

Applied Research in ArcObjects-based Hydrodynamic Analysis

Authors

Thomas Heinzer
Diane Williams

Abstract

This paper presents an overview of some of the research performed at the Bureau of Reclamation, Mid-Pacific GIS Service Center over the past two years related to the application of ArcObjects technology involving hydrodynamic modeling. Topics include finite element mesh generation and the representation of mesh components as a geometric network, construction of 3D fence diagrams in applications using the HydroGeoSphere model, and interface development between ArcGIS and various hydrodynamic models.

Introduction

From a philosophical perspective, perhaps one of the more important developments in recent years pertaining to hydrodynamic modeling and GIS (Geographic Information Systems) is ESRI's (Environmental Systems Research Institute) move to object-oriented architecture. By implementing COM+ (Component Object Model) technology, GIS features that historically represented real world objects in a feature-attribute environment, can now be described as objects possessing environmental awareness and behavior.

Some of this behavior can be set up through the off-the-shelf ESRI software interface. Much of the more challenging and interesting potential resides in the programmatic manipulation of the COM interfaces which comprise the geometric objects¹ used to model hydrologic/hydraulic systems.

Part 1: Finite Element/Volume Mesh Representation Theory

Geometric Objects

The first example we present is the use of geometric objects in the construction of non-uniform model meshes, such as finite element constructs. Historically in GIS, these mesh types have been represented by flat polygonal topologies that were draped onto various secondary elevation surfaces derived from the node elevation attributes for visualization.

In the current systems, the geometric objects have positions in space resulting from intrinsic properties of the objects themselves, not necessarily resulting from

¹ We use this term here loosely as 'object classes with member functions and inherent geometry'.

external operators. For example, a FEM (Finite Element Model) mesh can be represented as a collection of vector objects whose components are not functions of any external objects.

Messaging

An important feature of object-oriented programming is so-called 'messaging'. We hear the end effect of messaging in coffee shop discussions with statements like "VB is event-driven" or "My object has behavior and yours doesn't." Little known to the coffee patrons or probably anyone else in there is that behind the scenes, what enables this functionality are code sequences that set up 'listeners' between the objects. These listeners enable an object to respond to an event or state change that occurs on another object.

This functionality potentially has important application in the use of geometric objects in model mesh representation. For example, a simple control system whereby a gate is opened and closed as a function of flow in an upstream reach. Put another way, the gate object is listening to the reach object.

Geometric Networks

We are essentially searching for object sets and their interrelationships to build systems that lend themselves to hydrodynamic modeling. In the area of FEM mesh representations, one such object set that has promise is the so-called geometric network. The reasons are as follows:

1. The finite element nodes can be represented by point objects.
2. The finite element edges can be represented by polyline objects.
3. Geometric networks automatically set up a series of messaging functions that impose dynamic connectivity rules which govern the relative motion of the objects.

For example, once a geometric network is established to represent a FEM mesh, a node object which is moved in an edit session will message the element edges to rubberband with it. Additionally, a trigger can be set up on each node object that detects movement which triggers code to snap the node to a surface.

The Triangle FEM Mesh Generator

From a research perspective, we sought a FEM mesh generator that was fairly developed and robust, and that was written in a development platform that supported the COM architecture (compatible with ArcObjects). We decided to use the Triangle program (Shewchuk, 1996) as it was modifiable for institutional research and DHI (Danish Hydraulic Institute) was using it in their research and development. By default, the Triangle program takes flat files as input that specify various geometries representing boundaries, holes, enforced nodes² etc.,

² Specified points that the FEM mesh generator will ensure exists as nodes in the output mesh.

and generates (using certain constraints) output files that represent the constructed FEM mesh.

GIS Integration Methodology in General

The procedure is as follows: use data read directly from a GIS as input instead of flat files. This approach allows the input data to be manipulated in a GIS. Rather than writing the output FEM mesh to flat files, construct the FEM mesh as a set of geometric objects in the GIS and write the construct to a geodatabase.

