2005-11-07

An Automated Three-Dimensional Unstructured Mesh Generation Algorithm for Groundwater Modeling

James Edward Greer
Brigham Young University - Provo

AN AUTOMATED THREE-DIMENSIONAL UNSTRUCTERED MESH

GENERATION ALGORITHM FOR GROUNDWATER MODELING

by

James E. Greer

A thesis submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Master of Science

Department of Civil and Environmental Engineering

Brigham Young University

December 2005

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a thesis submitted by

James E. Greer

This thesis has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

Date	Norman L. Jones, Chair
Date	Alan K. Zundel
Date	E. James Nelson

BRIGHAM YOUNG UNIVERSITY

As chair of the candidate's graduate committee, I have read the thesis of James E. Greer in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

Date

Norman L. Jones
Chair, Graduate Committee

Accepted for the Department

A. Woodruff Miller
Department Chair

Accepted for the College

Alan R. Parkinson
Dean, Ira A. Fulton College of Engineering and Technology

ABSTRACT

AN AUTOMATED THREE-DIMENSIONAL UNSTRUCTERED MESH
GENERATION ALGORITHM FOR GROUNDWATER MODELING

James E. Greer
Department of Civil and Environmental Engineering
Master of Science

This thesis describes a new method to create three-dimensional finite element meshes using the horizons to mesh algorithm. The algorithm uses available geologic data and user-defined inputs to guide the mesh generation process. This new approach allows for material layer pinch outs and many different layer refinement options to create well-formed elements that better represent hydrogeologic formations. Two case studies are presented that demonstrate the application of the algorithm's options and capabilities. A graphical interface for the algorithm was developed in the Groundwater Modeling System.

ACKNOWLEDGMENTS

I would like to thank my committee, especially Dr. Jones for providing me this opportunity and working with me at the EMRL. I also need to thank Clarissa Hansen for providing me with the South Florida model data. I would lastly like to thank my wife Kaci for all of her support and patience in helping complete my thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 Introduction

Engineers and hydrologists use numerical models to build representations of groundwater systems to make predictions and to develop a greater understanding of hydrogeology. A subset of these numerical models is based on finite element methods which utilize a three-dimensional unstructured mesh (3D mesh). The 3D mesh defines both the overall geometry of the aquifer and the hydraulic attributes within the aquifer by assigning different properties to each element in the mesh. The advantage of an unstructured finite element mesh compared to a structured 3D grid is the ability to more accurately represent the complex subsurface stratigraphy. Developing a suitable 3D mesh that closely represents the hydrogeology of a complex aquifer is challenging because of the limited amount of physical data related to the subsurface geometry and properties. The objective of my research was to develop a 3D meshing algorithm called "horizons to 3D mesh" which uses the most typical hydrogeologic data (boreholes, cross-sections) to automatically create a 3D mesh that can represent complex stratigraphy. The resulting 3D mesh will then be used to run groundwater numerical models.

In my thesis, I first discuss previous research in 3D mesh generation for groundwater modeling and give background information on the horizons modeling approach. I then describe in detail the *horizons to 3D mesh* algorithm. The description is separated into three sections:

1) Algorithm inputs

2) 3D mesh generation process

3) Additional meshing options

To test and demonstrate the applications of the algorithm I will present two case studies. Lastly, I present the conclusions of my research. In the appendix I also include a section describing the graphical user interface for the algorithm that was developed in the Groundwater Modeling System (GMS).

2 Background

While the development of 3D meshes has been researched extensively in many fields of study, the focus of most research has not been in the field of geologic layered systems. One approach to stratigraphic modeling upon which my algorithm is based is known as the horizons method. This section also reviews previous research in 3D meshing and the fundamentals of the horizons modeling methodology.

2.1 Previous Research

Most of the literature and available programs that develop 3D meshes use a form of tetrahedral meshing since prism and hexahedral meshing are relatively difficult in three dimensions (Owen, 2005). One such tool, GEOMESH builds tetrahedral meshes of complex geology (Gable, 1996). However, the stratigraphic layering typically encountered in hydrogeology applications provides unique characteristics that may simplify prism and hexahedral meshing techniques. Prism and hexahedral meshing are the preferred meshing methods for groundwater modeling, because they reduce the amount of elements needed to closely represent the geology. Hexahedral meshing is usually classified as either a direct or indirect method (Quadros, 2002). The indirect method first generates a tetrahedral mesh that is later converted to hexahedral elements, while the direct method creates hexahedral elements directly from the geometry of the object (Owens, 2005). A branch in the direct method of meshing called "sweeping" has predominantly been used to develop 3D meshes for groundwater applications. This sweeping technique utilizes the simplifying characteristics of stratigraphy to develop the elements of the mesh. The sweeping method requires the opposite edges of the geometry to have equal numbers of divisions and each surface must share the same topology. Essentially, mesh elements are filled between two surfaces known as a "target" and "source surface" as shown in Figure 2-1 (Quadros, 2002). Numerous previously developed groundwater meshing algorithms use the sweeping technique to extruded element layers either from a 2D surface mesh (Tucciarelli, 1989) or filled elements between TIN surfaces or solid volumes (Owens, 1996).

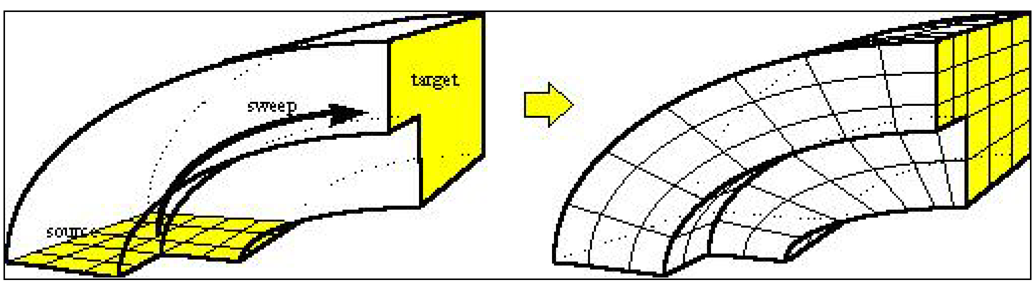

Figure 2-1 Elements created by sweeping(OWENS, 2005)

Methods involving filling between TINs or within solids are often used because of the simple layering often found in groundwater systems. A set of TINs is created to represent this layering and 3D elements are extruded between each layer, or solid volumes are developed of the subsurface and elements are created within each solid. These approaches require that each layer be represented throughout the entire model domain. Furthermore, these methods require complete 2D or 3D representations of the geology to develop the mesh when the the subsurface are most commonly derived from one-dimensional boreholes (Owens, 1996). Some work has been done to develop direct meshing schemes from borehole data, but these schemes are not robust and are limited to well-defined geologic layering that extends throughout the entire model domain (Owens, 1996.).

2.2 Horizons Method

The horizons method was originally developed by Alan Lemon in 2003 to build solid models of aquifer systems from borehole data. The algorithm described in this research extends the "horizons" modeling approach to create a 3D mesh from a set of horizons. A horizon is defined as a surface representing the top of a geologic unit in a depositional sequence. Borehole data are the most commonly available geologic data and are used to define the horizon surfaces. Borehole data are organized into segments and contacts. A contact is defined as the interface between two adjacent stratigraphic units. Segments occur between contacts and are associated with a material (silt, sand, clay, etc.). Each contact has a location (x, y, z) and a horizon ID. Horizon IDs can also be assigned to TIN surfaces which implicitly define the depositional sequence. The horizon IDs should start at 1 and increase from the bottom to the top, where 1 is associated with the top elevation of the bottom-most geologic unit. The horizons on the borehole contacts and TINs are interpolated to a 2D projection TIN which is then extruded to create 3D volumetric (solid) models of the stratigraphy. Figure 2-2 shows both a simple representation of horizons assigned to borehole data and a cross section of the resulting 3D stratigraphy created from the horizons method. The layers are built by extrapolating each horizon surface out to the point where it intersects any previously generated surfaces and then extruding the resulting surface down to form a solid. In addition to borehole data, the horizons method utilizes other user-defined inputs to provide more control over the mesh generation process. User-defined inputs include cross-sections built between boreholes to guide the extraploation process, horizon conceptual models created to constrain the bounds of individual layers, and TINs used to directly represent the layering of the stratigraphic units. These options are discussed in greater detail in the *Horizons to 3D Mesh* section.

2.3 Horizons to 3D Mesh

The objective of my research is to expand the horizons algorithm to use the horizon data defined with boreholes and TINs to create a 3D finite element mesh. The advantages of this method are that the layers of the resulting 3D mesh do not need to extend throughout the entire model domain, and the 3D mesh can be generated directly from borehole data. The 3D mesh is extruded using a sweeping method that minimizes the occurrence of ill-formed elements and accurately models the boundaries of geologic layers.

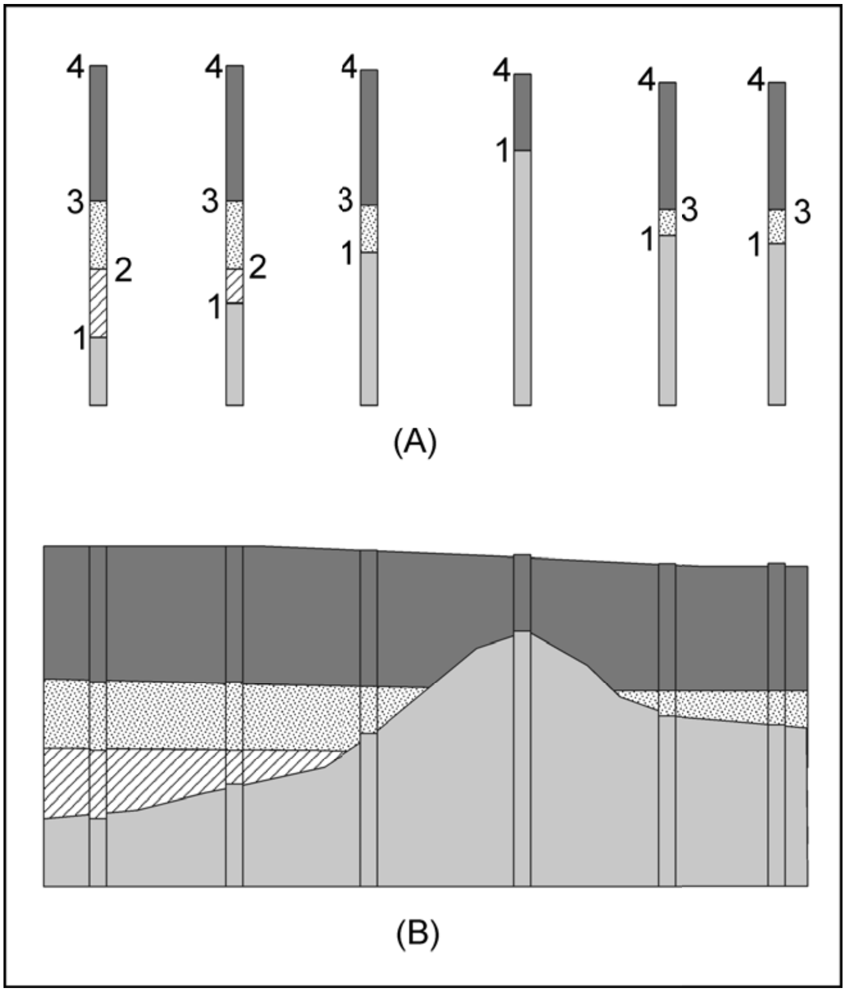

Figure 2-2 Horizons concept. (A) Horizon ID's assigned to borehole contacts. (B) Solids resulting from horizon assignments made in (A) (Lemon, 2003).

3 Horizons to 3D Mesh Algorithm

The Horizons to 3D Mesh algorithm consist of three main sections highlighted in Figure 3-1:

1. Inputs (highlighted in blue)
2. Mesh Generation (highlighted in yellow)
3. Post-Processing options (highlighted in green)

Each of these steps is discussed in more detail below.

3.1 Algorithm inputs

The first step sets up all of the inputs for the horizon algorithm. Many different input options exist to allow flexibility and control over the resulting the 3D Mesh. The inputs to execute the algorithm include:

1. 2D Projection Mesh / Meshing Conceptual Model
2. Horizon Surfaces
3. Top and Bottom Elevations
4. Interpolation Scheme
5. Horizons Conceptual model
6. Borehole Cross-Sections

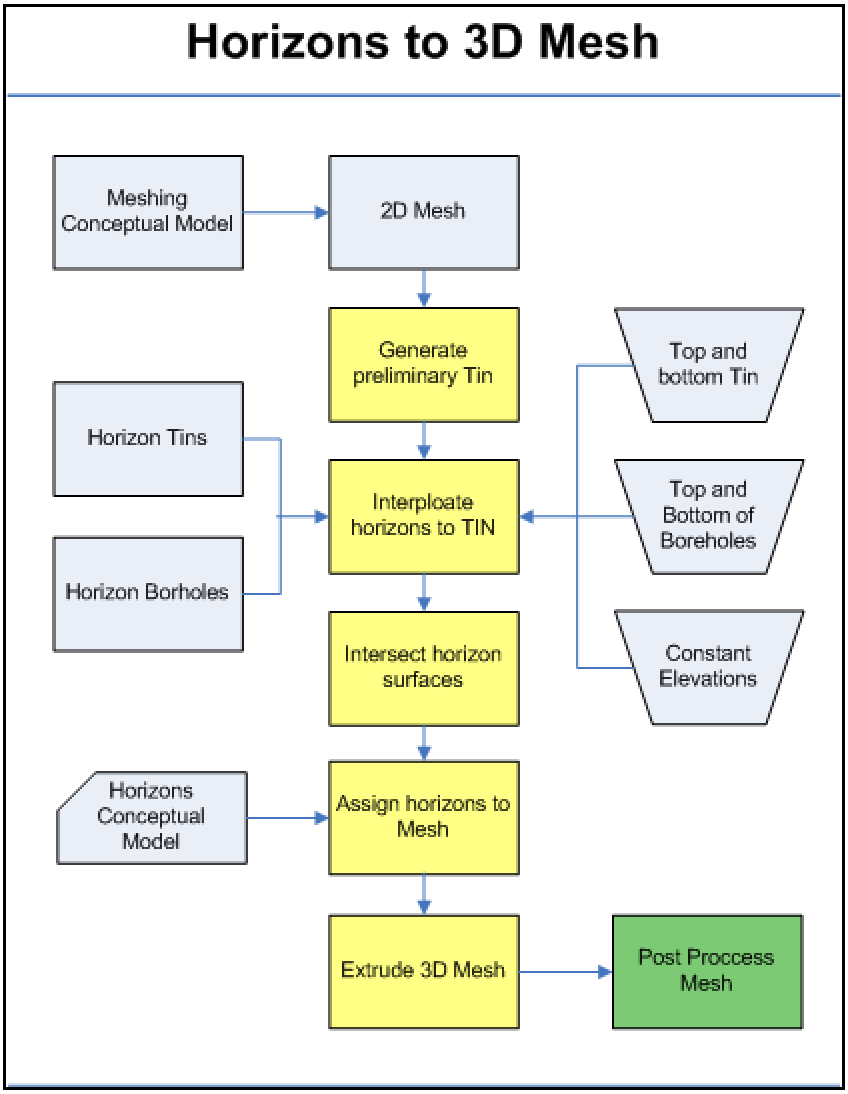

Figure 3-1 Horizons to 3D mesh algorithm

The 2D projection mesh is as a template to create the 3D mesh elements. This 2D projection defines the topology for the extruded layers in the 3D mesh. The 2D mesh must only contain triangular elements. Instead of a 2D mesh a conceptual model can be substituted to define the 2D projection. A conceptual model contains sets of feature objects (points, lines, polygons) and an associated set of meshing options. If a conceptual model is selected as the input then it is first used to create a 2D triangular mesh which then defines the 2D projection. A 2D mesh or conceptual model must be specified to execute the algorithm.

The second required input for the algorithm is the horizon surfaces that are defined by horizon IDs assigned to borehole contacts or TIN surfaces. The algorithm supports the use of both borehole and TIN horizons in conjunction or separately. In addition, a subset of either of the objects can also be selected and used for the operation.

The third required input is the top and bottom elevations of the 3D mesh. TINs, boreholes, or constant elevations can be used to define the top and bottom surfaces of the 3D mesh. TIN surfaces explicitly represent the top and bottom surfaces of the mesh. If the borehole option is selected then the top or bottom of the mesh is created by interpolating a surface from either the top or bottom elevations of the boreholes. Lastly, the algorithm can use specified values (constants) for the top and bottom elevations.

The last required input for the algorithm is a set of interpolation options. These options are used to extrapolate the horizons levels from the boreholes and TINs to the 2D projection defined by the 2D mesh. The implementation use to test this research provides *Inverse Distance Weighted* (Shepard, 1968) and *Natural Neighbor* (Watson and Phillip, 1987) as interpolation options.

