

Development of a Finite Element Mesh for the Chesapeake and Delaware Bays

Todd Jeffrey Wood

A selected project submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of
Master of Science

Dr. E. James Nelson, Chair
Dr. A. Woodruff Miller
Dr. Alan K. Zundel

Department of Civil and Environmental Engineering
Brigham Young University

April 2012

Copyright © 2012 Todd Jeffrey Wood

All Rights Reserved

ABSTRACT

Development of a Finite Element Mesh for the Chesapeake and Delaware Bays

Todd Jeffrey Wood
Department of Civil and Environmental Engineering, BYU
Master of Science

This project involved the creation of a finite element mesh for the Chesapeake and Delaware Bays using the WMS and SMS software. The Engineering and Research Development Center (ERDC) of the United States Army Corps of Engineers sponsored the project, and Aquaveo, LLC in Provo, Utah provided assistance.

I first defined and created the boundaries of the model domain with feature arcs using elevation data provided by ERDC that contained approximately two billion elevation data points. Using these feature arcs, I defined areas where mesh elements were to be created, taking into consideration the low points in the water channels. With the help of Aquaveo, I created a size function to dictate mesh element sizes based on surrounding elevation points. Finally, I converted the map data to mesh elements and assigned elevations and materials to the mesh.

During the course of the project, I identified several deficiencies in the software, as well as inefficiencies in the procedure I followed. The software deficiencies included bugs, lacking features, and limitations. Finally, I identified several features and enhancements to the Aquaveo software that would benefit users undertaking similar projects in the future.

Keywords: Todd Jeffrey Wood, finite element mesh, Chesapeake, Delaware, AdH, Aquaveo, SMS

ACKNOWLEDGMENTS

I'd like to give a special thanks to Dr. Alan K. Zundel for his patience in helping me through this project and his willingness to provide valuable insight. I'd also like to acknowledge and thank Lourdes Manley who helped with significant portions of the tedious manual editing process. Thanks also to Royd Nelson, Rusty Jones, Tom Moreland, and Clark Barlow at Aquaveo for sharing their expertise. Last but certainly not least, I want to thank my wonderful wife Alaina for all her support along the way.

TABLE OF CONTENTS

LIST OF TABLES	vii
LIST OF FIGURES	ix
1 Introduction.....	1
1.1 Background.....	2
2 Selecting and Creating the Mesh Boundaries with Feature Arcs.....	5
2.1 Generating Contour Arcs from Elevation Datasets	5
2.1.1 Displaying and Converting Contours to Feature Arcs	7
2.1.2 Cleaning and Editing the Contour Feature Arcs	10
2.1.3 Creating the Composite Shoreline	15
2.2 Defining the Model Domain Boundaries.....	17
2.2.1 Selecting the Ocean Boundary	17
2.2.2 Building Polygons.....	20
2.2.3 Assigning Properties to the Map Polygons	22
2.3 Creating Thalweg Arcs in WMS Using TOPAZ	23
2.3.1 Loading the Elevation Data Files into WMS	24
2.3.2 Running TOPAZ in WMS	24
3 Generating the Mesh.....	27
3.1 Creating the Size Function Scatter Sets	27
3.2 Converting the Map Data to a Finite Element Mesh	33
3.3 Assigning Elevations to the Mesh Nodes	35
3.4 Extending the Mesh to the One Meter Contour	38
3.5 Assigning Material Types to the Mesh Elements	43
4 Software Deficiencies and Suggestions for Future Improvement	49

4.1	Bugs in the Software.....	49
4.2	Software Limitations.....	52
4.3	Suggestions for Future Improvement and Software Development.....	54
4.3.1	Process Inefficiencies.....	54
4.3.2	Future Software Development	54
REFERENCES.....		57
Appendix A. Additional Images of the Final Mesh.....		59

LIST OF TABLES

Table 2-1: Approximate Areas, Depths, and Volumes of Chesapeake and Delaware Bays Used to Determine the Ocean Boundary	18
Table 2-2: Volumes of Bays/Tributaries and Ocean in the Final Mesh	20
Table 3-1: Target Values for Selected Depths for the Two Size Functions	29
Table 3-2: Elements and Nodes in and Elevation Range of the Final Mesh	42
Table 3-3: Elevation Ranges and Number of Elements of the Different Materials in the Final Mesh	45

LIST OF FIGURES

Figure 1-1: DEM Representing the Elevations of Tile Forty-One Loaded as a Raster in SMS 11.0 Beta.....	3
Figure 2-1: Location of Each of the DEM Files Provided by ERDC with the Elevation Data.....	6
Figure 2-2: Tile Forty-One in WMS 8.3 with the Default Linear Contours.....	7
Figure 2-3: Zero Meter Contour for Tile Forty-One in WMS 8.3	8
Figure 2-4: Zero Meter Contour of Tile Forty-One Before Smoothing in WMS 8.3.....	9
Figure 2-5: Zero Meter Contour of Tile Forty-One After Smoothing in WMS 8.3	9
Figure 2-6: Dangling Arcs Created when there is a Series of Neighboring Points with the Same Value as the Contour.....	11
Figure 2-7: Illustration of the (a) Snap Nodes and Vertices and (b) Remove Dangling Arcs Cleaning Tools in SMS 10.1	12
Figure 2-8: Pits Shown on the Land of a Portion of Tile Forty-One	13
Figure 2-9: Feature Arcs Representing a River Before (a) and After (b) Being Truncated	14
Figure 2-10: Illustration of Locations of Hard Points.....	15
Figure 2-11: Zero Meter (Black) and One Meter (Blue) Contour Feature Arcs.....	16
Figure 2-12: Close Up View of the One and Zero Meter Contour Arcs.....	17
Figure 2-13: First Candidate Ocean Boundary Arc that Met the Water Storage Requirements	19
Figure 2-14: Ocean Boundary Feature Arc Used to Generate the Final Mesh Georeferenced Above an Aerial Photo of the Site from Google Earth.....	20
Figure 2-15: Polygon Properties Window in SMS	22
Figure 2-16: A Portion of Tile Forty-One with Thalweg Cells (blue) and the Zero Contour (Black)	25
Figure 2-17: Feature Arcs Representing the Zero Meter Contour and the Thalwegs.....	26

Figure 3-1: Plot of the Fifty Meter and One-Hundred Meter Minimum Size Functions.....	30
Figure 3-2: Smooth Data Set Tool in the Dataset Toolbox in SMS 10.1	31
Figure 3-3: Final Size Function for the One-Hundred Meter Minimum Element Size Mesh Contoured According to the Values at Each Point.....	32
Figure 3-4 Close Up of the One-Hundred Meter Minimum Size Function Points Contoured According to their Values	32
Figure 3-5: 2D Mesh Options Window in SMS 10.1	33
Figure 3-6: The Six Major Subpolygons Used to Generate the Mesh Elements Inside the Zero Meter Contour Indicated by the Blue Feature Arcs.....	34
Figure 3-7: SMS 10.1 Dialog Asking Whether to Replace or Append the Existing Mesh.....	35
Figure 3-8: The Upper Piece of Elevation Data and the Two Polygons for Tile Forty-One in SMS 10.1	37
Figure 3-9: A Section of the One Meter Feature Arc to be Removed	39
Figure 3-10: One Meter Contour Arcs Connected to Zero Meter Contour Arcs.....	40
Figure 3-11: A Portion of the Mesh Between the Zero and One Meter Contours.....	41
Figure 3-12: Final Mesh with Color Fill Contours of the Elevations.....	42
Figure 3-13: Close Up View of the Final Mesh where the Potomac River Meets the Chesapeake Bay with Color Fill Contours Representing the Elevations.....	43
Figure 3-14: Select by Data Set Value Tool in SMS	44
Figure 3-15: Materials Data Window in SMS Used to Assign Materials to Mesh Elements.....	45
Figure 3-16: The Final Mesh with the Different Materials Displayed	46
Figure 3-17: Close Up View of the Final Mesh Near Tile Forty-One with Materials Displayed	47
Figure A - 1: Oblique View of the Final Mesh with Color Filled Contours of the Elevations and a Light Source	60
Figure A - 2: Plan View of the Final Mesh with a Light Source and a Background Image.....	60

Figure A - 3: Close Up View of the Final Mesh at the Mouth of the Chesapeake Bay	61
Figure A - 4: Close Up View of the Final Mesh where the Chesapeake Bay Meets the C&D Canal.....	62
Figure A - 5: Close Up View of the Final Mesh where the Delaware Bay Meets the C&D Canal.....	62
Figure A - 6: Close Up View of the Final Mesh at the North End of the Chesapeake Bay, with the Susquehanna River in the Upper Left	63

1 Introduction

A network of triangular or quadrilateral elements constructed from nodes (Aquaveo, LLC 2009), or a finite element mesh, often serves as the basis for hydraulic analyses of water bodies. The United States Army Corps of Engineers (USACE) requested the creation of a finite element mesh representing the Chesapeake and Delaware Bays. The request specified that the mesh should also include the Chesapeake and Delaware (C&D) canal. It further stated that for stability reasons, the mesh should include a portion of the ocean connecting the bays containing at least three times the volume of water contained in the bays. For the shoreline, the request specified that the mesh should include the area up to two meters above sea level. Higher resolution elements should be included along the thalwegs so that they were correctly represented.

The primary purpose for creating this finite element mesh was to verify the capability of the Adaptive Hydraulics Modeling (AdH) program of running successfully with large meshes. Secondary applications for this mesh included determining the residence time of the bays, studying how water flows into the marshes surrounding the bays, studying how long it takes for salt to flush out of the bay, and performing particle tracking analyses. Additionally, the project included identifying any deficiencies or bugs in the software used, as well as possible improvements.

1.1 Background

The Surface-water Modeling System (SMS), developed by Aquaveo, LLC in Provo, UT, is a graphical user interface used to create, visualize, manipulate, and analyze numerical data representing surface water (Aquaveo, LLC 2012). I used the mesh generation tools in SMS for this project.

AdH is a hydraulic modeling system developed by the Coastal and Hydraulics Laboratory of the Engineering and Research Development Center (ERDC), which is a division of the USACE. The program is capable of solving the Two-Dimensional Shallow Water, Three-Dimensional Navier Stokes, and Ground Water Flow equations. AdH uses a base mesh created by the user and modifies the size of the elements based on the results of each iteration. AdH also features rapid convergence of flows to steady state solutions, wetting and drying, completely coupled sediment transport, and wind effects (Coastal Hydraulics Laboratory, ERDC, USACE 2007).

ERDC provided bathymetric data as eighty-one .flt files with Digital Elevation Models, or DEMs, in each file (also referred to as tiles). Each of these files was between 61.3 and 95.4 MB in size. Figure 1-1 shows one of these DEMs loaded as an raster in SMS 11.0 Beta.

The DEMs provided included approximately two billion DEM points. During the course of the project, I found that SMS 10.1 was unable to load any of the .flt files as a triangulated scatter set, which was the only way to load these files into SMS 10.1. While searching for a work around to this problem, I found that I was able to load one of the files into version 8.3 of the Watershed Modeling System, or WMS (Aquaveo, LLC 2012). As such, I used WMS 8.3 to load the data and export them as a format that SMS 10.1 was able to read. I also filtered and smoothed

these points in WMS 8.3 to remove points in flat areas and remove zigzags in the contours, respectively.

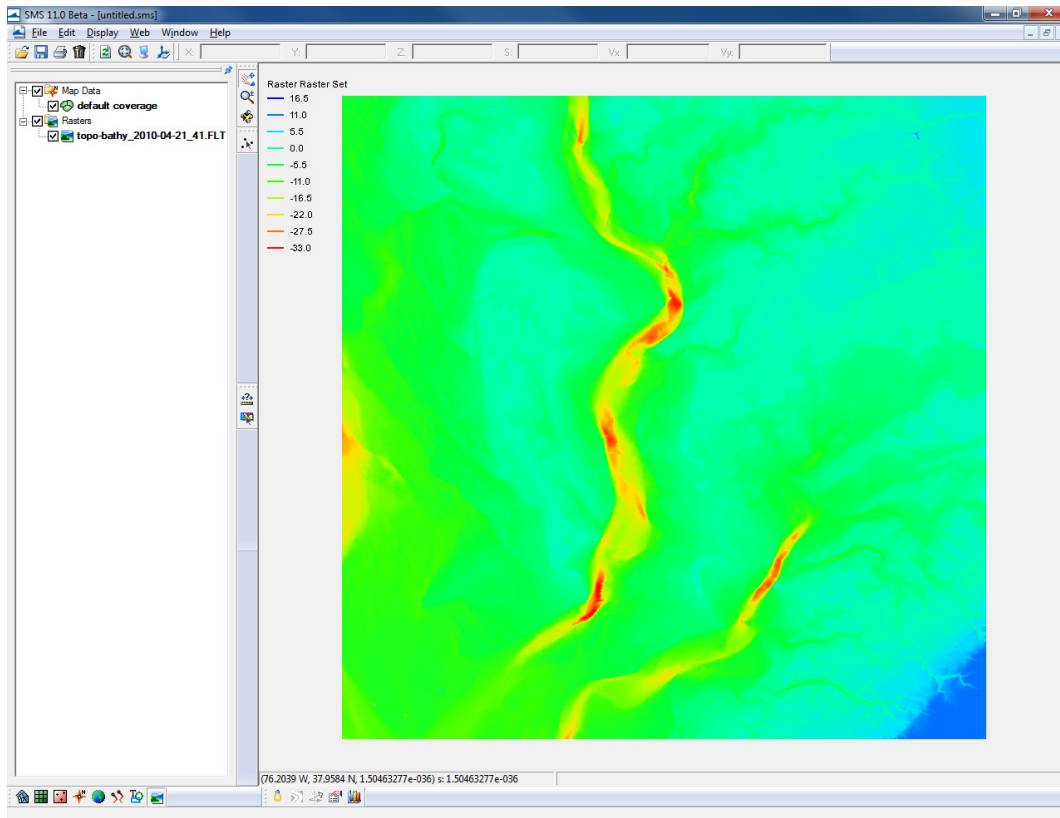


Figure 1-1: DEM Representing the Elevations of Tile Forty-One Loaded as a Raster in SMS 11.0 Beta

I then used these points to create feature arcs representing the zero and two meter contours. To ensure accurate model results, I bounded the domain in the ocean by an arc enclosing enough area in the ocean so that the ocean contained roughly three times the amount of water as the bays. These feature arcs constituted the model domain. I also created thalweg arcs to properly represent the channels using TOPAZ within WMS 8.3. TOPAZ, a topographic parameterization program which determines flow paths for a given scatter set (United States Department of Agriculture - Agricultural Research Service 2012).

I then created a size function based on the bathymetry so that deeper portions of the domain would contain larger elements, while the shallower portions would contain smaller elements. I then converted the map data to a finite element mesh and assigned elevations and materials to the elements.

The following list contains the general steps I followed to generate the mesh which will be discussed in detail in subsequent chapters:

- Generated contour arcs in WMS 8.3 from elevation datasets provided by ERDC
- Defined the mesh domain boundaries
- Created thalweg arcs in WMS 8.3 using TOPAZ
- Created the size functions used to guide the mesh generation
- Converted the map data to a finite element mesh in SMS 10.1
- Assigned elevations to the mesh nodes from the data provided by ERDC
- Extended the mesh to the one meter contour and assigned elevations to the mesh nodes between the zero and one meter contour
- Assigned material types to the mesh elements

2 Selecting and Creating the Mesh Boundaries with Feature Arcs

The first task in the project was to select the domain boundaries. I used the zero-meter contour, which represented the shoreline, as one of the boundaries. I also used the one-meter contours to define the domain extents of the Chesapeake and Delaware Bays, including their tributaries. The final boundary of the domain was the ocean boundary. This included the extents up and down the shoreline, which I also connected to the ocean boundary.

To define these extents, I used the elevation data provided by ERDC to generate feature arcs from the zero and one meter contours. I converted these contours to feature arcs in WMS. I created the ocean boundary so as to include all of the area inside the one-hundred meter depth contour and to provide sufficient water storage. This section describes the process of selecting and creating the boundaries of the finite element mesh using WMS and SMS.

