Generation of High Quality 2D Meshes for Given Bathymetry

by

Jorge Colmenero

Submitted to the Department of Mechanical Engineering

In Partial Fulfillment of the Requirements for the Degree of

Bachelor of Science in Mechanical Engineering

at the

Massachusetts Institute of Technology

June 2014

Signature of Author ……….
Department of Mechanical Engineering
May 28, 2014

Certified by …….
Pierre Lermusiaux
Associate Professor of Mechanical Engineering
Thesis Supervisor

Accepted by …….
Anette Hosoi
Professor of Mechanical Engineering
Chairman, Undergraduate Thesis Committee

Generation of High Quality Meshes for Given Bathymetry

By

Jorge Colmenero

Submitted to the Department of Mechanical Engineering
On May 28, 2014 in Partial Fulfillment of the
Requirements for the Degree of

Bachelors of Science in Mechanical Engineering

Abstract

This thesis develops and applies a procedure to generate high quality 2D meshes for any given ocean region with complex coastlines. The different criteria used in determining mesh element sizes for a given domain are discussed, especially sizing criteria that depend on local properties of the bathymetry and relevant dynamical scales. Two different smoothing techniques, Laplacian conditioning and targeted averaging, were applied to the fields involved in calculating the sizing matrix. The L^2 norm was used to quantify which technique had the greatest preservation of the original field. In both the reduced gradient and gradient cases, targeted averaging had a lower L^2 norm. The sizing matrices were used as inputs for two mesh generators, Distmesh and GMSH, and their meshing results were presented over a set of ocean domains in the Gulf of Maine and Massachusetts Bay region. Further research into the capabilities of each mesh generator are needed to provide a detailed evaluation. Mesh quality issues near coastlines revealed the need for small scale feature size recognition algorithms that could be implemented and studied in the future.

Thesis Supervisor: Pierre F. J. Lermusiaux

Title: Associate Professor

Acknowledgements

I would like to start off by thanking Professor Pierre Lermusiaux for giving me the opportunity to work with the MSEAS group and providing me with the guidance and support that I needed to succeed. I would also like to thank the MSEAS group for their support these past 2 years. In particular, I would like to thank Mattheus Ueckermann, who mentored me throughout my UROP. MIT, the Mechanical Engineering department, and the Undergraduate Research Opportunities Program also deserve my thanks for all of the support that they have provided. To everyone who helped me with my thesis: Pat Haley with coding issues; Chris Mirabito, John Aoussou and Pierre with proof reading and formatting, thank you for your time. My final thanks is to my family, who have always believed in me and supported me.

Table of Contents

List of Figures

1. Introduction

1.1 Background and Motivation

This thesis focuses on 2D mesh generation in complex ocean domains, with a focus on finite-element simulations. A mesh is a collection of nodes that are used to represent computational fields used in modeling physical processes. For a given bounded domain, the mesh representation may be structured or unstructured. Structured meshes consist of a uniform element size throughout the entire mesh. This uniformity is beneficial in finite difference calculations but its computational limitations, particularly near coastlines, makes unstructured meshes better suited for modeling over bathymetry that requires a 2D discretization (Pain et al., 2005). Unstructured meshes are flexible because they permit the varying of resolution within the domain. This allows the existence of small and large elements within a single mesh, thereby efficiently allocating computational resources. In the case of bathymetry, features that help in determining the element size at a particular point are,

- Coastline – distance to the coastline (Legrand et al., 2006) and coastline curvature (Conroy, 2010)
- Bathymetric Changes - gradients and reduced gradients of the bathymetry (Legrand et al., 2006 and Conroy 2010)
- Depth (Legrand et al., 2006; Hagen et al., 2002; and Westerink et al., 1994)
- Small scale features (Terwisscha et al., 2012)

Past methods of generating meshes involved a significant amount of user input. Manual alterations to the parameters determining the final mesh are time consuming and prone to errors. The meshing approach developed for this research tries to expedite the process of taking bathymetric data and a user identified domain within the bathymetry to reliably obtain a mesh that is high quality and adheres to the original representation of the bathymetry as closely as possible. The resulting meshing procedure and meshing scripts that were implemented utilize existing meshing schemes, comparing and combining these schemes as needed. Of course, while our results expedite the process of mesh generation, there are still various points where user input and iteration are useful and in some cases required. Finally, one of our sources of motivation is to investigate meshing schemes and techniques for multi-resolution ocean modeling with high-order finite element methods (Ueckermann and Lermusiaux, 2010; Ueckermann, 2013).

1.2 Past Work

Prior work in the field of 2D meshing includes the unstructured mesh generator created by Hagen et al. (2002) that uses a local truncation error analysis (LTEA) to adjust the resolution of the mesh itself. *A priori* error estimates are permitted for the finite element mesh generated initially but afterwards, *posteriori* estimates control the mesh element sizes. BatTri, created by Bilgili et al. (2006), is another 2D unstructured mesh generation package that utilizes a graphical mesh editing tool and several bathymetry-based refinement algorithms that are complemented by set of utilities to check and improve grid quality. GMSH, developed by Geuzaine and Remacle, is a fast 3D finite element grid generator that was made specifically to be fast, light and user friendly (Geuzaine et al, 2009). ADmesh (advanced mesh generator), developed recently by Colton Conroy (Conroy, 2010), is another advanced automatic mesh generator for hydrodynamic models. Both the scripts written for this thesis and ADmesh build upon the work of Persson's and his mesh generator, Distmesh (Persson, 2004). The element sizing in both frameworks is dictated by similar parameters. ADmesh incorporates tidal adaptation and local feature size adaptation, something that the process presented here currently does not. To output a mesh, ADmesh just needs bathymetry data, a boundary and the target maximum and minimum element sizes. The main differences between the work about to be presented and ADmesh come from coastline establishment and the sizing function restrictions. An important part of this paper discusses how to meld the features that influence element sizes at a particular point into a single sizing function, $h(x)$.

1.3 Overview

A brief outline of this paper is as follows: In section 2, we discuss the mesh generation process, including the selection of a closed domain and the role of the sizing function. After parameterizing the coast, each point is then assigned a target element size. In section 3, we discuss the mapping of gradients and reduced gradients to element sizes. The results of two smoothing methods are also presented. To obtain the final sizing function, $h(x)$, the element sizes due to curvature and gradients are melded together. We then present a series of meshes generated using Distmesh and GMSH. We discuss possible improvements to the methods used in this thesis in section 4. Finally, a summary and conclusions are given in section 5.

2. Mesh Generation Process

In the following sections, the mesh generation process and methodologies developed will be discussed. First, a boundary within the bathymetry is extracted. Each point along the coasts is then assigned an element size that may or may not be used. Then, the gradients and/or the reduced gradients are limited by one of the two smoothing methods. The element sizes assigned to the coast are then diffused through the contributions from reduced and or absolute gradients.

The end result is the sizing matrix, $h(x)$, which is used by Distmesh (Perssons, 2004) and GMSH (Geuzaine, 2009) to specify the target mesh element size(the ideal edge length) at a particular location point. As stated in the introduction, 2D meshes of bathymetry typically incorporate reduced/absolute gradients, coastline curvature, and tidal wavelengths. In MSEAS case, we would replace the tidal wavelengths by characteristic dynamical length-scales, which would become the tidal wavelengths if this were the main length-scale at the particular location considered. The user sets a minimum and maximum element size based on available computational resources and then the mesh generation process scales the factors mentioned previously to accommodate these bounds. Once scaled, the contributions of each factor are combined into a single sizing matrix, $h(x)$, that will ideally result in a high quality mesh that allocates elements efficiently without sacrificing critical information.

2.1 Coastline Adaptation

The first step in creating a mesh is defining a closed domain. If the domain of interest contains portions of a coastline, the sizing matrix should account for the curvature along that coastline. The scripts written for the present thesis take the bathymetry and find every contour at a given depth depending on the height (e.g. 0 m coastline) at which the user wishes to mesh. The default is set to find every contour at 0 m. Contours with an insufficient number of points are discarded. Each contour, in ($x(i)$, $y(i)$) form, is then parameterized via,

$$s(i) = \sum_{r=1}^{i} \sqrt{(x(r) - x(r-1))^2 + (y(r) - y(r-1))^2}. \quad \text{(eq. 1)}$$

This parameterization is useful in calculating the curvature of the coastline at each point,

$$k(i) = \sqrt{(\frac{d^2x}{ds^2})^2 + (\frac{d^2y}{ds^2})^2}. \quad \text{(eq. 2)}$$

Values for $\frac{d^2x^2}{ds^2}$ and $\frac{d^2y}{ds^2}$ in eq. 2 are calculated using a first order differencing scheme. Based on the curvature of the coast and the user specified upper and lower element size limits due to curvature, esc_{curv}^{upper} and esc_{curv}^{lower}, each point is linearly assigned an element size, $esc(i)$, via the following equation,

$$esc(i) = esc_{curv}^{upper} - \frac{k(i)}{\max(k)}\left(esc_{curv}^{upper} - esc_{curv}^{lower}\right). \quad \text{(eq. 3)}$$

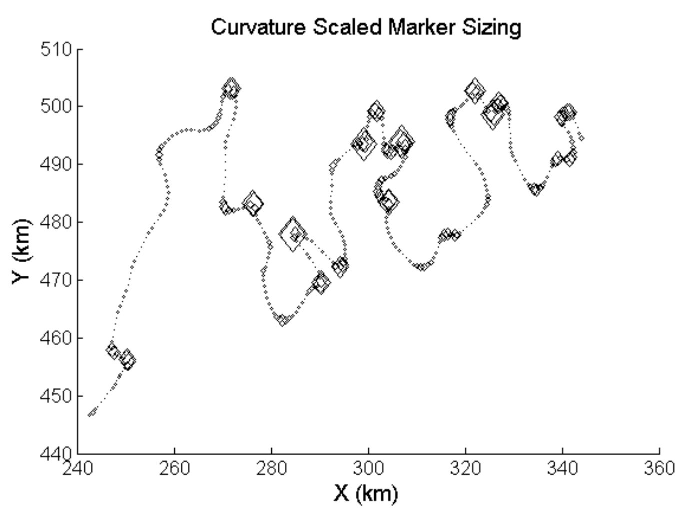

Figure 1: Curvature Scaled Marker Sizing. Curvature value mapped to size of diamond markers at each point along a subset of the Gulf of Maine coastline. (Large diamonds represent high curvature values. Note that coastline in Figure 1 is a smoothed version of the coastline extracted from the bathymetry.)