One of the main disadvantages with the horizons method is that it is highly dependent upon extrapolation because of the lack of physical data. The last two remaining optional inputs provide better user-control over the interpolation process.

First, a horizon conceptual model provides greater user control over the extents of the horizon surface. A horizons conceptual model contains user-defined polygons that provide horizontal boundaries to the stratigraphic layers developed for each horizon surface (Fugal, 2005). Second, borehole cross section data helps direct the creation of the horizon surfaces and guide the interpolation process. Borehole cross-sections manually sketched between boreholes supplement the borehole data to provide a better interpretation of the geology.

3.2 Algorithm description

The algorithm is similar to the horizons to solids algorithm. However, instead of solids a layered 3D mesh is created. The algorithm shown in Figure 3-1 has five main steps:

1. Generate primary TIN
2. Interpolate horizons to TIN
3. Intersect horizon surfaces
4. Assign horizons to mesh
5. Extrude 3D mesh

3.2.1 Generate Primary TIN

The first step creates the primary TIN. The primary TIN acts as an intermediate object to define all of the horizon surfaces. The TIN is created from the selected 2D mesh or meshing conceptual model. The primary TIN represents the topology for each layer of the 3D mesh. Thus, each horizon layer has the same number of nodes in corresponding XY locations. Using the same topology for each layer makes the horizon process simple and robust.

3.2.2 Interpolate Horizons to TIN

The second step in the algorithm extrapolates the defined horizons from the boreholes and TINs to the primary TIN. This results in a 2D representation of each horizon surface. A dataset on the primary TIN represents each horizon surface. The extrapolation creates a scatter point set for each horizon ID. A scatter point is added to each set for every borehole contact or TIN vertex with the same horizon ID. A scatter set created for a horizon is shown in Figure 3-2. Additionally, if user-defined borehole cross-sections are included, scatter points are created using the elevations defined from the cross-sections as illustrated in Figure 3-3. Each horizon scatter set is interpolated to the primary TIN.

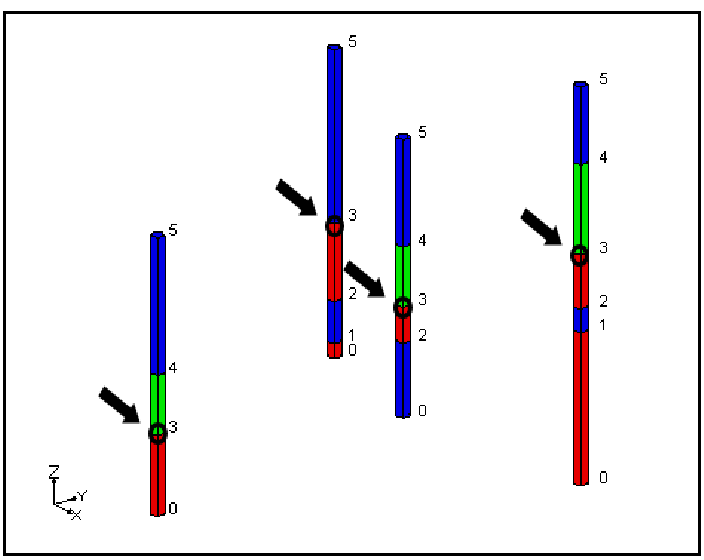

Figure 3-2 Scatter set created for horizon level 3

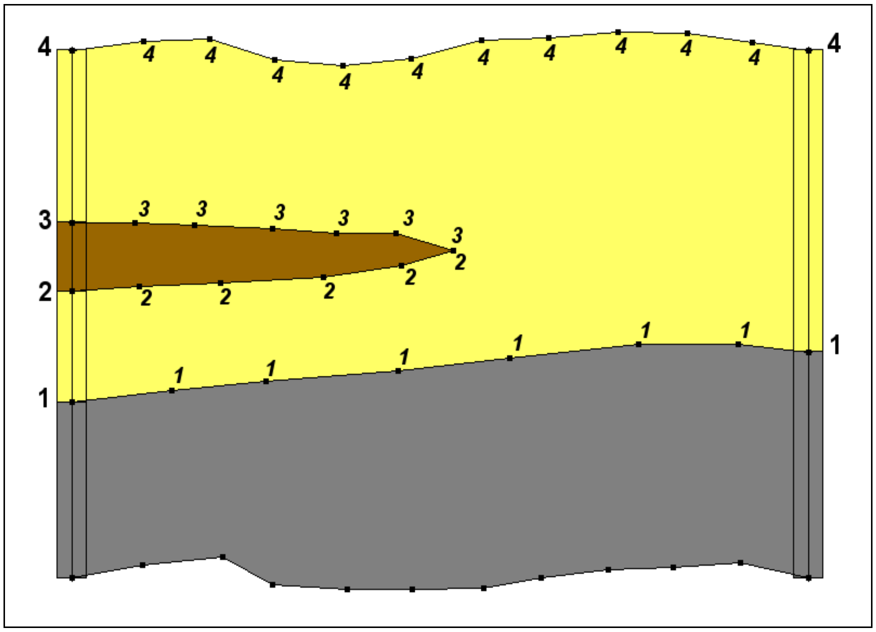

Figure 3-3 Scatter points are created from borehole cross-section data (Lemon, 2003).

3.2.3 Intersect Horizon Surfaces

Thirdly, the horizon surfaces are intersected to determine the boundaries for each layer. The intersection process is computationally fast because each horizon surface has the same topology. Working from the bottom to the top, the algorithm intersects each of the horizon surfaces. Since each layer has the same topology, this process is done by looping through each triangle on the current layer and checking the elevations of the nodes with the elevations of the nodes on the corresponding triangle on the layers above. For each node on the current triangle we then loop through all of the corresponding nodes on the triangles in the layers above and check their elevations. If a node's elevation is greater the elevation of an above layer then the elevation of the node is lowered to match the elevation of the lower overlying layer. This will force the layer that intersects the above layer to pinch out.

At this point, an optional horizons conceptual model is used to guide the interpolation process. A horizons conceptual model is a set of coverages that represent the horizontal extent of each horizon surface (Fugal, 2005). Each horizon coverage has polygons created by a user to give more control over the locations of each of the stratigraphic layers (Figure 3-4). For each horizon layer, if a node on the layer is not located inside of a polygon of the appropriate horizon coverage then the elevation of the node is set to elevation of the corresponding node on the surface below. This creates an intersection and pinches out the layer limiting it to the designated areas. Horizon coverages only limit the extents of the horizon coverages. They do not extend the horizon layers if they are pinched out within the boundaries of the coverage.

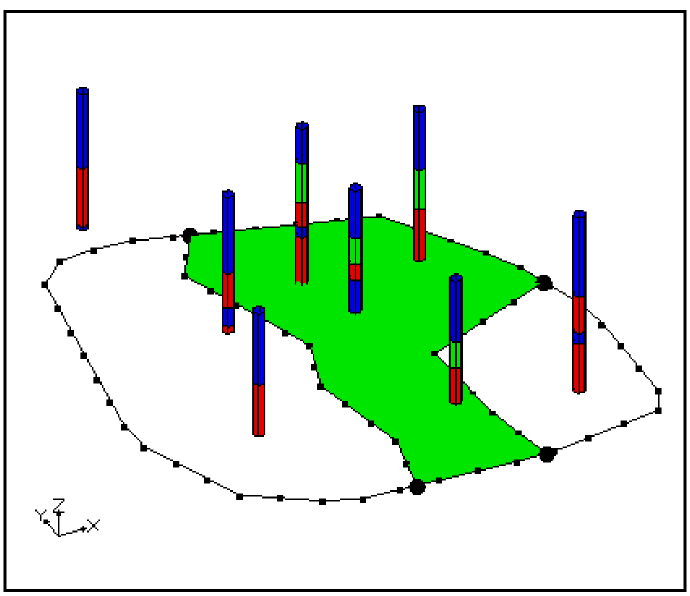

Figure 3-4 Horizon conceptual model polygonal boundary.

3.2.4 Assign Horizons to Mesh

All of the horizon datasets are next reassigned as datasets on the original 2D mesh. No interpolation is necessary because the primary TIN and the 2D mesh share the same topology. To ensure that the horizon elevations are consistent they are again checked and adjusted so that the elevations of each horizon layer are not below any of the lower horizon layers. This is performed by looping through each mesh node from the bottom horizon to the top comparing the elevation of each horizon with the next highest horizon. If the elevation of the horizon above is below the current horizon elevation then the elevation of the higher horizon is set equal to the elevation of the lower elevation. Each node is checked for each horizon (Lemon, 2003).

3.2.5 Extrude 3D mesh

Next, a 3D layered mesh is extruded from the 2D mesh with a layer for each horizon surface. The elevations of each layer of the 3D mesh are set from each of the horizon datasets from the 2D mesh. The node elevations outside of where a layer pinches out are set equal the elevations of the layer below. The elements that are created from these nodes will have zero volume and are marked to be deleted later. This extrusion process is completed layer by layer as illustrated in Figure 3-5.

The elements of the resulting 3D mesh consist of wedges, pyramids, and tetrahedra. Wedge elements are formed from 6 nodes (Figure 3-6), pyramid elements have 5 nodes (Figure 3-7), and tetrahedral elements only have 4 nodes (Figure 3-8). The pyramid and tetrahedral elements are created from degenerate wedge elements to more closely model the layer transitions. The material properties of each element are set depending on the horizon layer that is extruded to create the element.

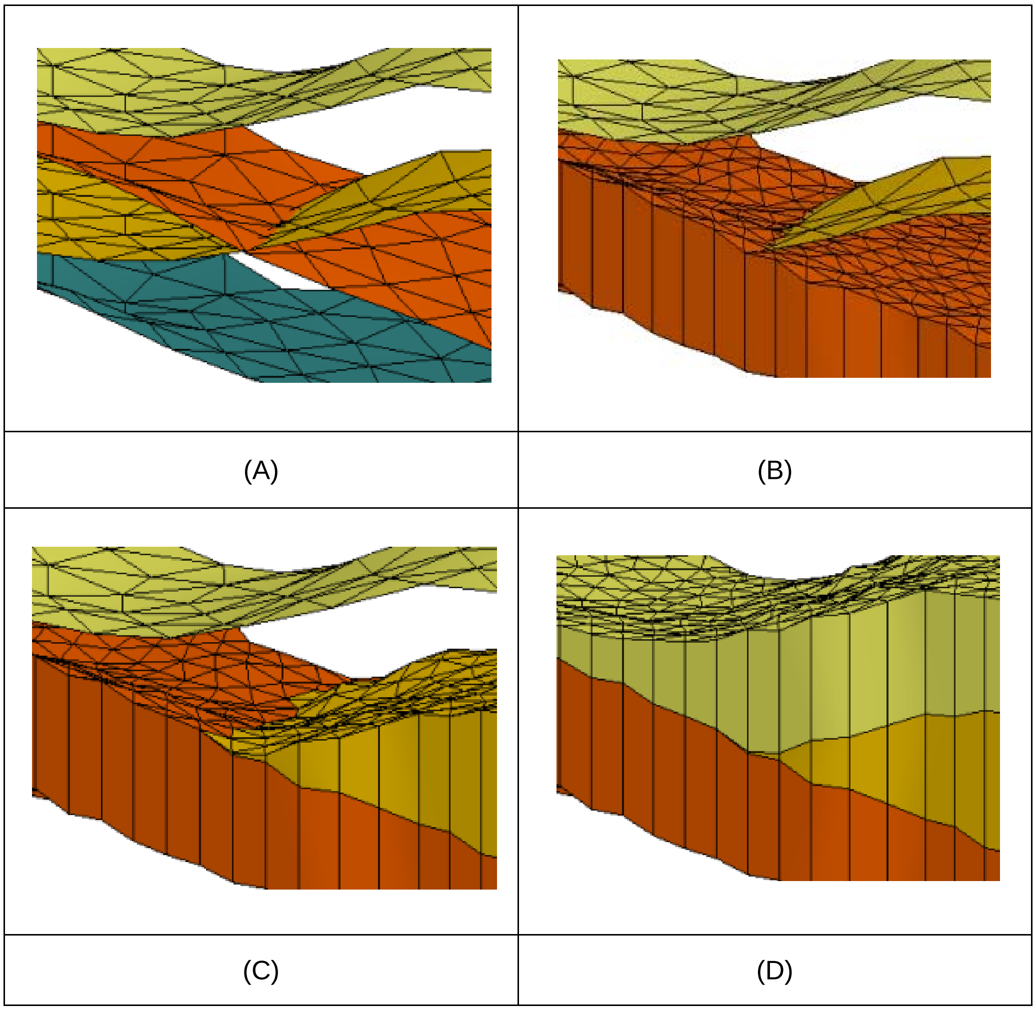

Figure 3-5 The 3D mesh extrusion process. A) A set of horizon layers. B) The bottom layer extruded between the blue and orange TIN. C) Second layer created and pinched out by the bottom layer. D) The complete extruded 3D Mesh

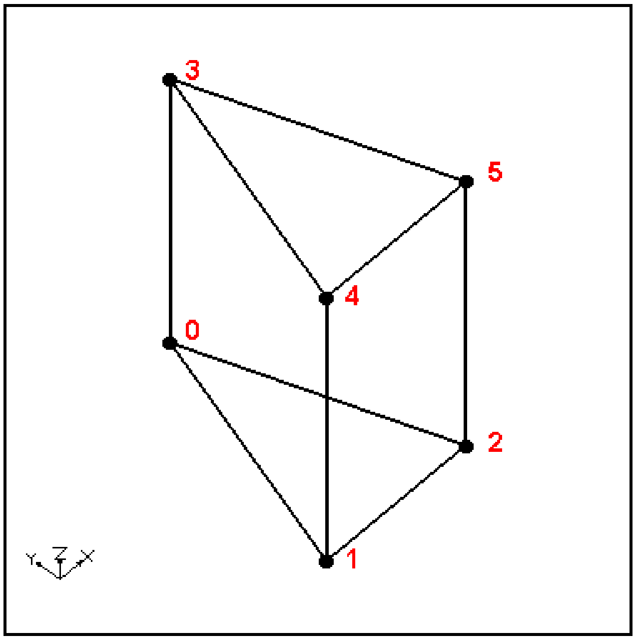

Figure 3-6 - Sample wedge element

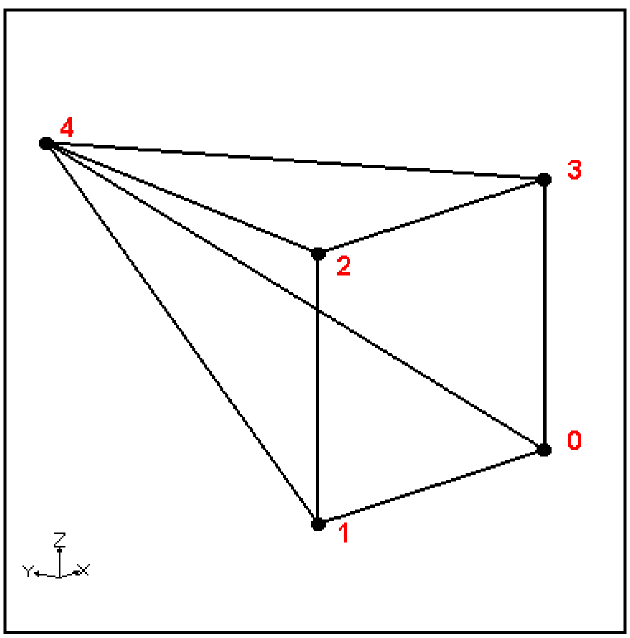

Figure 3-7 - Sample pyramid element

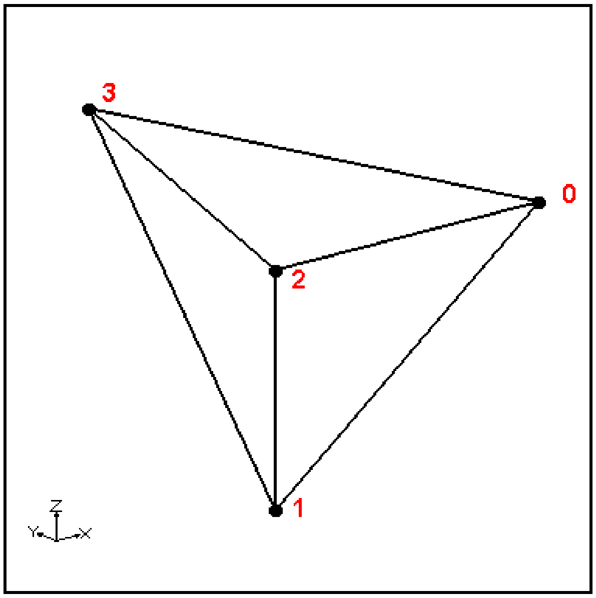

Figure 3-8 – Sample tetrahedral element

The final step resolves any problems that might have occurred during the mesh extrusion process. We start in the top layer of the mesh and once again loop through the nodes on each layer checking for coincident nodes on the layers below. If two nodes from different layers have the same elevation, the node in the lower layer is removed from the mesh. All of the elements attached to the removed node are also deleted and new elements are created linking the two layers together. In some cases the new elements that are created have fewer nodes than the original elements. If any of the deleted elements contained both of the coincident nodes then the new element that is created will have one less node. Thus a degenerate wedge element is converted to a pyramid, a degenerate pyramid is converted to a tetrahedron, and a degenerate tetrahedron is deleted. Along a layer pinch out, nodes are deleted and pyramid and tetrahedral elements are created to closely represent this boundary. Beyond the extents of a layer pinch out, the elements are deleted and the layer is removed from the mesh (Figure 3-9). This check is repeated for each layer in the 3D mesh. This process removes all of the duplicate nodes reforming the mesh elements.