2.1 Generating Contour Arcs from Elevation Datasets

The geometric (DEM) data provided by ERDC consisted of approximately two billion points about eight meters apart representing elevations of both land and water. Figure 2-1 shows the location of each tile provided.

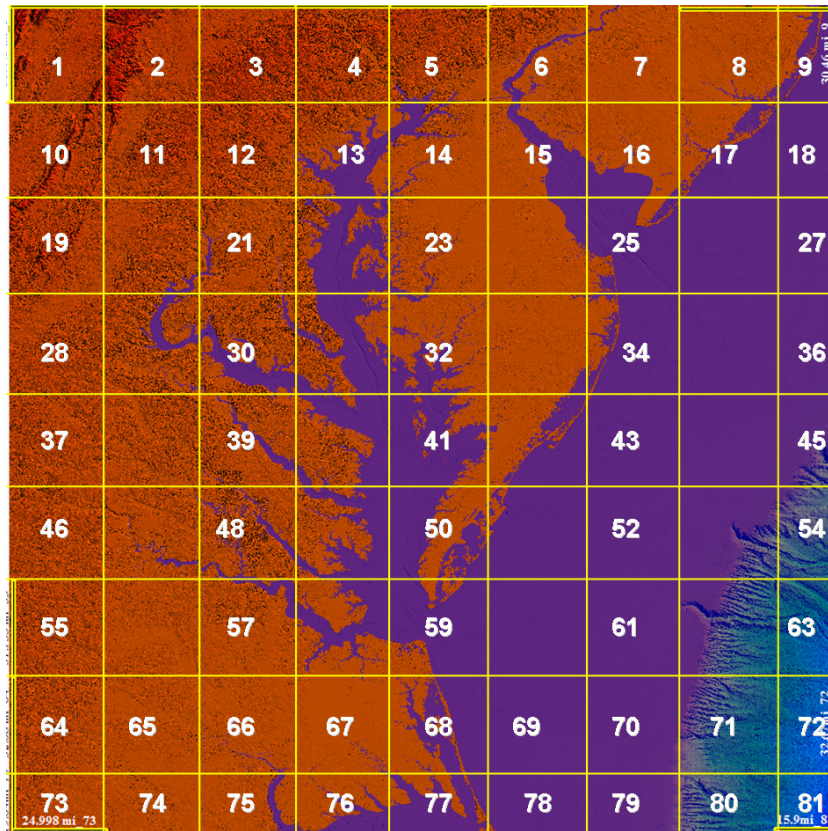


Figure 2-1: Location of Each of the DEM Files Provided by ERDC with the Elevation Data

I first used these data to generate feature arcs that represented the boundaries for the mesh. As mentioned previously, I used WMS 8.3 for this task because SMS was unable to load one of these files. I had to import these datasets representing elevation into WMS 8.3 one at a time due to hardware and software memory limitations. I used WMS in place of SMS for this task due to the more efficient manner in which WMS handled the DEM datasets. In fact, SMS 10.1 was unable to load one tile with a thinning factor of one (i.e. SMS does not remove any points from the data set). Figure 2-2 shows linear contours for one of these files loaded in WMS 8.3.

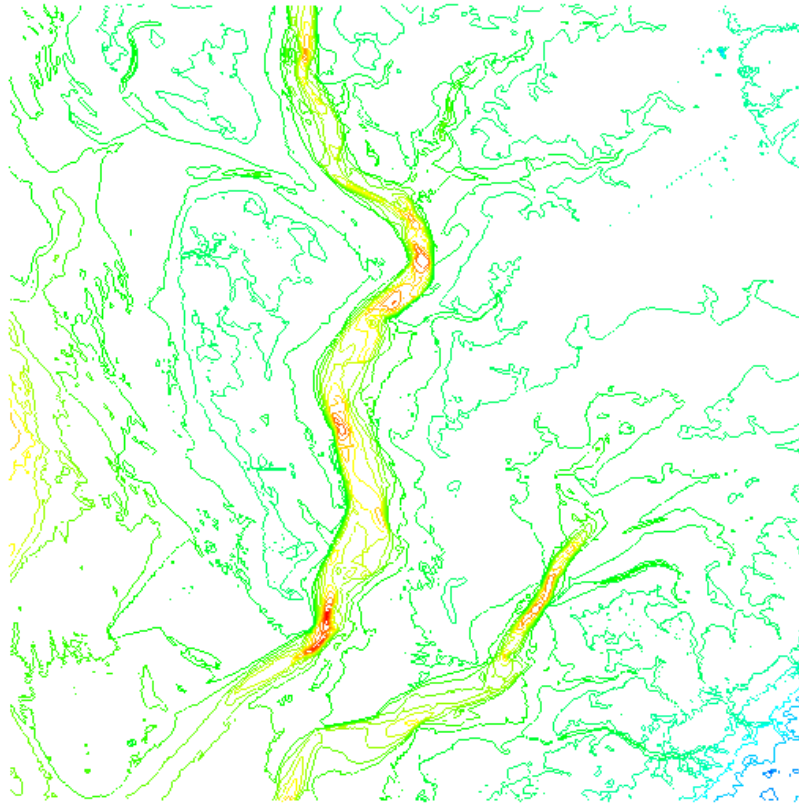


Figure 2-2: Tile Forty-One in WMS 8.3 with the Default Linear Contours

2.1.1 Displaying and Converting Contours to Feature Arcs

Once I loaded the DEM points into WMS, I then generated linear contours representing the zero and one meter elevation contours. I did this by using the normal linear contour method in the display options of WMS 8.3. To make the different contours, I entered the desired contour value in the contour options of the display options window. Once the contours were displayed in the main window, I converted the contour to feature arcs in WMS. Next, I saved the contours for each tile as a map file for later use in SMS. I used this process for the zero, one, and two meter contours, but we decided later to not use the two meter contours. Figure 2-3 shows the zero meter contour for tile forty-one.



Figure 2-3: Zero Meter Contour for Tile Forty-One in WMS 8.3

A few of the tiles caused WMS to crash when trying to convert the DEM contours to feature arcs. To get around this obstacle, I smoothed the problematic tiles first with the default options selected. The algorithm in WMS 8.3 for smoothing DEMs takes an inverse distance weighted average of a user-specified number of neighboring DEM points and uses this as a new elevation for the point (Aquaveo, LLC 2008). After smoothing, I was then able to convert the tiles to feature objects without the crash. At this point, I decided to go back and smooth the tiles I had not smoothed originally to be consistent. Figure 2-4 shows a portion of one of the tiles before smoothing the data, while Figure 2-5 shows the same portion after smoothing the data in WMS 8.3.

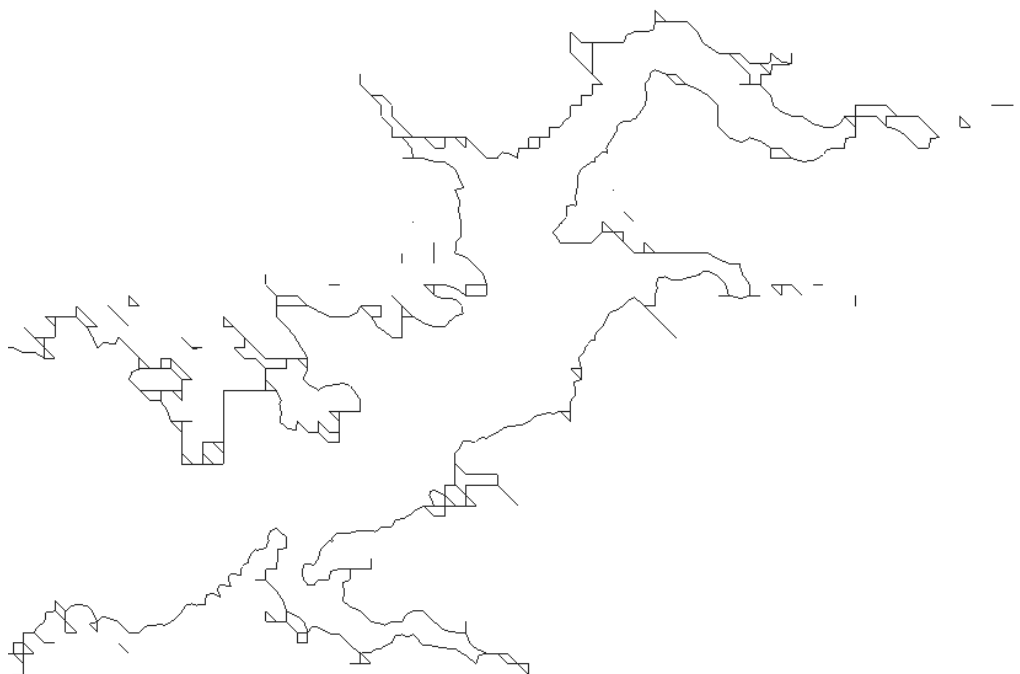


Figure 2-4: Zero Meter Contour of Tile Forty-One Before Smoothing in WMS 8.3

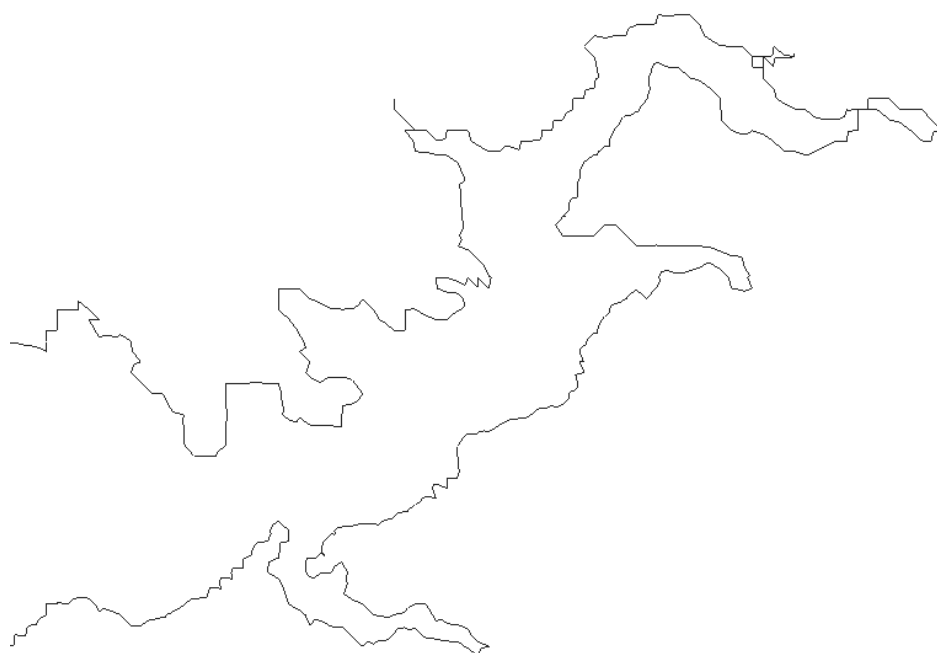


Figure 2-5: Zero Meter Contour of Tile Forty-One After Smoothing in WMS 8.3

I converted these feature arcs from contours in WMS and used them to define the mesh domain boundaries. To generate the boundary arcs for the bays and shoreline, I converted the contours representing the zero and one meter elevations to feature objects. I created the one and two meter contours in different folders and coverages in the Map Module of WMS 8.3. I did this with the intent of keeping the contours more organized when loading them into SMS. I created a folder for the contour arcs of each tile, and within each of these folders, I created a coverage for the zero and one meter contour feature arcs.

Finally, I exported the feature objects from WMS as a map file, which SMS could read later. I repeated this process for each of the tiles. Of the eighty-one tiles, forty-two were in the mesh domain. Therefore, I created forty-two coverages for both the zero and one meter contour feature arcs. This resulted in 5,012 zero meter and 6,510 one contour feature arcs.

2.1.2 Cleaning and Editing the Contour Feature Arcs

I then imported these map files into SMS 10.1 one at a time and cleaned them individually. This cleaning process involved both automated and manual editing techniques to remove unwanted arcs, modify boundaries, and remove and/or merge points along arcs to more accurately represent the water bodies.

The automated cleaning included snapping nodes together that were within a specified tolerance and deleting dangling arcs (Aquaveo, LLC 2007). I used a value of thirty meters for the tolerance for snapping nodes together since the project sponsor requested a thirty meter minimum element length for the mesh. The snap nodes option merges two nodes together if they are within the specified tolerance. The merged node will be at the location of one of the source nodes.

Dangling arcs were another problem that I addressed with an automated cleaning tool in SMS. When a contour extended from another contour and then ended, this resulted in dangling arcs in the converted feature arcs. This occurred when there was a series of neighboring data points in the DEM whose values were the same as the contour that I was extracting contours. This phenomenon is illustrated in Figure 2-6. The dangling arc goes through three points with a value of zero to the lower right of the page, which is the same value for which I generated the contours in this case.

Figure 2-6: Dangling Arcs Created when there is a Series of Neighboring Points with the Same Value as the Contour

The remove dangling arcs command deletes the arc that does not form part of a loop and that are shorter than a specified tolerance. I used a tolerance of thirty meters since that was the requested minimum element length. Figure 2-7 illustrates these two automated cleaning tools.

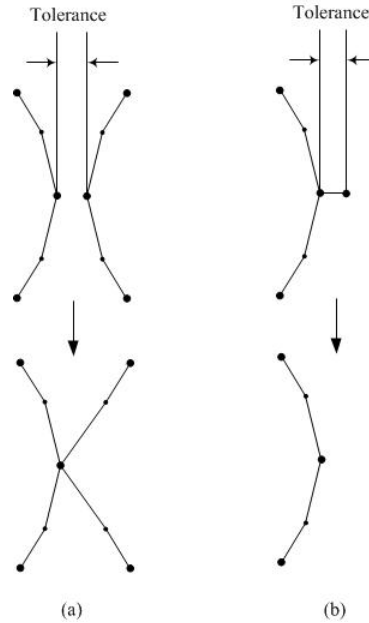


Figure 2-7: Illustration of the (a) Snap Nodes and Vertices and (b) Remove Dangling Arcs Cleaning Tools in SMS 10.1

I also manually cleaned the feature objects to fix problems that the automated cleaning tools did not address. The following are the manual cleaning processes that I used in addition to the automated cleaning tools:

- Pit/depression removal from the land side of the zero meter contour arcs
- Narrow waterway removal
- Small island removal
- Headland/peninsula preservation using hard points

To remove pits/depressions on land, I identified and deleted polygons that represented pits/depressions on the land side of the zero contour arcs. These areas below the zero or one meter elevation contours did not connect to the main water body and therefore did not make up part of the domain. Similarly, I deleted enclosed arcs that represented islands that were less than

thirty meters wide. Figure 2-8 shows one example of some of pits on the land side of the zero meter contour arcs which I manually deleted.



Figure 2-8: Pits Shown on the Land of a Portion of Tile Forty-One

The conversion from linear contours to feature arcs created a node at all intersections of contours, many of which were unnecessary. I converted nodes which were not endpoints of arcs to vertices so that the mesh resolution would not be affected by extraneous nodes. I did this by selecting all the nodes and using the convert nodes to vertices tool in SMS. This converted all extraneous nodes to vertices.

Water channel arcs, which can be seen in Figure 2-8 where the blue areas enter into the land, are pairs of arcs that represent a channel of water. Since the desired minimum element length of the mesh was thirty meters, I removed water channel arcs narrower than thirty meters

from the model. To do this, I created feature arcs across zero meter contour arcs representing water channels where they came within thirty meters of each other. I then deleted all the arcs remaining on the land side of zero contour arc. I utilized the measuring tool available in SMS 10.1 to determine where these feature arcs were created. Figure 2-9 shows a river which became narrower than thirty meters (a) and was therefore truncated (b).

