Algorithms for Unstructured Grids

John G. Manchuk and Clayton V. Deutsch

To move forward with the use of unstructured grids in modeling reservoirs, geostatistical methods to characterize them must be available. Much of the underlying theory from geostatistics is not constrained to regular grids. An algorithm to characterize unstructured grids using two popular techniques, indicator simulation and Gaussian simulation, is introduced in this work. Accompanying this is several existing tools to aid in the modeling process: triangulation and tetrahedralization tools to refine unstructured grids for characterization and upscaling purposes; basic equations involved in upscaling arithmetically averaging properties on such refinements; and a flexible software package for visualizing results.

Introduction

Geostatistical tools are developed for regular grids; however, the theory underlying them has no such restriction apart from spectral methods and multiple point statistics. The underlying estimator for practically all modeling algorithms is kriging or a variant of kriging, which has no dependence on grid specification. Only one aspect from geostatistical theory has potential complications, that being volume variance. Statistics such as the probability density function and variogram are dependent on the scale of the data; therefore so are the estimates from kriging or from any estimator. Considering that unstructured grids consist of elements with different volumes, the traditional approach to reservoir modeling cannot be applied. This approach takes various data sources and scales them to the regular grid element volume; therefore, all information integrated into the modeling process is consistent. With unstructured grids, this consistent element volume does not exist.

Theory and algorithms developed in this paper maintain a scale that is small enough to be considered point scale such as core data. Like the traditional approach, all sources of data to be integrated into modeling must be available at the same scale. The geometry of unstructured grids is basically defined by points and connectivity – points may be defined within elements, such as their centroid, or be the vertices of elements, both of which can be populated with reservoir properties. Under this framework, geostatistical methods are used to characterize distributions of points in space. A program that uses two algorithms – sequential indicator simulation and sequential Gaussian simulation – for populating these unstructured points is described in this paper.

Apart from geostatistics, several pre- and post-processing steps are involved. Assuming an unstructured grid design for a reservoir already exists, it must be refined so that detail within elements is accounted for during upscaling. From the refinement a set of points is extracted for property modeling. Points are commonly transformed to a stratigraphic or chronologic space prior to modeling, which is described by (Mallet, 2002 and 2004;Caumon et al, 2004) and not covered further here. After points are characterized upscaling is required. Different methods apply to different reservoir properties. Basic arithmetic averaging of facies and porosity is covered.

Background

In the context of reservoir characterization and flow simulation, concepts discussed in this paper are part of a larger workflow. Existing workflows (Deutsch, 2002) are designed for structured/regular grids, but some re-arrangement and additional components are involved for unstructured grids. This workflow is covered in Paper 112 of this report, which covers grid design in more detail. Sections of the workflow covered in this paper include:

4. Grid refinement
5. Facies modeling
6. Modeling continuous properties such as porosity and permeability
7. Upscaling to effective properties

An additional component covered is visualization. Many of the geostatistical plotting tools available are only for structured grids; however, there are several tools used in other fields that have the capabilities.

Several images of unstructured grids are generated in this paper using ParaView (©2008-09 Kitware, Inc. http://www.paraview.org), which is an open source data analysis and visualization application.

Algorithms

Grid refinement

Unstructured grids consist of a set of elements that in most cases will be far too coarse to approximate with one point. The spatial distribution of properties throughout elements must be accounted for and this is accomplished with refinement. Sampling an element with several points gives a better approximation to the distribution of facies, porosity, permeability and other reservoir properties therein. In existing workflows refinement was not done explicitly. The grid for geologic modeling is designed first and it implicitly discretizes the flow simulation grid, which is typically designed independently or at a later time. In ideal cases, resulting grids are both regular and conforming – the coarse grid for flow simulation and fine grid for geologic modeling align perfectly as in Figure 1. This greatly simplifies upscaling; especially flow based upscaling of permeability.