Additionally, as the mesh segments and nodes are being constructed, make them ZAware³, and assign values to the segment end points and nodes which are obtained from a specified digital elevation surface. This allows direct visualization of the results in the ArcScene (or ArcGlobe) viewer.

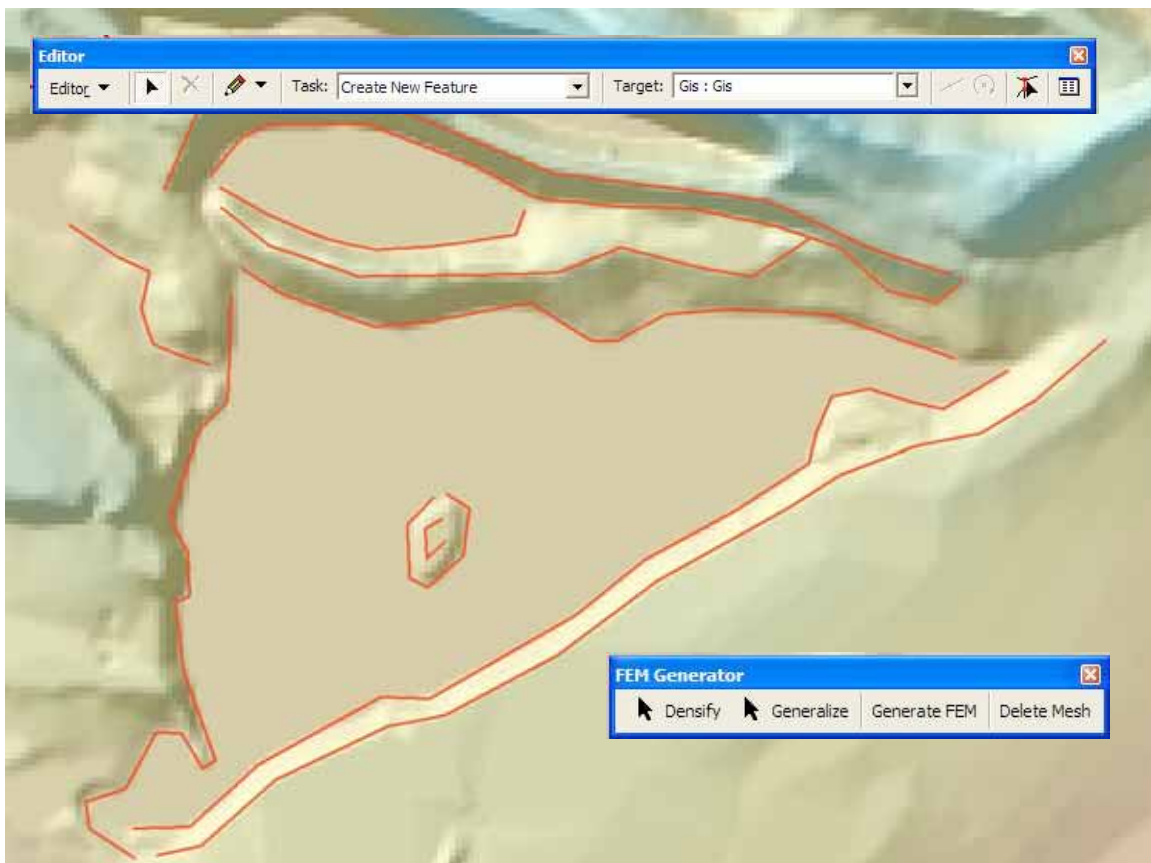


Figure 1. DEM of a project area with enforcement lines digitized in. Note the Densify and Generalize tools on the tool bar. These tools control mesh refinement in the vicinity of the polylines.

³ A COM interface that allows ESRI geometry objects to be intrinsically 3D.

Triangle is written in C++. It was decided to make the design platform Visual Studio C++ and reference the ArcObject libraries to bring GIS functionality into the application. ArcObjects were imported as smart pointers. Triangle uses structures to organize its engine's input and output. The normal file i/o code that populates the arrays associated with these structures were replaced by ArcObjects code that simply talk to the GIS geodatabase files instead of the flat files.