Many finite difference models, however, do not support pyramid elements therefore some additional options for refinement of pyramids elements are included in the algorithm as a post-processing step.

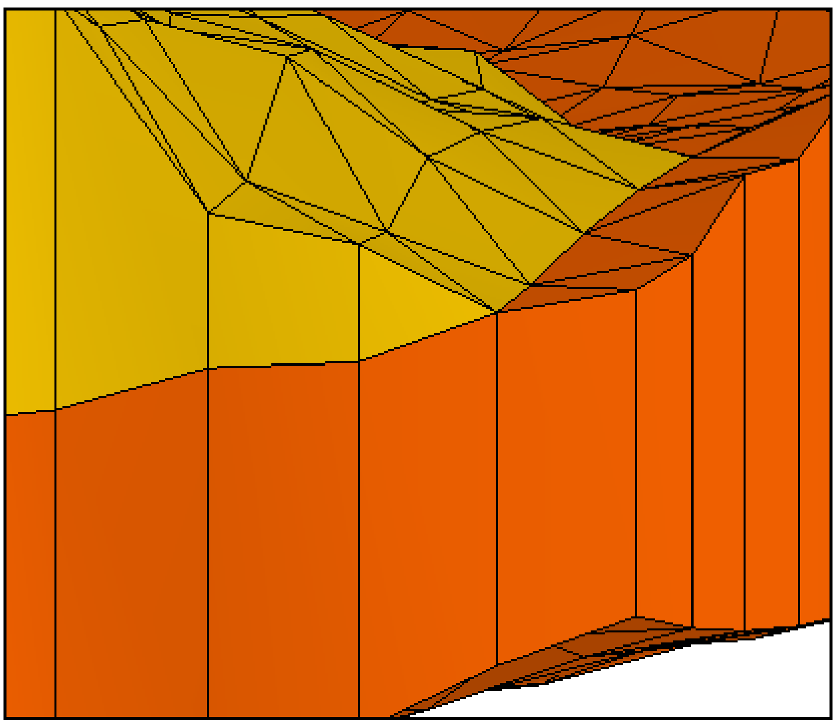

Figure 3-9 Pyramid and tetrahedral elements along a pinched out layer

The 3D mesh generation is now complete. A 3D mesh has been created which represents the horizon layering derived from the borehole and TIN horizon data. A sample application for building meshes is illustrated in Figure 3-10. The borehole data with horizons assigned is shown in Figure 3-10A. Figure 3-10B is a cutaway view of the horizon surfaces interpolated from the horizon elevations. The resulting 3D mesh extruded between the horizon surfaces is shown in Figure 3-10C.

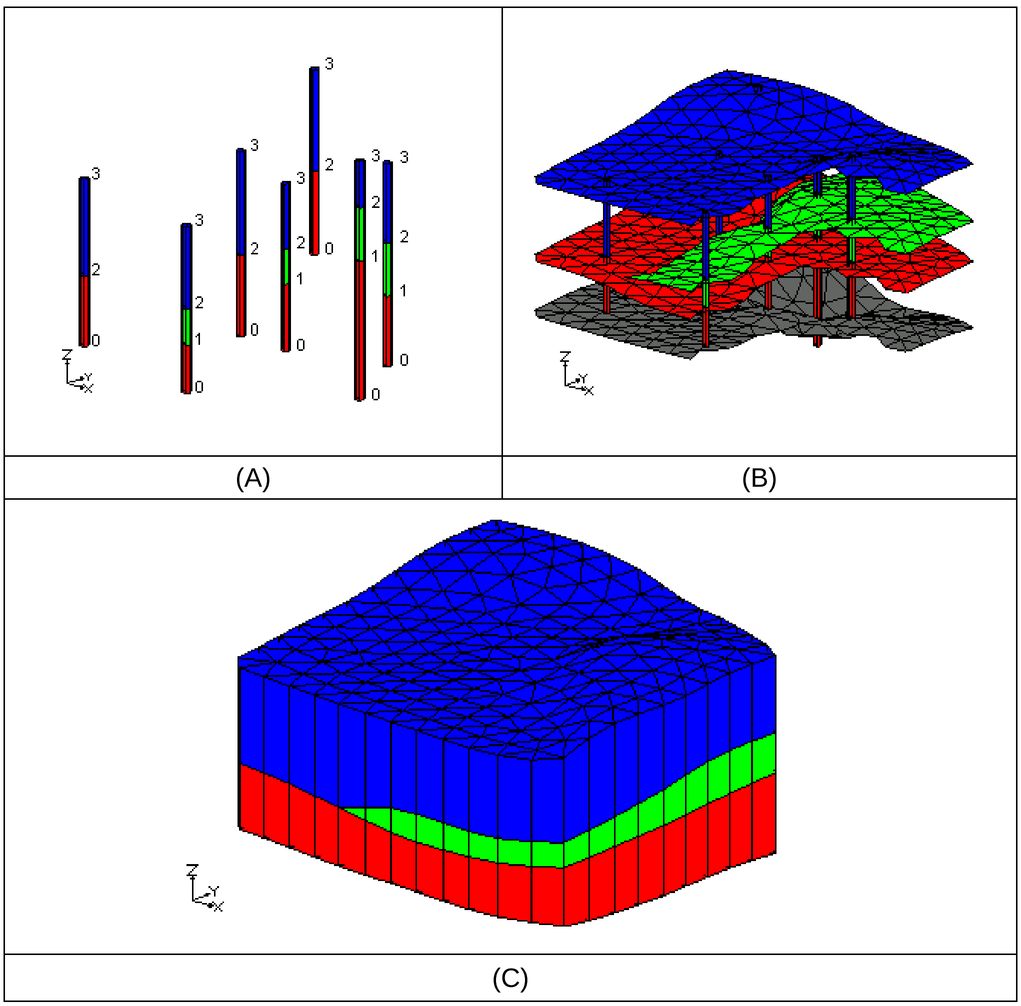

Figure 3-10 Horizons to 3D mesh application. A) Borehole data with horizon IDs assigned. B) Horizon surfaces interpolated from the borehole data. C) 3D mesh extruded from the horizon surfaces.

3.3 Additional Meshing Options

Four additional post-processing options, implemented to allow for more user control of the mesh generation process and to create better-formed 3D elements.

1. Minimum element thickness
2. Average element thickness between adjacent layers
3. Element target thickness
4. Refinement options

Each of these options is discussed in more detail below.

3.3.1 Minimum Element Thickness

A minimum element thickness can be applied to prevent the 3D elements from becoming too thin. Thin elements can result in numerical instabilities when running groundwater models. This option increases the thickness of the elements that are thinner than a specified value. When the elevations are being assigned to the 3D mesh from the horizons, each layer of elements is checked to determine its vertical thickness. The process involves checking the node of the current layer with the corresponding node of the layer above. If the thickness is less than the user-defined value the elevation of the node above is increased. This process is repeated on each node of every layer, except the top layer. If the top layer does not meet the minimum thickness requirement then the nodes below are moved down to prevent changing the specified top elevation.

After attempting all of the elevation adjustments, in some cases there is not enough elevation to include every layer in certain regions of the mesh. In this case the elements that are too thin are marked and later removed from the 3D mesh. Figure 3-11A shows a 3D mesh without a minimum thickness specified. For the case shown in Figure 3-11B a minimum thickness was specified that was too large to include all four of the layers. Thus the blue layer was removed from the 3D mesh.

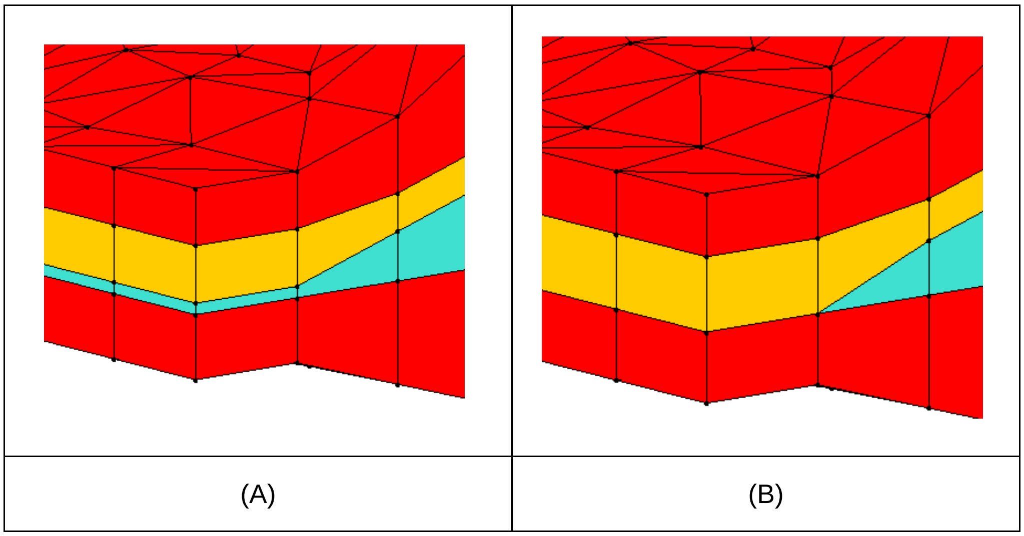

Figure 3-11 Minimum element thickness removal A) A 3D mesh created without specifying a minimum thickness requirement. B) A 3D mesh showing the pinch out of a layer because a minimum thickness was specified.

This illustrates how the minimum elevation thickness option either adjusts element thickness to the required value or removes elements that are too thin and cannot be adjusted. This option prevents the creation of small elements and results in a better 3D mesh.

3.3.2 Average Element Thickness

The average element thickness feature is always applied in the algorithm. During the extrusion of the 3D mesh it is possible to create adjacent layers of the same material. If this occurs, the algorithm averages the element thicknesses between the layers by distributing the total elevation evenly between the layers. An example of borehole data that would result in this scenario is shown in Figure 3-12. The red material is represented as two layers on some boreholes and one layer on other boreholes. The separate red layers are treated as separate stratigraphic units. The elements thicknesses are averaged at locations where these separate red layers are adjacent (Figure 3-13).

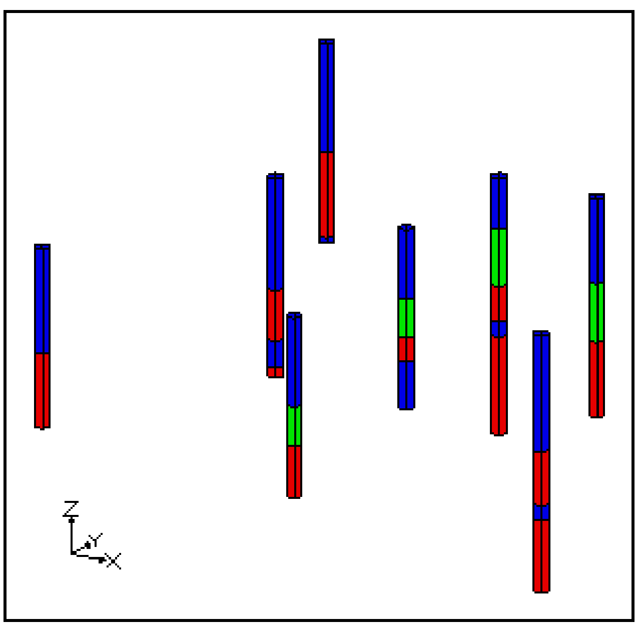

Figure 3-12 Borehole example of mesh layering that will need to be averaged

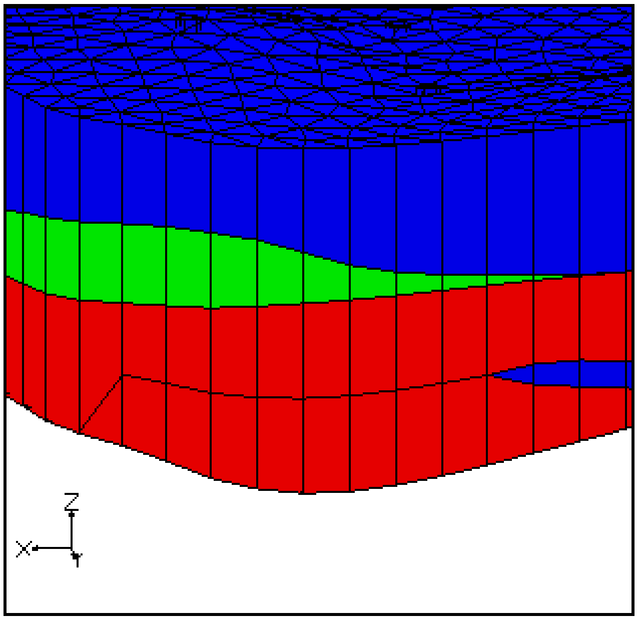

Figure 3-13 3D mesh generated showing the averaging of the layer thickness.

This operation is done by looping thought the nodes of each layer and checking the corresponding nodes on the layer below to see if they are both attached to an element of the same material. The check is performed from the top layer downward for each node until a node is found that is attached to an element of a different material. The number of layers is divided by the distance from the original node to the bottom node to determine an average thickness for the elements of that material type. The locations of the interior nodes along the vertical edges are then modified to form elements with a uniform average thickness.

3.3.3 Target Element Thickness

One of the more advanced options is the ability to assign a target thickness to elements based on their resulting material type. The elements in each mesh layer are assigned a material from the overlying horizon layer. These material layers can then be subdivided creating more layers per material. For each material the user specifies a target thickness. The appropriate number of layers is determined by dividing the element thickness by the target thickness and rounding the value to the closet integer.

Each material layer is divided one element at a time after the original mesh has been created. The algorithm first loops thought all of the mesh layers from top to bottom. For each layer it then loops through each mesh node and calculates the distance between it and the coincident node on the layer below. This distance is then divided by the target thickness for the given material type. The resulting value is then rounded to determine the optimal number of nodes to insert into this vertical element edge. Originally, element edges were not split until the thickness was twice as great as the desired thickness. This was found to create large element size transitions. The algorithm was then modified to split the elements if the edge length was greater than a 150 % of the desired thickness. This simple average helped smooth the transitions between the elements. This process is repeated for every node on each layer, and a table is created linking each node to the amount of nodes to insert below along the vertical edge of the elements.

Next, we loop thought all of the elements in each layer inserting the new nodes into the mesh and reforming the elements. The process is illustrated in Figure 3-14.