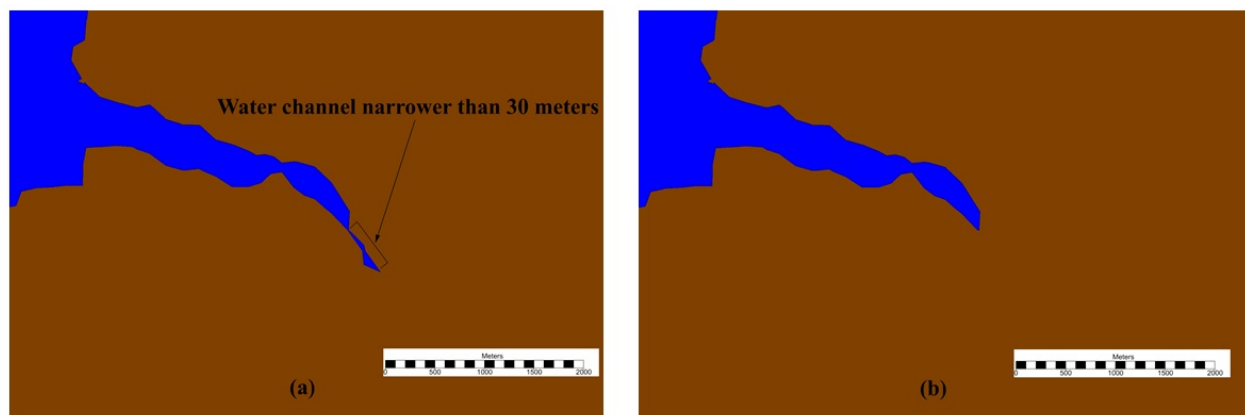


Figure 2-9: Feature Arcs Representing a River Before (a) and After (b) Being Truncated

I kept feature arcs that represented narrow peninsulas extending into water since they could potentially influence the circulation in the bays. Circulation must go around these peninsulas. However, I deleted the arcs representing water extending into land where they became narrower than thirty meters, since, as explained previously, I deleted arcs less than thirty meters apart because that was the target minimum mesh element length.

I then created nodes at specific points along the zero meter contour arcs where the boundary was to remain fixed (also referred to as hard points). I placed these points at the ends of feature arcs that extended into water and where water channel arcs intersected met the coast to preserve key features in the domain. This prevented the scalar paving mesh generation method

that was used to generate the mesh elements from changing the boundary at key locations. I created these points by converting existing vertices to nodes if there was already a vertex present, or creating new nodes at these specific locations along the feature arcs. Figure 2-10 shows an example of where I created hard points.



Figure 2-10: Illustration of Locations of Hard Points

2.1.3 Creating the Composite Shoreline

Next, I loaded all of the map files containing the zero meter contour arcs into. I then merged them into one coverage to represent the boundary between land and water. The feature arcs created in WMS for each elevation data set did not extend out to the boundary for the tiles. To fix this, I created feature arcs that connected feature arcs from the different tiles together to

make the zero meter contour arcs contiguous. I loaded aerial imagery into SMS when it was unclear how the feature arcs connected at the boundaries of the tiles with elevation data. Figure 2-11 shows the feature arcs in these combined coverages which determined the landward extents of the mesh.

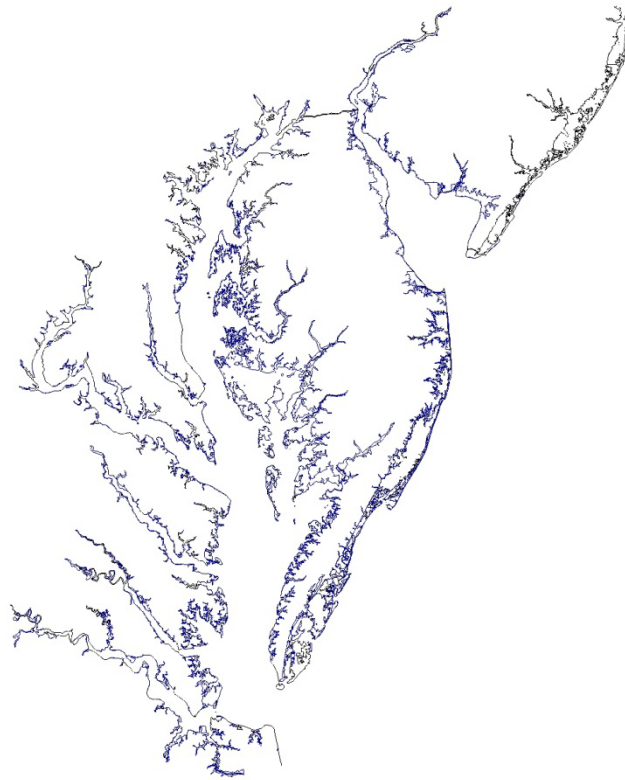


Figure 2-11: Zero Meter (Black) and One Meter (Blue) Contour Feature Arcs

Figure 2-12 shows a close up view of these feature arcs. The zero meter contour feature arcs are represented by the thicker black lines, while the one meter contour arcs are represented by thinner blue lines.

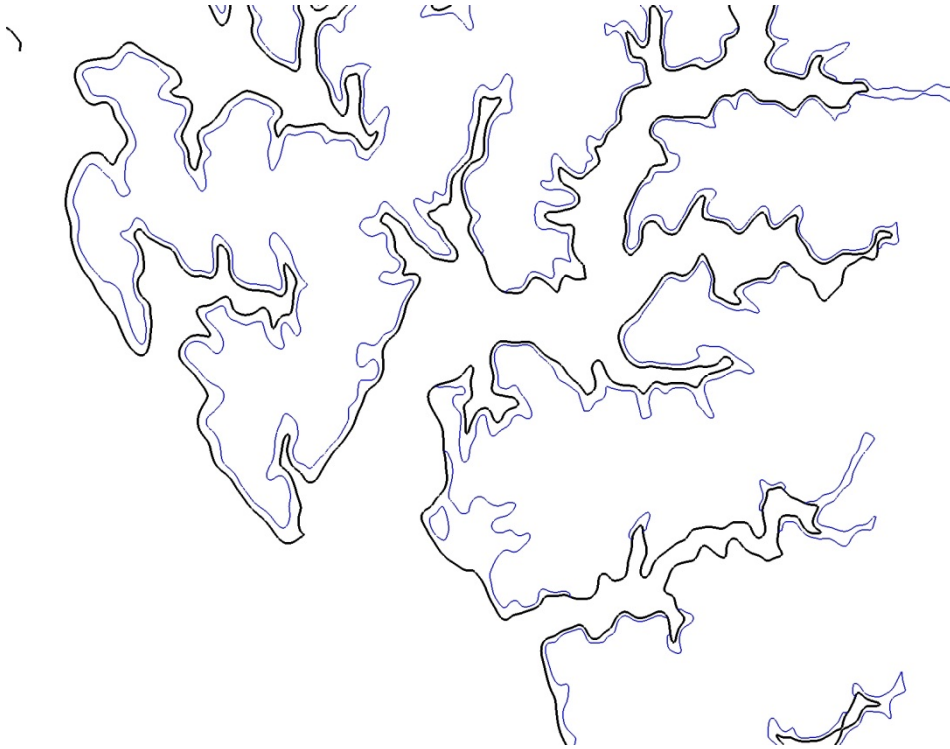


Figure 2-12: Close Up View of the One and Zero Meter Contour Arcs

2.2 Defining the Model Domain Boundaries

The next task involved defining domain boundary in the ocean to define rest of the domain, building polygons, and assigning attributes to polygons to specify where mesh elements would and would not be generated. This section explains this task.

2.2.1 Selecting the Ocean Boundary

To obtain accurate circulation results when running numerical models with an enclosed body of water, the mesh domain needs to extend out far enough into the ocean to include sufficient water storage.

To fulfill this requirement, I created the ocean boundary so that the volume of water in the portion of the mesh representing the ocean would be three times as much as the volume in the bays and their tributaries. An online search for information about the two bays, along with an analysis of the elevation data provided by ERDC, revealed the approximate surface areas (Chesapeake Bay Program 2012) (THANKYOUDELAWAREBAY.ORG 2008). I used these estimates to calculate the approximate volume of the Chesapeake and Delaware Bays and their tributaries. Table 2-1 shows these approximated and calculated values.

**Table 2-1: Approximate Areas, Depths, and Volumes of Chesapeake and Delaware Bays
Used to Determine the Ocean Boundary**

	Chesapeake Bay	Delaware Bay
Area, km²	12,000	2,000
Average Depth, m	6.4	6.2
Approximate Volume, km³	74	13

I then read the DEM files and coastline arcs into SMS, created a candidate ocean polygon, and computed the area and volume of this candidate. The select feature polygon tool in SMS allowed for an easy selection of all points in the ocean polygon. SMS displays the average value of the selected data set for selected entities. I used this functionality to estimate the average depths.

I then compared this ocean volume to the volume of the bays, and adjusted the candidate boundary until the ocean volume was at least three times the volume of the bays. The ratio of volume in the ocean to volume in the bays and tributaries was 3.42 for the first candidate boundary that met the requirement. Figure 2-13 shows this candidate boundary with a georeferenced aerial photo of the area.

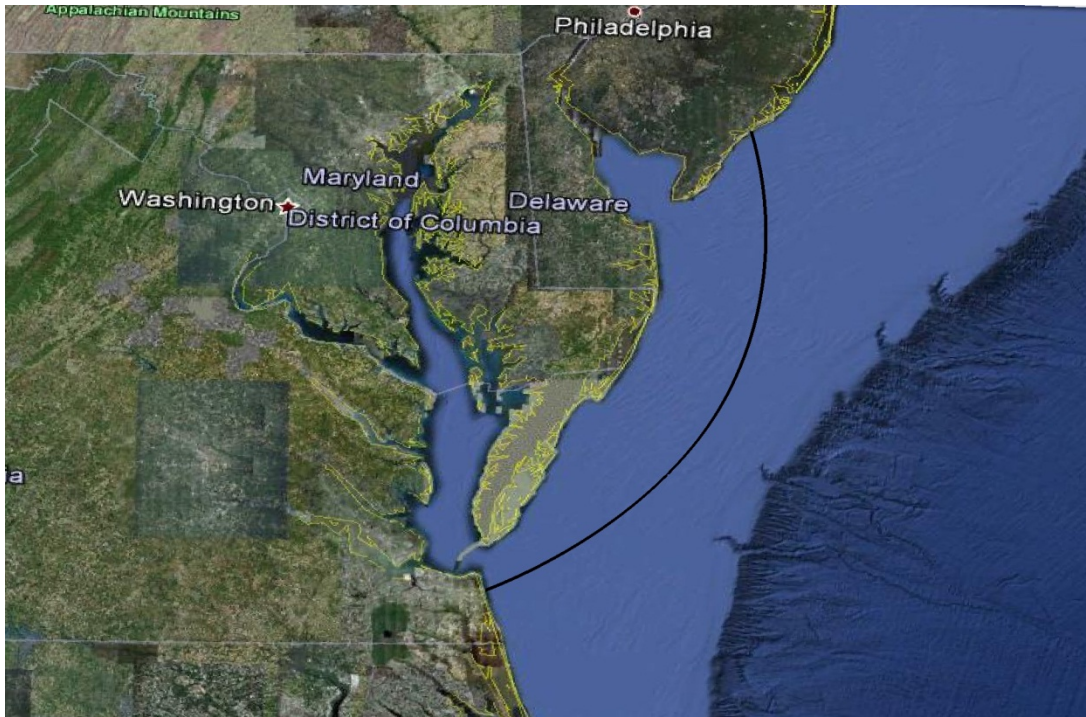


Figure 2-13: First Candidate Ocean Boundary Arc that Met the Water Storage Requirements

After submitting these boundaries to the sponsor of the project, they requested that the ocean boundary be extended. I therefore extended the ocean boundary close to the ocean trench. The sponsor also requested that the ocean boundary arc be rectangular in shape, rather than circular. Therefore, I changed the ocean boundary arc to an open rectangle instead of a curved arc. Figure 2-14 shows the feature arc that was used to represent the ocean boundary in the final mesh.

Table 2-2 shows the volumes in both the bays and tributaries, as well as the ocean. The volume ratio of ocean to bays and tributaries of the final mesh was 8.23, which exceeds the required minimum ratio of 3.



Figure 2-14: Ocean Boundary Feature Arc Used to Generate the Final Mesh Georeferenced Above an Aerial Photo of the Site from Google Earth

Table 2-2: Volumes of Bays/Tributaries and Ocean in the Final Mesh

	Bays and Tributaries	Ocean	Ocean to Bay/Tributaries Ratio
Volume, km³	89	733	8.23

2.2.2 Building Polygons

I then read the map files with feature arcs representing the shoreline into SMS and merged the coverages. I also connected these feature objects to form closed loops. Polygons then needed to be built using these combined feature arcs. I used SMS to build polygons from the closed loops formed from the feature arcs to represent the mesh domain (Aquaveo, LLC 2012).

Building polygons for the entire domain at once did not work because of problems with the creation of feature arcs from the DEM contours. The automated cleaning tools discussed previously were used to automatically remove obvious errors with the feature objects. However, the following issues also caused problems with building polygons with the cleaned feature arcs:

- degenerate arcs in the map files
- arcs with duplicate IDs
- arcs that did not form a closed loop because their end nodes were less than the tolerance for the snap nodes cleaning tool

Degenerate arcs are objects that are not properly formed as feature arcs. They lack an arc that has nodes at each endpoint. As such, SMS does not recognize them as feature arcs. The degenerate arcs in the zero and one meter feature arc files were most likely introduced into the map files when the contours from the tiles were converted to map files. There were also duplicate IDs for many of the feature objects. Duplicate IDs might have resulted from the merging process in SMS. Both degenerate arcs and multiple feature objects with identical identification numbers cause problems in the polygon building process.

To work around these problems, and to identify where the problems were, I created subareas and built smaller polygons. In the process of building these subpolygons, I found many problems in the arcs that I generated from the DEM contours that caused problems while attempting to build polygons. By splitting the domain up into smaller areas, I was able to identify and fix most of the problems.

To identify the problematic arcs, I split the polygons into subpolygons with feature arcs, and then tried to rebuild polygons for one of the subpolygons. This sometimes required several iterations of this process, but this led me to the problematic arc. However, in some cases, I was

unable to determine why the build polygon command failed. I submitted these issues to the SMS development team and they addressed these issues. The process of building polygons for the entire domain of the final mesh resulted in 205 polygons.

2.2.3 Assigning Properties to the Map Polygons

In SMS, polygons are assigned properties that dictate how the polygons will be used in the mesh generation process. These properties include the mesh generation type, source for elevation, and the material that will be assigned to the mesh elements in that area (Aquaveo, LLC 2009). Figure 2-15 shows the polygon properties window in SMS.

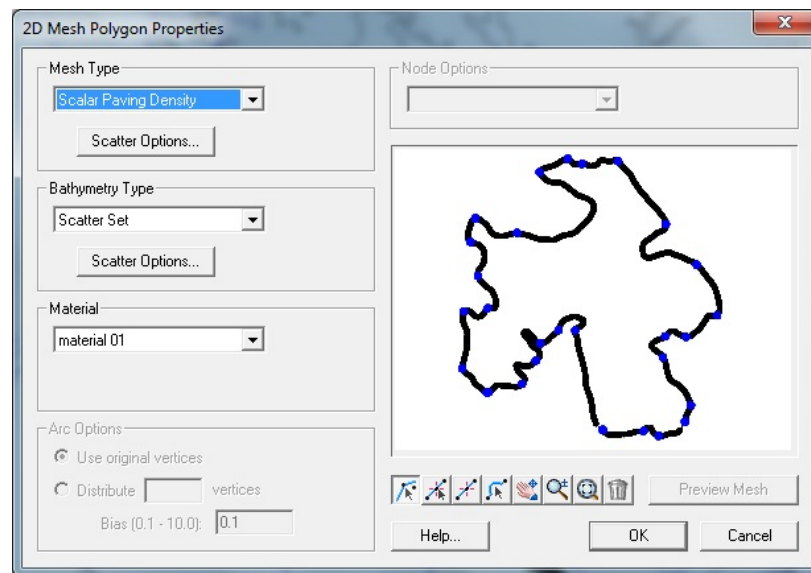


Figure 2-15: Polygon Properties Window in SMS

I assigned properties to each polygon. For polygons where mesh elements were to be generated, I assigned polygon attributes that specified scalar paving density as the mesh type and a combined but low resolution scatter set with elevation data the bathymetry type. I assigned

elevations to the mesh with the high resolution elevation data from ERDC since I was unable to load all of that data at once. I also specified a scatter set to be used in the scalar paving density mesh generation, which is discussed in section 3.1. Finally, I entered an extrapolation value of -9999 so as to be able to easily identify areas that did not have valid bathymetric values.