Element sizes at and near the coastline should be inversely related to the curvature because the higher the curvature, the smaller the mesh resolution needs to be to accurately discretize the domain near the coastline. Figure 1 displays the relative magnitude of curvature along a portion of the Gulf of Maine coastline. From looking at the graph, the linear assignment of element sizes based on curvature will result in $esc(i)$ at or near the upper limit. To ensure that the few areas of higher curvature, the larger diamonds in Figure 1, are not suppressed by the abundance of low curvature values, regions of low $esc(i)$ were expanded by replacing the element size at every node with the minimum $esc(i)$ of N of its neighbors and itself,

$$esc_{expand}(i) = \min\left\{esc\left(i-\frac{N}{2}\right), esc\left(i-\frac{N}{2}+1\right), esc\left(i-\frac{N}{2}+2\right), \dots, esc\left(i+\frac{N}{2}\right)\right\}.$$

(eq. 4)

Element resolution at the coastline is typically higher than resolution elsewhere in the domain, thus a representation of the boundary that takes into account the minimum and maximum element sizes tolerated is also important. If $x(i)$ and $y(i)$ are to be fixed then two scenarios arise. The first happens when the target element sizes at and near the coast are equal to or smaller than the distance between the points making up the coastline. If this is the case, GMSH and Distmesh have little issues creating a high quality distribution of elements near the coastline. As the target element size decreases relative to the spacing between coastline points, the quality of the mesh near the coast also increases. In the scenario where the target element sizes are larger than the distances between coastline points, mesh quality and convergence issues occur. To address this, there are two options: reducing the target element sizes near the coast or recreating the coastline with fewer points, ensuring increased spacing. The first option requires replacing $esc(i)$ with a weighted average of the distance to its direct neighbors, perhaps with a higher weight towards the lower distance as shown in equation 4.

$$esc_{avg}(i) = k_{esc}\min\{s(i+1), s(i-1)\} + (1-k_{esc})\max\{(s(i+1), s(i-1)\}$$

(eq. 5)

$$k_{esc} \in [0,1]$$

A value of $k_{esc} > .5$ will ensure that the smaller distance gets a higher weight. The second option of increasing the distance between points was implemented by using the parameterized coastline, $s(i)$, used in the initial curvature calculations. A re-parameterized curve was created by iterating through equations 5a and 5b. The initial conditions $s_{new}(1) = s(1)$ and $esc_{new}(1) = esc(1)$ were first set, then starting at $i = 2$ the following equations were solved:

$$s_{new}(i) = s_{new}(i-1) + esc_{new}(i-1)$$

(eq. 6a)

$$esc_{new} = interp(s(i), esc(i), s_{new}(i))$$

(eq. 6b)

$s_{new}(i)$ and $s(i)$ were then used to interpolate for a new set of values: $x_{new}(i)$, $y_{new}(i)$ and $k_{new}(i)$. The results of the equations above are shown in Figure 2 along with a comparison between using the original $esc(i)$ and the expanded version, $esc_{expand}(i)$.

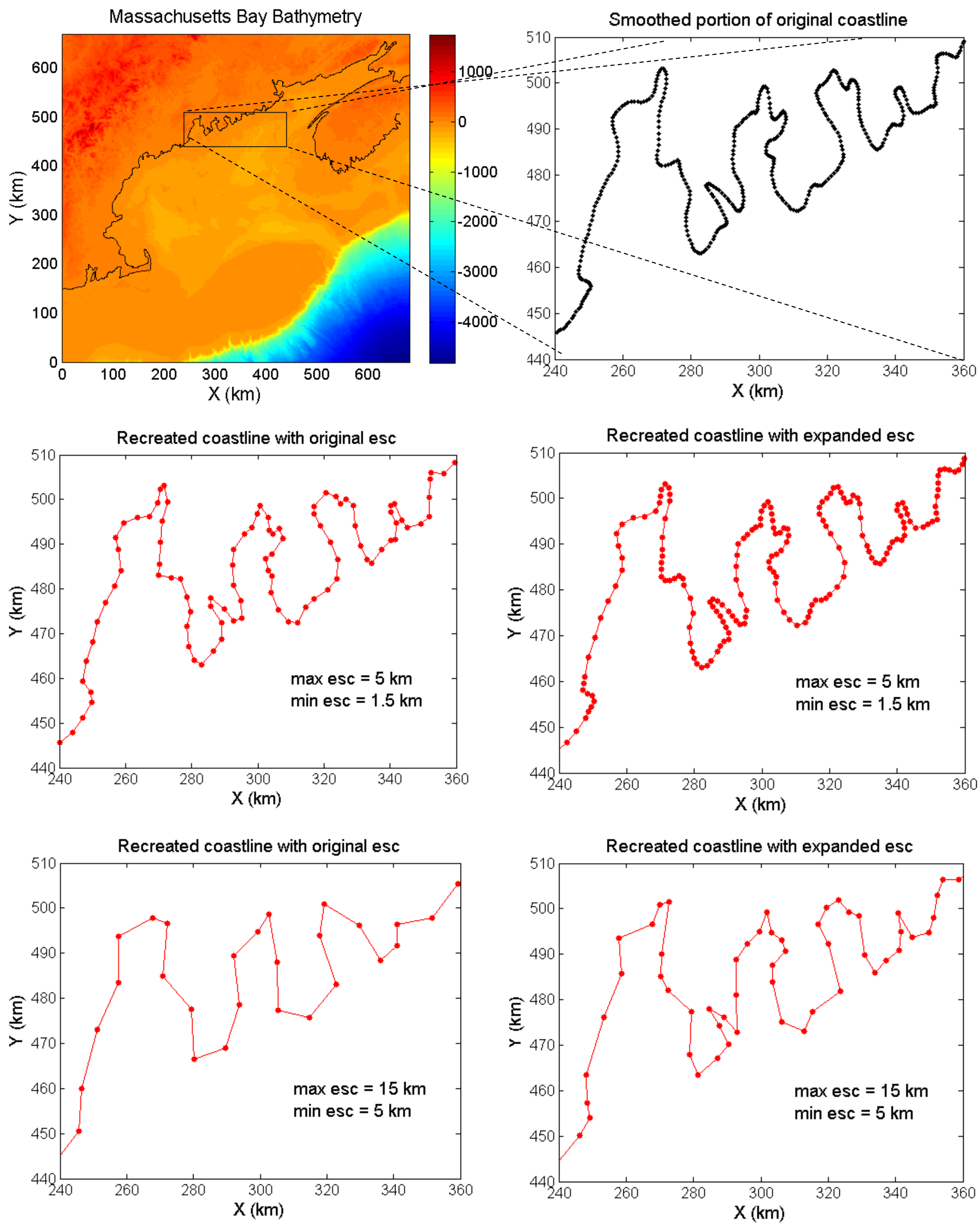

Figure 2: Recreation of a portion of the Gulf of Maine - Massachusetts Bay Coastline The top left plot shows the bathymetry used while the top right shows a smoothed version of the original coastline. The plots in the bottom left show the recreation of the original $esc(i)$ values while the plots in the bottom right show the recreation of the expanded , $esc_{expand}(i)$

Recreating the coastline with $esc_{expand}(i)$ results in a coastline with more points and a more accurate representation of the original coastline. While more features of the original coastline are removed as the average $esc(i)$ increases, the overall shape of the coastline remains intact. If certain features were critical to the simulations/models, the user could manually fix parts or the entire coastline. If fixing only by parts is employed, a scheme to meld the parts that were fixed with parts that were recreated may be required. This is discussed further in the future work section 5.

2.2 Gradient Adaptation

To calculate the gradients of the bathymetry, a pseudo-staggered grid was utilized.

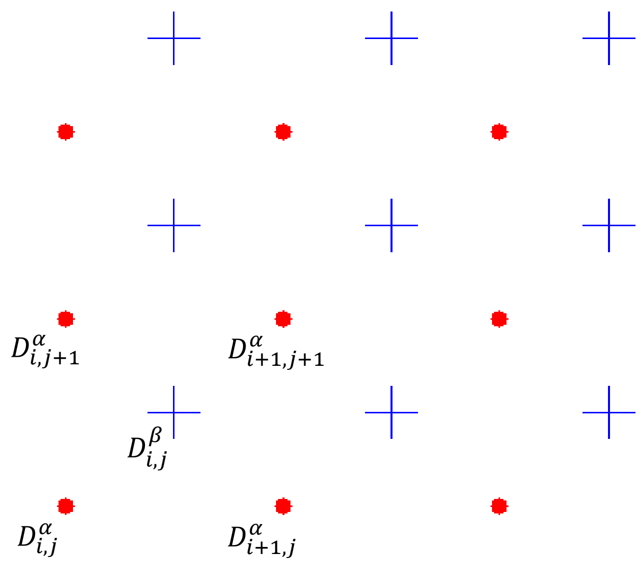

Figure 3. Staggered Grid Indexing. The bathymetry is located von the red α nodes. The gradients were calculated on the blue β

The red markers (α) represent the bathymetric nodes while the blue markers (β) show where the gradients where calculated. This means that the gradients were calculated half a unit to the north and half a unit to the east of each bathymetric node. In the equation below, $D_{i,j}$ represents the depth at node $(i,\ j)$. The quantities Δx and Δy are calculated using a MATLAB package that uses a plane approximation for the distance between two latitude and longitude coordinate pairs.

$$\|\nabla D\|^{\beta}_{i,j} = \sqrt{((\frac{1}{2\Delta x}[(D_{i+1,j+1} - D_{i,j+1}) + (D_{i+1,j} - D_{i,j})])^2 + (\frac{1}{2\Delta y}[\left(D_{i+1,j+1} - D_{i+1,j}\right) + (D_{i,j+1} - D_{i,j})])^2)}$$

(eq. 7)

The gradients are then linearly mapped to element sizes. The issue with this is the potential for outlier gradient values to skew the linear mapping.

This issue is illustrated in Figure 4. The plot on the left shows the gradients calculated on the β grid and the plot on the right displays the gradient values in histogram form to show the skewed distribution. The maximum gradient is 547.96 but the mean is much lower at 16.93. While a linear mapping results in an ideal representation of the gradients in terms of element sizes, without some form of smoothing, the mesh will appear mainly uniform. When smoothing bathymetry, the goal is to keep alterations to the original bathymetry to a minimum. Thus, the smoothing process has to selectively target areas in violation of user identified conditions and the altering methods have to preserve the original bathymetry as much as possible. The two smoothing methods tested for smoothing bathymetry were targeted averaging and Laplacian conditioning with a no normal boundary condition. To compare the magnitude of preservation, the L^2 norm was used.