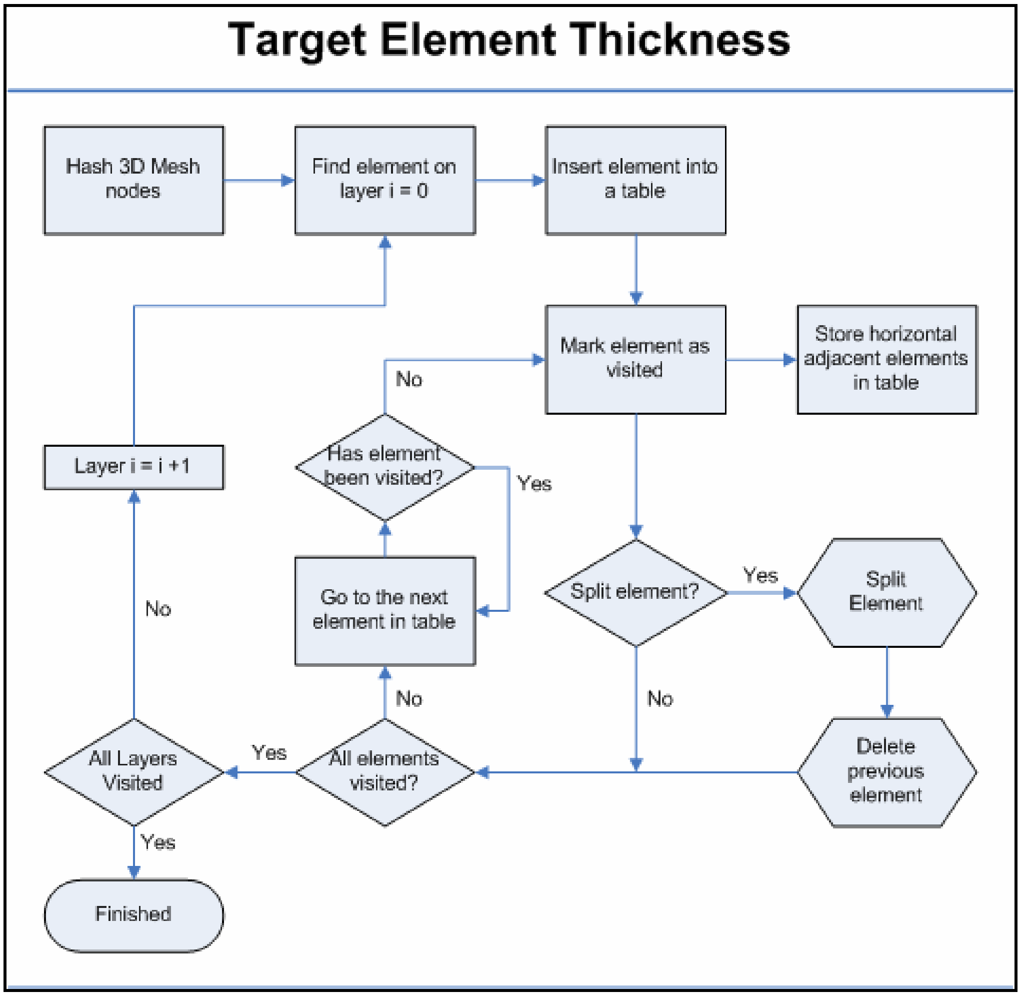

Figure 3-14 Target element thickness algorithm

The nodes of the existing 3D mesh are first placed in a hash table with the node location as the key to the table. This table is created so that nodes are not duplicated in the 3D mesh allowing the splitting of individual elements without duplicating nodes on neighboring elements.

Next we begin the splitting process by first locating any element in the bottom layer. This first element is added to a table, and all of the horizontal elements that neighbor this element are also added to the table. The element is marked as having been visited and is split depending upon the number of nodes to be inserted into each of its vertical edges. We then loop to the next element in our table and repeat this process by adding its neighboring elements to the table and splitting it if necessary. This element is in turn marked as visited and we move on to the next element in the table. This is continued until all of the elements in a layer have been visited. This process is repeated for each layer in the 3D mesh.

Elements are subdivided depending upon the number of new nodes that will be place upon each vertical edge of the element. The process is different for each type of element. The algorithm supports the subdividing of wedge, pyramid and tetrahedral elements.

Previously, we created a table that linked each node in the mesh to the number of new nodes to insert along each vertical edge of the element. For each element type there are three nodes that define the top of the element. For example, the top face of a wedge element is defined by the nodes 3, 4, and 5 as shown in Figure 3-15. These nodes are accessed in the node table and the number of nodes to create along each vertical edge is retrieved. The new nodes are inserted along each edge and added to the hash table so that nodes inserted on adjacent elements are not duplicated. Once the nodes have been inserted new elements are then created. The number and type of elements created are dependent upon the number of new nodes inserted on each edge of the element. This is done differently for each element type and is discussed separately for each below.

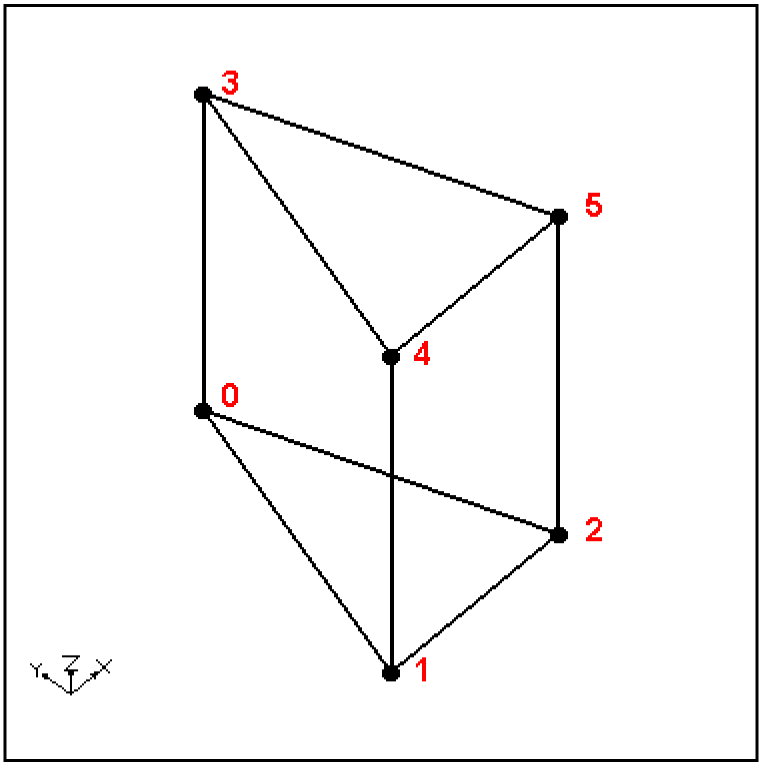

Figure 3-15 Wedge element node numbering.

3.3.3.1 Wedge Element Splitting

The three vertical edges of a wedge element can have different thicknesses, thus a wedge element can be split into a combination of wedge, pyramid, and tetrahedral elements. The number of wedge elements to create is equal to the maximum number of nodes inserted along any of the three edges. The number of pyramids created equals the number of nodes along the edge with the median umber of nodes minus the number of nodes on the edge with the least number of nodes. The number of tetrahedra is lastly determined by subtracting the number of nodes on the edge with the most nodes from the number of nodes on the edge with median umber of nodes inserted.

For example, two nodes were inserted along the edge defined by nodes 1 and 4 in Figure 3-16A, one node was inserted along the edge defined by nodes 2 and 5, and no nodes were inserted on the final edge. Thus the maximum number of nodes inserted along an edge is two, the median number of nodes inserted is one, and the minimum number of nodes is zero. The number of new wedge elements to create is one, the minimum number of nodes plus one. A new wedge element is then created between nodes 4, 4, 5, 0, 7 and 8 as shown in Figure 3-16B. Next, the number of pyramid elements created is one, determined by subtracting the median number of nodes from the max number. The new element is then created between nodes 6, 7, 8, 2, and 0 as shown in Figure 3-16C. In most cases tetrahedral elements need to be inserted in order to finish the splitting process. In our example one tetrahedral element needs to be inserted, calculated by subtracting the medium number of nodes from the max number of nodes. The tetrahedral element is created from nodes 0, 1, 2, and 6 (Figure 3-16D). To finish the splitting process the original element is deleted, resulting in three new elements, each with a uniform thickness (Figure 3-17).

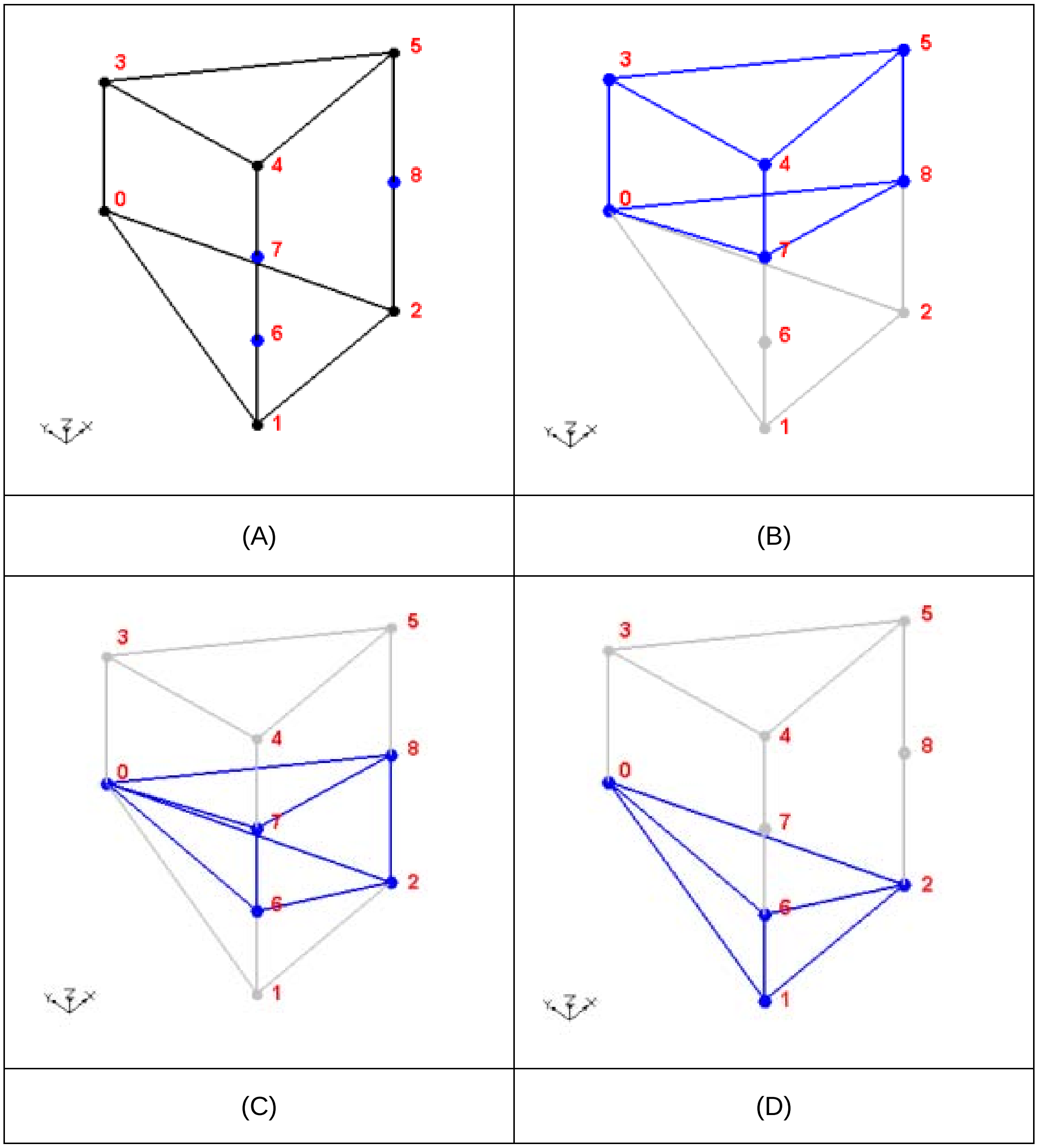

Figure 3-16 Wedge element splitting. A) New nodes inserted along vertical edges of the element. B) Wedge element created. C) Pyramid element created. D) Tetrahedral element created.

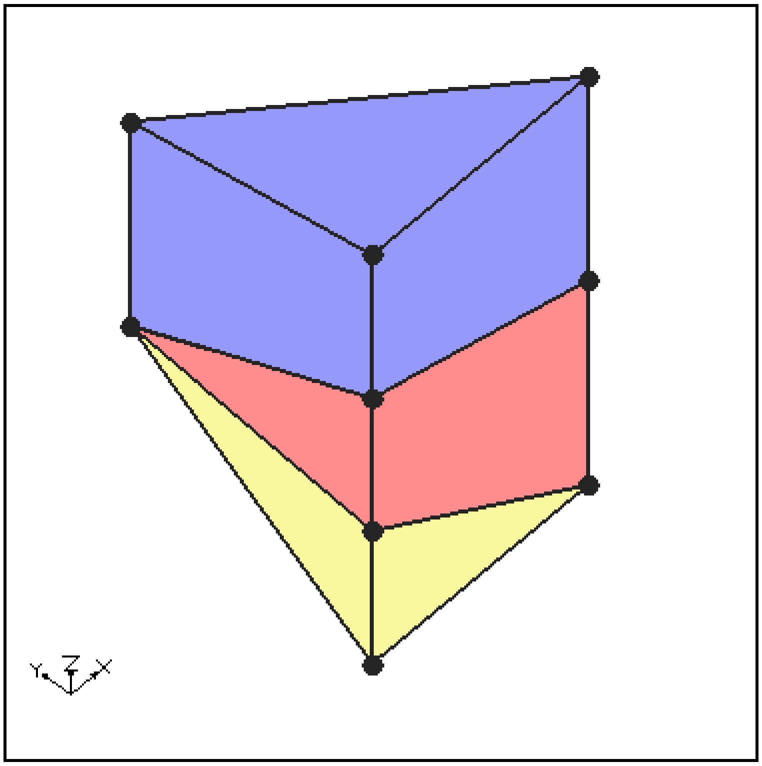

Figure 3-17 Wedge element split by adding interior nodes

3.3.3.2 Pyramid Element Splitting

Pyramid elements have two vertical edges on which nodes are inserted as shown in Figure 3-18. The number of new nodes on these edges can vary depending on their lengths, thus the element can be split into additional pyramid and tetrahedral elements. The number of pyramid element to create is equal to the number of nodes inserted along the edge with the least nodes of nodes inserted. The number of tetrahedral elements is then determined by the difference in the number of nodes inserted on the two edges.

For example, in Figure 3-18A two nodes were inserted along one edge while only one node was inserted on the other edge. The max number of nodes is two and the min number of nodes inserted is one. Two new pyramid elements are created in this case. The two new pyramid elements are created between nodes 4, 2, 3, 6, and 7 and between nodes 4, 6, 7, 5, and 8 as shown in Figure 3-18B and Figure 3-18C. One tetrahedral element also needs to be created. The tetrahedron is created from the nodes 4, 1, 0, and 5 (Figure 3-18D). In our example the original element was split into three elements consisting of two pyramids and one tetrahedron element (Figure 3-19). Depending on the lengths any combination of elements can be created.

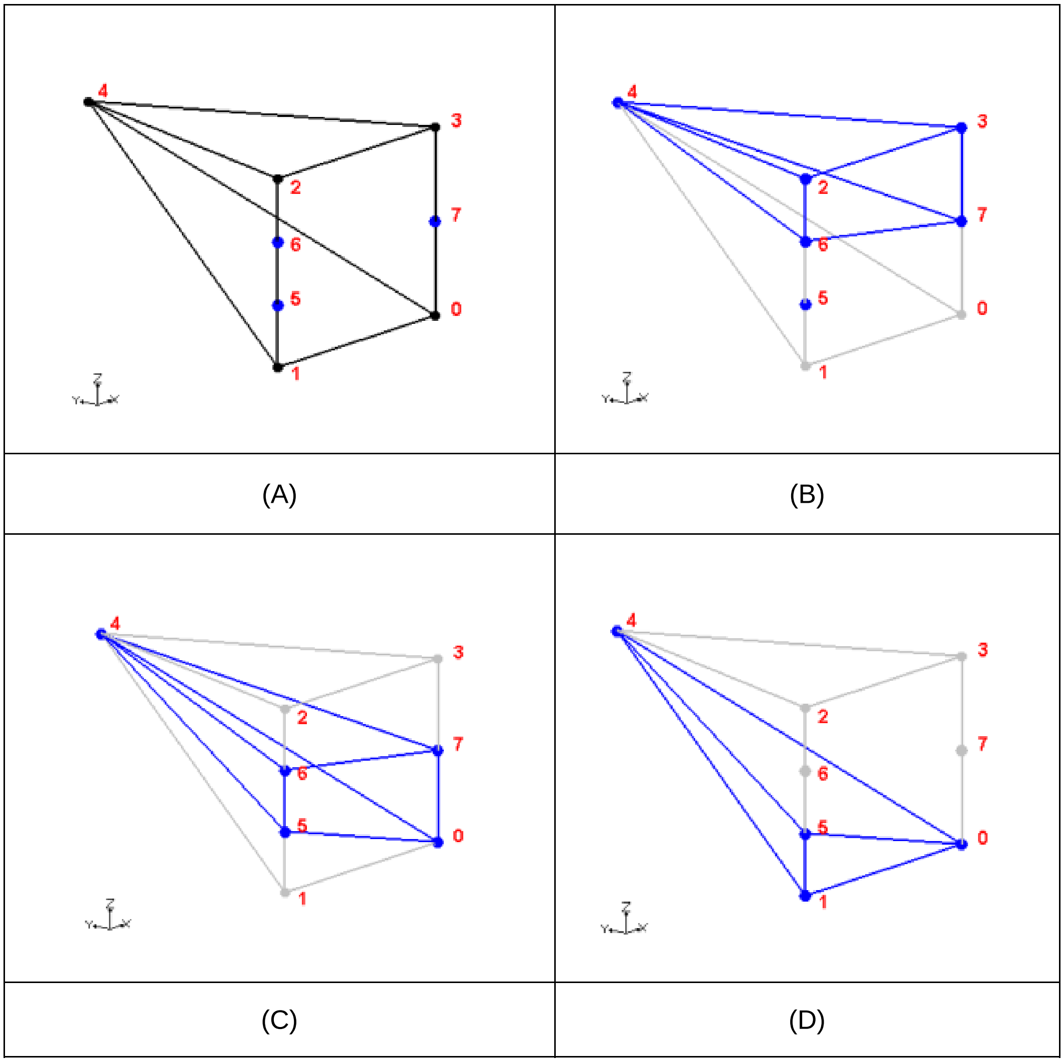

Figure 3-18 Pyramid element splitting. A) New nodes inserted along vertical edges of the element. B) Top pyramid element created. C) Middle pyramid element created. D) Tetrahedron element created.