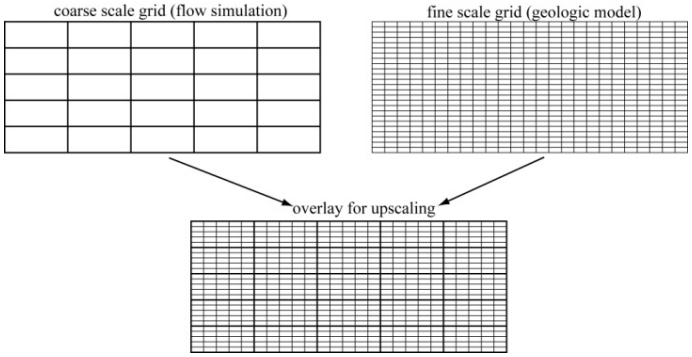

Figure 1: Conforming regular grids

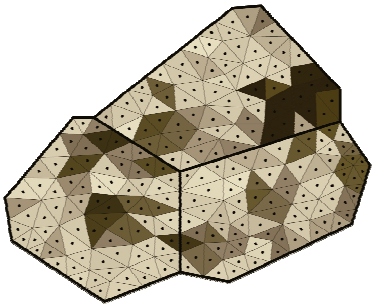

Figure 2: PEBI elements with conforming triangulation

The goal of unstructured grid refinement is identical to the ideal case for regular grids: to achieve conforming grids. Additional conditions to meet include obtaining an unbiased discretization and one that is adequate for upscaling. These concepts were discussed in Paper 111 of this report. To summarize, the discretization is primarily controlled by permeability since it must meet requirements of the flow simulator used in flow based upscaling. Obtaining conformance between grids can be achieved for any linear element using triangulation as shown in Figure 2 or tetrahedralization.

Algorithms for refining unstructured grids were not developed at CCG. Software already exists for triangulation (*Triangle* by Shewchuk J.R., 1996) and tetrahedralization (*TetGen* by Si H., 2006). Both programs generate constrained triangular/tetrahedral grids so conforming grids are achievable and both accept mesh quality and element area/volume constraints to control the degree of refinement. To provide an example, a small radial grid consisting of 180 elements is used. The grid has a radius of 225 meters and covers a depth of 45 meters, which in reality may correspond to a reservoir zone undergoing production. Refinement is done to achieve a trend in element volumes similar to that of the radial grid: decreasing volume with proximity to the well bore. Achieving this with TetGen requires a two stage process. The radial grid is initially refined with a coarse tetrahedralization and then a target volume is assigned to each tetrahedron for a second pass. Volume was set as the $1.5d$ with d being the distance from a tetrahedron barycentre to the well. Results of all stages are shown in Figure 3. The coarse tetrahedral grid has 13,482 elements and the fine grid has 104,553 elements. Once refined, characterization of the grid is carried out using the set of barycentres, which are extracted as the average of the four vertices of each tetrahedron.

A few diagnostics to check the refinement should be done. The quality of tetrahedral elements, number of tetrahedra per unstructured element, and distribution of tetrahedral elements can be checked. TetGen uses the radius-edge ratio quality measure (Miller et al, 1995), the radius being of the circumsphere and the edge being the shortest edge of the tetrahedron. In most cases, smaller ratios indicated higher quality; however, slivers can have small ratios as well and a different measure such as the

aspect ratio may be better. Three tetrahedra from the refined grid in Figure 3 and their quality measures are shown in Figure 4 along with the circumsphere. A histogram of quality measures is shown along with a plot of the number of tetrahedra per unstructured element in Figure 5. The later can be used to ensure elements have received adequate refinement for upscaling. Because the unstructured grid is radial with only four radial cells, this plot shows four distinct unstructured element volumes.