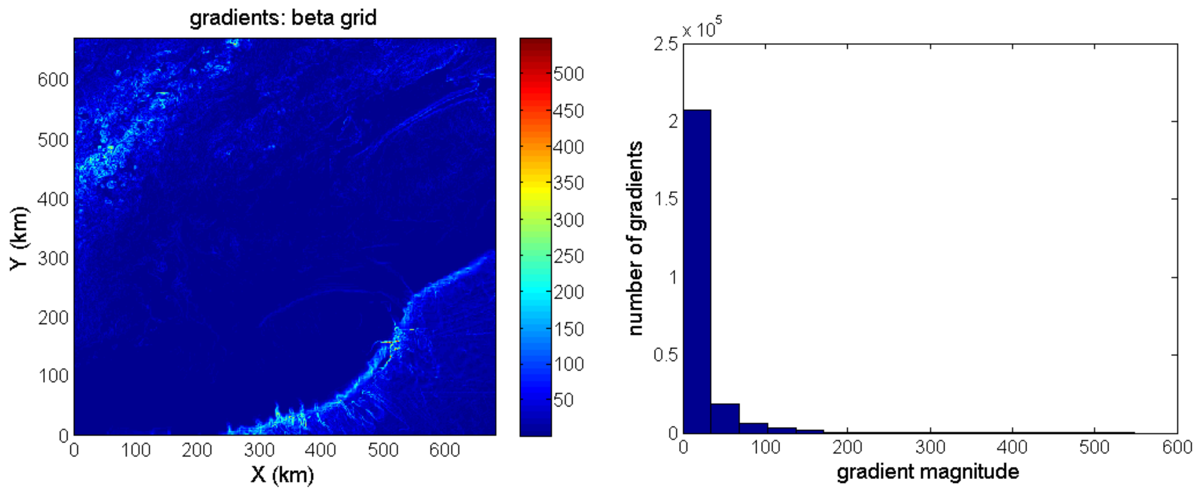

Figure 4. Gradients for the Gulf of Maine - Massachusetts Bay Region.
Maximum gradient = 547.96, average gradient = 16.93

Targeted averaging works by replacing every node that violates the gradient constraint, $(\nabla D)^{\beta}_{i,j} > G_{max}$, with a weighted average of the surrounding nodes and itself. For nodes inside the boundary, the equation is,

$$D^{new}_{i,j} = \frac{1}{4}D_{i,j} + \frac{1}{8}\left(D_{i+1,j} + D_{i,j-1} + D_{i-1,j} + D_{i,j+1}\right) + \frac{1}{16}\left(D_{i+1,j+1} + D_{i+1,j-1} + D_{i-1,j-1} + D_{i-1,j+1}\right).$$

(eq. 8)

For nodes on one of the outer edges, but not the corners, the five surrounding nodes and the node itself were used in calculating a new depth. The equation below applies to the left edge:

$$D^{new}_{i,j} = \frac{3}{8}D_{i,j} + \frac{1}{8}\left(D_{i-1,j} + D_{i-1,j+1} + D_{i-1,j-1} + D_{i,j+1}D_{i,j-1}\right).$$

(eq. 9)

The corners use the surrounding three nodes and the node itself. The equation below shows the convention used for the upper left corner:

$$D_{i,j}^{new} = \frac{1}{3}D_{i,j} + \frac{2}{9}\left(D_{i-1,j} + D_{i-1,j-1} + D_{i,j-1}\right).$$

(eq. 10)

Since the gradients are calculated on a staggered grid, the bathymetric nodes in violation of the gradient constraint are the four nodes used in calculating the original gradient itself,

$$\left(D_{i+1,j}, D_{i,j}, D_{i,j+1}, D_{i+1,j+1}\right)^{\alpha}.$$

After a few iterations, depending on how low the maximum gradient constraint is, the number of nodes in violation of the maximum gradient constraint stagnates. A way to further smooth the field and reach the target maximum gradient constraint is to expand the areas in violation. By tagging the nodes that surround the node in violation, the code will replace those values with the weighted average of its surrounding neighbor values and itself. If stagnation still occurs, the area in violation is expanded again until eventually the maximum gradient constraint is achieved. Figure 5 shows the results of targeted averaging on the Gulf of Maine - Massachusetts Bay region for the gradient constraints, G_{max}, 100 and 50.

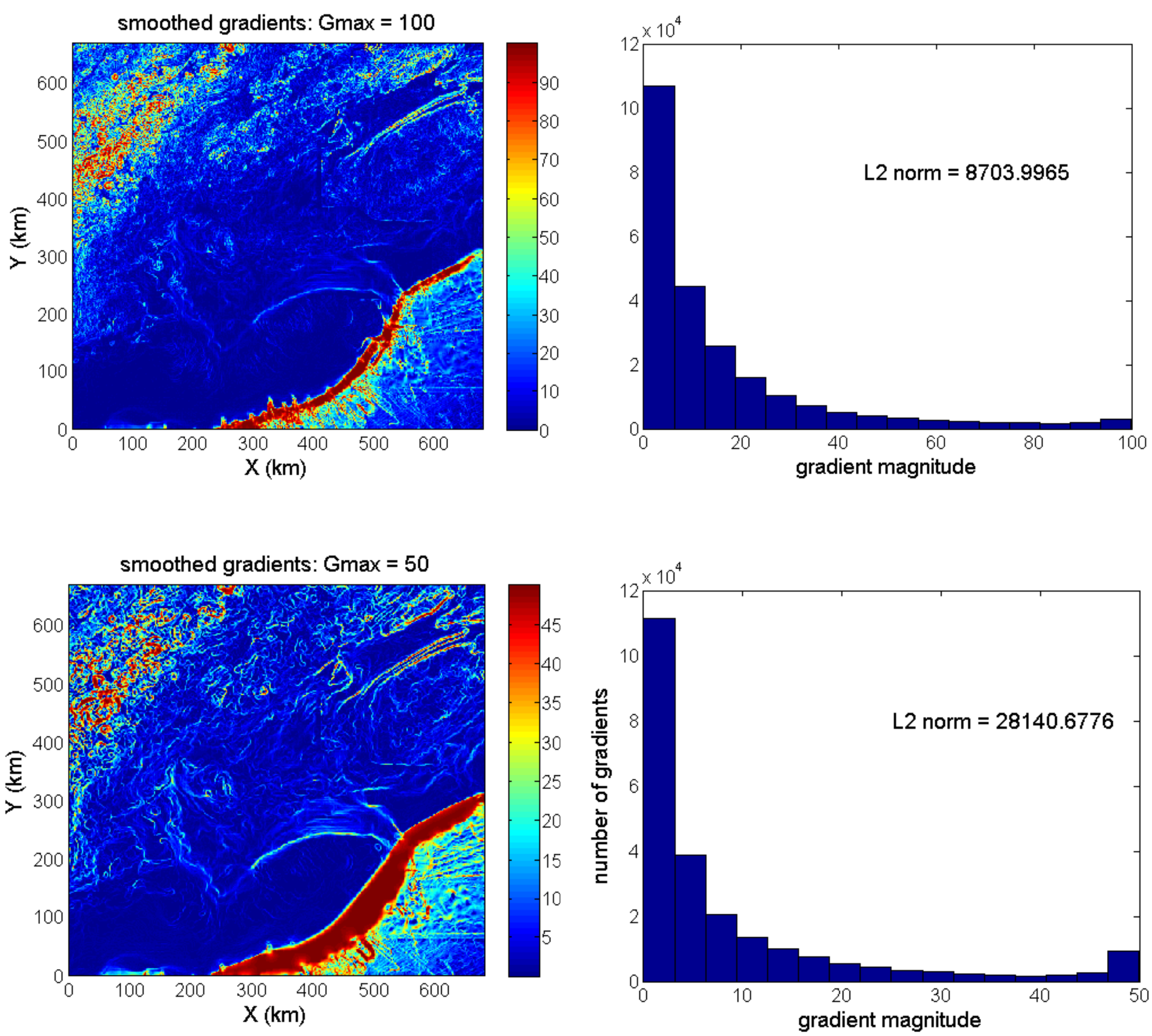

Figure 5. Targeted averaging of the Gulf of Maine - Massachusetts Bay region

In Figure 5, the plots on the left show the gradients after targeted averaging is applied. Note the change in the distribution of the gradients shown in the histograms to the right of each gradient plot. The distribution is still skewed but to a lesser extent than the original distribution. The L^2 norm's for the 100 and 50 gradient constraints were 8907 km and 28141 km, respectively.

Laplacian conditioning is an iterative scheme that works by first tagging the areas in violation of the gradient constraint, G_{max}. If $(\nabla D)_{i,j}^{\beta} > G_{max}$, then $\left(D_{i+1,j}, D_{i,j}, D_{i,j+1}, D_{i+1,j+1}\right)^{\alpha}$ are all considered to be in violation. A Poisson equation is then set up to condition the bathymetry,

$$\nabla^2 D_{i,j} = F(D). \quad \text{(eq. 10)}$$

The Laplacian of the bathymetry is evaluated as follows:

$$\nabla^2 D_{i,j} = \frac{1}{2\Delta x}\left(\left(\frac{dD}{dx}\right)_{i,j}^{\beta} - \left(\frac{dD}{dx}\right)_{i-1,j}^{\beta} + \left(\frac{dD}{dx}\right)_{i,j-1}^{\beta} - \left(\frac{dD}{dx}\right)_{i-1,j-1}^{\beta}\right) + \frac{1}{2\Delta y}\left(\left(\frac{dD}{dy}\right)_{i,j}^{\beta} - \left(\frac{dD}{dy}\right)_{i,j-1}^{\beta} + \left(\frac{dD}{dy}\right)_{i-1,j}^{\beta} - \left(\frac{dD}{dy}\right)_{i-1,j-1}^{\beta}\right). \quad \text{(eq. 11)}$$

The gradients in the Laplacian are discretized in a manner similar to the gradient discretization shown in equation 6**.** If the gradient at a point is above G_{max}, the partial derivatives at that point are scaled by a factor,

$$\frac{G_{max}}{\sqrt{\left(\frac{dD}{dx}\right)^2 + \left(\frac{dD}{dy}\right)^2}}. \quad \text{(eq. 12)}$$

These rescaled partial derivatives are then substituted back into eq. 11 to give the right hand side of eq. 10**.** The fact that some of the partial derivatives in the Laplacian for a single point will be scaled while others will not is what allows for the conditioning.

When solving eq. 10**,** the following conditions are also applied:

- $D_{i,j} = D_{i,j}$ is set for interior points that are not in violation, and,
- For the boundaries the no-normal condition below is applied:

$$\begin{gathered} D_{1,j} - D_{2,j} = 0\,; D_{nx-1,j} - D_{nx,j} = 0\,, j \in [2, ny-1] \\ D_{i,1} - D_{i,2} = 0\,; D_{i,ny-1} - D_{i,ny} = 0\,, i \in [2, nx-1] \\ D_{1,1} - D_{2,2} = 0 \\ D_{1,ny} - D_{2,ny-1} = 0 \\ D_{nx,1} - D_{nx-1,2} = 0 \\ D_{nx,ny} - D_{nx-1,ny-1} = 0 \end{gathered} \quad \text{(eq. 13)}$$

The bathymetry used here is the same one used for targeted averaging. The same two gradient constraints of 50 and 100 were again imposed for 100 iterations.