GIS Input to Triangle

This paper will focus on the use of polyline objects as input enforcement nodes to the Triangle engine. We start with a DEM (digital elevation model) of the project area as in Figure 1.

The GIS is then used to digitize polylines into a feature class (the red lines in Figure 1). Custom tools were developed to control the vertex densification of the polylines. After the polylines have been placed, the Triangle code is executed by pressing the 'Generate FEM' button. This generates the FEM mesh and writes it to a geodatabase. Results are shown in Figure 2.

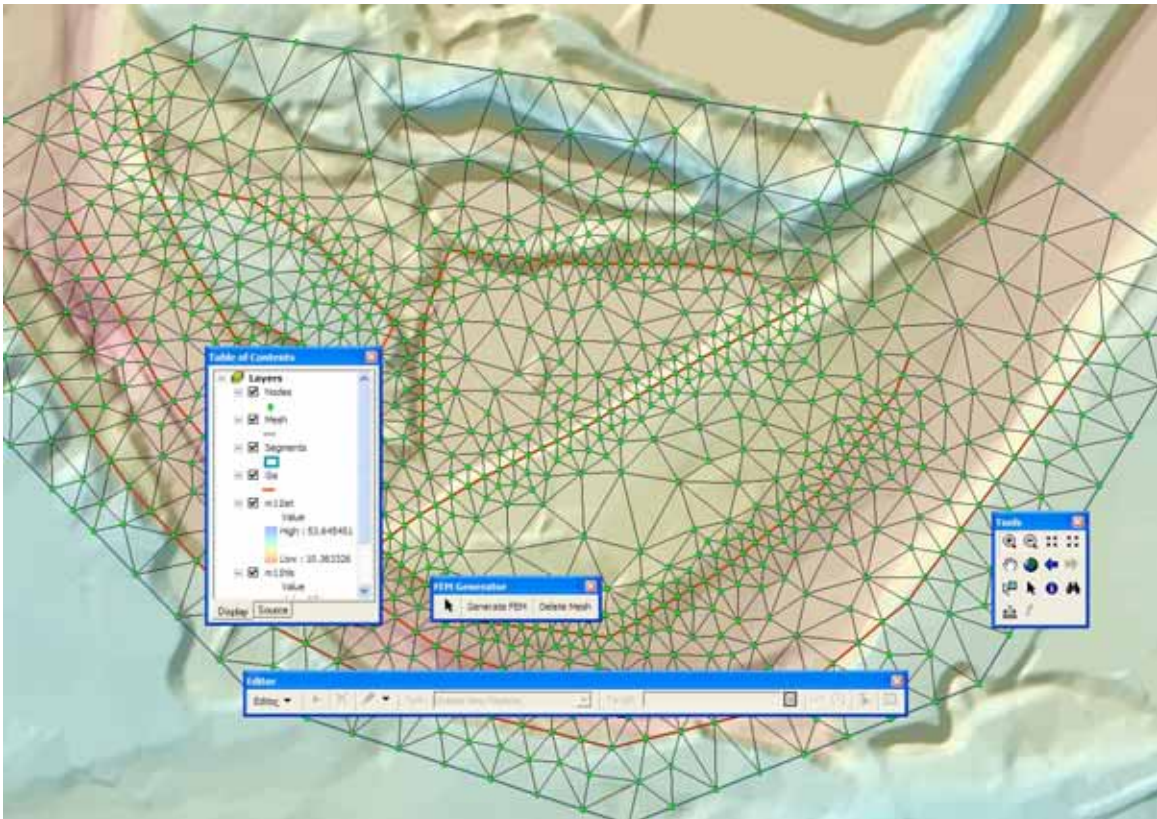


Figure 2. FEM mesh (Triangle output) as GIS objects. The mesh can be deleted, the GIS lines modified, and then regenerated.

