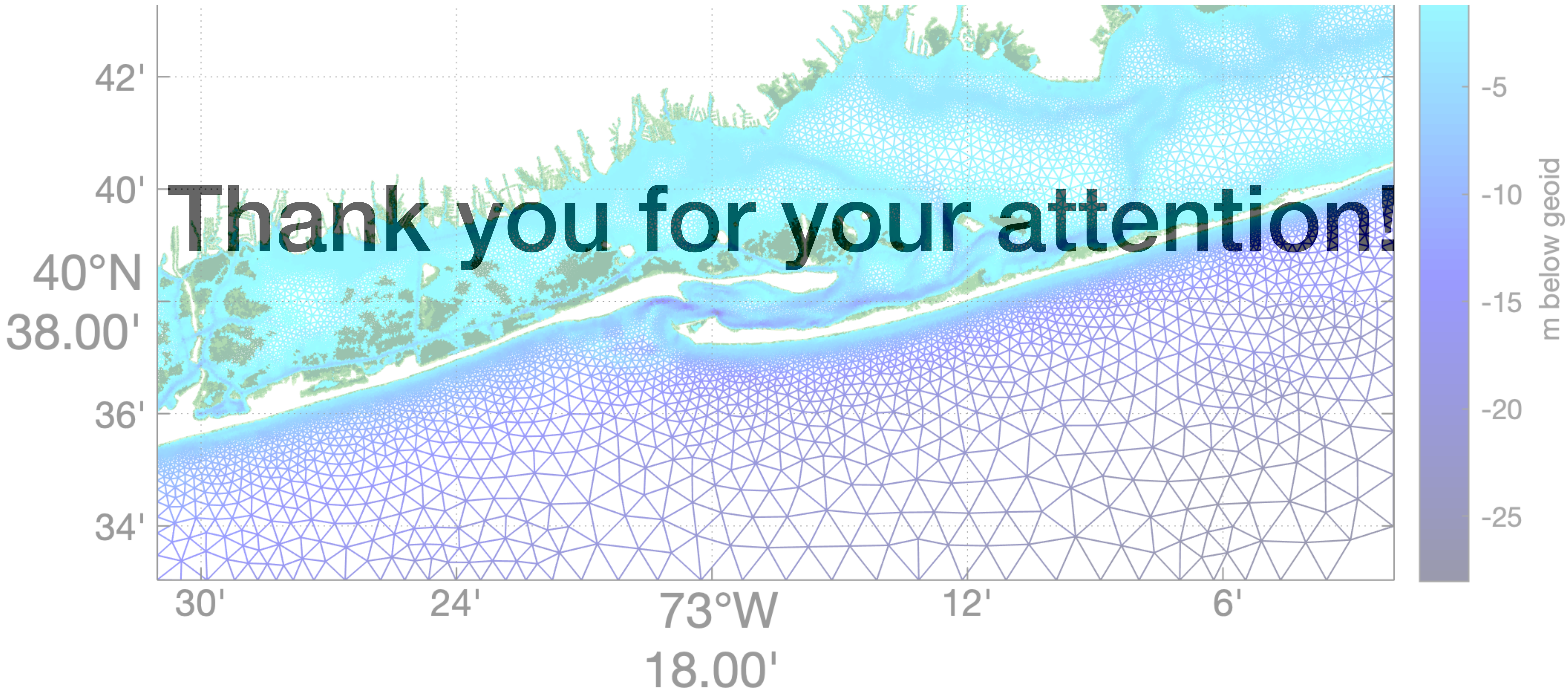


Private Repo:

```
Simply execute: tests/HSOFS_Florence/run_HSOFS_Florence.sh
```

https://github.com/WPringle/Storm-Surge-Mesh-Subsetting

OceanMesh2D


Thank you for your attention!
42'
40'
40°N
38.00'
36'
34'
30'
24'
73°W
18.00'
12'
6'
-5
-10
-15
-20
-25
m below geoid