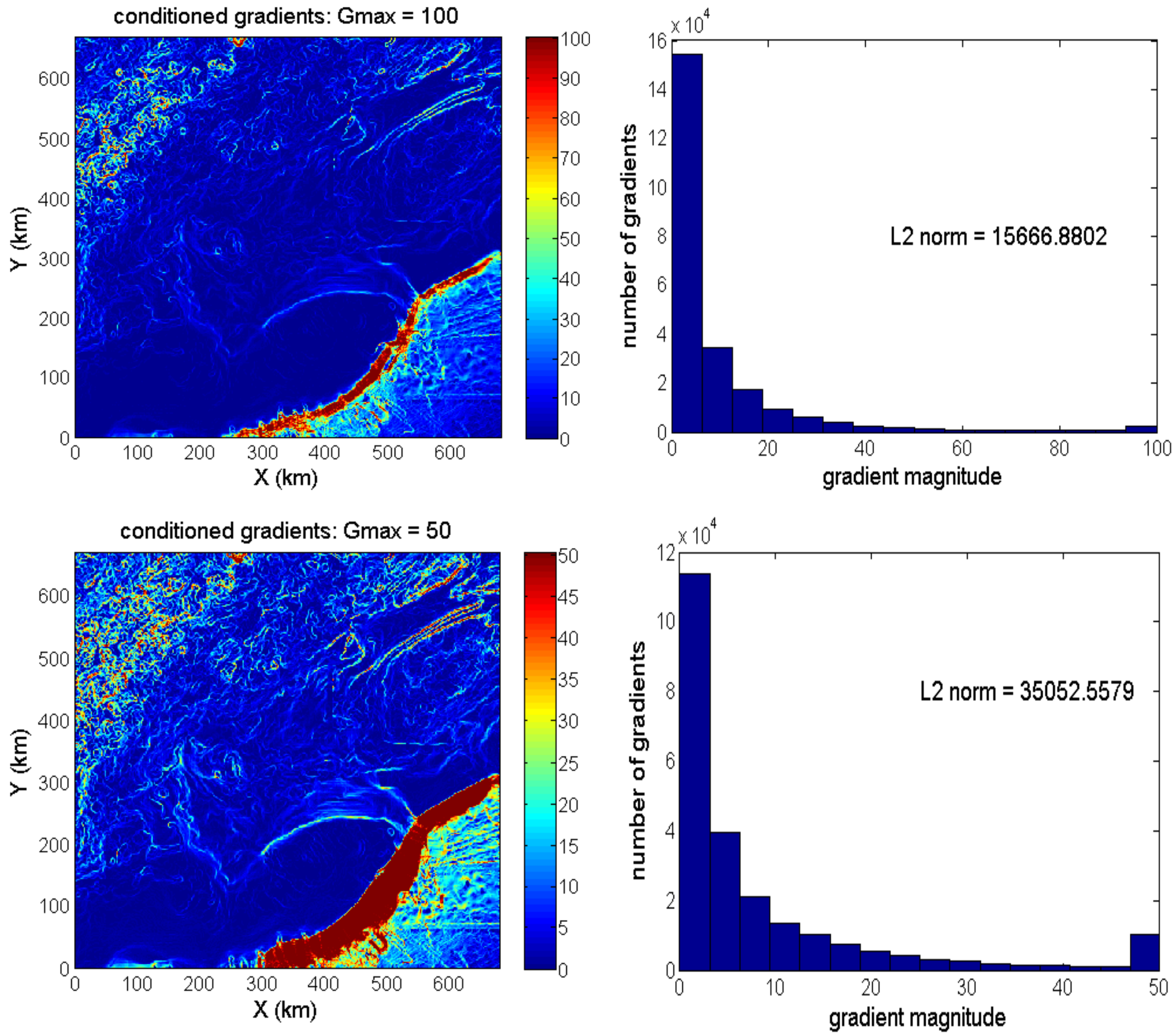

Figure 6. Laplacian Conditioning of the Gulf of Maine - Massachusetts Bay region
100 iterations, reduced gradient constraint of 100 for the top plot, 50 for the bottom

For both of these cases, the maximum gradient achieved after 100 iterations was slightly above the intended value. The maximum gradient for both cases was within .4% of the intended value, thus comparisons between these plots and the plots in Figure 5 still valuable. After 100 iterations, the maximum gradient for the top two plots was 100.03 with a L^2 norm of 15780km. For the bottom plots, the maximum gradient was 50.209 with a L^2 norm of 35365km. Comparing these two values with the targeted averaging values of 8907km and 28141km shows that targeted averaging has greater success in preserving the bathymetry. Examination of the histograms, however, reveals there is a noticeable distinction between targeted averaging and Laplacian conditioning. The Laplacian conditioning distributions maintain a higher peak at lower gradient values, thus showing a more skewed distribution. This is indicative of the fact that Laplacian conditioning avoids operating on nodes that are not in violation of the gradient constraint. Having a less skewed distribution for targeted averaging is expected since the underlying principle of targeted averaging is to bring values closer to a mean and thus closer to each other.

2.3 Reduced Gradient Adaptation

The reduced gradient was calculated as the magnitude of the gradient divided by the depth at each node,

$$\frac{\left\| \nabla D_{i,j}^{\beta} \right\|}{D_{i,j}^{\beta}}.$$

(eq. 14)

Since the gradients were calculated on the β grid, the depth at each node was taken as the average of the four nodes in the α grid surrounding it (see Fig. 3 for staggered grid definitions),

$$D_{i,j}^{\beta} = \frac{1}{4}(D_{i,j}^{\alpha} + D_{i+1,j}^{\alpha} + D_{i,j+1}^{\alpha} + D_{i+1,j+1}^{\alpha}).$$

(eq. 15)

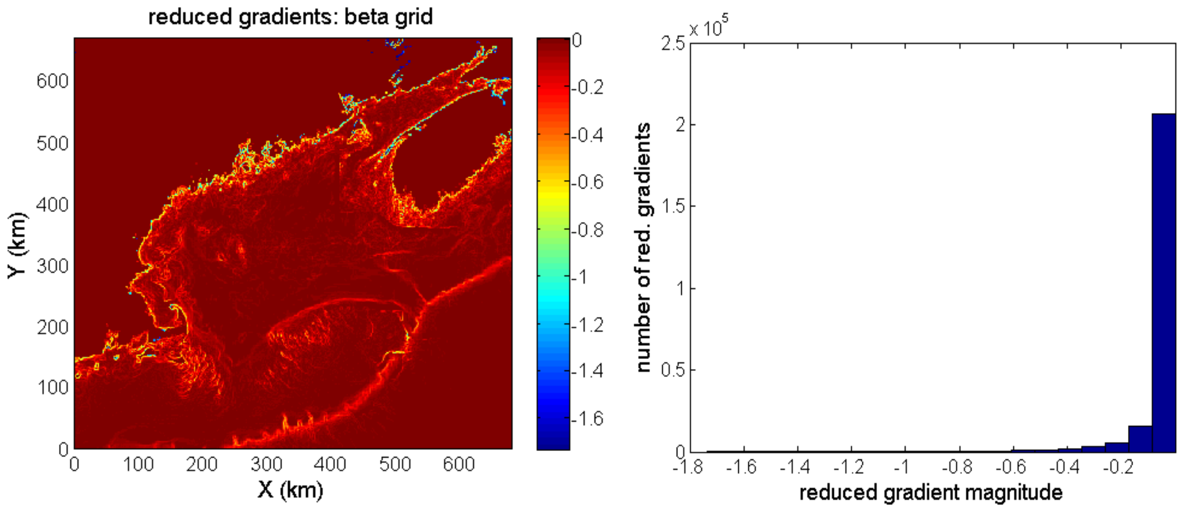

Figure 7. Reduced Gradients of the Gulf of Maine- Massachusetts Bay region

The plots in Figure 7 show the reduced gradients for the Gulf of Maine- Massachusetts Bay region that was used in section 2.3. Note that because the reduced gradient calculation involves dividing by a depth, the bathymetry was clipped at -5km to avoid dividing by unnecessarily large numbers (small magnitude). A linear mapping of reduced gradients to element sizes would again result in a mainly uniform mesh. The higher the reduced gradient, the lower the element size and vice versa. Depending on the minimum element size set by the user, the traces of low reduced gradients (small element sizes); may either not be present in the final mesh or cause large element size gradients leading to low quality meshes. The goal of smoothing/conditioning the bathymetry would then be to reduce the skewness of the reduced gradient distribution.

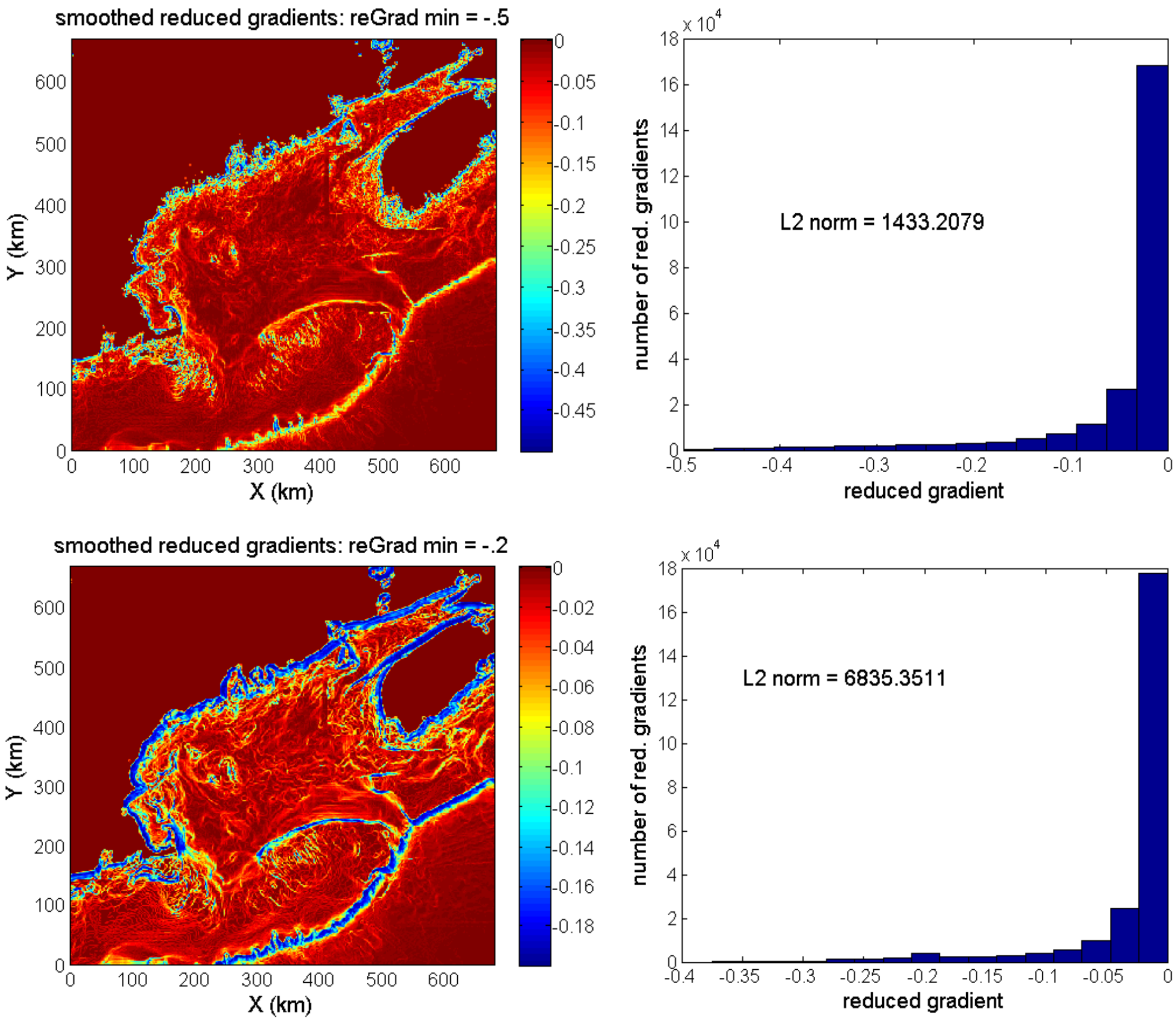

Figure 8. Targeted averaging in the Gulf of Maine – Massachusetts Bay Region based on reduced gradients. The minimum reduced gradient of the original clipped bathymetry was -1.7343.