The mesh is a 3D construct, so it is immediately viewable in ArcScene as shown in Figure 3. Subsequent code sets take the node and segment feature classes and generate a geometric network. At this point, nodes can be moved in the editor and the mesh lines will 'rubberband' with it. A procedure then resamples the elevation model to correct the node's xyz position. Standard FEM mesh files (node-element files) can be output if desired.

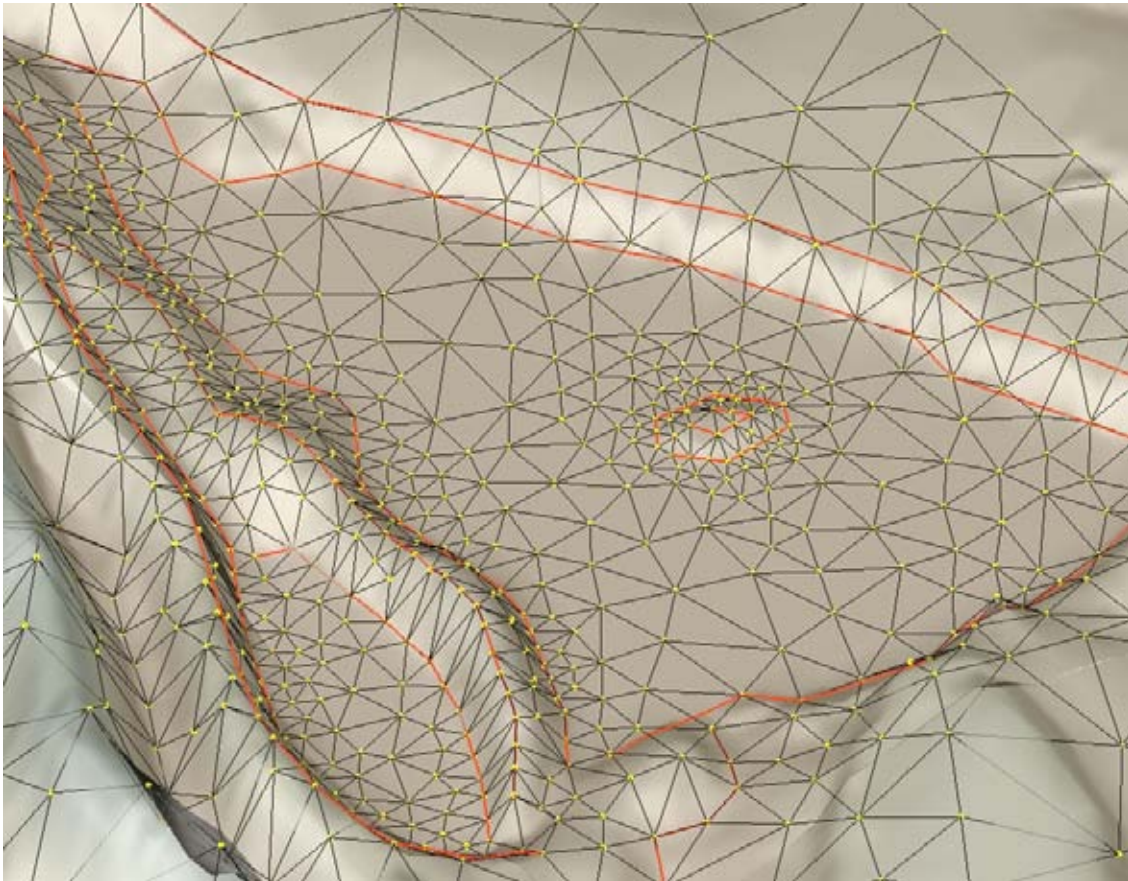


Figure 3. A FEM mesh viewed with its parent DEM.

Part 2: 3D Fence Diagrams

Reclamation is proceeding with modeling studies to prepare for the possible raise of Shasta Dam. One of these models uses HydroGeoSphere (Therrien et al., 2004), which is a groundwater/surfacewater model that solves the system's differential equations as a whole instead of using coupling techniques.