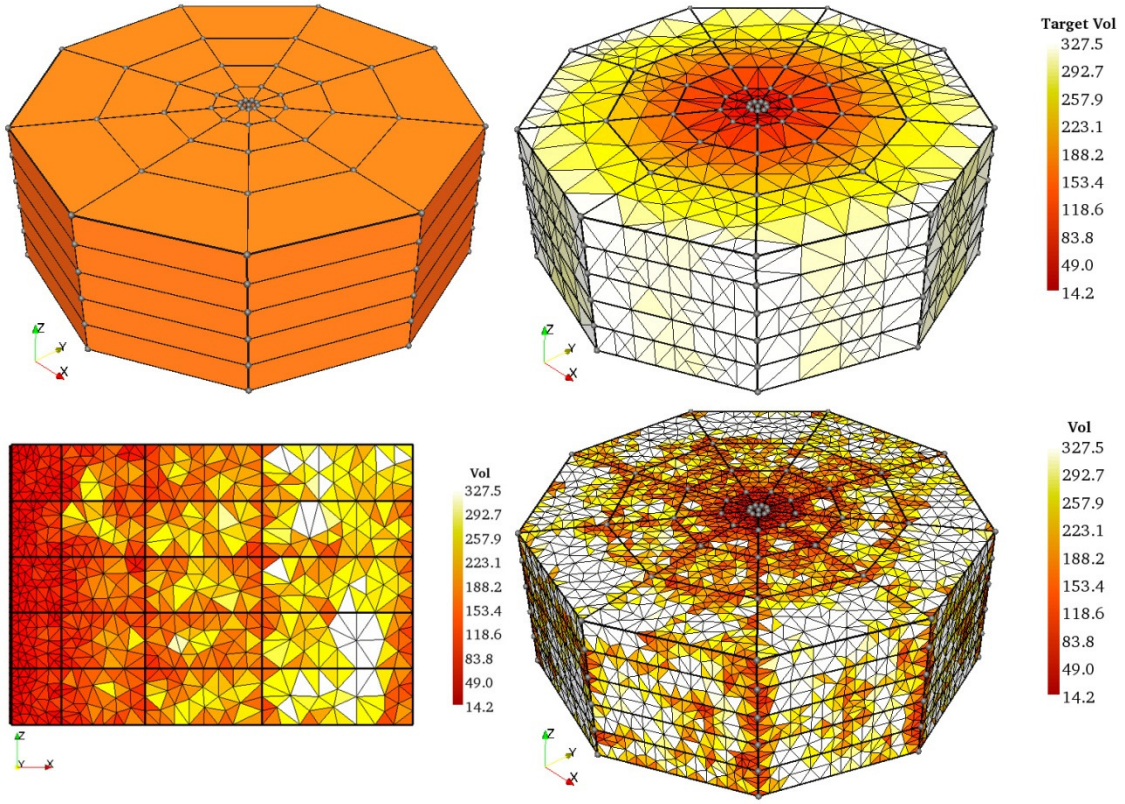

Figure 3: Refinement of a radial grid: clockwise from top left: radial grid; coarse tetrahedral grid; fine tetrahedral grid; *xz* cross section from well bore to full radius.

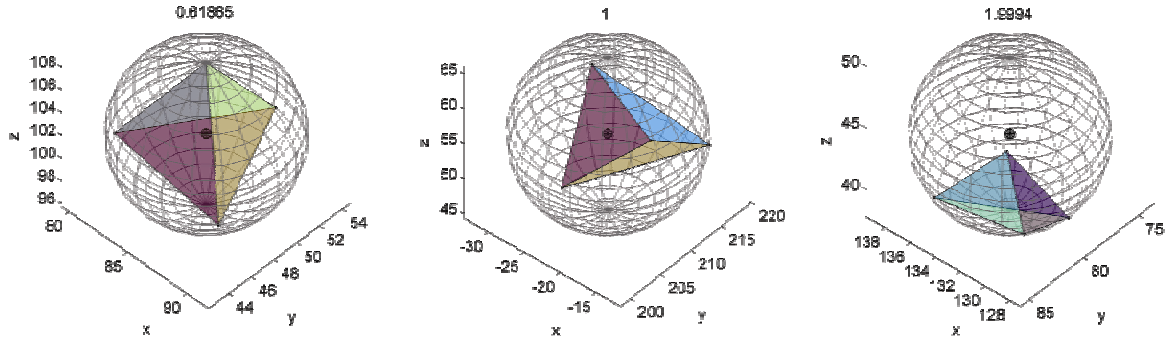

Figure 4: Tetrahedra and their radius-edge ratio quality measures

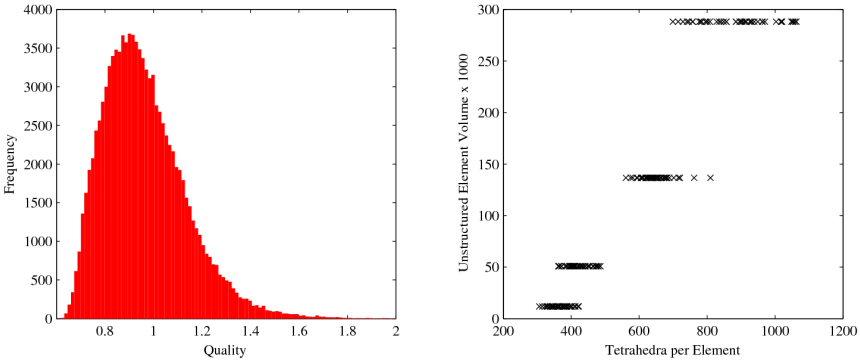

Figure 5: Distribution of quality (left) and number of tetrahedra per unstructured element (right)