In Figure 8, the first row of plots shows the results of targeted averaging with a minimum reduced gradient constraint of -.5. The second row shows the results for a reduced gradient constraint of -.2. The L^2 norms for targeted averaging were 1434km for -.5 and 6836km for -.2. As the constraint is increased, the skewness of the distribution decreases, as expected. From the reduced gradient plots, the two areas necessitating a high number of elements are the coastline and the shelf break.

For Laplacian conditioning, the only thing that differs from conditioning on absolute gradients is the scaling factor. In reduced gradient conditioning, each node in violation has its corresponding partial derivatives scaled by the following factor,

$$\frac{Re_{max}\left|D^{\beta}_{i,j}\right|}{\sqrt{\left(\frac{dD}{dx}\right)^2+\left(\frac{dD}{dy}\right)^2}},$$

(eq. 16)

where Re_{max} is the minimum reduced gradient allowed.

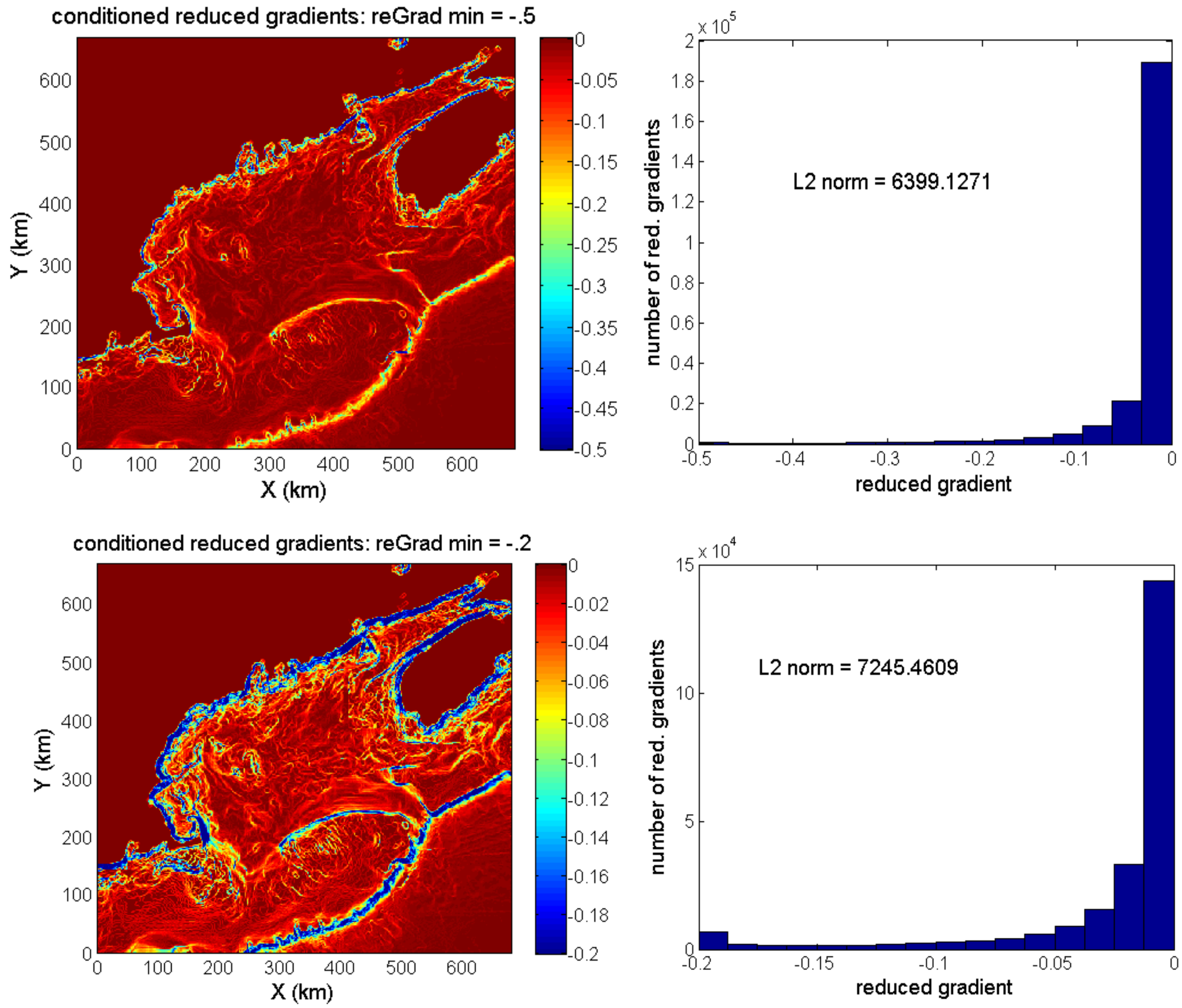

Figure 9. Reduced gradient based Laplacian conditioning on the Gulf of Maine-Massachusetts Bay region (100 iterations).

Figure 9 shows the results of Laplacian conditioning on the Gulf of Maine region. The minimum reduced gradient for both runs was within .2% of the intended goal. The L^2 norms for reduced gradient constraints of -.5 and -.2 (recall that the minimum reduced gradient was originally -2.1961) were 6400km and 7246km respectively. For both reduced gradient constraints, targeted averaging again preserves the bathymetry better than Laplacian conditioning.

2.4 Finalizing $h(x)$

The final $h(x)$ must be able to incorporate any combination of the factors discussed in sections 2.1 through 2.3. To combine the element sizes set by the absolute and reduced gradients, a minimum condition is applied;

$$h_{grad}(x) = \min\{(h_{absolute\ gradient}(x), h_{reduced\ gradient}(x)\}$$

(eq. 17)

Given the maximum and minimum element sizes due to absolute and reduced gradients, $elem_{grad}^{upper}$, $elem_{grad}^{lower}$, $h_{absolute\ gradient}(x)$ and $h_{reduced\ gradient}(x)$ are calculated as,

$$h_{absolute\ gradient}(x) = \ elem_{grad}^{upper} - \left(\frac{\|\nabla D(x)\|}{\max(\|\nabla D(x)\|)}\left(elem_{grad}^{upper} - elem_{grad}^{lower}\right)\right),$$

(eq. 18a)

$$h_{reduced\ gradient}(x) = \ elem_{grad}^{upper} - \left(\frac{\frac{\|\nabla D(x)\|}{D(x)}}{\min\left(\frac{\|\nabla D(x)\|}{D(x)}\right)}\left(elem_{grad}^{upper} - elem_{grad}^{lower}\right)\right).$$

(eq. 18b)

A minimum condition is appropriate because it preserves the areas where assigned resolution is high. The goal is to capture the effects of both gradients and reduced gradients, thus an average of the two, even if weighted, runs the risk of masking areas of high resolution.

To incorporate the element sizes due to curvature, a sizing matrix, $h_{curvature}^{temp}\{i\}$, is created for each point along the coastline via the following steps. First, the distance matrix P is defined as,

$$P = \left(X - x(i)\right)^2 + \left(Y - y(i)\right)^2,$$

(eq. 19a)

where X and Y are the matrices containing the x and y coordinates respectively, of each bathymetric node. The expression $h_{curvature}\{i\}$ is then defined as,

$$h_{curvature}^{temp}\{i\} = esc(i) + k_e P,$$

(eq. 19b)

where k_e serves as a scaling factor to achieve a desirable distribution of $esc(i)$ across the domain. The sizing matrix due to curvature, $h_{curvature}(x)$, is set equal to the node-wise minimum of every $h_{curvature}^{temp}\{i\}$.

The final sizing matrix is defined using another minimum condition,

$$h(x) = \min\{h_{curvature}(x), h_{grad}(x)\}.$$

(eq. 20)

With a boundary defined and $h(x)$ calculated, Distmesh and GMSH have the necessary components needed to generate a mesh.

3 Meshing Results

In this section, we present the results of our meshing procedure for a range of meshes across different areas of the Gulf of Maine. The purpose of the meshes in this section is two-fold: the first is to compare the quality of meshes generated by GMSH and Distmesh. The second is to observe the effect of certain parameters used in creating $h(x)$, on the mesh itself. To quantify the quality of a mesh, a commonly used metric evaluated on finite elements is:

$$q = \frac{2R_{in}}{R_{out}} \quad \text{(eq. 21)}$$

where R_{in} represents the inscribed radius while R_{out} represent the circumscribed radius. The closer a triangle is to being equilateral, the close q will be to 1.

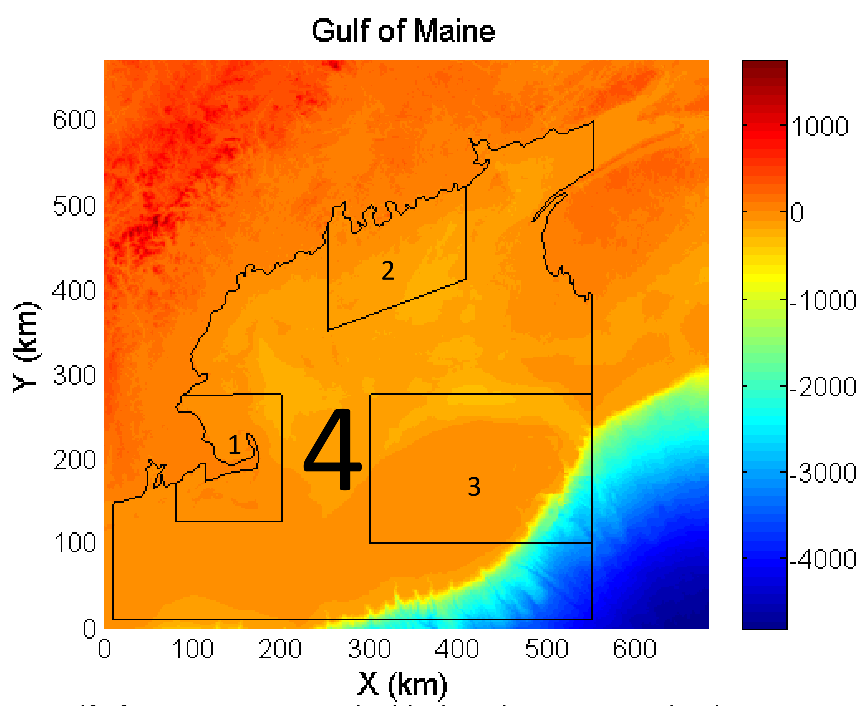

Figure 10. Gulf of Maine Domains. The black outlines contain the domains meshed. Domains 1 and 2 were chosen to examine the meshes near the coastline. Domain 3 was chosen to capture the shelf while domain 4 (the outer polygon) was chosen to show $h(x)$.