The study area is below Shasta Dam, in the northern Central Valley of California. This part of the paper focuses on the development of 3D fence diagrams from interpretive geologic cross section maps derived from well data. These diagrams are used to construct formation tops that assist in the 3D placement of the FEM mesh layers. Techniques to develop the FEM meshes from these surfaces are similar to the ones described in Part 1 above.

One of the 6 cross sections used in this analysis is shown in Figure 4. The blue vertical lines are the well locations. The original plan was to interpolate the formation tops from the well log information. It was later decided to bring in the interpreted information also. A feature class was created and points were digitized along the formation boundaries. It was agreed that this approach would give a better representation of the actual stratigraphy. The cross section was digitized in x,y space, where x is the distance along the transect and y the actual z (elevation) value of the point. After the points were digitized, an ArcObjects routine looped through each point and sampled the formation polygon above and below it and assigned the formation types to attributes associated with that point. This procedure was performed on all 6 cross sections.

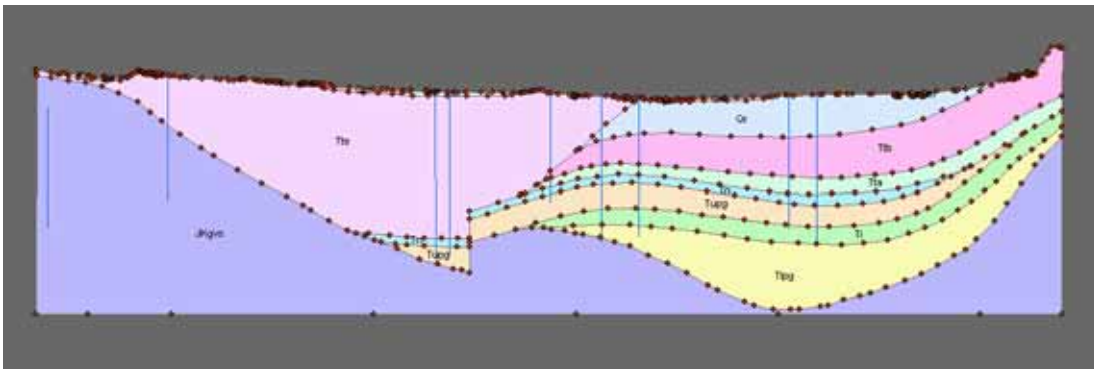


Figure 4. One of the geologic cross sections used in the analysis.

The individual cross section point sets were then brought in to real word x,y,z coordinates. This was achieved by using ArcObjects routines to perform a coordinate transform of the form:

$$\begin{aligned}x' &= \text{QP}(x) \\y' &= \text{QP}(x) \\z' &= y\end{aligned}$$

where x',y' and z' are the real word coordinates and QP(x) is the QueryPoint interface function operating on the cross section feature in plan space. QueryPoint takes the distance passed to it, then starting from a given geometry's 'from node', measures that distance along a curve geometry and returns the point (x,y) in plan space. In addition to the point sets generated for each cross

section, the code also generates polygons whose vertices are composed of 3D point geometries.

Figure 5 depicts the resulting point sets associated with the transects. At this point, it is possible to select out various formation tops for further analysis. Formation surfaces are created using a managed interpolation process.



Figure 5. Transect locations.

Since the polygons representing the formations in each cross section are inherently 3D, the cross sections can be directly visualized in 3D. Figure 6 shows the 3D geologic cross sections and the associated vertex points viewed in ArcScene. A surface level FEM mesh is shown in purple.