Property modeling

Populating grid refinements with reservoir properties is possible with traditional geostatistical methods, but new software is required to handle the irregular distribution of points introduced by an unstructured grid refinement. A program to perform sequential indicator and sequential Gaussian simulation called *psgsim* was developed to handle unstructured points. It is a traditional GSLIB (Deutsch and Journel, 1998) executable with a parameter file. The parameter file is broken into different blocks and global variables, all of which are not required depending on the type of simulation and output required. Blocks and variables are identified by keywords described in Table 1.

All input and output files apart from that for visualization follow the standard geoeas format, where the first line is a description, the second line is the number of colums, the next lines contain column names, and following this is the actual data. In the SIMP block: the nodes file contains the set of 2D or 3D points where simulation is to be done; the path file can be used to specify a particular order for simulation to follow, which may be useful for diagnostic purposes; the output file will contain all point coordinates and resulting simulated values for categorical and continuous variables. Specifying a path is done using a file with 1 column of node numbers.

Table 1: Keywords for *psgsim*

Keyword	Description
#	Identifies a comment in the parameter file. These can be placed anywhere, but if they are found on the same line as and prior to a keyword, that keyword is considered a comment. #UNDEF -999.9 #this whole line is a comment
UNDEF	Specify the value that is associated with undefined numbers in data files and for output files. The default value is -999 if not set UNDEF -999.9
VTK	If this keyword is present in the parameter file, an output file will be generated for visualization in ParaView (©2008-09 Kitware, Inc. http://www.paraview.org), which is described later.
KOUT	If present, a file with the kriging mean and variance is generated
DATA	Indicates a block of parameters for conditioning data file and associated parameters DATA conddata.dat #file 1 2 3 0 #columns for x,y,z,w 3 #number of variables 4 5 6 #column numbers 1 0 0 #variable type (0=continuous,1=categorical) 0 1 1 #transform

Keyword	Description
PROPS	Indicates a block of parameters for indicator (facies) variable proportions, which can be global constants or locally varying. As many proportions or column numbers must be specified as there are facies, which is automatically determined from the datafile in the DATA block if used. Constant case: PROPS 0.3 0.7 #global proportions Locally varying case: PROPS LVM facieslvm.dat #file with locally varying proportions 1 2 #columns
TAILS *n*	Indicates parameters for describing the limits (distribution tails) of continuous variables. The number of tail descriptions, *n*, must be specified. TAILS 2 1 0.0 0.40 #variable, min, max 2 0.001 100.0 #variable, min, max
ILMC *on/off*	Specify if an LMC is to be used for indicator variables. In the *off* case, indicator variables are treated as independent (null cross variograms) and only *n* are required, with *n* being the number of indicators. In the *on* case, $n(n+1)/2$ variograms are needed. ILMC *on* is not implemented in this version.
IVARGS	Indicates parameters for indicator variograms. Multiple variograms are specified by repeating the variogram parameters sequentially as shown IVARGS #All indicator variograms 1 0.0 1 1 1.0 #nst nugget, head-facies, tail-facies, sill 1 1.0 45.0 0.0 0.0 #type cc ang1 ang2 ang3 50.0 30.0 10.0 #range 1 2 3 1 0.0 2 2 1.0 #nst nugget, head-facies, tail-facies, sill 1 1.0 45.0 0.0 0.0 #type cc ang1 ang2 ang3 50.0 30.0 10.0 #range 1 2 3
LMC *on/off*	Identical to ILMC for continuous variables. LMC *on* is not implemented in this version; however, for collocated cokriging, dummy variograms must be entered since the correlation coefficients are derived from them.
CVARGS	Similar to IVARGS, but for continuous variables CVARGS #All continuous variograms 1 0.0 1 1 1 1.0 #nst, nugget, facies, head, tail, corr 1 1.0 0.0 0.0 0.0 #type cc ang1 ang2 ang3 50.0 30.0 10.0 #range 1 2 3
SEARCH	Indicates parameters for a search ellipsoid SEARCH 0.0 0.0 0.0 #Search angles 100.0 100.0 10.0 #Search radii
SIMP	Indicates parameters for simulation SIMP nodes.dat #file with point locations 1 2 3 #columns for x,y,z 20 #max data for conditioning (previous sim + known) 1 #number of realizations 624 #random number seed none #file with specified path nodes.out #file with simulated output
KRIG	Used to specify a kriging type. The current version supports three types: 0 – simple kriging, 1 – ordinary kriging; 2 – collocated cokriging. KRIG 2 #Kriging type