In the next four sections, a series of meshes will be presented along with a histogram showing the distribution of q. Each section tries to isolate the effect of a few parameters on the overall mesh. GMSH and Distmesh outputs are compared. The following color convention will be followed: Red meshes were generated using Distmesh. Blue meshes were generated using GMSH.

3.1 Distance Based Meshing

To observe the influence of the slope factor in diffusing the element sizes due to curvature, the following meshes were created. Recall from section 2, equation 19b, the slope factor, k_e essentially determines the rate of diffusion of the element sizes assigned to each point along the coastline. The meshes bellow disregard any contribution except distance in deciding the target element size.

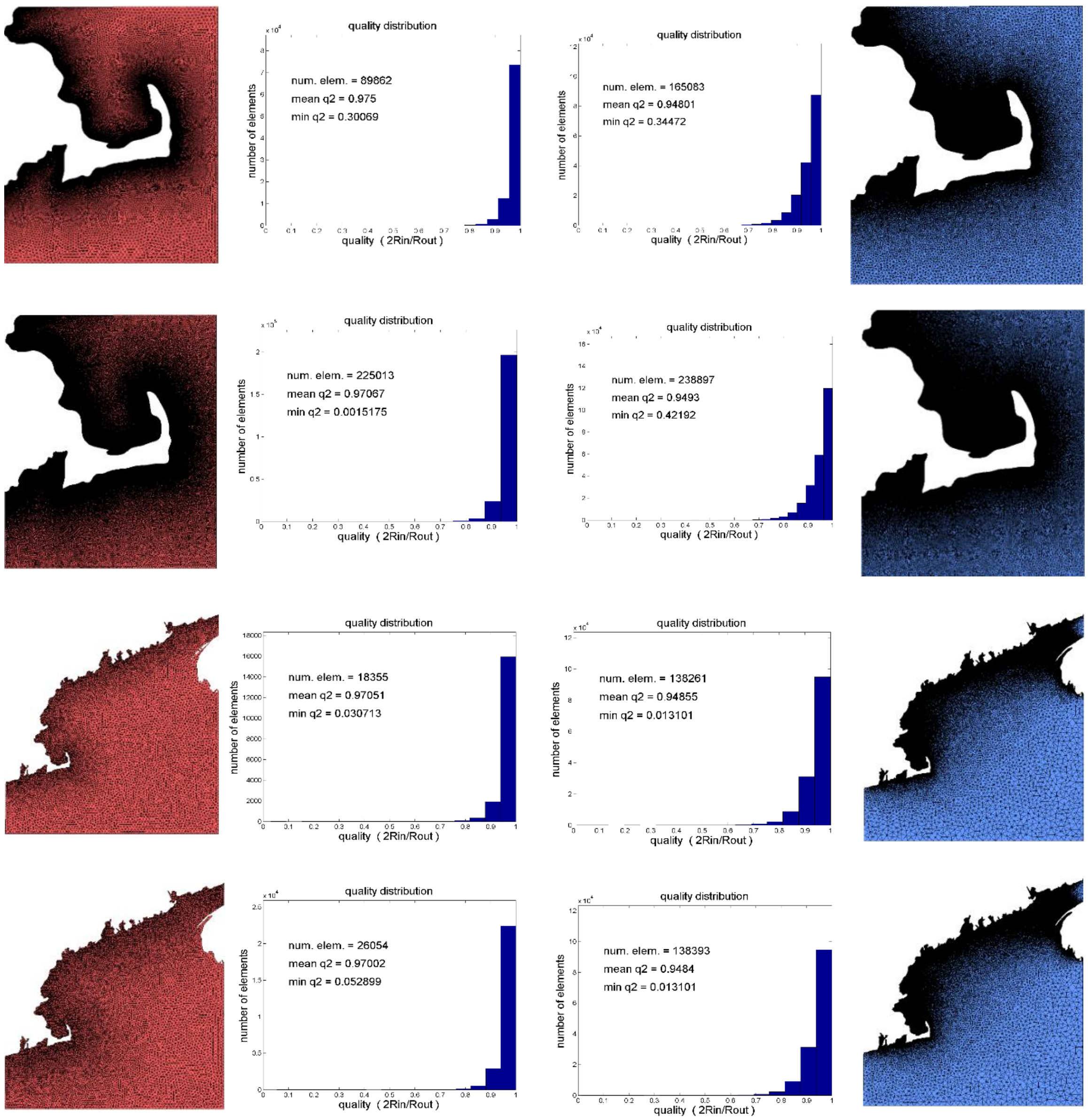

Figure 11. Distance Based Meshes. These meshes show the radial diffusion of element sizes on domains 1 and 4. The meshes above were made by assigning a constant value along each point along the coast. The values used were .2 for the top 4 plots and 5 for the bottom 4 plots. The radial fall off was set to .02 to .04 for the first 4 meshes and .045 to .09 for the bottom 4 meshes.

Since the distribution of element sizes is fairly smooth for the 4 cases above, the quality of the mesh is expected to be high. In each case, the mean quality is high for both Distmesh (.97) and GMSH (.94). GMSH has slightly higher minimum q values, in particular row 2 where Distmesh has a min q value of .0015 while GMSH has a min q value of .42. This is probably explained by GMSH's ability to increase the target element size near the coast, the area where Distmesh has the most trouble. By replacing the target element sizes near the boundary by significantly smaller values, the issues of small scale features and curvature generally disappear.

3.2 Coastline Curvature Meshing

The following meshes show the impact of expanding the element size via equation 4, followed by diffusing them via equation 19.b. Recall that equation 4 expanded the number of low element size values assigned to each point along the coastline. (eq. 4 reassigned a new $esc(i)$ to every node equal to the minimum of its N neighbors and itself.

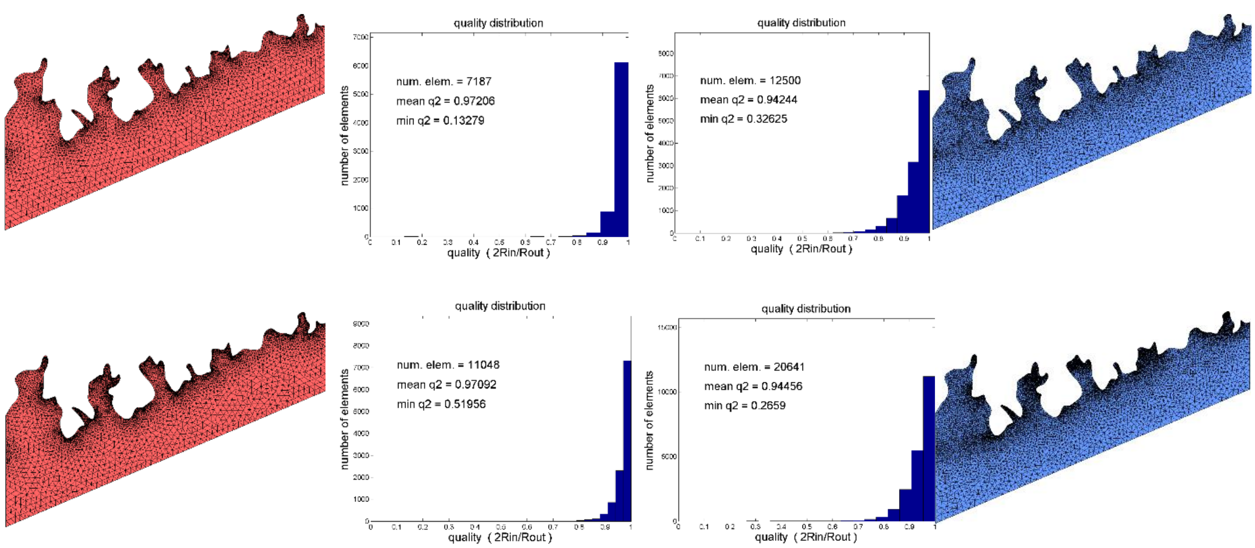

Figure 12. Coastline Curvature Meshes. The meshes above were created using only the element sizes set by curvature, esc, and diffusing them throughout the domain in a radial manner (see eq. 19b) Then esc_{expand} was used to test the effectiveness of spreading out minimum values, (see eq. 4). The radial scaling factor (k in eq. 18b) in this case was .18.

In both cases, using esc_{expand} increases the number of elements as expected. The mean quality for Distmesh and GMSH is above .94, while the minimum quality is much lower for Distmesh than GMSH. Note that in this case, the element sizes assigned to the coastline were smaller or at least comparable in magnitude to the spacing's of the coastline. If not, there would have been little difference if at all in diffusing esc or esc_{expand}. If the radial scaling factor was decreased to .1 for example, the effect of esc_{expand} would be less pronounced. The way equation 19b works makes it such that a smaller scaling factor increases the reach of a single small-sized element, thus decreasing the impact of esc_{expand}.

3.3 Reduced and Absolute Gradient Based Meshing

This section meshes every domain in Figure 10. The $h(x)$ for this set of meshes is shown on Figure 13.

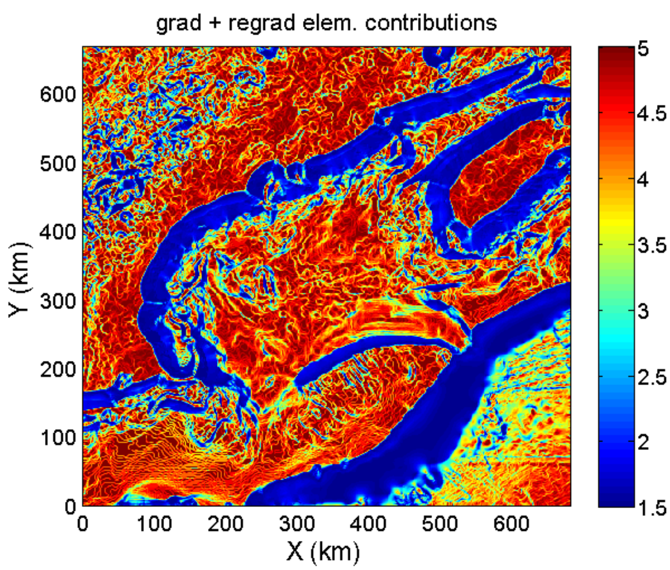

Figure 13. Gradient + Reduced Gradient Contributions. The above $h(x)$ combines reduced gradient and gradient contributions. The absolute gradient constraint was 50. The reduced gradient constraint was .1. The bathymetry was smoothed via targeted averaging.