To ensure that elements spanned the entire width of the C&D Canal, I assigned the polygons in that area the patch mesh type. The patch mesh type was favorable to the scalar paving density mesh type for the C&D Canal due to the Canal banks being parallel to each other (Aquaveo, LLC 2008). The patched elements in C&D Canal can be seen in both Figure A - 5 and Figure A - 6 in the Appendix.

I assigned polygons that were not to be included in the mesh (islands) a mesh type of “None” in the polygon attributes window. In most cases, I selected these polygons by selecting the polygons to be meshed and then inverting the selection since there were many “island” polygons and only a few large “water” polygons. In other cases, I manually searched for polygons representing islands and specified a mesh type of “None” in the polygon attributes so that the mesh would not be created within these areas.

2.3 Creating Thalweg Arcs in WMS Using TOPAZ

A thalweg is a line that defines the bottom of a water channel or valley. These lines consist of the lowest points through a given set of elevation points. I created thalweg arcs using TOPAZ in WMS to properly represent the deepest parts of the bays and channels. Creating a thalweg arc in the mesh domain causes SMS to create mesh elements along this arc, thus ensuring that the sides of the mesh elements align with the deepest parts of the channel. This helped ensure that conveyance areas in the channel were accurate.

2.3.1 Loading the Elevation Data Files into WMS

I first loaded the elevation datasets which covered the determined model domain into WMS one at a time. During this process of loading the elevation data into WMS, I found that the elevation data loaded into WMS in just a few seconds, which was significant because they would cause SMS to crash or freeze and would not load at all. Also, I found that TINs in a gdm file format exported from WMS could be successfully loaded into SMS. This provided a way to load the elevation datasets into SMS since attempting to load the raw elevation datasets would cause SMS to crash or freeze.

2.3.2 Running TOPAZ in WMS

TOPAZ, or the Topographic Parameterization Program, was developed by the Agricultural Research Service, which is an agency of the United States Department of Agriculture, by Dr. Jurgen Garbrecht and Dr. Lawrence W. Martz (Martz n.d.). It is a software system that analyzes landscape topography from DEMs, or Digital Elevation Models. It determines the flow path of water based on the relative elevations of each point in the DEM. To do this, TOPAZ compares the elevation of each cell to the eight surrounding cells to determine the steepest slope from each cell (United States Department of Agriculture - Agricultural Research Service 2012).

I ran TOPAZ for each of the tiles with elevation data to find the bottom of the channels, or thalwegs. TOPAZ determines the cells that represented streams (or an accumulation of water) for a given tolerance and outputs these cells as streams. Figure 2-16 displays the stream or thalweg cells that TOPAZ determined for one of the tiles.

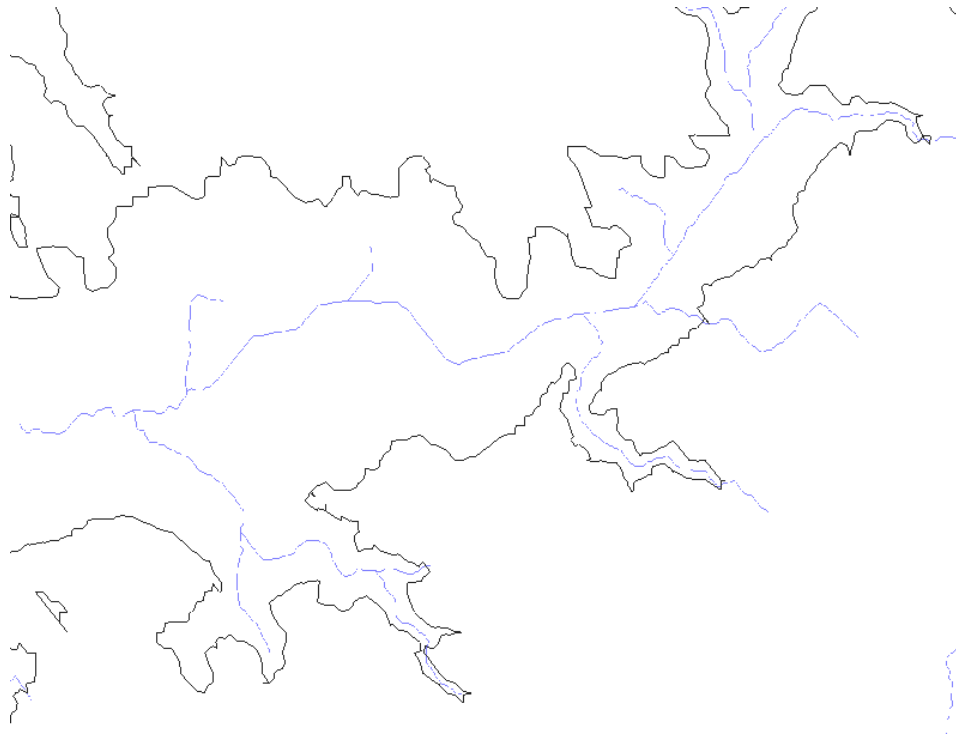


Figure 2-16: A Portion of Tile Forty-One with Thalweg Cells (blue) and the Zero Contour (Black)

Using the DEM Contours to Feature Arcs command in WMS 8.3, I converted these thalweg cells to feature arcs. I then exported them from WMS as map files (one for each tile) so I could use them in SMS to generate the mesh. Once I loaded them into SMS, I cleaned each of them using both the automated and manual cleaning tool techniques discussed previously. After I generated the thalweg arcs, I loaded them into SMS and cleaned them. I then merged them with the feature arcs representing the domain boundary described previously. Figure 2-17 shows the resulting coverage which represents the model domain.



Figure 2-17: Feature Arcs Representing the Zero Meter Contour and the Thalwegs

3 Generating the Mesh

After generating the coverage to represent the domain of the mesh, I then converted that data to a finite element mesh. This chapter describes this process, which is summarized in the following list:

- Created the size function scatter sets
- Converted the map data within the zero meter contour to a 2D mesh
- Assigned elevations to the mesh nodes
- Created additional mesh elements between the zero and one meter contour
- Assigned material types to the mesh elements

3.1 Creating the Size Function Scatter Sets

When using the scalar paving density mesh generation option in SMS, the mesh element lengths are determined using a scatter set called a size function, or size data set. The value assigned to each scatter point in the size function determines the length of elements in the vicinity of that point. SMS interpolates the desired size at a location from surrounding size function points. For additional information regarding size functions, see (Howlett 2005).

We created two different size functions for consideration in the mesh generation process. These functions considered the local depth and the desired minimum size. Towards the end of the project, we decided to increase the minimum element length from the original value of thirty

since we determined that it would create more elements than desired. The first size function had a minimum element side length of fifty meters, while the second had a minimum element side length of one hundred meters. The steps I took to create these size function scatter sets are outlined below:

- Created the scatter set to which I assigned the size function values
- Selected ranges for the size function values based on elevation values
- Created a function in Microsoft Excel to linearly interpolate size function values between the selected values
- Assigned size function values to the scatter points by entering the Excel function in the Data Calculator in SMS
- Smoothed the size function values

I first created a scatter set to which I could then assign size function values. I created this scatter set from the elevation data provided by ERDC. I removed all but every tenth point from the elevation data set so that I could load it into SMS while having points covering the entire model domain. To properly define the element sizes at key features of the domain, I merged the scatter set with the low resolution elevation points with the points along the zero meter contour and thalwegs at the high resolution of the original elevation data. Using the zero meter and thalweg high resolution scatter sets in conjunction with the thinned scatter set ensured that the elements properly reflected the coastlines and the deepest parts of the channels.

Next, I determined the minimum and maximum values of this elevation scatter set. We selected a few elevations within this range of elevations, and their corresponding target values for desired mesh element edge lengths to guide the mesh resolution. Table 3-1 shows the values

we selected for the target size function values. We interpolated the values followed by an asterisk between target values.

Table 3-1: Target Values for Selected Depths for the Two Size Functions

Depth (m)	Target Values for 50m Size Function (m)	Target Values for 100m Size Function (m)
0	50	100
1	50	100
3	100	121*
5	200	143*
15	1044*	250
30	2311*	500
50	4000	4000

*Values resulting from interpolation between target values

We designed and refined equations to create the two size functions in Microsoft Excel. The equations that we designed used the depth at a point as the input and output the size function value at that point. The equation linearly interpolated the size function values between the selected values shown in the previous table. Equation (3-1) shows the fifty meter minimum size function equation, while Equation (3-2) shows the one-hundred meter minimum size function equation, both of which were created in Microsoft Excel.

$$\begin{aligned}
 \text{Size, min 50} = & \max(50, \max(50 + (-a - 1) * (100 - 50)/(3 - 1), \\
 & \max(100 + (-a - 3) * (200 - 100)/(5 - 3), \\
 & 200 + (-a - 5) * (4000 - 200)/(50 - 5)))
 \end{aligned}
 \tag{3-1}$$

$$\begin{aligned}
 \text{Size, min 100} = & \max(100, \max(100 + (-a - 1) * (250 - 100)/(15 - 1), \\
 & \max(250 + (-a - 15) * (500 - 250)/(30 - 15), \\
 & 500 + (-a - 30) * (4000 - 500)/(50 - 30)))
 \end{aligned}
 \tag{3-2}$$

The values grow slowly at lower depths and then rapidly as the depths increase. Figure 3-1 shows the depths and their corresponding size function values. The points indicate the target values that we selected, and the lines show the linearly interpolated values between those points.

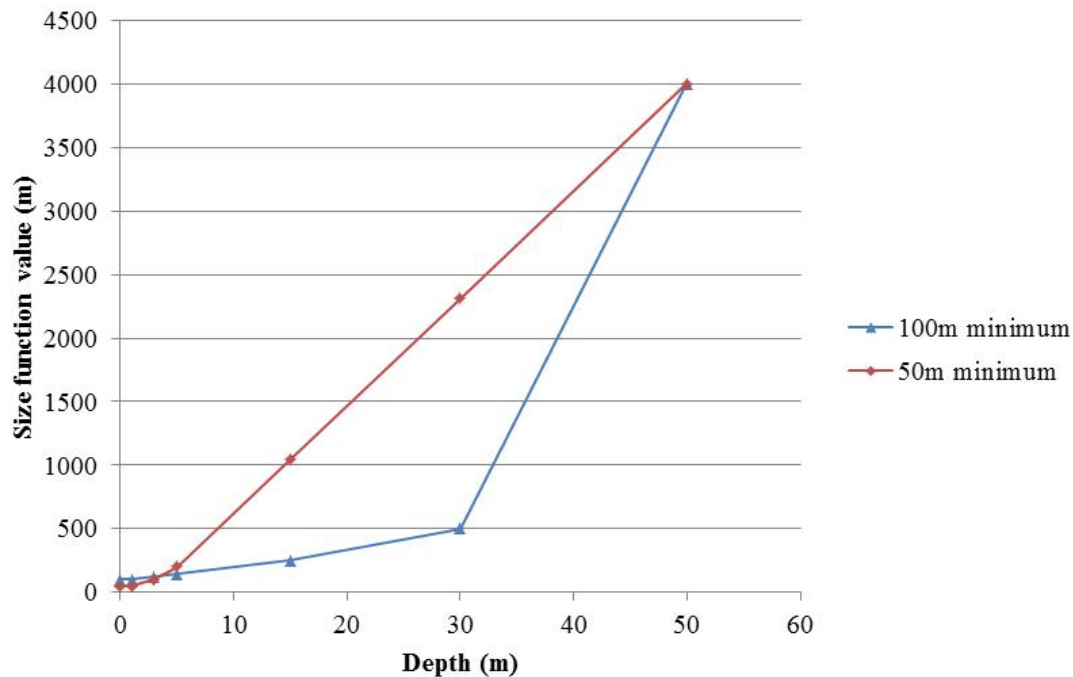


Figure 3-1: Plot of the Fifty Meter and One-Hundred Meter Minimum Size Functions

We designed the equations so that it would output higher values for scatter points with larger depths and lower values for scatter points with smaller depths. This caused the mesh resolution to be higher in shallow areas and lower in deeper areas, which was necessary to accurately represent the bathymetry in the shallower areas. We created a second equation by modifying the Excel spreadsheet that was designed for the first equation.

I entered these equations that we designed and modified in Excel in the Data Calculator in SMS 10.1 to create a base size function for each scatter point for the two size functions. This

resulted in two size function datasets associated with the scatter set I created from the elevations, zero meter contours, and thalwegs.

I then smoothed the resulting size function datasets to ensure that there was a smooth transition from smaller elements to larger elements. Smoothing a size function data set is desirable because abrupt changes in area of adjacent mesh elements can cause poor element shape, thus creating numerical stability problems when running numerical models.

I smoothed the datasets in SMS using a maximum area change limit of 0.5 and anchoring the size data set values to the minimum value. The maximum area change value defines the maximum ratio between adjacent points based on the distance between points. Specifying the minimum value as the anchor causes the data set values to increase and results in a more refined mesh with smaller mesh elements (Aquaveo, LLC 2010). Figure 3-2 shows these input parameters in the Dataset Toolbox in SMS.

Figure 3-3 shows the final smoothed size function data set for the one-hundred meter minimum element size mesh.

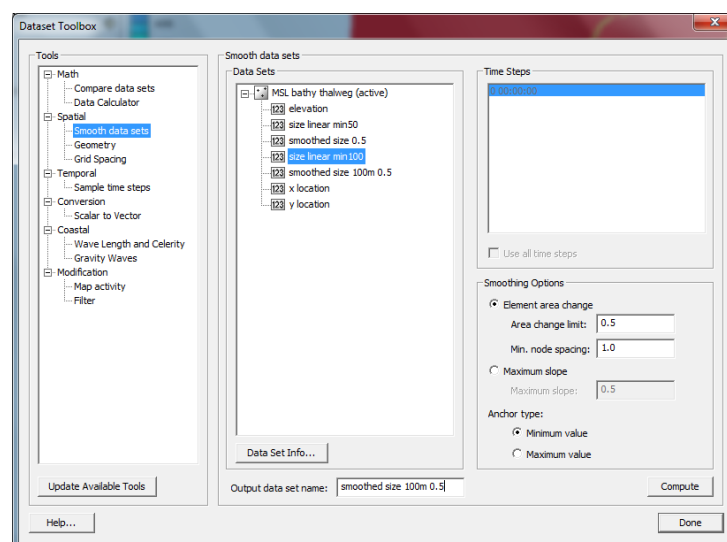


Figure 3-2: Smooth Data Set Tool in the Dataset Toolbox in SMS 10.1

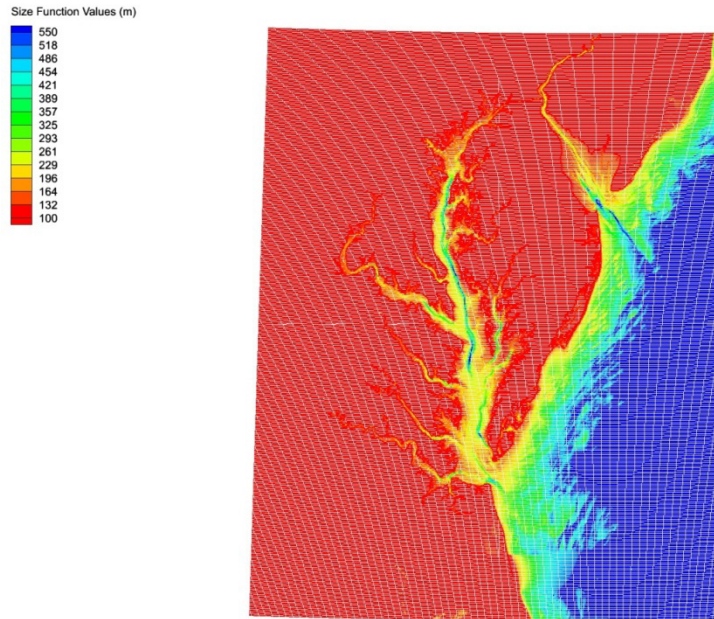


Figure 3-3: Final Size Function for the One-Hundred Meter Minimum Element Size Mesh Contoured According to the Values at Each Point

Figure 3-4 shows a close up of the points in the one-hundred meter minimum size function contoured according to the size function value at each point.