When conditional simulation is done, most of the information is determined from the DATA block including the number of variables, which sets how many continuous variograms are expected; the number of categories is determined from the data file, which sets how many indicator variograms are expected; and continuous data are automatically partitioned by category. However, if no DATA block is found, unconditional simulation is assumed. In this case, multiple categories can be simulated by specifying the PROPS block where the number of categories is determined by either the number of proportions listed in the global case or by the number of column numbers specified in the locally varying case. Multiple continuous variables can be unconditionally simulation by specifying multiple variograms.

For an example, the refined radial grid from Figure 3 is used. Unconditional simulation of three arbitrary facies and two variables was used to generate a realization. Continuous variables were transformed to Weibull and Lognormal distributions to represent porosity and permeability respectively. Resulting facies, porosity, and permeability fields are shown in Figure 6. Facies were assigned global proportions of 0.2 (dark gray), 0.3 (medium gray) and 0.5 (light gray).

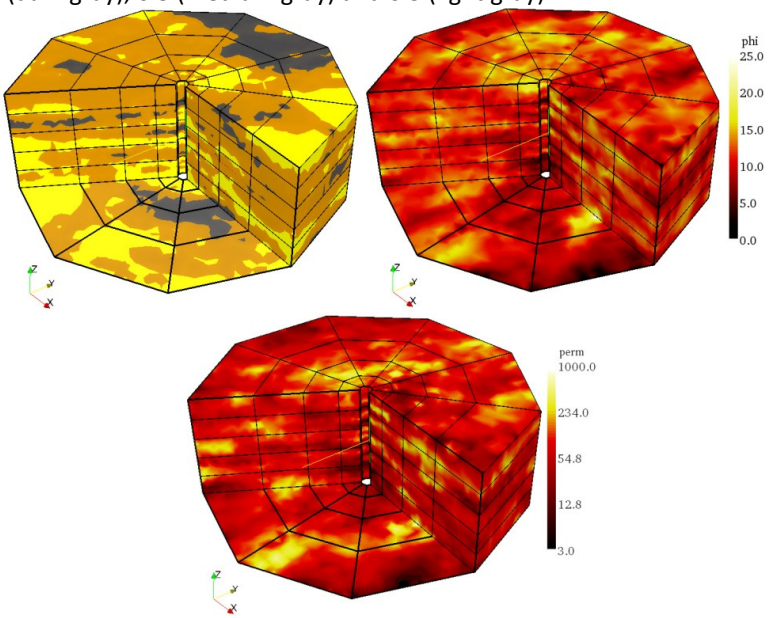

Figure 6: Indicator simulation of three arbitrary facies (top left), porosity (top right) and permeability (bottom)

Upscaling

Different upscaling regimes are needed for different variables encountered in modeling. Facies averages arithmetically and results in a facies proportion vector; porosity averages arithmetically; and permeability scales dynamically according to flow equations. Using tetrahedral refinement, upscaling variables such as facies and porosity is straightforward. The weighted average of the set of tetrahedral elements within an unstructured element is computed via (1), where V is the volume of the unstructured element, w_k is the volume of a tetrahedron, Z_V is the equivalent property and $Z(x_k)$ is the property value assigned to a tetrahedron. In the case of facies, $Z(x_k)$ is converted to an indicator variable and the result is a set of proportions. Tetrahedron volume is calculated using (2), where P_1 to P_4 denotes the spatial locations of the four vertices, • is the dot product and × is the cross product.

$$Z_V = \frac{1}{V}\sum_{k=1}^{n} w_k \cdot Z(x_k) \qquad \sum_{k=1}^{n} w_k = V \tag{1}$$

$$V = \frac{1}{6}\left|(P_1 - P_4)\bullet\left((P_2 - P_4)\times(P_3 - P_4)\right)\right| \tag{2}$$

Determining which tetrahedra are within each unstructured element can be accomplished using TetGen. There is an option to input region attributes, which can be set to an element index for example. A region is any facet bounded volume and is specified in TetGen using a point that is within the region such as an

unstructured element centroid. All tetrahedra within a region are assigned the region attribute. More thorough explanation and usage can be found in the user's manual (Si, 2006). Upscaled facies proportions and porosity are shown in Figure 7. For comparison purposes with Figure 6, the average facies code was computed and scaled with a similar color range so that elements having mostly one type of facies are shaded accordingly.