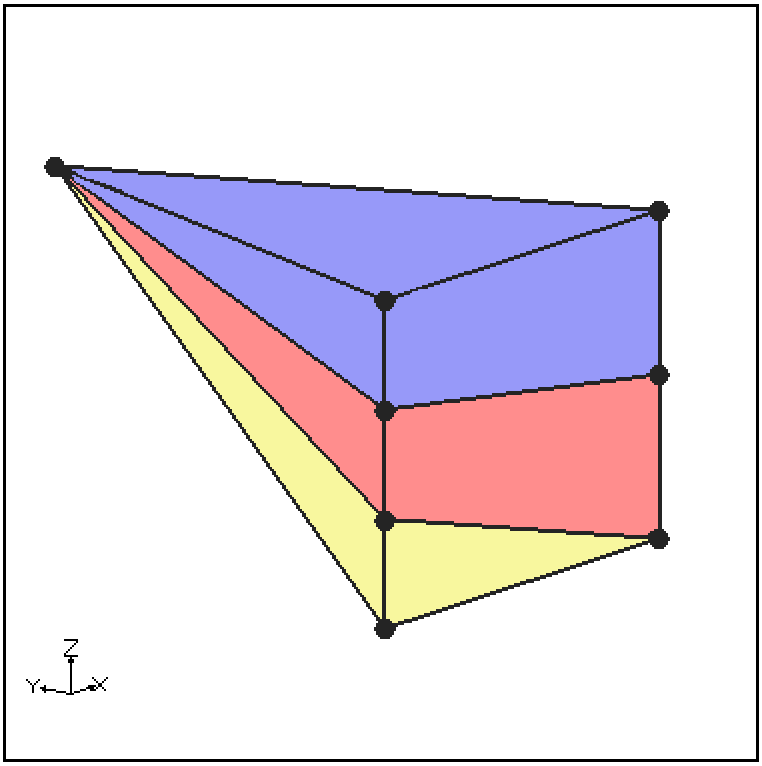

Figure 3-19 Pyramid element split by adding interior nodes

3.3.3.3 Tetrahedral Element Splitting

Tetrahedral elements only have one vertical edge on which nodes are inserted. The element can only be split into more tetrahedral elements. The number of new tetrahedral elements to create is equal to one less than the the number of nodes inserted along the vertical edge. An example of this is shown in Figure 3-20.

In this example only one node is inserted along the vertical edge Figure 3-20A. Two new tetrahedral elements need to be created from nodes 3, 0, 2, and 4 (Figure 3-20B), and from nodes 3, 0, 1, 4 (Figure 3-20C). The original element is removed from the mesh and the two newly formed elements are inserted (Figure 3-21).

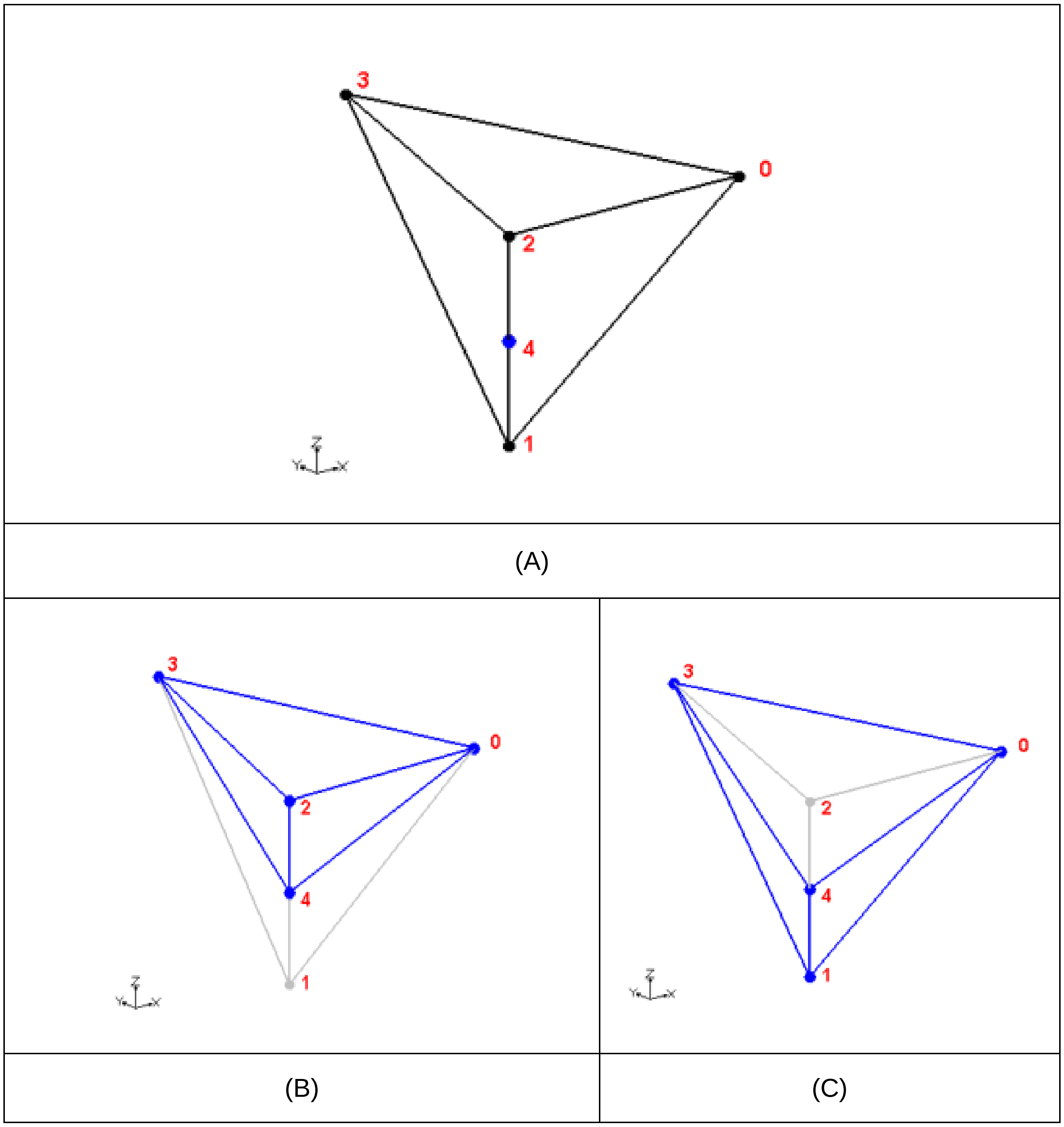

Figure 3-20 Tetrahedral element splitting. A) New nodes inserted along vertical edge of the element. B) Top tetrahedral element created. C) Bottom tetrahedral element created.

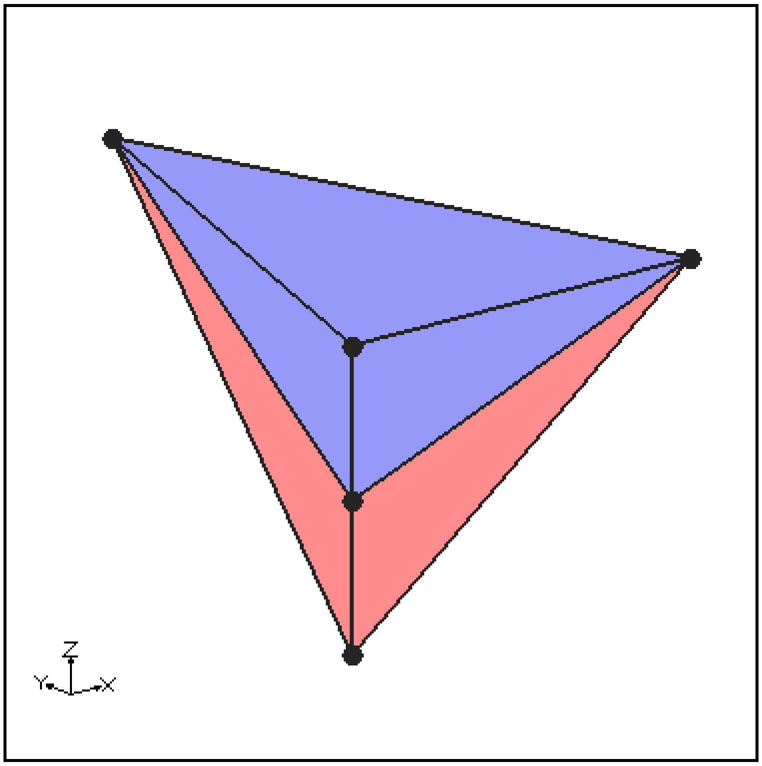

Figure 3-21 Tetrahedral element split by adding interior nodes

3.3.4 Refinement options

The algorithm also includes refinement options that apply to the 3D mesh. During the meshing process some of the wedge elements degenerate to pyramid and tetrahedral elements to represent the pinch out boundaries. Some finite element based numerical models do not support pyramid elements. Therefore, when using one of these types of models, pyramid elements need to be first refined to tetrahedral elements. The algorithm includes this common task as an option.

The first refinement option converts all of the 3D mesh elements to tetrahedral elements. This greatly increases the number of elements in the mesh without improving the stratigraphic representation. A second option is to refine only the mesh layers that contain pyramid elements. If a layer has a pyramid element then the whole layer would then be converted to tetrahedral elements to remove the pyramids. The whole layer needs to be converted because transitioning from tetrahedral elements directly to wedge elements in a layer is not possible. Thus any of the layers that contain pyramid elements from a pinch-out boundary will be refined to tetrahedral elements. Layers that do not contain pyramid elements will remain as either wedge or tetrahedral elements. This optimally eliminates pyramid elements and decreases the total amount of elements needed to represent the geology.

The *refine layers with pyramids* algorithm works by first looping through each element layer adding all of the pyramid elements to a list. It then adds all of the horizontal neighboring elements of the pyramid elements to the same list. This process results in a list of all elements in a layer with a pyramid element. The algorithm them converts all of the elements in the list to tetrahedral elements. The refinement process is done using a coarse refinement scheme (Staten, 1997).

Figure 3-22 shows a set of horizon surfaces defined by TINs. The horizons to mesh algorithm with no refinement was used to create a 3D mesh from the TIN data. As can be seen in Figure 3-23 the second layer of the 3D mesh is pinched out by the bottom layer. Sixteen pyramid elements were created along the border between these layers to represent the pinch out of the second layer. The number of elements in the unrefined mesh totaled 1,580.

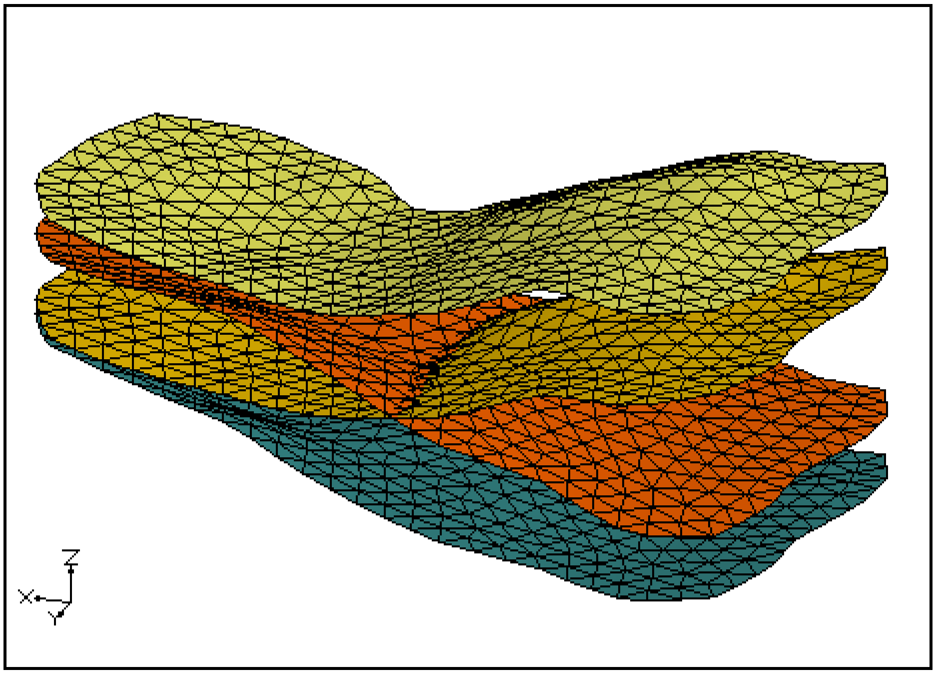

Figure 3-22 Horizon TINs used to define layered stratigraphy

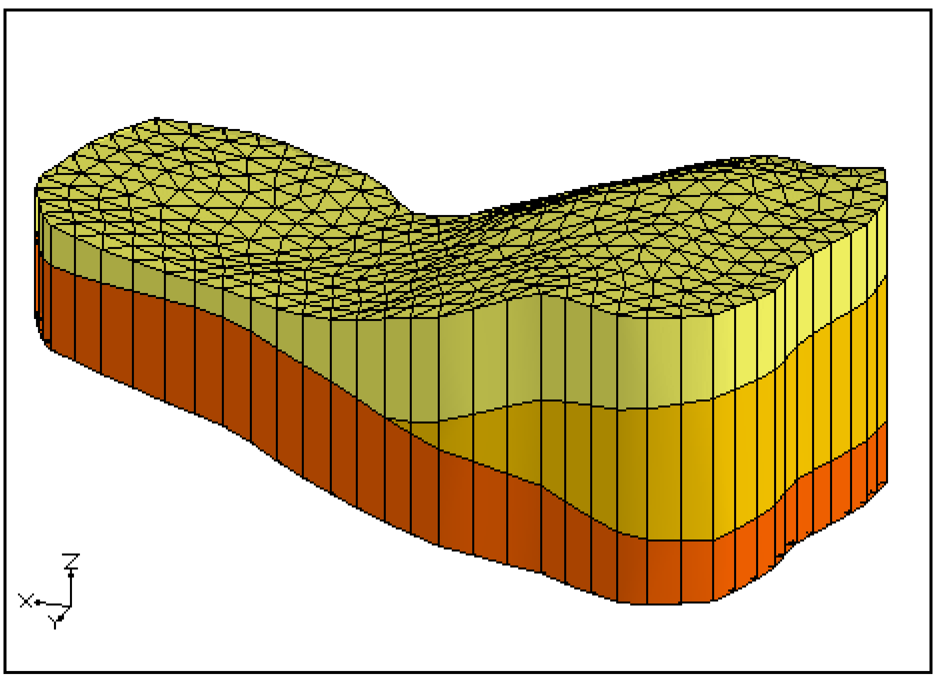

Figure 3-23 3D mesh without element refinement

Another 3D mesh was created using the *refine all elements* option (Figure 3-24). Each element was converted to a tetrahedral element. This resulted in a 3D mesh with 4,769 elements (67 % increase). Finally, Figure 3-25 shows the results of the option to only refine the layers with pyramid elements to create a 3D mesh. Only the second layer which contained the pyramid elements was refined, and the resulting mesh included a total of 2,271 elements (1,027 tetrahedral elements and 1,244 wedge elements). The results are summarized in Table 3-1.