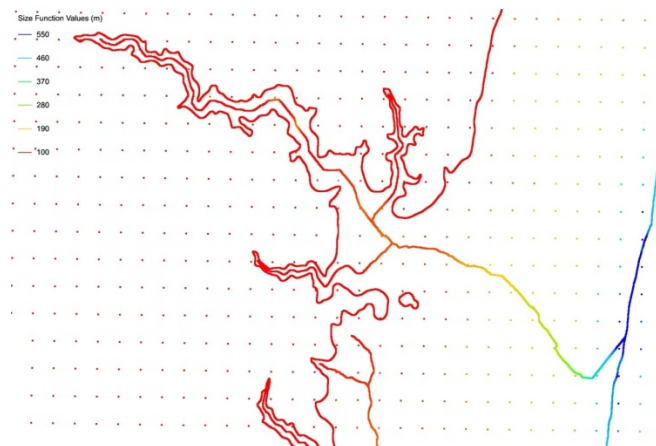


Figure 3-4 Close Up of the One-Hundred Meter Minimum Size Function Points Contoured According to their Values

Once I created the size function, I specified it as the scatter set to be used for scalar paving density for each polygon within the mesh domain. I did this by going into the polygon properties and selecting the scatter set representing the size function to be used for the mesh while it was loaded in the project.

3.2 Converting the Map Data to a Finite Element Mesh

After defining the mesh boundaries, creating the thalweg arcs, and assigning the polygon attributes, I was then able to create the finite element mesh from the map data associated with the polygons. The process of converting the map data to a mesh involved invoking the map to 2D mesh command in SMS with the map data properly set up as described previously. Figure 3-5 shows the 2D Mesh Options window that this command opens.

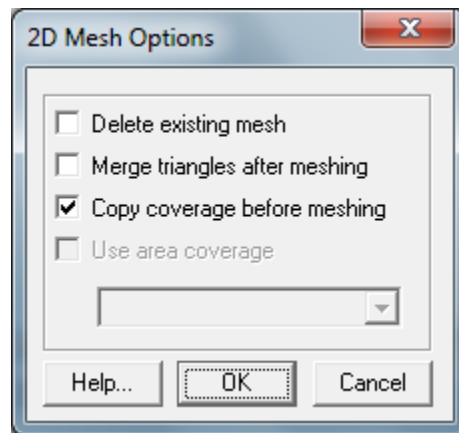


Figure 3-5: 2D Mesh Options Window in SMS 10.1

Due to the large area and complexity of the model, I was unable to generate the mesh for the entire domain at once. Consequently, I divided the domain into six subpolygons by creating arcs defining these subareas, rebuilding polygons, and specifying the polygon attributes for the

new polygons built. The trials described in section 2.2 influenced the location and number of these subpolygons. I also attempted to divide the domain in a way that would separate key geographical features such as the Delaware Bay and C&D Canal, while also creating approximately equal subareas.

I then converted the subpolygons to 2D meshes one at a time and saved these submeshes as 2dm (two dimensional mesh) files. Figure 3-6 shows the six major subpolygons, with the boundary arcs in blue.

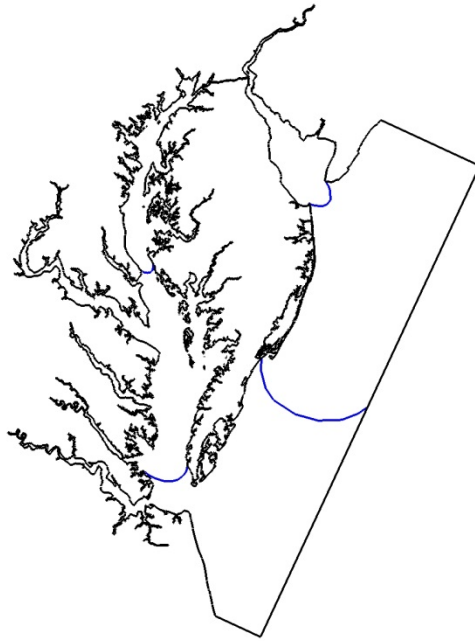


Figure 3-6: The Six Major Subpolygons Used to Generate the Mesh Elements Inside the Zero Meter Contour Indicated by the Blue Feature Arcs

I did not assign elevations to the mesh in the same step as the conversion of the map data to a 2D mesh in an attempt to reduce the time required to generate the mesh. I did not assign

materials to the mesh in this step because the project sponsor did not request that materials be assigned to the mesh in the beginning of the project.

To join these submeshes together, I first loaded one of the 2dm files into SMS and then loaded another in the same instance of SMS. This brought up an SMS 10.1 dialog asking whether it should replace or append the existing mesh. Figure 3-7 shows this dialog in SMS 10.1.

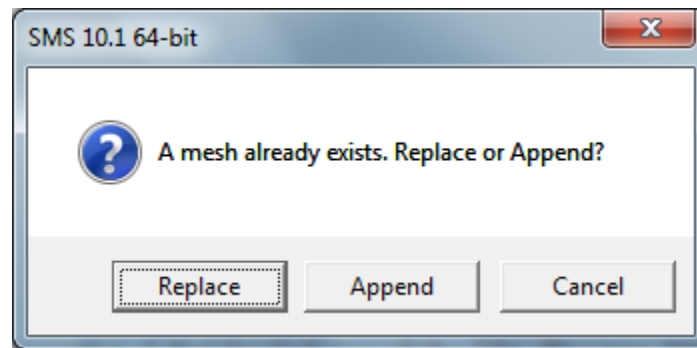


Figure 3-7: SMS 10.1 Dialog Asking Whether to Replace or Append the Existing Mesh

I selected the option to append the meshes together when loading in a mesh file while another was already loaded. SMS then merged the two meshes together. I followed this process for the entire domain, piecing together the 2dm files until I had created mesh elements within all of the zero meter contour arcs.

The final chapter of this report discusses the problems I faced while building the mesh and suggests possible improvements.

3.3 Assigning Elevations to the Mesh Nodes

I assigned elevations to each mesh node from the high resolution elevation data provided by ERDC. I did this in a separate step from generating the mesh elements since multiple

elevation data set files could not be loaded into one instance of SMS. Therefore, the mesh nodes up to this point all had an elevation of zero.

Since I could not load the high resolution data files into SMS 10.1, I had to load them in one at a time in WMS 8.3 and then export them as files that SMS would then be able to load. We found that we were able to load files with an extension of .gdm (a DEM file) that we exported from WMS 8.3 into SMS 10.1. WMS 8.3 did not triangulate the elevation data when I loaded it into the program, whereas SMS 10.1 did. This seemed to be the main reason why I could load these data into WMS 8.3 and not SMS 10.1.

After loading an elevation data file, I exported the data as a .gdm file so I could then load the data into SMS and interpolate the data to the mesh. During this process, I found that SMS could not load some of these .gdm files. For the .gdm DEM files that SMS could not load, I loaded these files back into WMS 8.3 and trimmed them into two approximately equally sized pieces. In some cases, I had to trim them further into three or four pieces before SMS was able to load them. While undergoing this task, I split almost all the tiles into at least two pieces, and many were split into additional pieces as needed.

While testing this procedure to verify that it would properly assign bathymetry to the mesh nodes, I discovered that SMS 10.1 assigned zeros (or a specified extrapolation value) to mesh nodes outside of the boundaries of the DEM from which I was interpolating elevations. I noticed this while checking values interpolated to the mesh and seeing zero elevations in all but the area to which I had just assigned elevations. Furthermore, I also noticed that there were mesh nodes along the boundaries of the tiles that were not being assigned elevations. This was clear when I saw mesh nodes near the borders of the tiles with elevations of zero in areas where the elevation obviously was not zero, such as in the middle of the Chesapeake Bay.

To work around these problems, I created new polygons that overlapped and extended just beyond the edges of the data points in WMS 8.3 for each tile. This ensured that all mesh nodes would be assigned elevation points from the elevation data provided by ERDC. In addition, it allowed me to select the mesh nodes within the boundaries of a tile. I was then able to only affect those selected nodes during the interpolation. I exported these polygons as map files that SMS would also be able to load. I did this for each tile of elevation data. Figure 3-8 shows one of the two pieces of elevation points and the two overlapping polygons for tile forty-one.

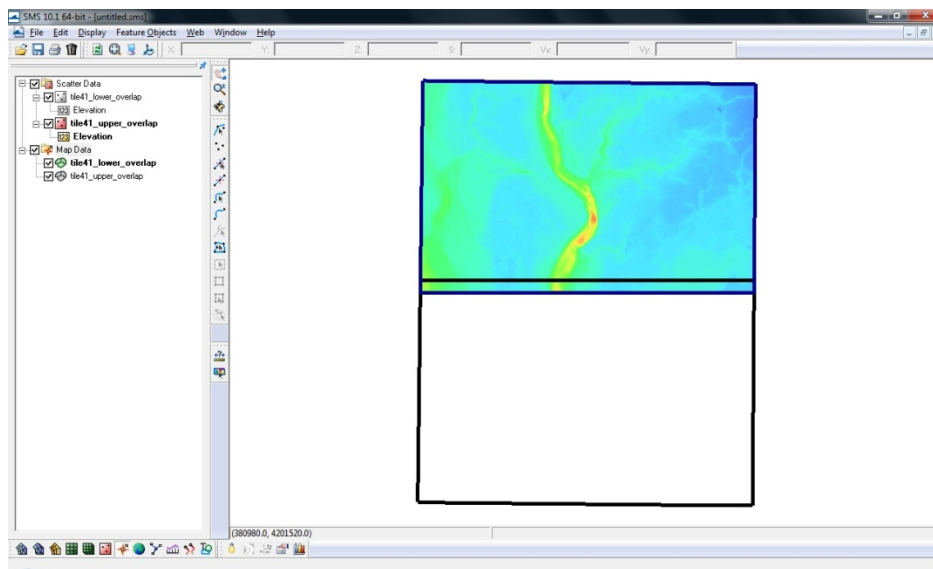


Figure 3-8: The Upper Piece of Elevation Data and the Two Polygons for Tile Forty-One in SMS 10.1

After I created all of the polygons and .gdm files, I loaded them into SMS with the mesh one at a time. I triangulated the elevation points in each of the pieces of the elevation data so that SMS could interpolate their elevations to the mesh nodes. Next, I used SMS 10.1 to automatically select the mesh nodes within the polygon corresponding to the elevation data for that piece. Finally, I interpolated the elevation data from the piece that was loaded to the selected

mesh nodes. I repeated this process for each piece that I created in WMS 8.3 to interpolate the elevations from the elevation data provided by ERDC to the mesh.

3.4 Extending the Mesh to the One Meter Contour

At this point, the finite element mesh included elements within the zero meter contour arcs. The sponsor of this work also desired to apply this mesh for marsh storage and wetting and drying simulations. To accommodate this, I generated mesh elements between the zero and one meter contour arcs and appended these meshes to the previously created mesh. To fulfill this request, I performed the following steps:

- Cleaned the one meter contour arcs
- Connected the one meter contour arcs to the zero meter contour arcs
- Built polygons for the areas between the zero and one meter contour arcs (hereafter referred to as side cars)
- Assigned the mesh attributes to the new polygons
- Converted the map data to side car meshes
- Assigned elevations to the mesh nodes in the side car meshes

I first loaded the one meter contour arcs that I created in WMS 8.3 into SMS and connected them together, just as I did for the zero meter contour arcs. This was necessary because there were small gaps between the one meter contour arcs along the boundaries of the tiles with elevation data. I then cleaned the one meter contour arcs using both the automated clean tools in SMS and manual editing, both of which I described in section 2.1.2.

Next, I connected the one meter contour arcs to the zero meter contour arcs representing the boundary between land and water. This involved going through each tile and creating feature

arcs from the one meter arcs to the zero meter arcs when they were within a distance of a few meters from each other. Figure 3-9 shows a section of the one meter contour arcs that approached the zero meter contour arcs within a few meters. The zero meter contour arcs are the thicker black lines, while the one meter contour arcs are the thinner blue lines.

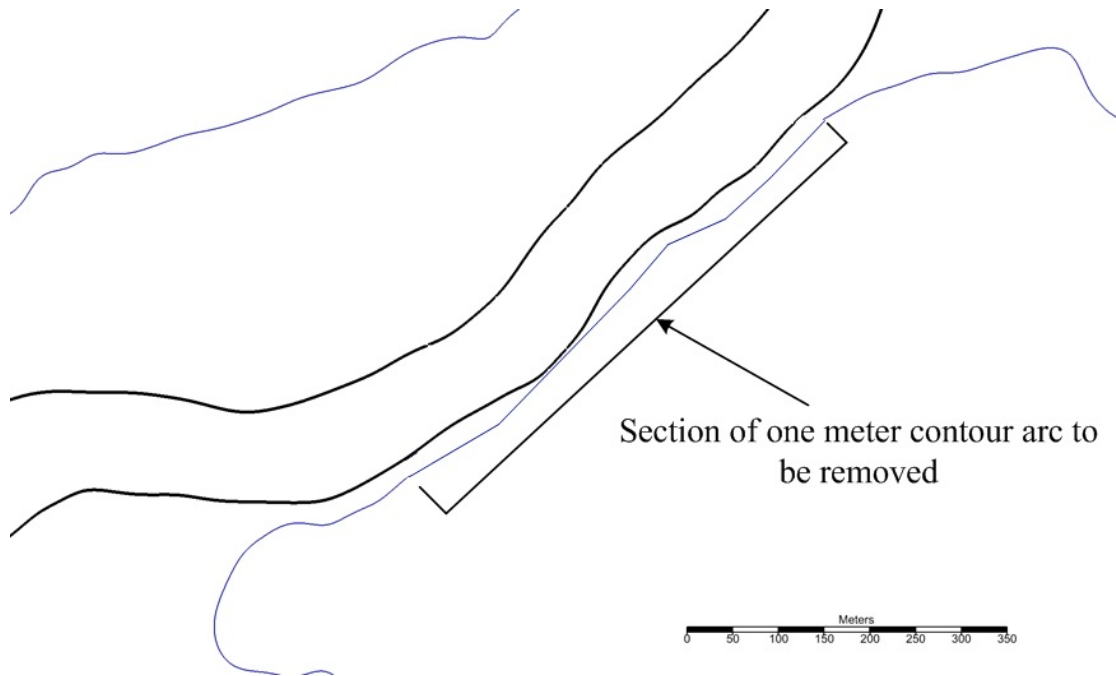


Figure 3-9: A Section of the One Meter Feature Arc to be Removed

Figure 3-10 shows the same area, as well as the surrounding feature arcs, after I connected the one meter contour arcs to the zero meter contour arcs. The black dots are the nodes I inserted to connect the one meter contour arcs with the zero meter contour arcs. Again, the zero meter contour arcs are the thicker black lines, while the one meter contour arcs are the thinner blue lines.