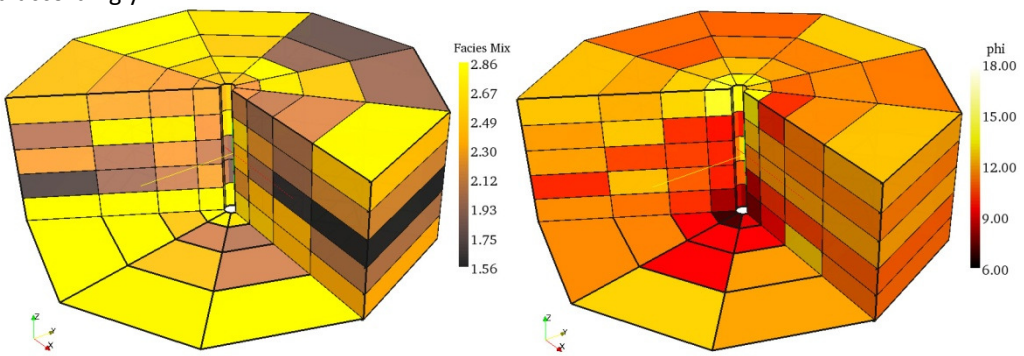

Figure 7: Upscaled facies code (left) and porosity (right)

Visualization

Actively seeing results of geostatistical modeling and upscaling is an important component to any application. Areas of interest, problematic areas, and errors can often be picked up quickly by visually exploring a model. A software package called ParaView was used throughout this paper for visualizing tetrahedral grids and radial grids composed of hexahedrons, although a much wider variety of grids and geometries could be explored. Grids can be loaded into this program from text files not much different from Eclipse keyword-based grdecl files. The layouts of both grids in this paper are explained based on a grid file involving a single tetrahedron and hexahedron in Table 2.

The first line of the file defines the version of VTK (Visualization Toolkit); the following two lines contain a file description and data type descriptor, which can be either ASCII or BINARY. The fourth line involves the keyword DATASET. Several built in datasets are possible including STRUCTURED_POINTS, STRUCTURED_GRID, POLYDATA, and UNSTRUCTURED_GRID. For an unstructured grid, line 5 contains keyword POINTS followed by the number of points *n* to be used in the grid, 9 in this case, and the data type. Lines 6 to 14 contain the 3D points which cycle by (x_0, y_0, z_0), (x_1, y_1, z_1), to (x_{n-1},y_{n-1},z_{n-1}). Line 16 defines the unstructured CELLS, which is followed by the number of cells *m* and the total number of records involved. In this case 2 cells, one hexahedron with 8 points and one tetrahedron with 4 points are used, so the number of records is 8 + 4 + 2 = 14. Cells on lines 17 and 18 are defined by the number of points first followed by the point indices, which are indexed from zero.

VTK unstructured grids can consist of several different cell types, which are defined on line 20 using the keyword CELL_TYPES followed by the number of cells. Cell types range from a vertex (type 1) to a quadratic hexahedron (type 25). The hexahedron and tetrahedron in the example are types 12 and 10 respectively. Data can be defined at the vertex locations of the elements or as cell data. The later is the typical case in geostatistical modeling. Line 24 indicates CELL_DATA followed by the number of cells and line 25 describes the type of data (SCALARS), variable name (z), data type (float) and number of tuples. The LOOKUP_TABLE on line 26 is for color mapping. Scalar values on lines 27 and 28 in this example have been set to the z coordinate of the centroid of each cell. The resulting 2 cell grid is shown in Figure 4. Note the arrangement of vertices for the hexahedron must be specified in the order shown.