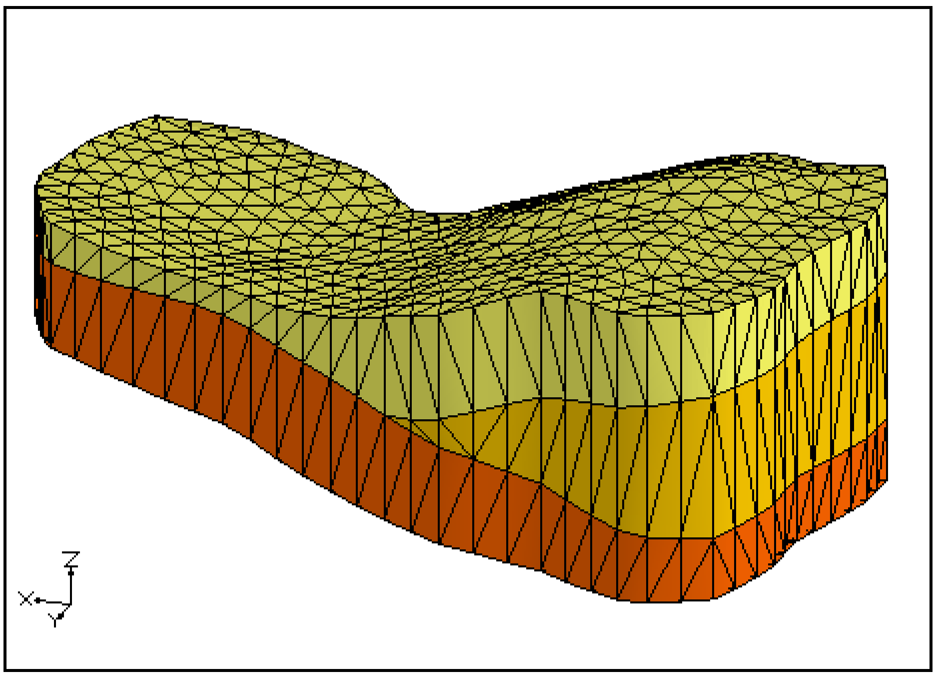

Figure 3-24 3D mesh with all element refined

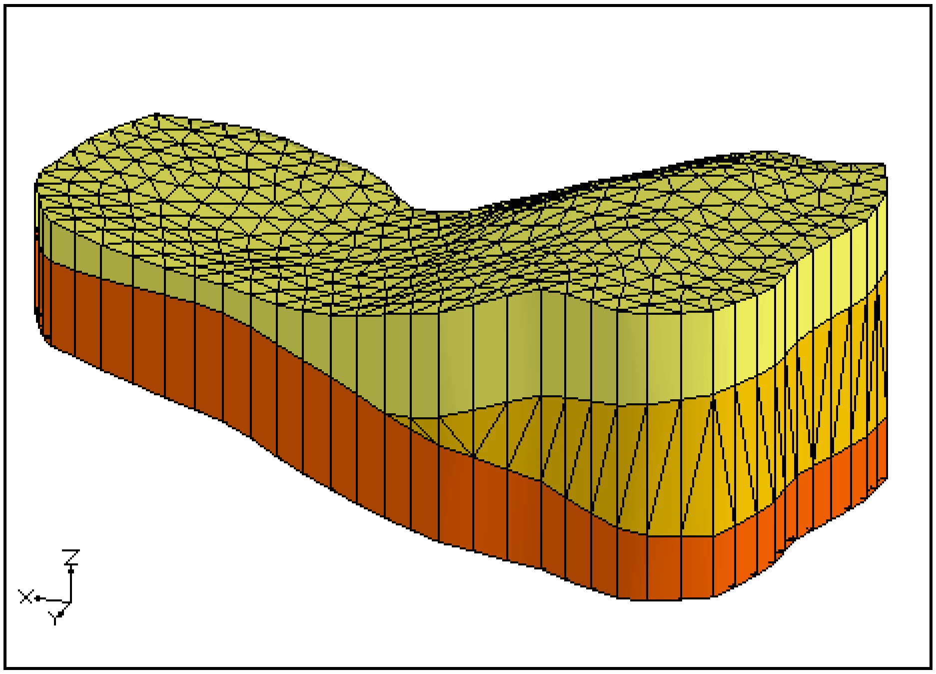

Figure 3-25 3D mesh with elements refined in the layers with pyramid elements

The refinement options allow for quick redefinition of element types to create a mesh compatible with various numerical models. The option to refine only the layers with pyramids provides the ability to closely represent the layer pinch outs without having to increase significantly the total number of elements.

Table 3-1 Refinement Options

Refinement Options	Number of elements			Total No. of elements	% increase
	Wedge	Pyramid	Tetrahedron		
No refinement	1,549	16	15	1,580	-
Refine all elements	0	0	4,769	4,769	67%
Refine layers with pyramid elements	1,244	0	1,027	2,271	30%

4 Case Studies

This research includes two case studies to demonstrate the application of the horizons to 3D mesh algorithm. The first case study, a simple academic model, illustrates the layer distribution options. The second case study, a real world example, demonstrates the scope and robustness of the algorithm.

4.1 Case Study 1

The data for case study 1 was prepared previously as an academic exercise for a tutorial. The tutorial originally used a method that would extrude a 3D mesh between TIN surfaces. This research recreated the 3D mesh from the same data using the horizons algorithm to demonstrate the advantages of the layer distribution options.

The model site, a small costal aquifer, consists of two main stratigraphic zones, an upper and lower aquifer. Figure 4-1 shows a quad map of the model boundaries. The stratigraphy of the model is simple with the upper aquifer extended over the whole model area, and the lower aquifer pinching out on the Western side from the thinning of the aquifer due to a high bedrock elevation. Figure 4-2 shows each of the layers represented by horizon TINs: the ground surface in yellow, the bedrock in blue, and the boundary between the two aquifers in green.

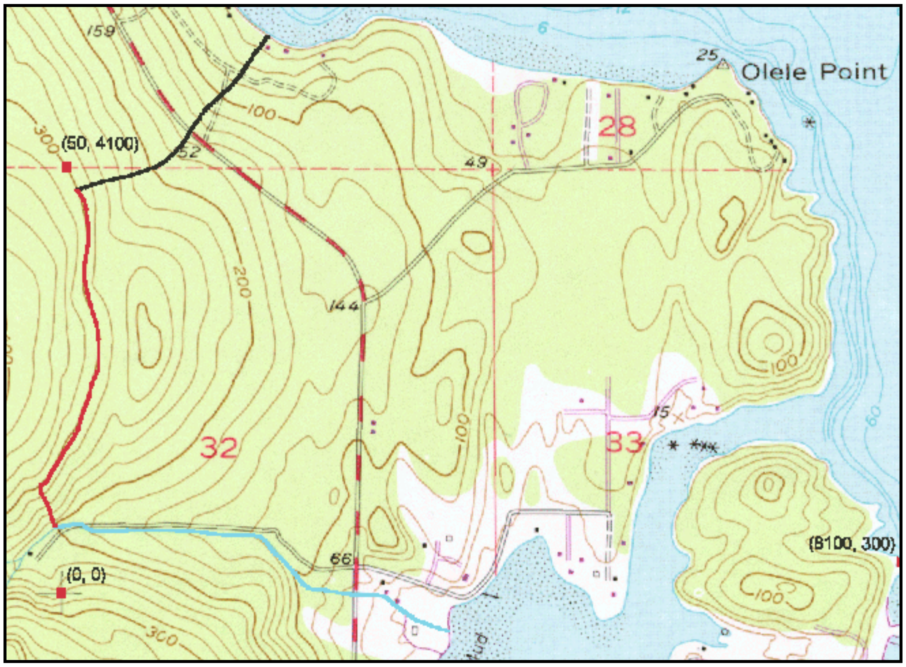

Figure 4-1 Case 1 Site to be modeled

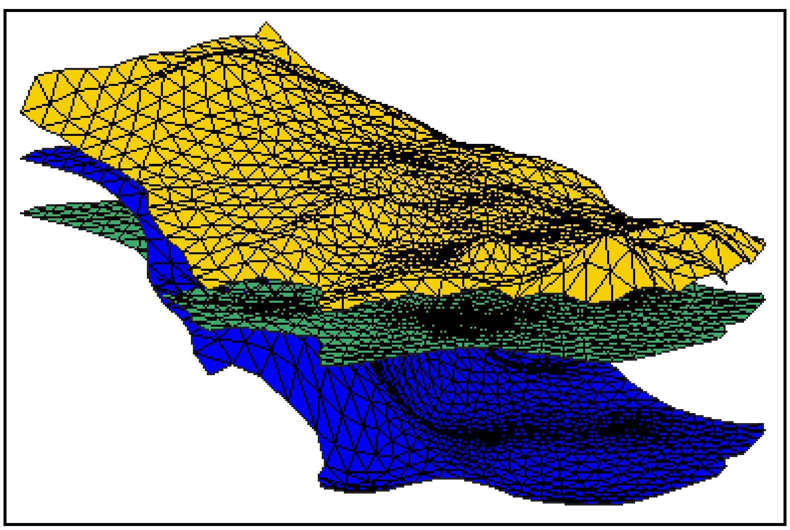

Figure 4-2 Case Study 1 Tin Stratigraphy

First, all of the necessary inputs for the algorithm were prepared. A 2D projection mesh created using a tessellation scheme with some refinement around the three well locations marked in red is shown on Figure 4-3. Horizon IDs were also assigned to the TIN layers, the ground surface = 10, the lower aquifer top surface = 5, and the bedrock layer = 0. The top elevation of the 3D mesh was set to use the ground surface TIN, and the bottom of the 3D mesh was set to use the bedrock TIN. Also the option to distribute the layers by elevation was set to 10 ft for the upper aquifer and 20 ft for the lower aquifer.

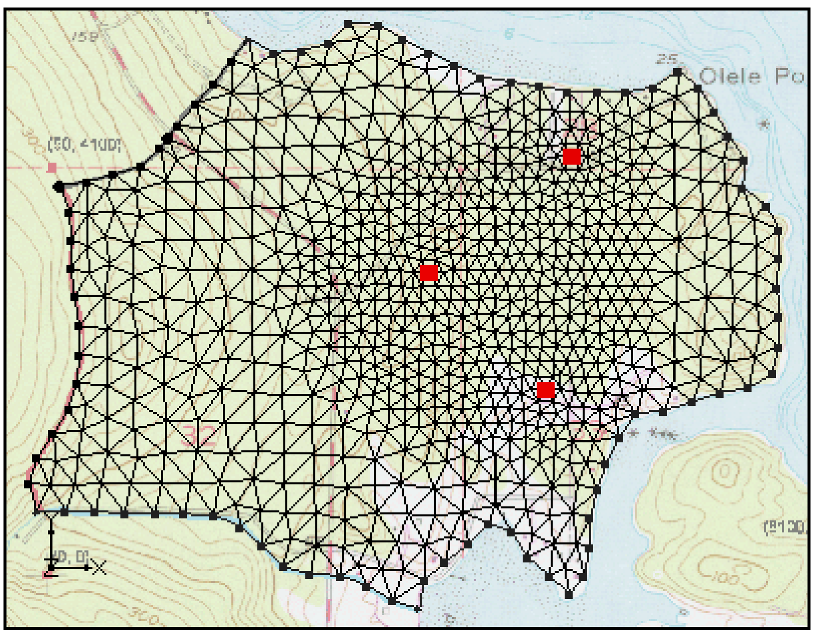

Figure 4-3 Case study 1 2D projection mesh

The resulting 3D Mesh is shown in Figure 4-4. The upper aquifer was split into layers approximately 10 ft thick in a maximum of four layers. The lower aquifer was split into layers approximately 20 ft thick. This resulted in four layers that taper off to the West where the layer pinches out into the bedrock. Examining the 3D mesh reveals well-formed elements and smooth transitions between each of the layers. The previous method of extruding the mesh between TINs results in an inferior distribution of layers and did not simulate pinch out layers

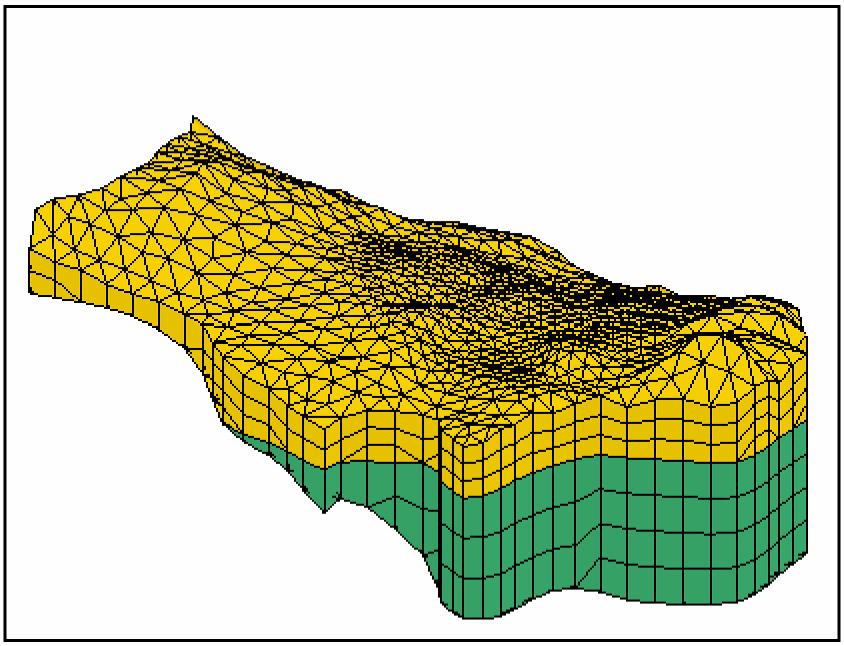

Figure 4-4 Case Study 1 3D Mesh

4.2 Case Study 2

Case study 2 used the horizons algorithm to match the geology of a much larger real world site with more complex geology. The borehole data were provided by Clarissa Hansen of the U.S. Army Engineer Research and Development Center (ERDC). This data was collected as part of the Regional Engineering Model for Ecosystems Restoration (REMER) project in South Florida. This project is part of the Compressive Everglades Restoration Plan (CERP) whose focus restores natural flow patterns through the everglades. The borehole locations and the aquifer boundary are shown in Figure 4-5. The 3D mesh created from these data extends from the northern boundary of Lake Okeechobee down to the southern tip of the Florida peninsula. The area of the model is approximately 8,000 square miles. There are 1037 boreholes which characterize the stratigraphic layers of the aquifer.

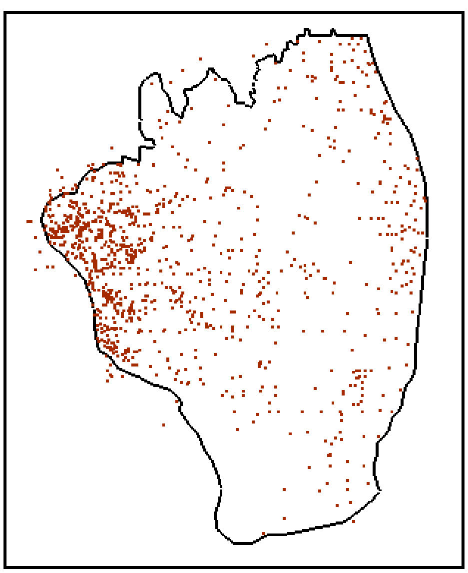

Figure 4-5 Case study 2 site boundary and borehole locations

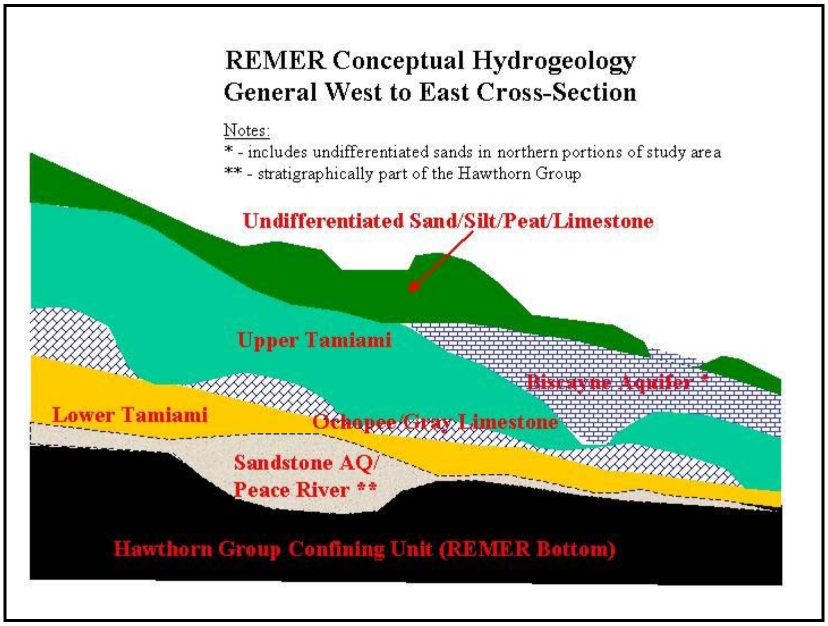

Figure 4-6 Case Study 2 geologic cross-section

Geologists from the Jacksonville District of the Army Corps of Engineers have delineated the hydrogeologic zones from the borehole data. They determined that the geology consists of seven distinct layers. A typical cross-section is shown in Figure 4-6. Figure 4-6 does not include the top thin topsoil layer. The next layer is miscellaneous undifferentiated sands. This layer continues through the aquifer except between the Eastern Everglades National Park and the city of Maimi where the Lower Biscayne layer comes to the surface. This next Lower Biscayne layer only exists in the Eastern two thirds of the aquifer. It is pinched out by the underlying Upper Tamiami formation which extends over the whole area except for some small areas. The next layer down, called the Grey Limestone layer, is also non-continuous throughout the model domain. The last two layers are both continuous. The upper layer is the Lower

Tamiami formation and the bottom layer is a combination of a Sandstone Aquifer and a thin Peace River formation. The aquifer is confined underneath by the continuous Hawthorn formation.