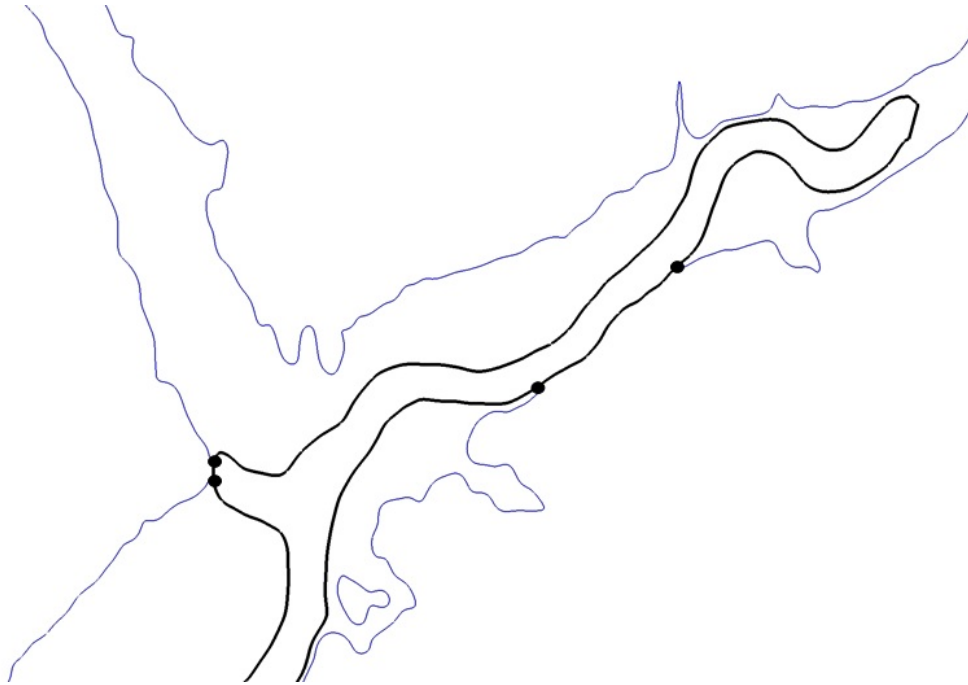


Figure 3-10: One Meter Contour Arcs Connected to Zero Meter Contour Arcs

After connecting all of the one meter contour arcs to the existing zero meter contour arcs, I then built polygons for the areas between those arcs in SMS. We referred to these polygons formed by the zero and one meter contour arcs as side cars in the context of this project. I had to build polygons in SMS 11.0 Beta since SMS 10.1 was not able to build the polygons. I then specified the appropriate polygon attributes to the side car polygons so mesh elements would be created using the scalar paving density method. I followed the same procedure discussed in section 3.2 to perform these tasks.

Finally, I converted the map data that I created between the zero and one meter contour arcs to mesh elements. Because of software and hardware limitations, I was unable to generate mesh elements for all of the side car areas at once. As such, I converted one, or sometimes several of these side car polygons to mesh elements at a time, following the same steps described in section 3.2. After I generated one group of side car meshes, I then saved that data as a 2D

Mesh (.2dm) file. Once I created mesh elements for all the side car polygons, I then appended these files to the existing mesh one at a time, also using the same procedure described in section 3.2. Figure 3-11 shows a close up of the mesh that I created around the area shown in Figure 3-10.



Figure 3-11: A Portion of the Mesh Between the Zero and One Meter Contours

I then assigned elevations to the mesh elements created between the zero and one meter contours. I did this by loading in the scatter data and map polygon files I created to interpolate elevations to the area within the zero meter contour, and then interpolating the data to the new mesh nodes. I followed the same procedure outlined in section 3.3 to perform this task. However, I did not load the files that were only inside the zero meter contour, as I had assigned elevations to those nodes previously.

Figure 3-12 shows the final mesh with color fill contours of the elevations.

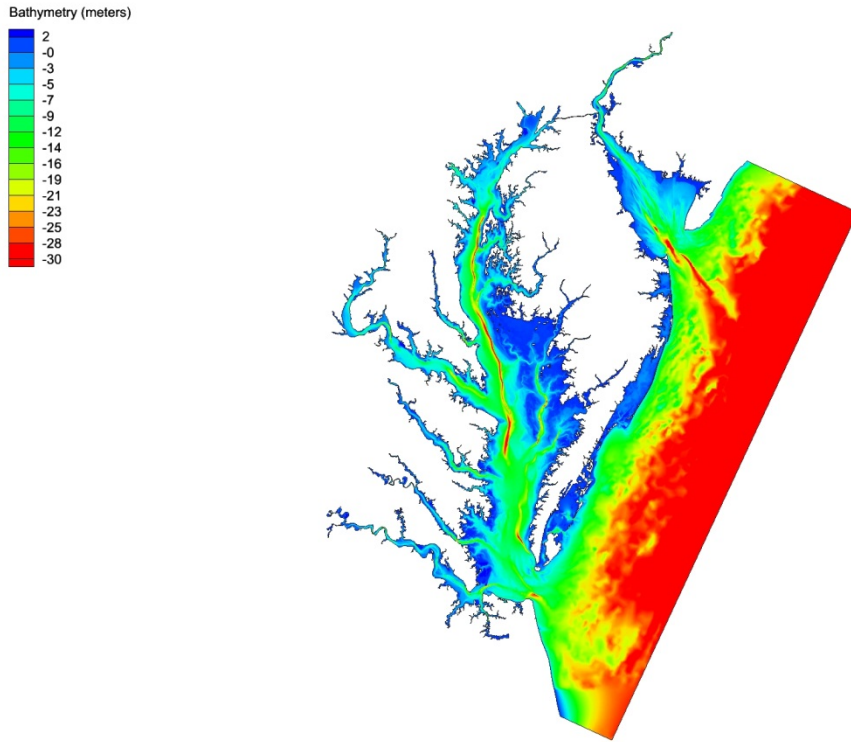


Figure 3-12: Final Mesh with Color Fill Contours of the Elevations

Figure 3-13 shows a close up view of the final mesh with color fill contours of the elevations. The edges of the mesh elements are also displayed to illustrate the varying resolution of the mesh. Additional images of the final mesh can be found in the appendix.

Table 3-2 shows the number of elements and nodes in the final mesh, as well as the minimum and maximum elevations of the mesh.

Table 3-2: Elements and Nodes in and Elevation Range of the Final Mesh

Elements	Nodes	Minimum Elevation (m)	Maximum Elevation (m)
1,273,389	686,094	-59.2	21.1

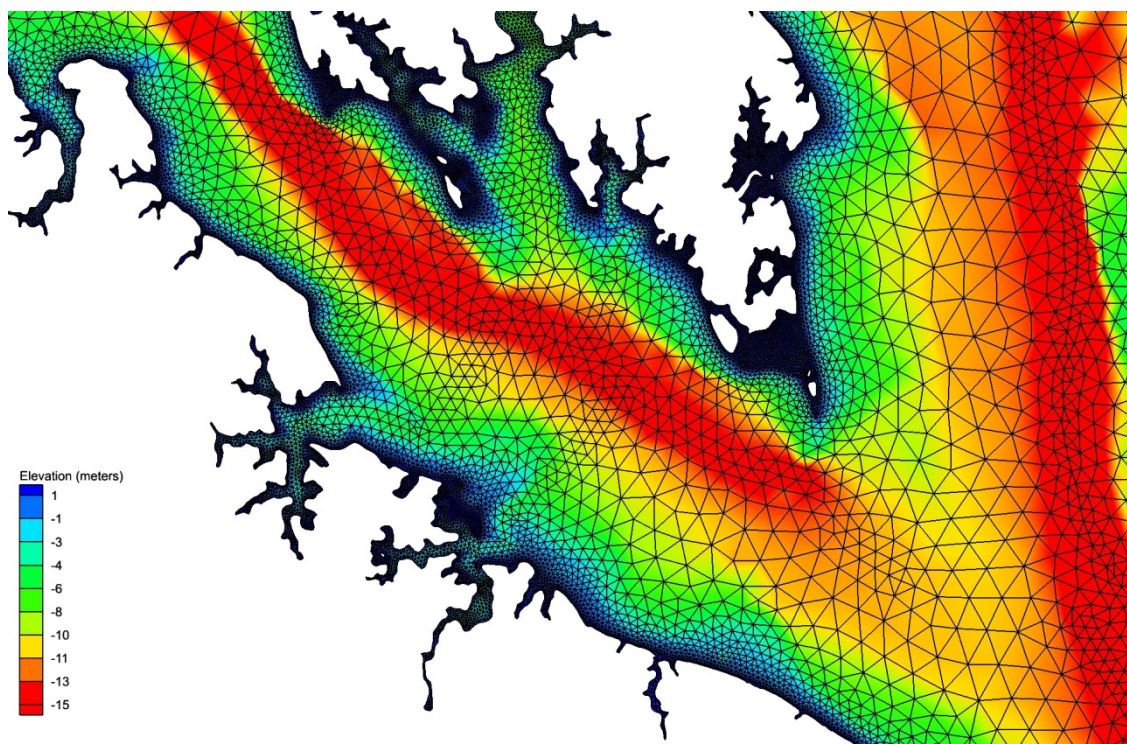


Figure 3-13: Close Up View of the Final Mesh where the Potomac River Meets the Chesapeake Bay with Color Fill Contours Representing the Elevations

3.5 Assigning Material Types to the Mesh Elements

After submitting an initial version of the finite element mesh to the project sponsor, they requested that materials be assigned to the mesh elements. Assigning materials to elements is helpful in defining areas of the mesh that have common properties. These properties can be assigned to the materials rather than to each element. Once each element is assigned a material, parameters such as a Manning's roughness coefficient can be assigned to those materials. These parameters can then be used in finite element numerical models to calculate flow fields.

The sponsor requested that one material be assigned to the C&D Canal, and that the rest of the elements be assigned materials based on their elevations. This resulted in seven different materials to be assigned to the mesh elements. The elevation of an element is determined in SMS

by a simple average of the nodes that are connected to the element. A negative elevation indicated that the element was below mean sea level, whereas a positive elevation indicated that the element was above sea level.

To assign materials to mesh elements, I selected the elements using a tool in SMS that automatically selects elements between specified minimum and maximum value. Figure 3-14 shows this tool which I used in SMS to select mesh elements based on their elevations.

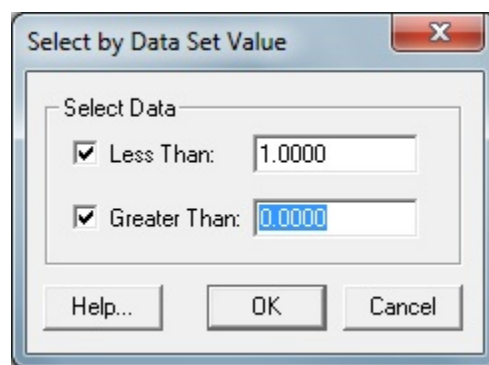


Figure 3-14: Select by Data Set Value Tool in SMS

Once I selected the elements within a certain range using the Select by Data Set Value tool, I then assigned those elements their corresponding material in SMS. Figure 3-15 shows the Materials Data window of SMS which I used to assign material types to elements with the materials the sponsor defined.

We discretized the materials based on depth to reflect changes in roughness. For example, elements assigned the deepest material (Material 7) will have a significantly higher roughness than those assigned shallower materials because of interactions with the ground. The materials between the deepest and shallowest materials will vary in the amount that those interactions occur.

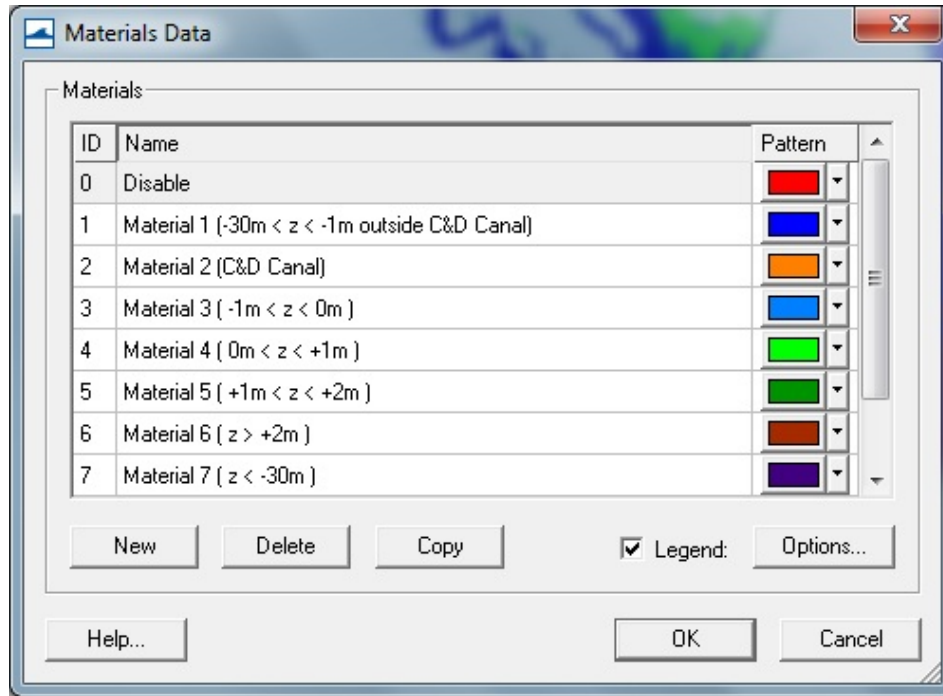


Figure 3-15: Materials Data Window in SMS Used to Assign Materials to Mesh Elements

We assigned the C&D Canal a unique material because it is a manmade channel and will thus have a unique roughness. Table 3-3 shows the elevation ranges for each of the seven materials we assigned to mesh elements, along with the number of elements assigned each material in the final mesh.

Table 3-3: Elevation Ranges and Number of Elements of the Different Materials in the Final Mesh

Material	Elevation Range (meters from sea level)	Number of Elements
1	-1 to -30 (outside the C&D Canal)	426,771
2	C&D Canal	7,088
3	-1 to 0	401,772
4	0 to +1	734,866
5	+1 to +2	111,192
6	Greater than +2	14,423
7	Less than -30	2,783

I assigned materials to the C&D Canal last since they were the only elements whose material was not based on elevation, but rather location. I manually selected these elements and then assigned them the C&D Canal material, or Material 2.

Table 3-3 indicates that there were a significant amount of elements with elevations above one meter above sea level. This was unexpected since we created the domain boundaries only using the zero and one meter contours. After identifying these elements and investigating further, it appears that the main cause for these elements being created is human errors. During the process of manually cleaning and connecting the one meter contour arcs (see sections 2.1.2 and 3.4, respectively), I connected the contour arcs using my best judgment, but this often resulted in creating the domain boundaries outside of the one meter contour. I also failed to assign several island polygons the mesh type of “None.” However, these extraneous elements only accounted for 14,423 of the 759,160 (about 1.9%) of the total elements in the mesh. Figure 3-16 shows the final mesh with materials displayed. Figure 3-17 shows a close up view of the final mesh near tile forty-one, again with the materials assigned to the mesh elements displayed.