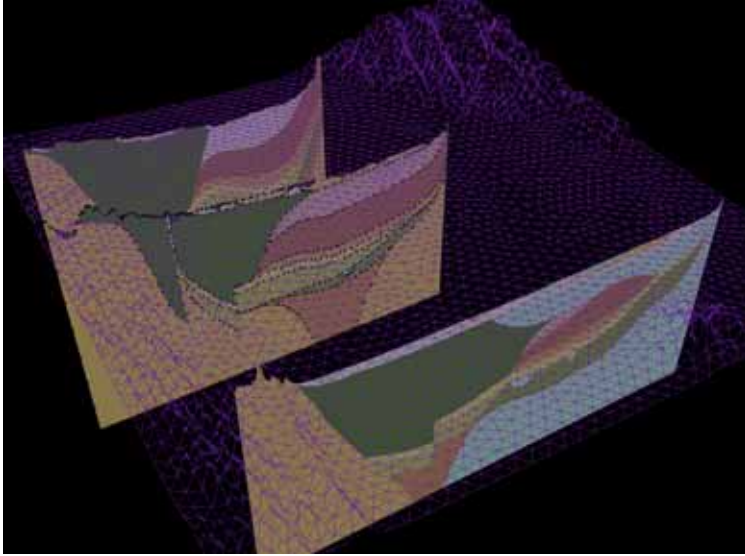


Figure 6. 3D geologic cross sections, vertex points and FEM mesh.
Custom tools were developed to aid in the interpolation of surfaces, which is beyond the scope of this paper. Formation surfaces can then be viewed in conjunction with the cross sections, as shown in Figure 7.

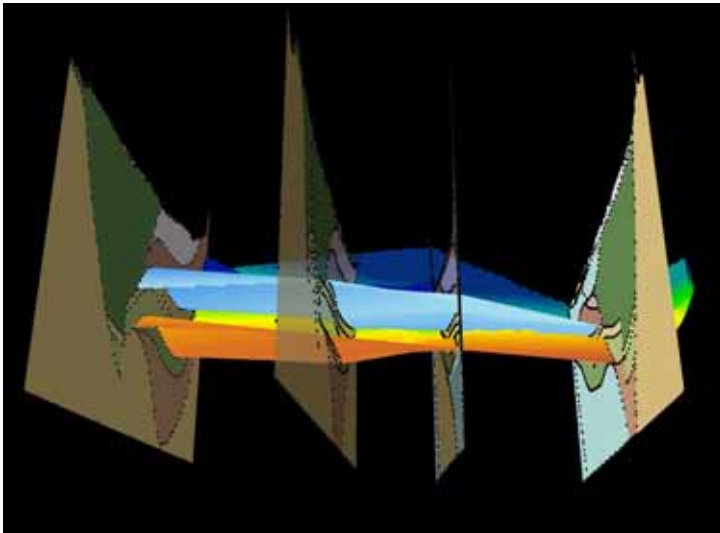


Figure 7. Geologic cross sections and a formation surface.

Finally, the node objects of a FEM mesh can then be associated with a given surface (Figure 8). This procedure is repeated for all of the formations (along with some manual modifications) to generate the model layers. ArcObjects routines as subsequently executed to generate files for model engine input.

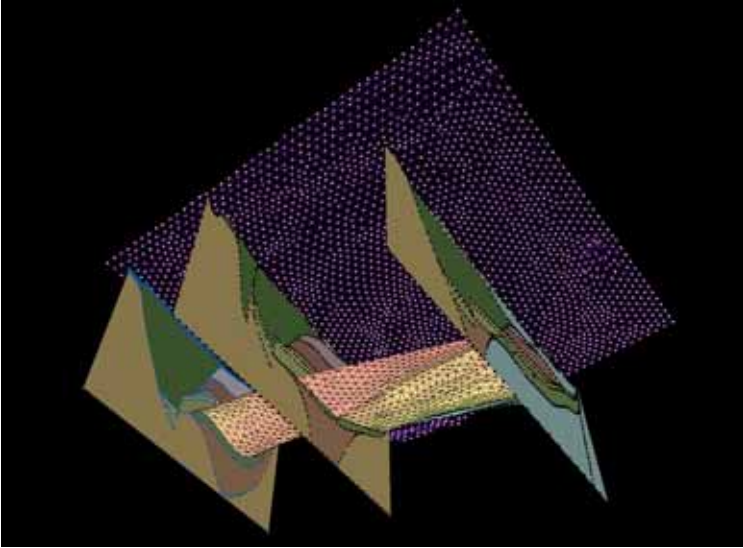


Figure 8. Geologic cross sections with an interpolated formation surface. One FEM mesh is at the ground surface, a second mesh has assumed the elevation values of the interpolated formation surface.