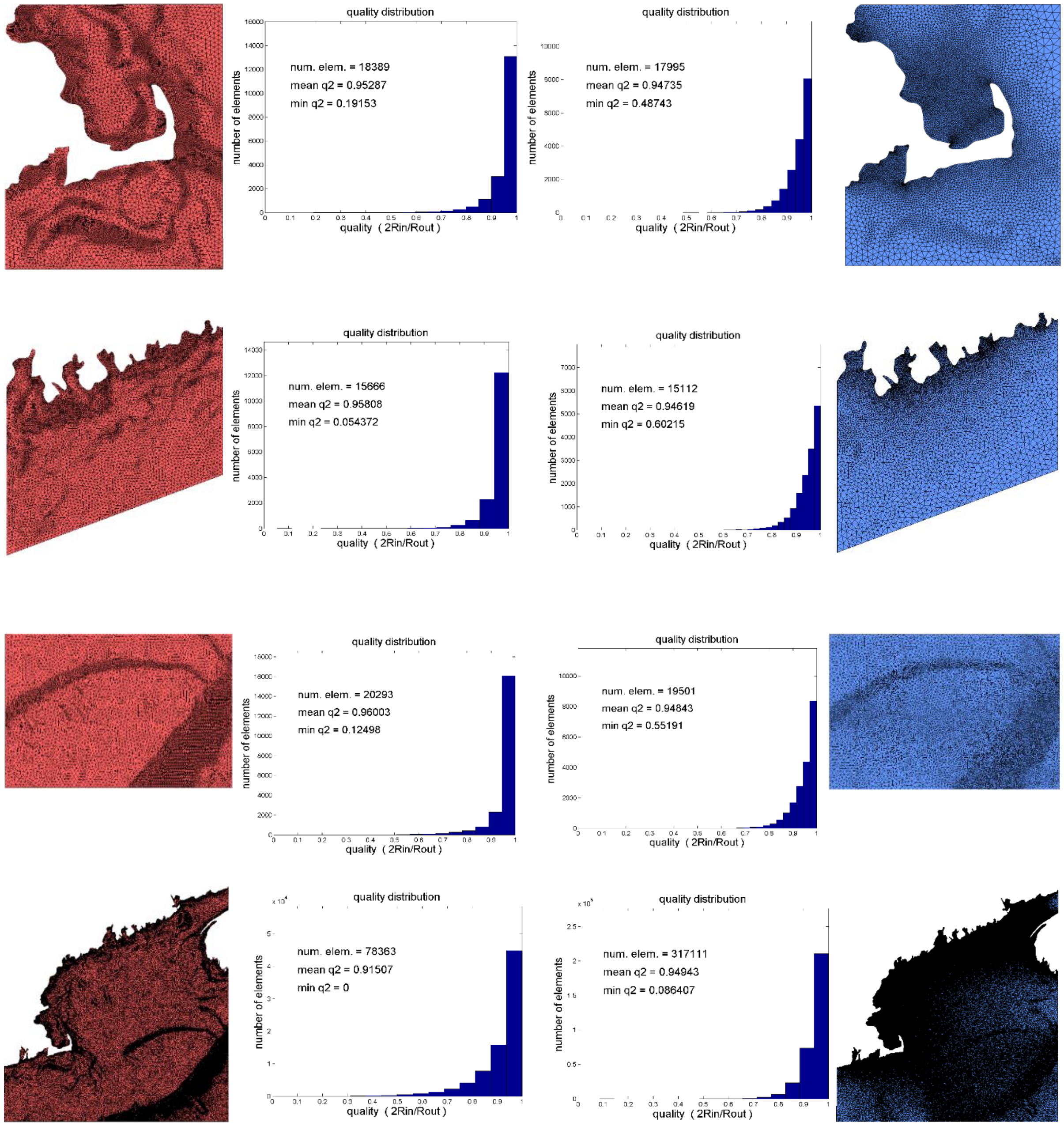

Figure 14. Reduced + Absolute Gradient Meshing (via targeted averaging). The sizing matrix used for the eight meshes above only incorporates reduced and absolute gradients. The absolute and reduced gradients were capped via targeted averaging to 50 and .1 respectively. Element size limits based on gradients were 5 and 1.5.

From the meshes in Figure 14, the minimum quality for GMSH meshes is higher in every case. Studying the red meshes, it would seem that Distmesh creates meshes that represent the underlying sizing matrix with greater fidelity. GMSH trades this fidelity for higher resolution within a large band around the coast. GMSH may be taking the spacing's at the boundary and overriding $h(x)$ in those areas. This explains why the first two rows of meshes, concentrated near the coast, show GMSH meshes that have higher resolutions near the coast. The third row of meshes further supports this by showing two meshes far away from the coast that look fairly similar.

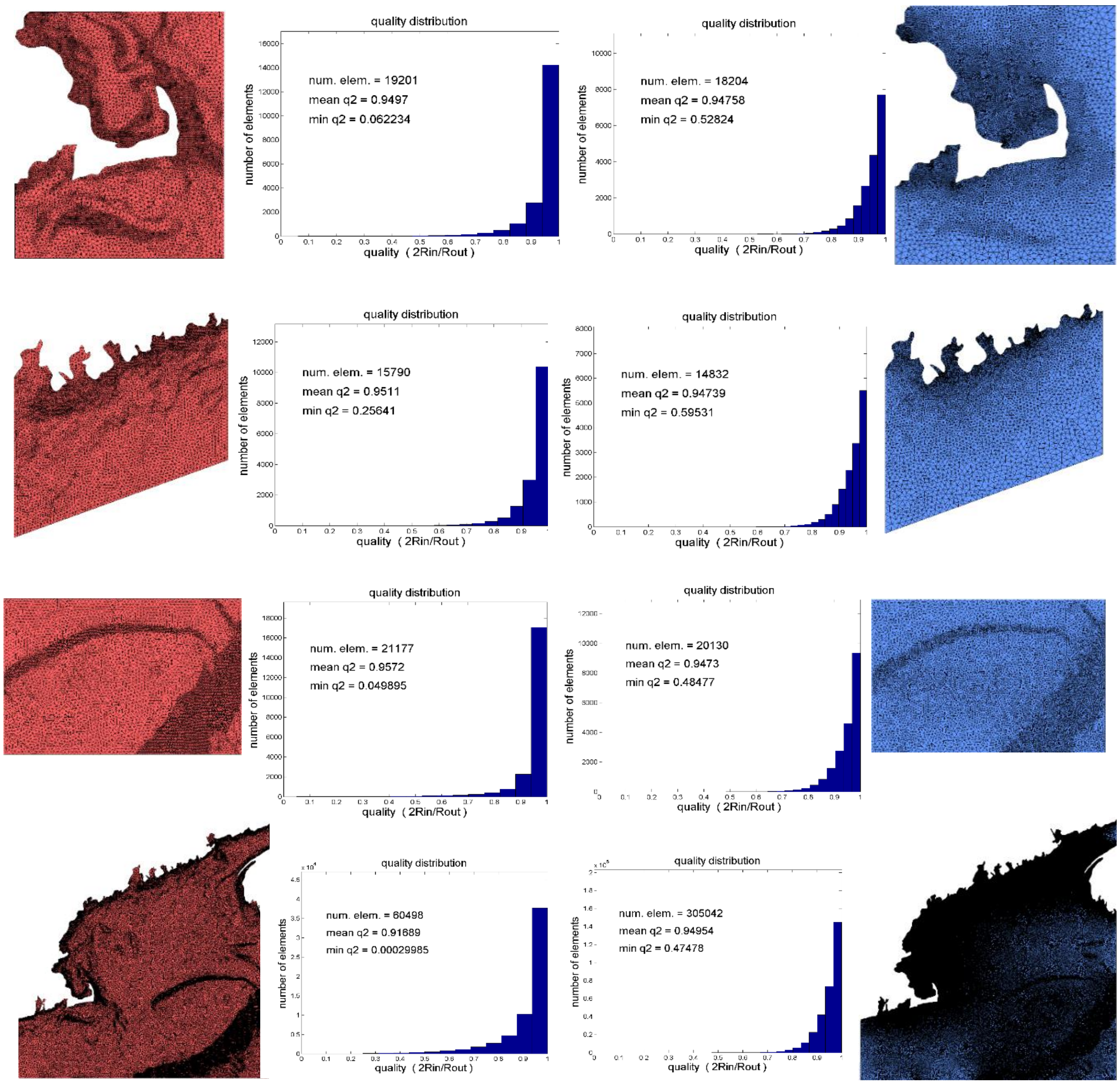

Figure 15. Reduced + Absolute Gradient Meshes (via Laplacian conditioning). Same plots as in Figure 14, except this time, the gradient and reduced gradients constraints are applied via Laplacian conditioning. The same limits of 1.5 to 5 on element sizes were placed.

Now, looking at Figure 15, where the reduced gradient and absolute gradient constraints were enforced through Laplacian conditioning, slight changes can be seen in the red meshes. In terms of quality, the values stay about the same for both GMSH and Distmesh, with the one exception being the bottom blue mesh. The minimum quality jumped from .0864 to .47 even though the mean and the shape of the distribution stayed roughly the same. Examining the number of elements in the GMSH for domain 4 and assuming equilateral triangles, the average length of a triangle would be about 1.8km, much lower than the average value of $h(x)$, 3.54km. This suggests that, under the conditions GMSH ran during the making of these plots, the boundary node spacing took precedence over $h(x)$ near the boundary. Further tests and familiarization with the workings of GMSH are needed to analyze these results in further depth.