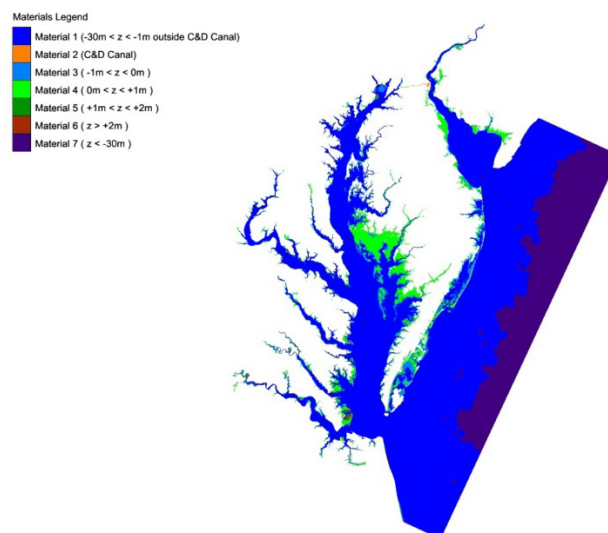


Figure 3-16: The Final Mesh with the Different Materials Displayed

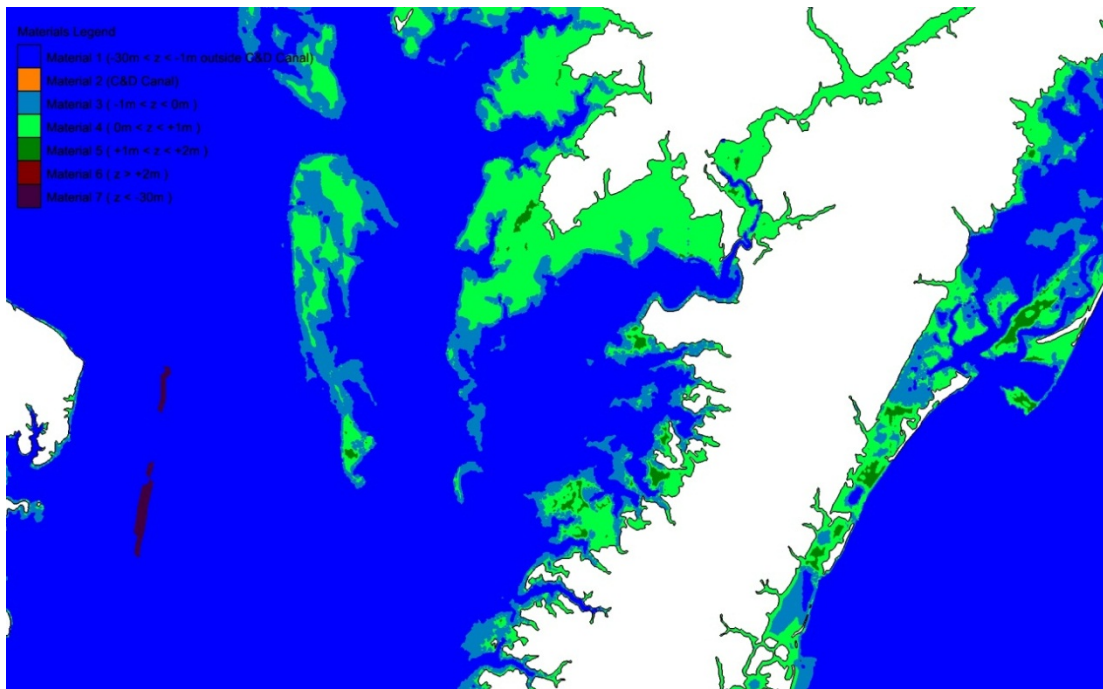


Figure 3-17: Close Up View of the Final Mesh Near Tile Forty-One with Materials Displayed

4 Software Deficiencies and Suggestions for Future Improvement

Several obstacles arose during the course of this project. Many of these obstacles required modifications to the steps we originally planned, while others required entirely new steps. These included bugs and limitations in the software, as well as many repetitive and time intensive processes. While performing the various tasks associated with this project, I identified several features for the Aquaveo developers to consider implementing in the software that could benefit users doing similar tasks in the future.

4.1 Bugs in the Software

There were numerous bugs that I encountered in SMS 10.1, SMS 11.0 Beta, and WMS 8.3 during the course of this project. I was unable to consistently replicate many of these bugs, which made it difficult to address them. However, I submitted those that I could replicate to the Aquaveo development team for them to address in future releases of the software. Below is a list of the bugs that I did submit, as well as the decision made by the Aquaveo developers for each bug:

- SMS 11.0 Beta crashed while attempting to generate a mesh using the scalar paving density mesh generating technique during a test with the project files. The SMS development team determined that this particular crash was caused by an unreasonable number of nodes to be created. They decided not to fix this problem.

- SMS 10.1 crashed while attempting to redistribute vertices along an arc. The SMS development team found that there was a problematic arc in the files used to replicate the problem. They also determined that SMS 11.0 had a feature that gives the user a warning that tells the user which arc is causing the problem so the user can then fix it. They decided not to put this feature in SMS 10.1.
- SMS 10.1 froze while using the zonal classification tool during an analysis of the elevation data from ERDC. The SMS developers found that the program was still running and just needed more time to complete the task. They commented that another group within Aquaveo had a task to investigate possible speed improvements which might address this problem. They also said that they made a change that should result in marginal time improvement.
- SMS 10.1 crashed while attempting to redistribute vertices along a feature arc. The developers found that the cause was a small arc that was invalid. They implemented a fix that displayed a message to the user to run the clean tool if an invalid arc is found.
- SMS 10.1 reports that the progress is 113% while attempting to convert map data to a 2D mesh. The SMS developers fixed this problem.
- SMS 10.1 crashed while attempting to save a project file. The SMS developers also fixed this problem.
- SMS 10.1 reported errors while reading two scatter sets saying that a duplicate GUID (globally unique identifier) was found and the scatter set did not read correctly. The SMS developers determined that this issue did not require a fix.

- SMS 10.1 reported an error saying there was a problem redistributing polygon boundaries and that the user should try to clean the feature objects while attempting to convert map data to a 2D mesh. The SMS developers decided that the method to prevent this issue was to choose the option to copy the coverage before generating the mesh. They also submitted a feature request to always copy the map coverage.
- SMS 10.1 did not display inactive coverages. The SMS developers found that the cause of the problem was that there were duplicate GUIDs. They implemented a fix to check for duplicate IDs in SMS 10.1.
- SMS 10.1 froze while attempting to build polygons with a particular map file. I found that the problem did not occur in SMS 11.0. After reporting this to the SMS developers, they decided they would not fix the problem in SMS 10.1 since it worked in SMS 11.0 and as far as they could tell it only occurred in extreme cases.
- SMS 10.1 reported an error stating that branching coastline arcs exist while attempting to run the define domain tool. They concluded that this was not a bug and that issue was with a branching arc in the file. They submitted a feature request to improve the feedback to users to help them isolate the problem.
- WMS 8.3 crashed when selecting the display options macro. While trying to replicate the problem on another computer, the display options window appeared, but was blank. The WMS development team could not replicate the problem. They asked me to show them the problem, but when I attempted to replicate it, the

crash did not occur. They determined that the problem must have been that I was using too much memory.

- WMS 8.3 froze with a message about sorting contours in the status bar while attempting to convert DEM contours to feature arcs. I was able to circumvent this problem by smoothing the DEM and then converting the contours to feature arcs.
- WMS 8.3 crashed while attempting to convert DEM contours to feature arcs. The WMS developers resolved this issue.

The fixes and improvements that were implemented for these bugs should allow users attempting to generate large meshes in SMS to accomplish their task with less problems. The development teams at Aquaveo have also implemented many other improvements that were apparent while working with SMS 11.0 Beta that will be beneficial for users performing similar tasks to those described in previous chapters.

4.2 Software Limitations

While there have been many fixes and improvements made to the WMS and SMS software, many of the limitations discussed in previous chapters still exist. This section discusses several of these limitations.

As mentioned previously, a major issue that caused significant delay in the project was that the tiles with elevation data points could not be loaded into SMS without removing points. This has been addressed somewhat with the addition of the raster module in SMS 11.0, which can easily load the elevation data files. However, there are limitations to the types of operations that can be performed with raster datasets compared to those that are available for scatter datasets. The work around that I used to load the tiles into SMS was to first load the tiles into

WMS and then export them from WMS as *.gdm files, which SMS could then read as scatter sets.

Another limiting factor in the project was that building polygons did not work in many cases with the large map files. The SMS developers implemented fixes for this problem in SMS 11.0 Beta, but I still had difficulties building polygons with these improvements. The work around for this problem was subdividing the domain into different areas with arcs and then building polygons for those individual subdomains separately.

Along those same lines, because of the large size of the domain and the high target resolution of the mesh, I could not create all the mesh elements for the entire domain all at once. When I attempted to convert all of the map data to a two dimensional mesh at once in SMS the program either froze or crashed. I therefore had to convert pieces of the map data to separate meshes and then append them together.

As mentioned in section 4.1, SMS did not display inactive coverages in many cases while I manually cleaned feature arcs. This proved to make things extremely difficult during the manual cleaning process. For instance, the task of manually cleaning feature arcs representing the one meter contour was much easier when the zero meter contour arcs that I already cleaned were correctly displayed at the same time.

The large size of the datasets provided by ERDC caused many of the processes to take much longer than originally anticipated when planning the tasks to complete the project. For many of these processes that took a significant amount of time, multi-core processors or multiple computers were used to multi task so that additional tasks could be completed while the computer was performing a different process.

4.3 Suggestions for Future Improvement and Software Development

This section contains suggestions for improving the process of generating large meshes, as well as software improvements that would assist in performing similar projects in the future.

4.3.1 Process Inefficiencies

A major impediment to the progress of the project was poor communication with the project sponsor. Improved communication could have avoided repeating steps by making a decision on what would be required with the client before beginning the project. A few examples of this are creating the main mesh twice at two resolutions, interpolating elevations to side cars after already doing it for the main mesh, and generating contour arcs for the two meter contour even though it was not used in the end. Improving communication with the sponsor before the work began would have limited repeating steps and helped streamline the tasks.

Also, spending additional time analyzing the data before beginning the mesh generation process would have ended up saving significant time. For instance, if I had spent more time reviewing the elevation data, I would have realized that smoothing the elevation data at the beginning of the project would have avoided several problems that we encountered while defining the domain. I also could have tried generating meshes with similar spacing to the ERDC elevation data to estimate the maximum number of mesh elements I could generate with the particular computer I was using.

4.3.2 Future Software Development

This section suggests features for and enhancements to Aquaveo software that would better facilitate similar tasks to those discussed in previous chapters.

Both WMS and SMS update or rebuild the display more often than seems necessary. For example, when switching tools or dragging SMS to a different location on the screen the display refreshes, which doesn't seem necessary. Limiting the instances that SMS updates or rebuilds the display would save significant time over the course of a similar project involving a large amount of data. Another potential solution to this problem is to allow the user to specify the frequency or specific actions that trigger the display to be rebuilt.

A feature that would have saved significant time would be to include an option in the window that appears when interpolating scatter data to a mesh to only interpolate data to nodes within the bounds of the scatter set. Section 3.3 discusses this problem in the context of this project and the additional steps it made necessary to interpolate elevation data to the mesh.

While waiting for long processes to finish such as building polygons or converting data to different formats, I sometimes inadvertently pushed the escape key on my keyboard, which cancels the current operation. This caused hours of delays. To prevent this from occurring, the result of pushing the escape key could be modified to reduce the likelihood of a user accidentally cancelling an operation. Another way to remedy this problem would be to include a dialog that requires the user to confirm that they want to cancel an operation.

Many of the tasks that we performed as part of this project were repetitive and involved extensive idle time while waiting for the process to complete. To address this issue, the Aquaveo development team could develop a script editor that would allow the user to run frequent or repetitive tasks automatically with different sets of data or objects.

A feature enhancement that would benefit projects with long tasks would be to improve the messages in the status bar of WMS and SMS to more accurately predict how long a task will take. For tasks that take longer than about a minute, the percentages reported in the status bar do

not seem to be accurate. The percentages in the status bar typically stop updating at a certain point depending on the operation. Towards the end of the operation, the “Updating display...” message appears before the operation suddenly finishes. By improving the accuracy of the messages reported in the status bar, users performing time intensive tasks would be able to better budget their time.

Finally, a popular feature request that many users of Aquaveo software have requested is an undo button. This also would have been a useful tool in this project in many cases, especially during the manual editing process where mistakes due to human error were common. Instead, I saved often and reverted to a previous save when I made a mistake, which slowed down the process significantly. The Aquaveo developers have considered this feature, but it has proved to be unfeasible and problematic up to this point. However, they do have long term plans to implement an undo button in the future.

REFERENCES

- Aquaveo, LLC. *SMS:2D Mesh Polygon Properties*. September 16, 2009.
http://www.xmswiki.com/xms/SMS:2D_Mesh_Polygon_Properties (accessed March 15, 2012).
- . *SMS:Build Polygons (Feature Objects Menu)*. February 7, 2012.
[http://www.xmswiki.com/xms/SMS:Build_Polygons_\(Feature_Objects_Menu\)](http://www.xmswiki.com/xms/SMS:Build_Polygons_(Feature_Objects_Menu)).
- . *SMS:Clean (Feature Objects Menu)*. August 8, 2007.
[http://www.xmswiki.com/xms/SMS:Clean_\(Feature_Objects_Menu\)](http://www.xmswiki.com/xms/SMS:Clean_(Feature_Objects_Menu)).
- . *SMS:Mesh Generation*. September 16, 2009.
http://www.xmswiki.com/xms/SMS:Mesh_Generation.
- . *SMS:Patches*. February 5, 2008. <http://www.xmswiki.com/xms/SMS:Patches> (accessed March 9, 2012).
- . *SMS:Smooth Data Set (Data Menu)*. July 13, 2010.
[http://www.xmswiki.com/xms/SMS:Smooth_Data_Set_\(Data_Menu\)](http://www.xmswiki.com/xms/SMS:Smooth_Data_Set_(Data_Menu)).
- . *SMS:SMS*. April 19, 2012. <http://www.xmswiki.com/xms/SMS:SMS>.
- . *The Surface-water Modeling Solution | Aquaveo.com*. 2012. <http://www.aquaveo.com/sms>.
- . *Watershed Modeling System | Aquaveo.com*. 2012. <http://www.aquaveo.com/wms>.
- . *WMS:Smoothing DEMs*. September 27, 2008.
http://www.xmswiki.com/xms/WMS:Smoothing_DEMs.
- Chesapeake Bay Program. *A Watershed Partnership - Chesapeake Bay Program*. 2012.
<http://www.chesapeakebay.net/> (accessed February 23, 2012).
- Coastal Hydraulics Laboratory, ERDC, USACE. *Adaptive Hydraulics Model*. January 4, 2012.
https://adh.usace.army.mil/new_webpage/main/main_page.htm (accessed February 23, 2012).
- . *ADH: Main Page*. 2007. <https://adh.usace.army.mil/> (accessed February 23, 2012).
- Howlett, J.D. *Size Function Based Mesh Relaxation*. Thesis, Provo: Brigham Young University, 2005.

- Kozel, S.M. *Chesapeake and Delaware Canal (C & D Canal)*. May 30, 2010.
http://www.pennways.com/CD_Canal.html (accessed 2011).
- Martz, L.W., Ph.D., P.Geo. *TOPAZ -digital terrain analysis for watershed analysis and hydrologic modeling*. n.d. <http://homepage.usask.ca/~lwm885/topaz/index.html>.
- THANKYOUDELAWAREBAY.ORG. *Welcome to Thank You Delaware Bay*. 2008.
<http://www.tydb.org/> (accessed 2011).
- The Center for Land Use Interpretation. *Spring 1998 Lay of the Land Newsletter - Chesapeake Bay Hydraulic Model*. 1998. <http://www.clui.org/newsletter/spring-1998/chesapeake-bay-hydraulic-model> (accessed 2011).
- United States Department of Agriculture - Agricultural Research Service. *Great Plains Agroclimate and Natural Resources Research Unit : TOPAZ: Digital Landscape Parameterization*. November 9, 2012.
<http://www.ars.usda.gov/Main/docs.htm?docid=21167>.
- Weaver-Missick, T. "TOPAZ Is a Topographic Gem." *Agricultural Research Magazine*, 1999: 9.

Appendix A. Additional Images of the Final Mesh

This appendix contains more detailed imagery of the final mesh that I generated in SMS

10.1. The images include combinations of the following:

- Oblique views of the mesh
- A light source
- Background imagery
- Various levels of zoom in different locations

Figure A - 1 shows the entire mesh with color fill contours representing the elevations assigned to the mesh nodes. The mesh is also viewed from a rotated angle to better visualize the spatial variation in elevation. In addition, I added a light source in SMS 10.1 to emphasize the depths in the channels.

Figure A - 2 shows the final mesh again with color fill contour and a light source to emphasize the elevations. It also includes a background image to provide a geographical reference for the location and extents of the model domain.

Figure A - 3 displays a close up view of the mesh at the mouth of the Chesapeake Bay. The figure illustrates the transition from high resolution mesh elements along the coast and shallow regions to the low resolution mesh elements in the ocean.