The modeling process began with the creation of a TIN for each of the stratigraphic layers from the borehole data. These TINs are shown in Appendix B with each TIN's color representing the top of corresponding stratigraphic layer in Figure 4-6. A horizon Id was assigned to each of the layers to define the depositional sequence (Table 4-1).

Table 4-1 - Case study 2 TIN horizon ID assignments

Layer Name	Horizon Id
Top Soil	70
Undifferentiated Sand	60
Biscayne	50
Upper Tamiami	40
Grey Limestone	30
Lower Tamiami	20
Sandstone/Peace River	10
Hawthorn	0

Next, a 2D triangulated projection mesh was created from the boundary polygon to guide the 3D mesh generation (Figure 4-7). To aid the meshing process, two horizon coverages were also created to limit the extents of the Grey Limestone and the Lower Biscayne layers (Figure 4-8).

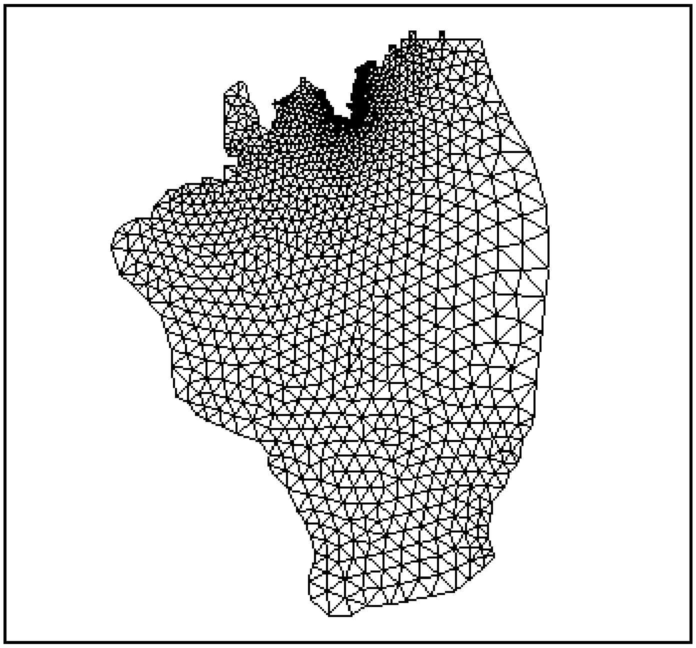

Figure 4-7 - Case study 2 2D project mesh

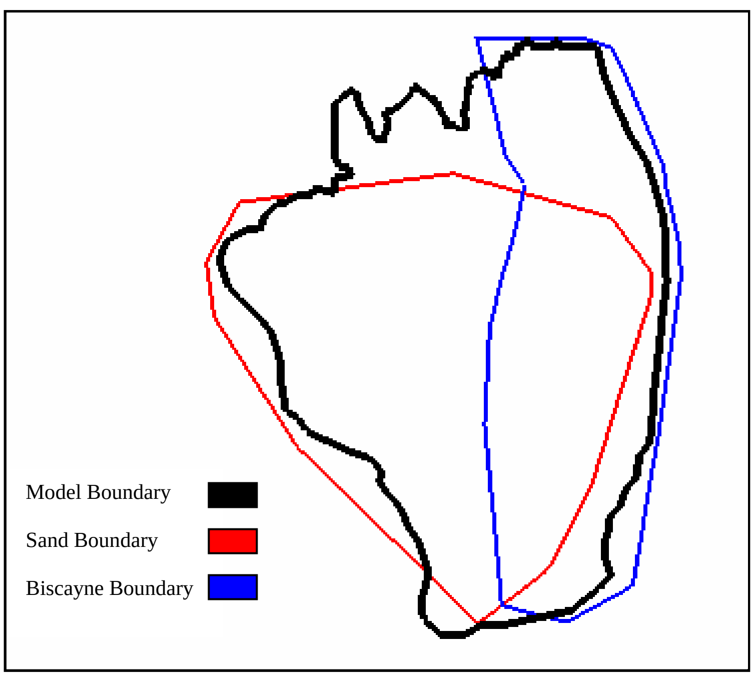

Figure 4-8 - Case study 2 horizon coverages

A 3D mesh, generated using the horizons algorithm and the specified inputs, is illustrated in Figure 4-9. Each stratigraphic layer is correctly represented in the 3D mesh. The extents of the Biscayne and Grey Limestone layers are also limited from the horizon conceptual model. A figure of each layer is presented in Appendix B. This case study illustrates the algorithm is robust in that it can handle large projects of complex geology.

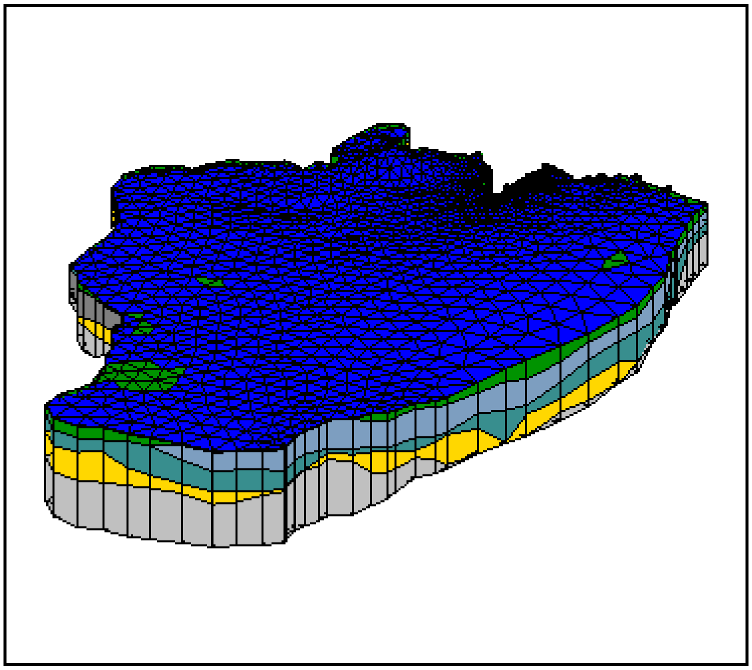

Figure 4-9 - Case study 2 3D mesh

5 Conclusions

The horizons to mesh algorithm effectively develops well-formed 3D unstructured finite-element meshes for numerical analysis. The first advantage of the algorithm is that it uses available groundwater data to guide the meshing process so that the resulting mesh will more closely match the stratigraphic layering of geology. Second, the algorithm supports borehole cross-sections and horizon conceptual models to guide the interpolation process. Thirdly, many options are included to allow control over the meshing process to create better-formed elements. Lastly, the algorithm supports layer pinch-outs so that every layer does not need to exist through the entire extent of the model.

Two future areas of research in this area would be an element smoothing option and to expand the algorithm to support hexahedral elements. The smoothing option would be used to post-process the resulting mesh and help create better element transitions. When using the "set element thickness" option some large element size transitions can occur, and this can cause some numerical instability. These transitions can be modified and smoothed in the z direction to reduce the difference in the element sizes. To support hexahedral elements the algorithm could be expanded to allow the 2D projection mesh to contain not only triangular elements but also quadrilateral elements. Thus the projection of the 2D mesh would be extruded to create hexahedral elements. However, the post-processing functionality to create better formed elements would need to be expanded to support the refining and splitting of hexahedral elements.

6 References

Carl W. Gable, Harold Trease, and Terry Cherry (1996). "Geological Applications of Automatic Grid Generation Tools for Finite Elements Applied to Porous Flow Modeling." *Numerical Grid Generation in Computational Fluid Dynamics and Related Fields* , edited by B. K. Soni, J. F. Thompson, H. Hausser and P. R. Eiseman, Engineering Research Center, Mississippi State Univ. Press

Fugal, Adam, C. (2005). “A New Automated Approach to Three-Dimensional Soil Stratigraphy Modeling.” *Masters Project*, BYU, Provo, Utah

Jones, Norman L. and Stephen G. Wright (1993). “Subsurface characterization with solid models.” *ASCE Geotechnical Engineering Journal,* Vol. 119, No. 11, November, 1993, pp. 1823-1839.

Lemon, Alan M. and Jones, Norman L. (2003). “Building solid models from boreholes and user-defined cross-sections.” *Computers & Geosciences*, Vol. 29, Issue 5, pp. 547-555

Owens, S. J. (2005). “A Survey of Unstructured Mesh Generation Technology.” http://www.andrew.cmu.edu/user/sowen/survey/index.html

Owen, Steven J., Norman L. Jones, and Jeffrey P. Holland (1996). "A Comprehensive Modeling Environment for the Simulation of Groundwater Flow and Transport." *Engineering With Computers*, Vol. 12, pp. 235-242

Quadros, W.R. and Kenji, Shimada (2002). "Hex-Layer: Layered All-Hex Mesh Generation on Thin Section Solids via Chordal Surface Transformation." *Proceedings, 11th International Meshing Roundtable,* pp 162-182

Roca, Xevi, Sarrate, Josep, and Huerta, Antonio (2004). "Surface Mesh Projection for Hexahedral Mesh Generation by Sweeping." *Proceedings, 13th International Meshing Roundtable,* pp 169-180

Shephard, D. (1968). "A two-dimensional interpolation function for irregularly spaced data." *Proceedings of the 23rd National Conference of the ACM*, pp. 517-523.

Staten, M.L. and N.L. Jones (1997). "Local Refinement of Three-Dimensional Finite Element Meshes." *Engineering with Computers*, Springer-Verlag, Vol 13, pp.165-174

Tucciarelli, Tullio (1989). "A Semiautomatic Mesh Generation Algorithm for Three-Dimensional Groundwater Tetrahedral Finite Element Models." *Water Resources Research*, Vol. 25. No. 3 pp. 573-576

Watson, D.F. and G.M. Phillip (1987). "Neighborhood based interpolation." *Geobyte*. Vol. 2, No. 2, pp. 12-16.

Appendix

Appendix A - GMS Interface

As part of the research, a user interface was developed in the Groundwater Modeling System (GMS). The interface is a wizard that allows a user to set up the necessary options to run the *horizons to 3D mesh* algorithm. The wizard is accessed either through the *Tins* or *Borehole* menu in GMS. The wizard is made up of three steps:

1. Horizon elevations
2. Primary mesh and top and bottom elevations
3. Mesh building and interpolation options.

After all of the options on each step are assigned the user selects the *Next* button to bring up the next set of options. Upon completion of the wizard the user selects finish and a 3D mesh is created from the specified inputs and options.

In the first step in the horizons wizard the user specifies the horizon elevations (Figure A-1). As stated previously, horizons layers are specified either on borehole contacts or TINs, thus the dialog allows the user to choose either option separately or together to define the horizon elevations. The user specifies to use only a subset of the boreholes or TINs by selecting a folder in the corresponding tree windows. An option to include borehole cross-sections to guide the interpolation process is also found on this step under the borehole section of the dialog. There is also an option to utilize a horizon conceptual model to constrain the boundaries of the horizon surfaces.

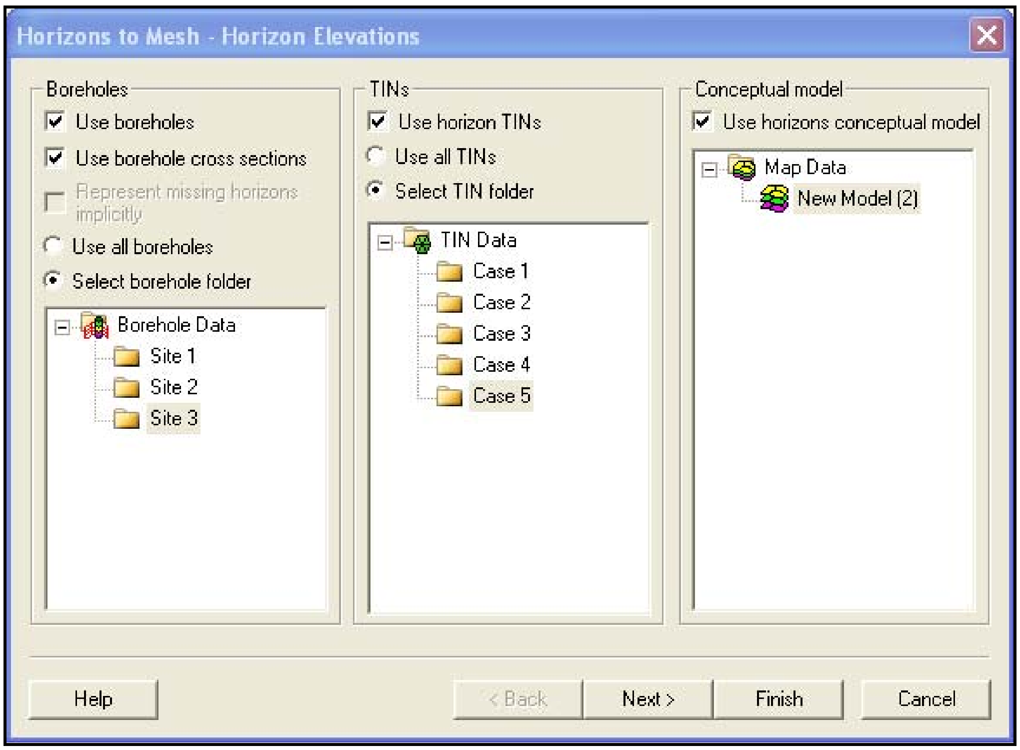

Figure A-1 Step 1 of the horizons to 3D mesh user interface

The second step of the wizard is broken up into two sections (Figure A-2). The first section is used to select the primary 2D mesh to be used as the topology of the 3D mesh. The user can either use the current 2D mesh that is in memory or select a meshing coverage from the tree window. The second section on the dialog allows the user to select how the top and bottom elevations will be defined for the 3D mesh. The three options are to use the top or bottom of the boreholes, specify a constant elevation, or select a TIN to define the elevations.

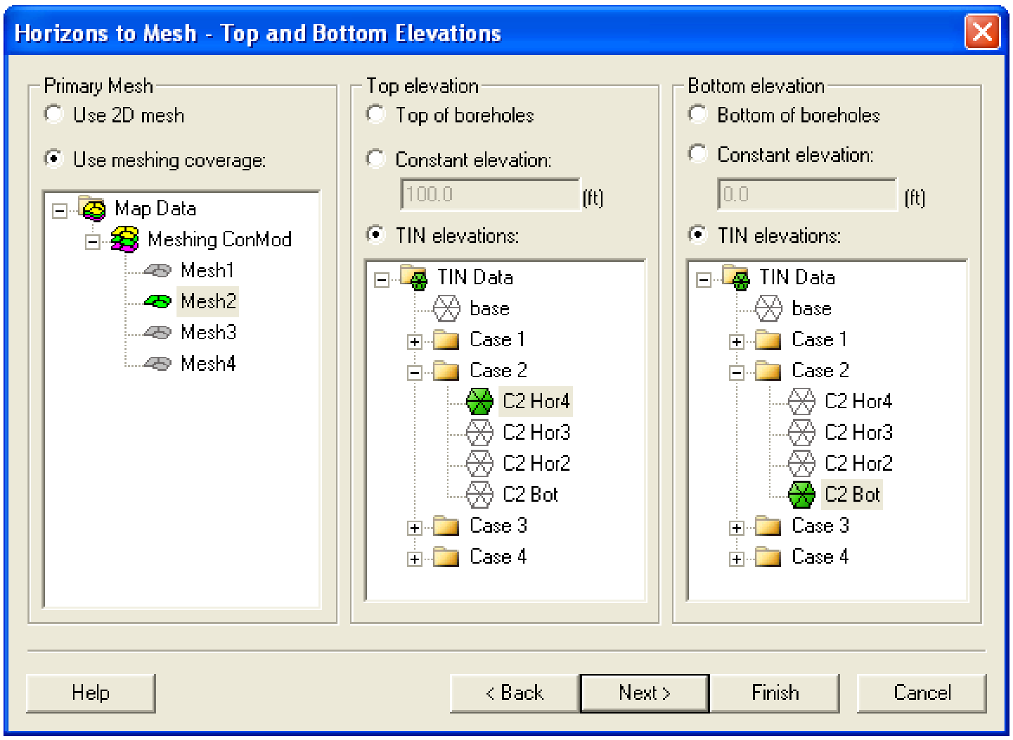

Figure A-2 Step 2 of the horizons to 3D mesh user interface

The final step of the horizons to 3D mesh wizard is used to specify all of the interpolation options of the algorithm and the post processing options that can be used to modify the resulting 3D mesh (Figure A-3). The three optional mesh post-processing options allow the user to specify a minimum element thickness, an element refinement scheme, and a layer distribution. If the layer distribution option is selected, the user can specify a different maximum layer thickness for each horizon material.