3.4 Smoothing $h(x)$

Figure 16 displays a sizing function, $h(x)$, with and without smoothing. The $h(x)$ shown takes into account the gradients and reduced gradients of the bathymetry as well as the curvature, and the distance to the coastline. The gradient based element size limits imposed were 15 and 4. The curvature based element size limits imposed were 2.5 and 1.25. Targeted averaging with a gradient constraint of .5 was applied to $h(x)$ to obtain the plot on the left. (The original $h(x)$ had a maximum gradient of 8.08)

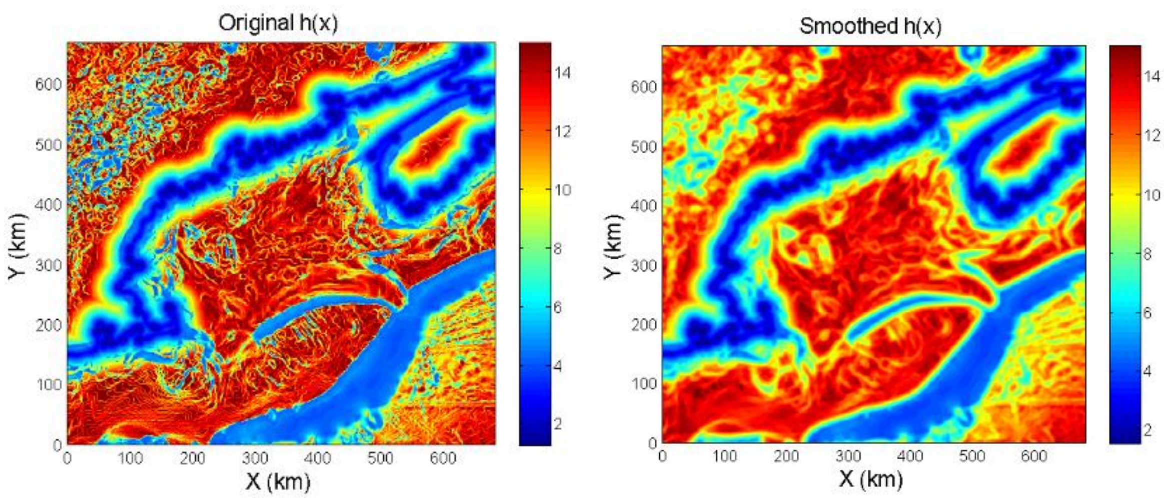

Figure 16. Sizing function, $h(x)$, for the Gulf of Maine. This $h(x)$ takes into account the gradients, reduced gradients, curvature and the distance to the coastline. The absolute gradient was constrained to 40. The reduced gradient was constrained to .08. The radial scaling factor was .25. For the smoothing, targeted averaging was applied.

Figure 17 shows an example of the effectiveness of smoothing $h(x)$. The goal of smoothing $h(x)$ is to ensure that the maximum gradient of $h(x)$ is kept to a value such that the highest ratio of neighbor element edge lengths is roughly 3 or less throughout the entire field (Conroy 2010). The mean quality jumps from .91 to .96 after smoothing$h(x)$. Note that this particular run was stopped as soon as the minimum quality passed .1.

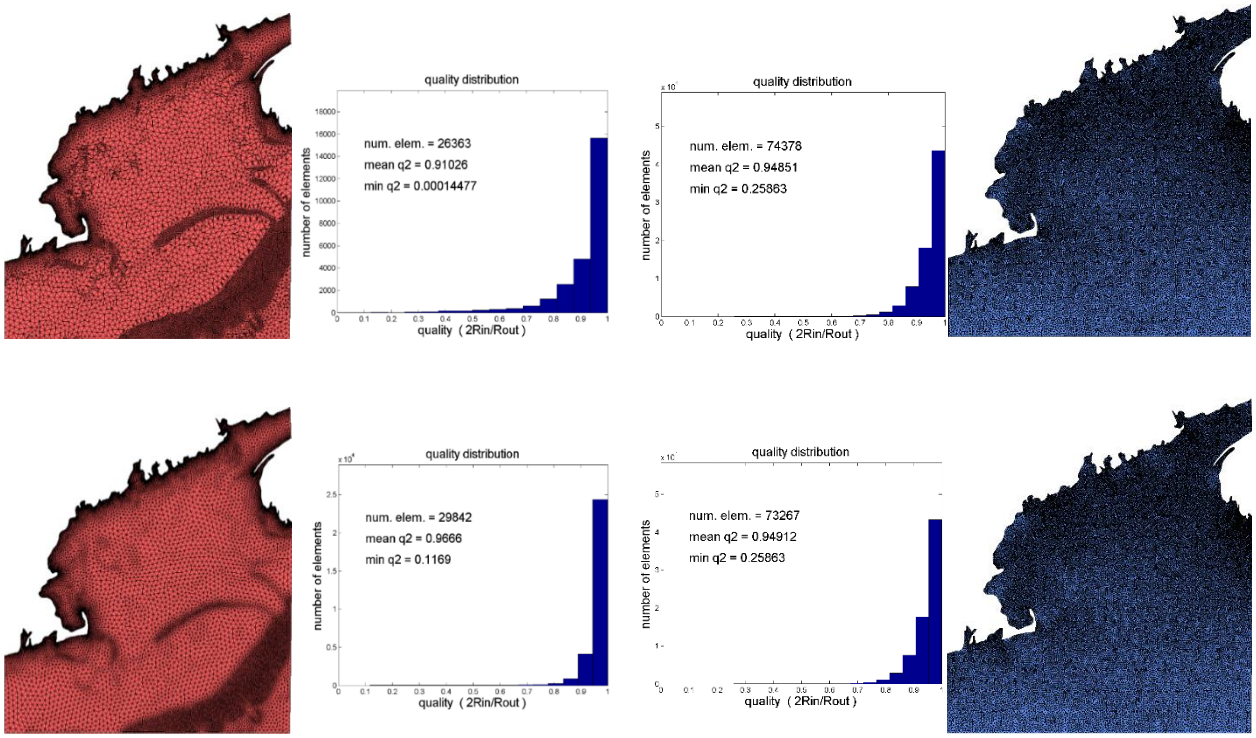

Figure 17. Effect of Smoothing vs. Not Smoothing $h(x)$. These domain 4 meshes (see Figure 10) take into account curvature, gradients, reduced gradients, and the distance to the coast. The top row of meshes show the result of a $h(x)$ without smoothing. The bottom row of meshes show the result of a smooth $h(x)$.

Figure 17 shows an example of the effectiveness of smoothing $h(x)$. The goal of smoothing $h(x)$ is to ensure that the gradient of $h(x)$ is kept to a value such that the highest ratio of neighbor element edge lengths is roughly equal to 3 or less throughout the entire field (Conroy 2010). The effect of smoothing on the Distmesh generated mesh was a .05 increase in average quality and about a .1 increase in the minimum quality. The effects of smoothing on the GMSH generated mesh are less pronounced, if there at all, for this particular example. The average quality increased by about .0005 and the minimum quality actually stayed the same. Further research and testing on both of these mesh generators is required we are able to critically evaluate these results.

4. Future Work

4.1 Small Scale Feature Detection

A major issue that came up with the coastline was the inability to detect small scale features that require a high resolution discretization. The only solution the current mesh generation has is to either manually assign greater resolution near these small scale feature or increase the resolution near the entire boundary. Current methods involve finding the medial axis and setting the element size as a function of the distance to the medial axis.

4.2 Smoothing

While the two smoothing methods where able to limit the gradients/reduced gradients, there are probably other smoothers/ conditioning techniques that do a better job at preserving the original field. For example, a simple improvement to targeted averaging would be to weight the nodes about to be averaged, with a weight that varies according to the normalized distance to the violating node. An improvement to Laplacian conditioning, at least for the reduced gradient case, is to implement the no normal condition implemented at the boundaries, at the height clipped. This would ensure that the bathymetry “floats” instead of having deal to with a static boundary.

4.3 Adaptive meshing/ Error Analysis

The methods presented so far try to accurately display the magnitude of gradients, reduced gradients, distance, and curvature in mesh format. Whether the accuracy of these representations matter would be the next step in this research. The sizing matrix of the meshing algorithm should also utilize the dynamical scales of the ocean fields to be modeled (i.e. the smallest expected and relevant dynamical scales at each location in the domain).

5. Conclusion

Each step of the mesh generation process was discussed. The curvature of the coast was approximated from a parameterization of the original coastline and was then mapped to element sizes. To reduce the skewness of gradient magnitudes, some form of smoothing was necessary. The two smoothing methods tested were targeted averaging and Laplacian conditioning with a no normal boundary condition. Comparing the L^2 norms of both methods, targeted averaging did a better job of preserving the bathymetry. Element sizes where then linearly mapped to the clipped gradients and reduced gradients. The sizing matrix, $h(x)$, was then calculated though a series of minimum conditions. Two different mesh generators were tested, GMSH and Distmesh. GMSH is quick and consistent while Distmesh produces slightly higher quality elements on average. However, further detailed studies of each mesh generator, its properties and its options should be completed. This would allow us to fully explore their respective capabilities and provide a more detailed evaluation.

Bibliography

Bilgili, A., Smith, K.W., Lynch, D.R. (2006) A two-dimensions bathymetry-based unstructured triangular grid generator for finite element circulation modeling*, Computers and Geosciences*, 32, pp. 632-642

Conroy, C.J. (2010). ADMESH: An Advanced Mesh Generator for Hydrodynamic Models. Masters Thesis, The Ohio State University, Department of Civil Engineering

Geuzaine, C., Remacle, J.F. (2009) Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering* 79(11), pp. 1309-1331

Hagen, S.C., Hortsmann, O., Bennett, R.J. (2002). An Unstructured Mesh Generation Algorithm for Shallow Water Modeling, *International Journal of Computational Fluid Dynamics*, 16:2, 83-91

Legrand, S., Deleersnijder, E., Hanert, E., Legat, V., Wolanski, E. (2006) High-resolution, unstructured meshes for hydrodynamic models of the Great Barrier Reef, Australia. *Estuarine, Coastal and Shelf Science* 68:36-46

Pain, C.C., Piggott, M.D., Goddard, A.J.H., Fang, F., Gorman, G.J., Marshall, D.P., Eaton, M.P., Power, P.W., Oliveira, C.R.E. de (2005). Three-dimensional unstructured mesh ocean modeling. *The Second International Workshop on Unstructured Mesh Numerical Modelling of Coastal, Shelf and Ocean Flows* 10:5-33

Persson, P.O. (2005) *Mesh generation for implicit geometries,* PhD Thesis, Massachusetts Institute of Technology

Terwisscha, A.D. van Scheltinga, Myers, P.G., Pietrzak, J.D. (2012) Mesh generation in archipelagos. *Ocean Dynamics* , 62:1217:1228

Ueckermann, M.P., and Lermusiaux, P.F.J. High order schemes for 2D unsteady biogeochemical ocean models. Ocean Dynamics 60, 6 (2010), 1415–1445.

Ueckermann, M.P. High Order Hybrid Discontinuous Galerkin Regional Ocean Modeling. PhD thesis, MIT, Cambridge, MA, September 2013.

Westerink, J.J., Leuttich, R.A. JR. (1995). Continental shelf scale convergence studies with a barotropic tidal model, *Quantitative Skill Assessment for Coastal Ocean Models Coastal and Estuarine Studies*, 47, pp. 349-371, 1995.

Westerink, J.J.,Leuttich, R.A., Hagen, S.C. (1994). Meshing requirements for large scale ocean tidal models, *Computational Methods in Water Resources*, 10, pp. 1323-1330