Figure 9 is an interesting GIS view of the fence diagram with a cut away view of the digital elevation model. Mount Shasta is the peak in the background.

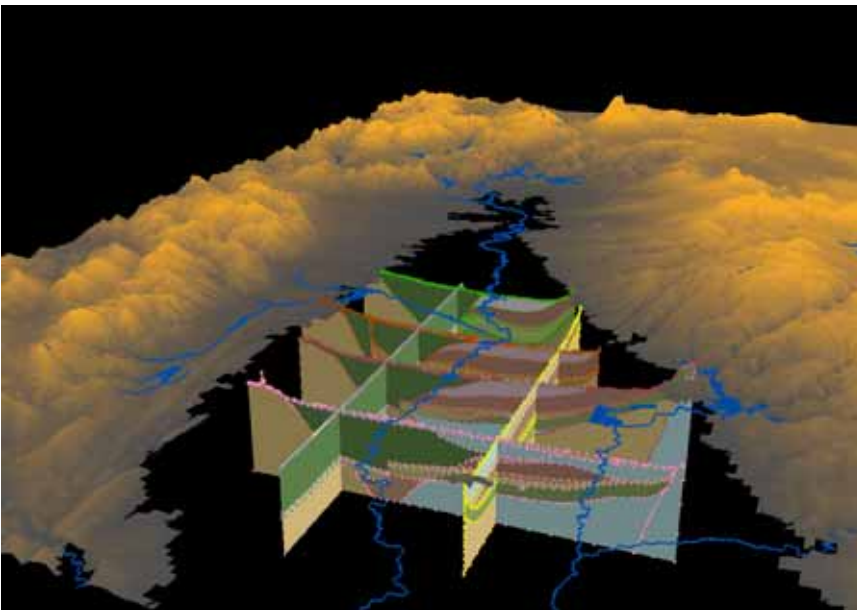


Figure 9. Fence diagram with ancillary data.

Conclusion

This paper presents recent research using geometric objects to represent both model meshes and physical parameters associated with those models. The specific examples were FEM meshes and fence diagrams.

It was found that the use of ArcObjects code to manipulate these data types is essential to performing high level operations. The ability to reference ESRI libraries and utilize the object sets from external modeling platforms is extremely powerful. Bringing ESRI objects into the Triangle executable code is an example of what can be accomplished. The construction of model meshes using 3D geometric objects, and the exploitation of object messaging to form object sets like geometric networks shows promise in various modeling applications.

Data preparation using ArcObjects coding techniques offers new approaches previously unavailable for model design. The methods used to generate the 3D geologic cross sections and subsequent fence diagram are an example of how various ArcObjects coding methods can be employed to automate and assist model parameterization.

Acknowledgements

We would like to thank George Matanga and Lisa Gessford for their assistance and financial support.

Thanks also to Chuck Johnson for his financial support and words of wisdom.

References

Shewchuk, Jonathan, 1996, Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator, First Workshop on Applied Computational Geometry, Philadelphia, PA, ACM, pp. 124-133.

Therrien, R., R.G. McLaren, E.A. Sudicky and S.M. Panday. 2004, HydroGeoSphere: A Three-dimensional Numerical Models Describing Fully-integrated Subsurface and Surface Flow and Solute Transport, Manual (Draft), HydroGeoLogic Inc., Herndon, VA.

Author Contact Information

Thomas J. Heinzer
Manager, MPGIS Service Center
GIS Developer; Chemical Engineer
U.S. Bureau of Reclamation

2800 Cottage Way
Sacramento, CA 95825
916-978-5273
theinzer@mp.usbr.gov

Diane Williams
Principal Engineer
Mid-Pacific GIS Service Center
U.S Bureau of Reclamation
2800 Cottage Way
Sacramento, CA 95825
916-978-5272
diwilliams@mp.usbr.gov