Table 2: Unstructured grid file structure for ParaView

```
1   # vtk DataFile Version 2.0
2   example grid
3   ASCII
4   DATASET UNSTRUCTURED_GRID
5   POINTS  9 float
6    0.000  0.000  0.000
7    7.000  0.000  0.000
8   10.000 10.000  0.000
9    0.000 10.000  0.000
10   0.000  0.000 10.000
11  10.000  0.000 10.000
12  10.000 10.000 10.000
13   0.000  7.000 10.000
14   5.000  5.000 14.000
15
16  CELLS 2 14
17  8 0 1 2 3 4 5 6 7
18  4 4 5 6 8
19
20  CELL_TYPES 2
21  12
22  10
23
24  CELL_DATA  2
25  SCALARS z float 1
26  LOOKUP_TABLE default
27  5.000
28  12.000
```

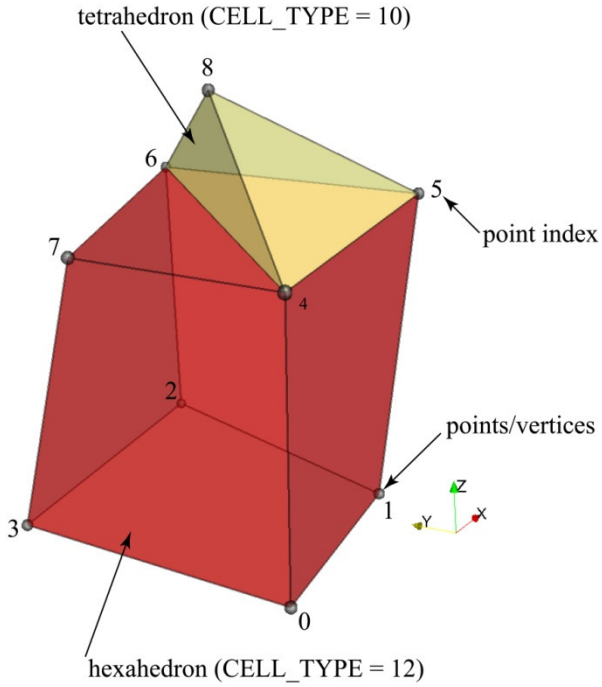

Figure 8: Unstructured grid from ParaView based on Table 2

Conclusions

Geostatistical modeling on unstructured grids can be accomplished with the aid of several existing tools and with a new version of sequential indicator and Gaussian simulation for unstructured point sets. Grid refinement is accomplished in 2D using triangulation and a program called *Triangle* and in 3D using tetrahedralization and a program called *TetGen*. Characterization of the refinement is done with *psgsim*. A program called *ParaView* for visualization and the basic file structure for unstructured grids was also covered. Several tools for file conversions are required: unstructured grids and output from Triangle and TetGen must be reformatted for *ParaView*; results from psgsim need to be appended to ParaView data files.

References

Caumon G, Grosse O, and Mallet J-L (2004) High resolution geostatistics on coarse unstructured flow grids. In O. Leuangthong and C.V. Deutsch (eds.), Geostatistics Banff, 703-712

Deutsch CV (2002) Geostatistical Reservoir Modeling. Oxford University Press, 384

Deutsch CV and Journel AG (1998) GSLIB: geostatistical software library and user's guide. Oxford University Press, 384

Shewchuk JR (1996) Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In Applied Computational Geometry: Towards Geometric Engineering (Ming C. Lin and Dinesh Manocha, eds.), vol. 1148 of Lecture Notes in Computer Science, 203-222, Springer-Verlag, Berlin.

Si H (2006) TetGen: A quality tetrahedral mesh generator and three-dimensional Delaunay triangulator. Numerical Mathematics and Scientific Computing Research Group, Weierstrass Institute for Applied Analysis and Stochastics

Si H (2006) TetGen: A quality tetrahedral mesh generator and three-dimensional Delaunay triangulator, Version 1.4 user's manual. Numerical Mathematics and Scientific Computing Research Group, Weierstrass Institute for Applied Analysis and Stochastics

Mallet J-L (2002) Geomodeling. Oxford University Press, 624

Mallet J-L (2004) Space-time mathematical framework for sedimentary geology. Mathematical Geology, 36(1), 1-32

Miller GL, Talmor D, Teng S-H, Walkington N, Wang H (1995) A Delaunay based numerical method for three dimensions: generation, formulation, and partition. Proceedings of the 27th Annual ACM Symposium on the Theory of Computing, 683-692