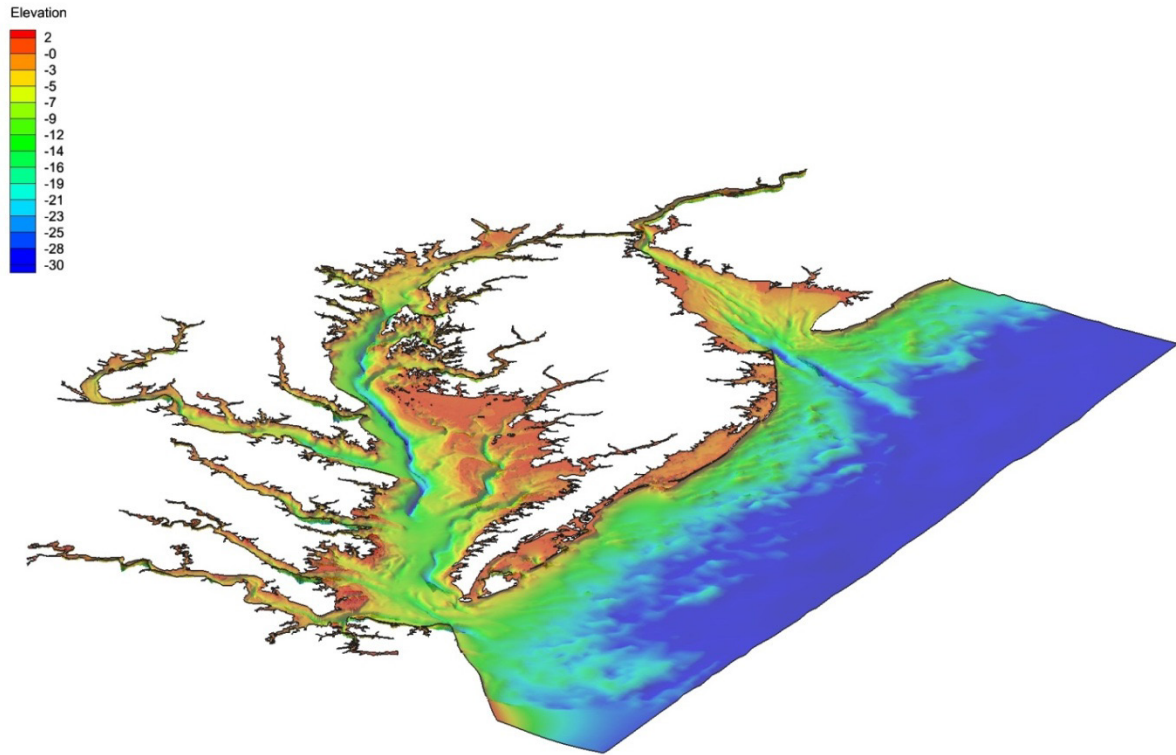


Figure A - 1: Oblique View of the Final Mesh with Color Filled Contours of the Elevations and a Light Source

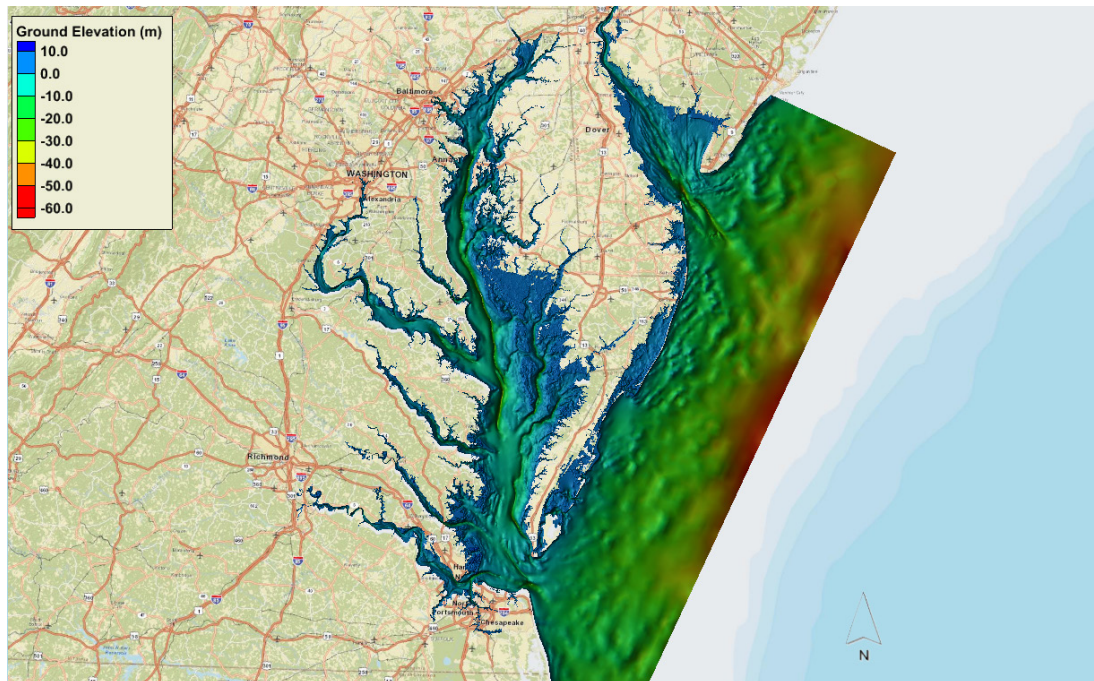


Figure A - 2: Plan View of the Final Mesh with a Light Source and a Background Image

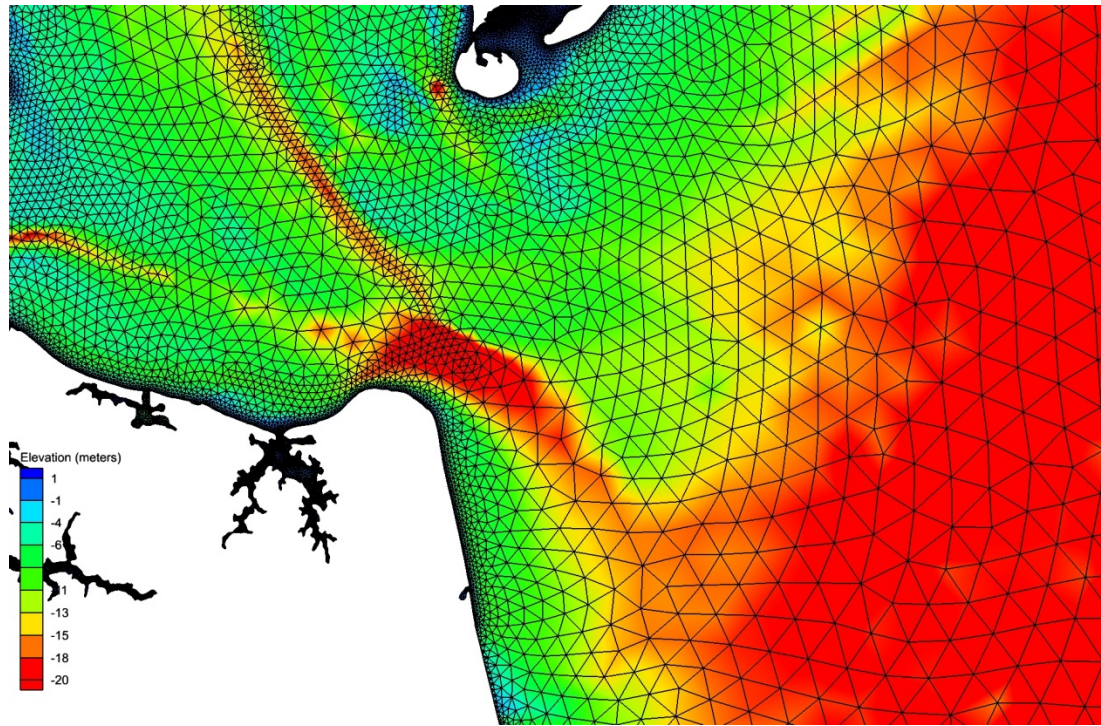


Figure A - 3: Close Up View of the Final Mesh at the Mouth of the Chesapeake Bay

Figure A - 4 shows the mesh in the area where the Chesapeake Bay meets the C&D Canal. The figure illustrates how the elements generated using the scalar paving density mesh generation technique in the Chesapeake Bay connect with elements created using the patch mesh generation technique. This same phenomenon can be seen in Figure A - 5, which shows the mesh on the other end of the C&D Canal where it meets the Delaware Bay.

Figure A - 6 shows the mesh at the north end of the Chesapeake Bay. The portion of the mesh extending out to the upper left is the Susquehanna River.

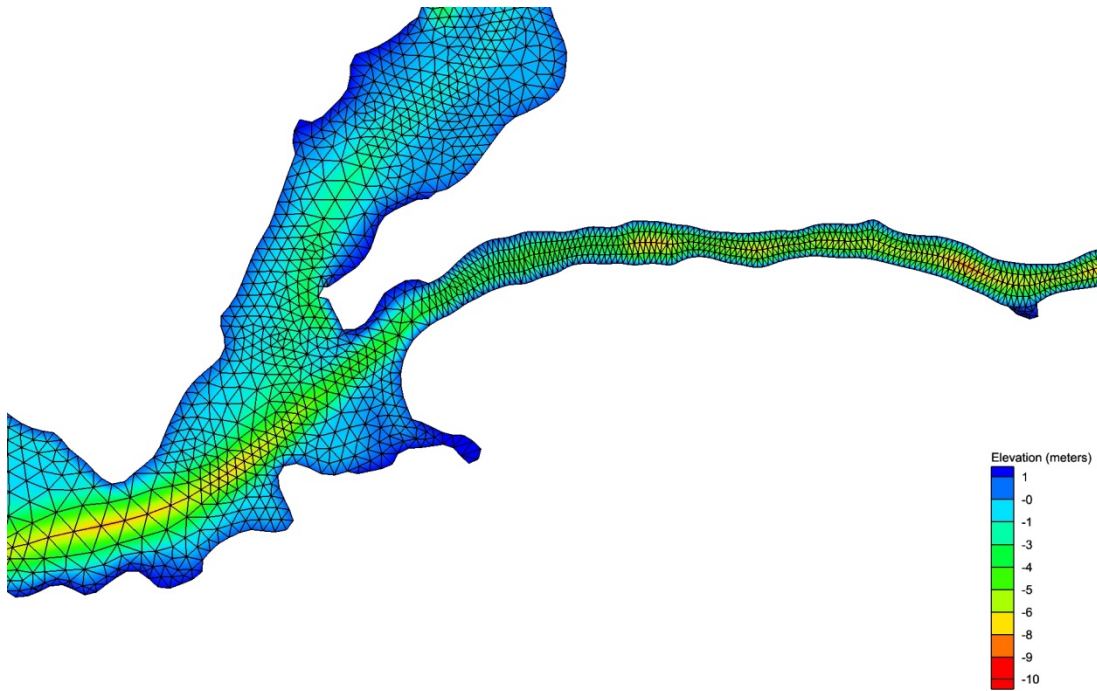


Figure A - 4: Close Up View of the Final Mesh where the Chesapeake Bay Meets the C&D Canal

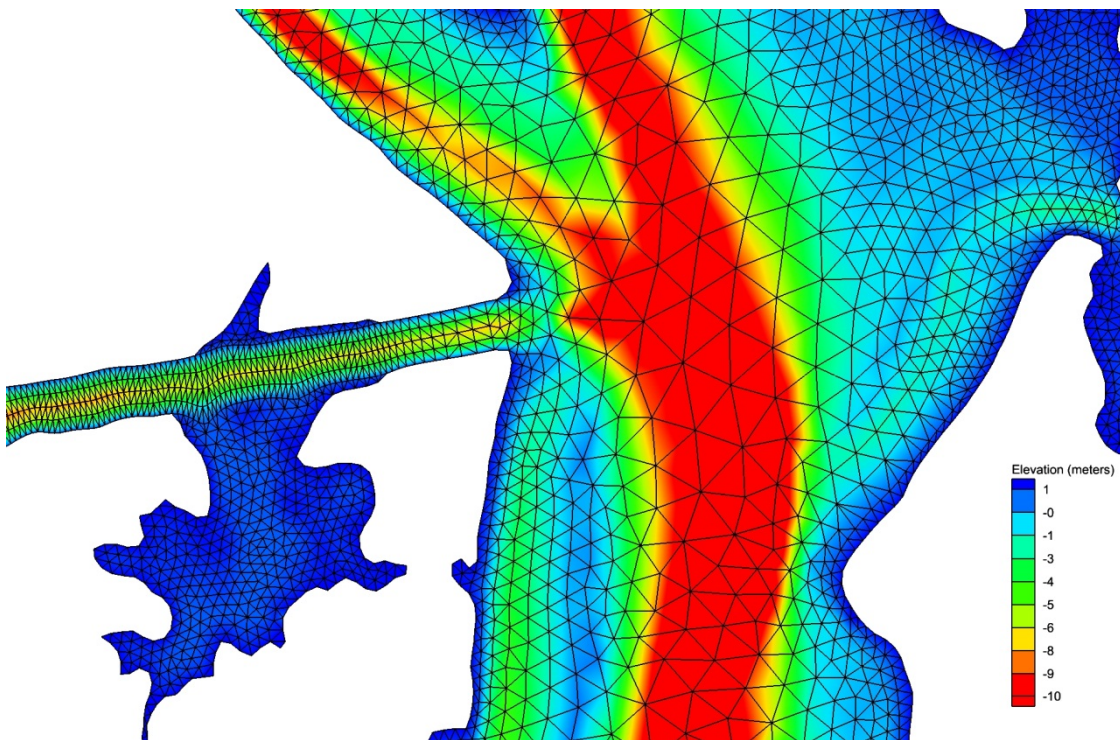


Figure A - 5: Close Up View of the Final Mesh where the Delaware Bay Meets the C&D Canal

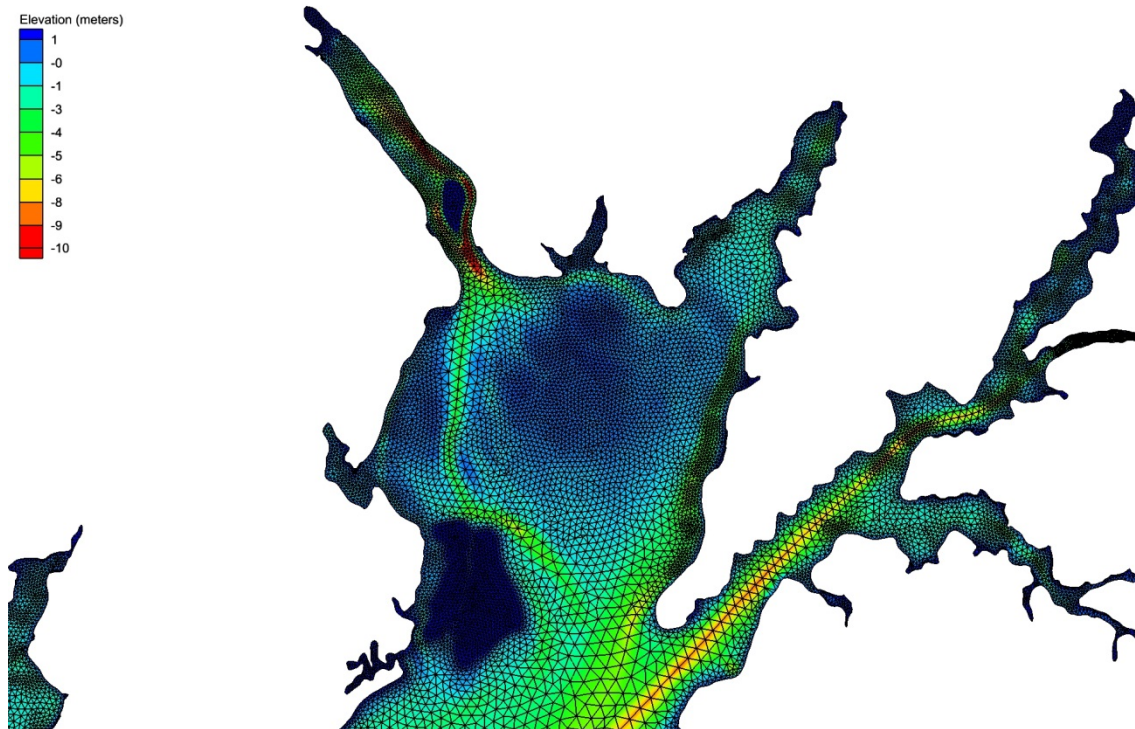


Figure A - 6: Close Up View of the Final Mesh at the North End of the Chesapeake Bay, with the Susquehanna River in the Upper Left