This allows the user distribute the layer depths differently through the resulting material layers in the 3D mesh.

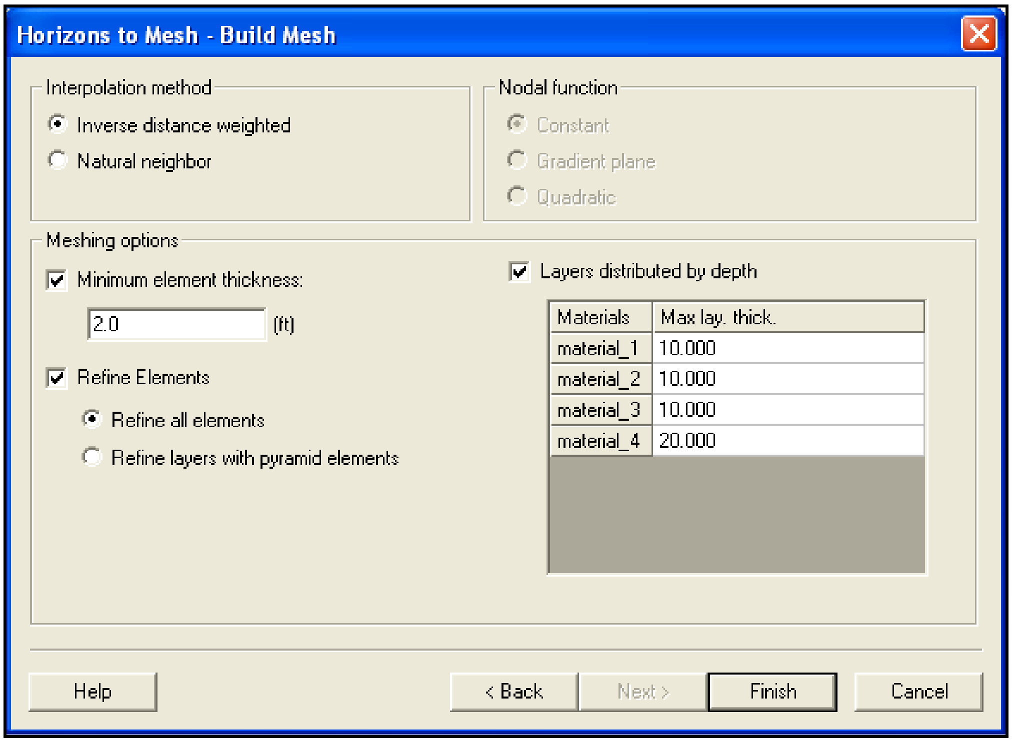

Figure A-3 Step 2 of the horizons to 3D mesh user interface

Appendix B - Case Study 2 Figures

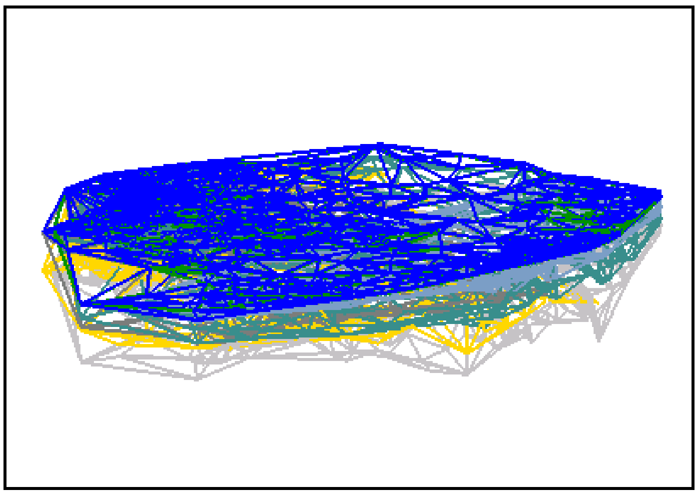

Figure B-1Tin surfaces created for each of the stratigraphic layers

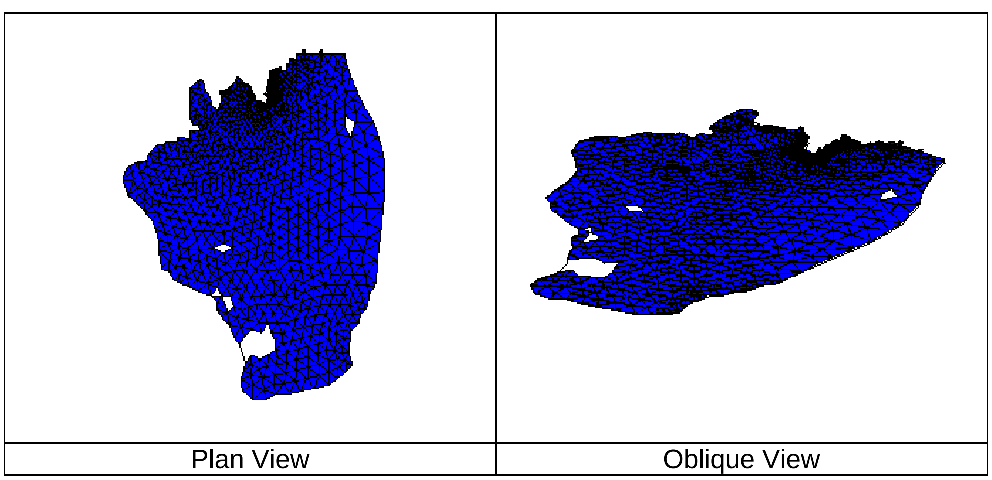

Figure B-2 Top Soil layer of 3D mesh created for case study 2

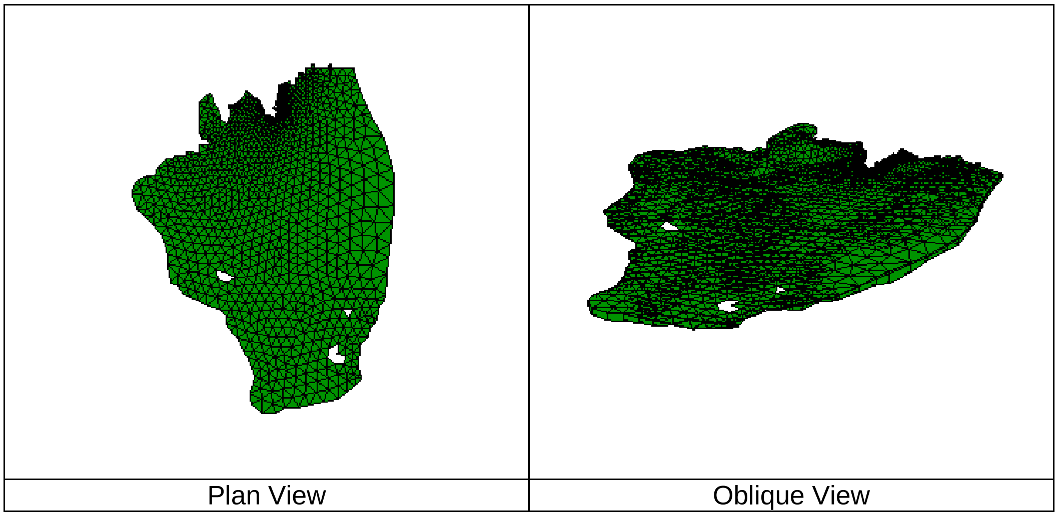

Figure B-3 Sand layer of 3D mesh created for case study 2

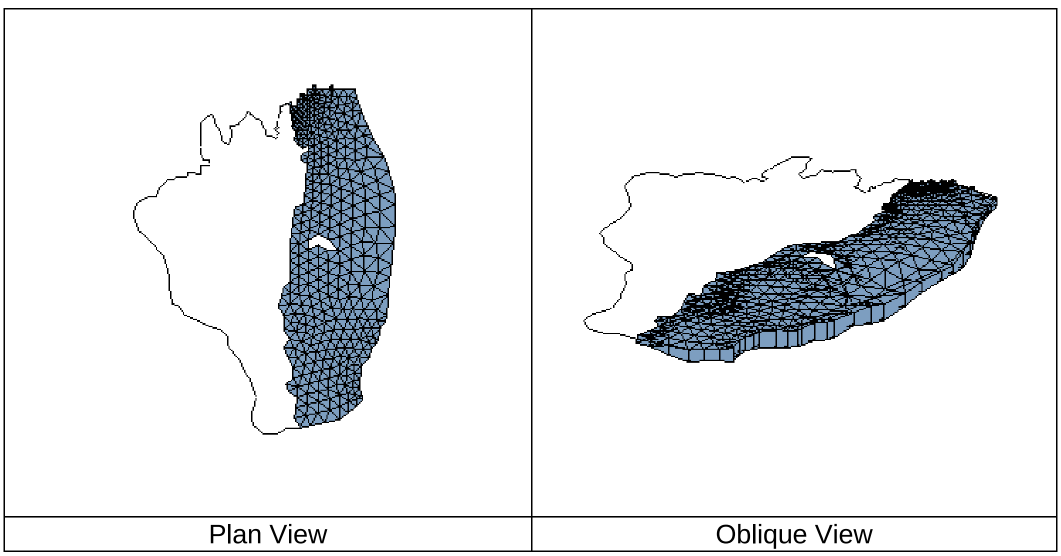

Figure B-4 Biscayne layer of 3D mesh created for case study 2

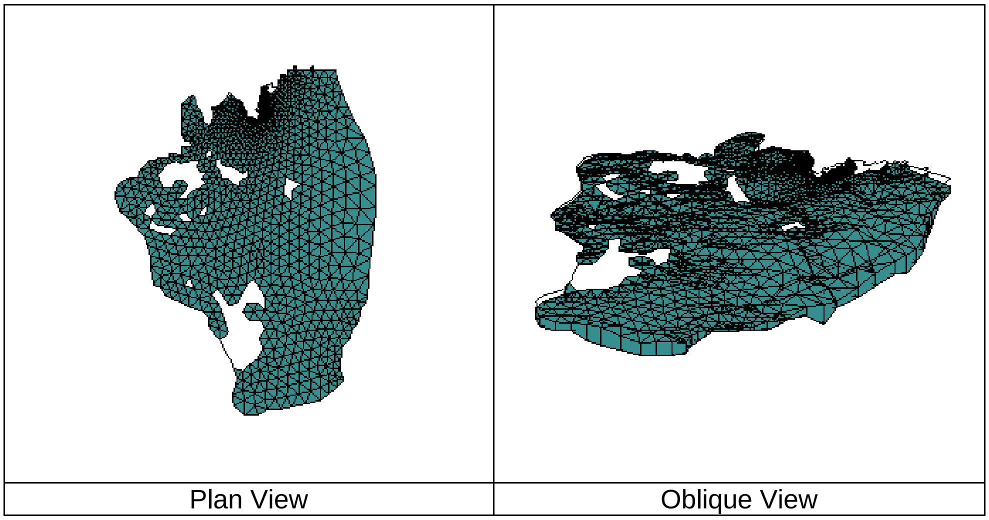

Figure B-5 Upper Tamiami layer of 3D mesh created for case study 2

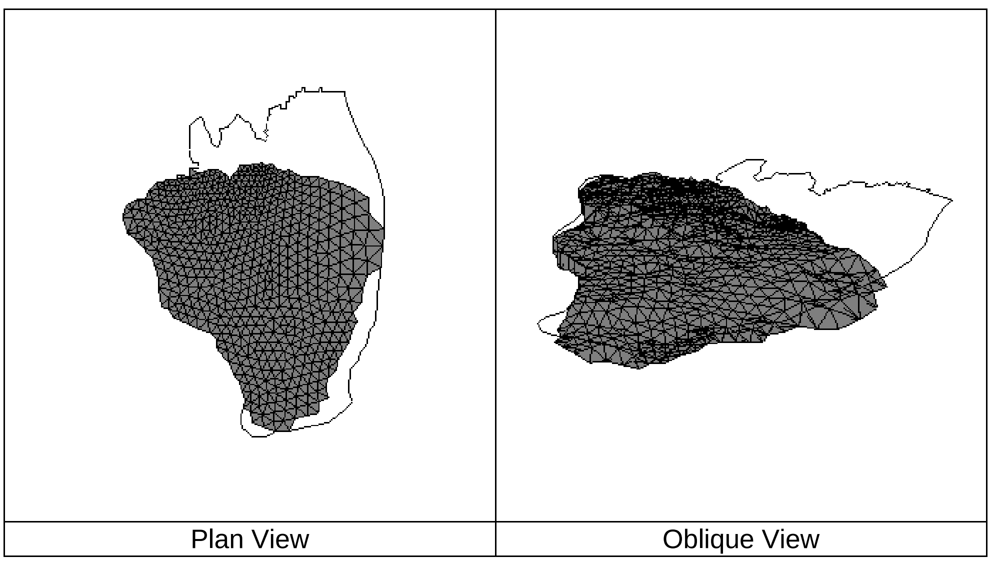

Figure B-6 Gray limestone layer of 3D mesh created for case study 2

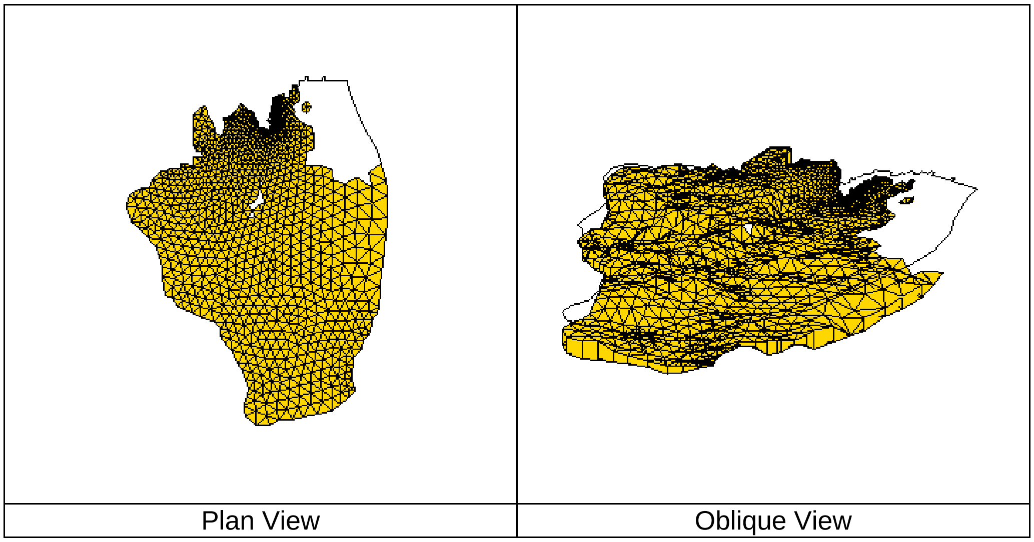

Figure B-7 Lower Tamiami layer of 3D mesh created for case study 2

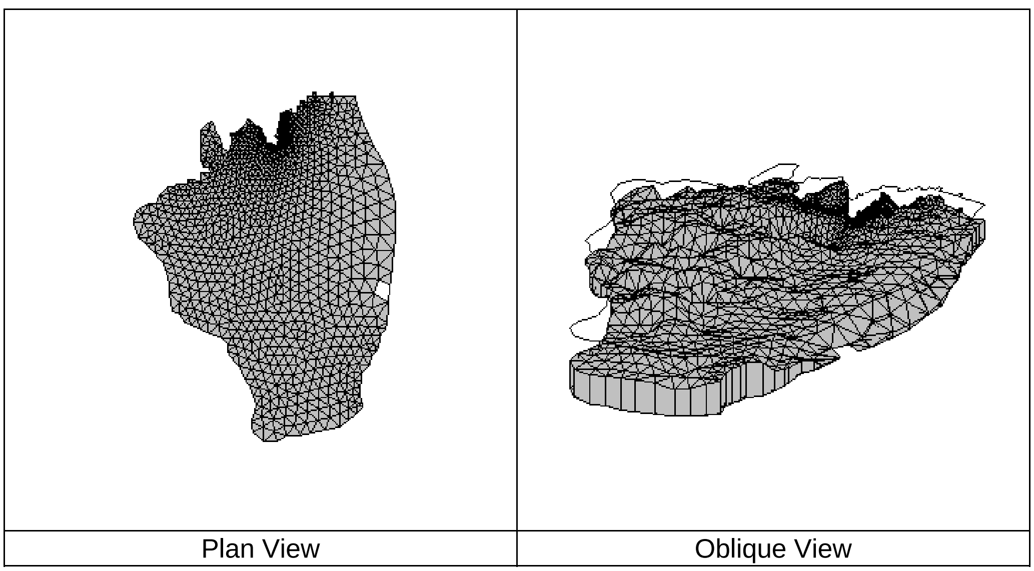

Figure B-8 Sandstone/Peace River layer of 3D mesh created for case study 2