

## FINITE ELEMENT MESH GENERATION FOR NUMERICAL SIMULATION OF HYDROELECTRIC POWER PLANT RESERVOIR FILLING

### **Abrão Macandi Dongala**

Universidade do Estado do Rio de Janeiro, UERJ, Faculdade de Engenharia.  
Rua São Francisco Xavier, 524 – 20550-013 – Rio de Janeiro, RJ, Brasil.  
[abramado1@hotmail.com](mailto:abramado1@hotmail.com)

### **Leon Matos Ribeiro De Lima**

Universidade do Estado do Rio de Janeiro, UERJ, Faculdade de Engenharia.  
Rua São Francisco Xavier, 524 – 20550-013 – Rio de Janeiro, RJ, Brasil.  
[leonmatos@hotmail.com](mailto:leonmatos@hotmail.com)

### **Norberto Mangiavacchi**

Departamento de Engenharia Mecânica, Universidade do Estado do Rio de Janeiro, UERJ.  
Rua São Francisco Xavier, 524 – 20550-013 – Rio de Janeiro, RJ, Brasil.  
[norberto.mangiavacchi@gmail.com](mailto:norberto.mangiavacchi@gmail.com)

### **Cássio Botelho Pereira Soares**

Departamento de Engenharia Ambiental, Furnas Centrais Elétricas, S.A.  
Rua Real Grandeza, 219 – Rio de Janeiro, RJ, Brasil.  
[cassiobp@furnas.com.br](mailto:cassiobp@furnas.com.br)

**Abstract.** *This paper presents the application of the Finite Element Method on the numerical simulations of hydroelectric power plant reservoir filling, focusing on two-dimensional mesh generation. The problem of the heterogeneity of the data sets in hydrodynamic simulations, which differ in precision, type, structure, among other factors, is addressed. The methodology employed in the 2D mesh generation is shown, as well as its relation to the construction of the 3D terrain models. Four different available representations of the terrain are presented, with methods of conversion from one representation to another. The mesh triangulation is made from a set of Cartesian points, using Delaunay triangulation. The present methodology is part of a larger project, based on the object oriented programming paradigm, which has been an effective improvement in this kind of software development.*

**Keywords:** *mesh generation, finite element, object-oriented programming, hydrodynamic simulations.*

### **1. Introduction:**

The filling process of a hydroelectric power plant reservoir can cause several problems, such as under water biomass decomposition, which may compromise the quality of all the water affected by the reservoir. In order to avoid such consequences, a precise analysis is necessary, and studies based on numerical methods are a powerful tool that allows a reliable prediction analysis. This is an important issue given that water supply and water quality is an increasing problem that sooner or later we will have to face.

Many projects of energy generation for hydroelectric power plants have been elaborated and concluded. As the number of such projects increases a larger body of scientific studies on the ambient and socio-economic impacts is necessary. Especially in Brazil, where the hydrographic conditions are an advantage for the exploitation of hydroelectric reservoirs, the area covered by them is very extensive, and so the problems they bring. We can minimize these problems by obtaining information that provides a more consistent forecast capability of the consequences of a hydrographic basin flooding. A useful tool to gather forecast information about the consequent ambient impacts is the numerical simulation.

The main contribution of this work is a 2D mesh generation procedure for the application on numerical simulation of the hydrodynamic phenomena present in a hydroelectric reservoir. The presented system of mesh generation is part of a larger project of studies and simulations on the ambient impacts related to the flooding of a certain region for hydroelectric power plant construction. There are other similar works on hydrodynamic simulations, which make use of two-dimensional meshes, such as FESWMS (Finite Element Surface Water Modeling System). The difference to the method we are presenting is that, although the focus of this paper is grid assembling for applications on 2DH modeling for shallow water problems, the generation of 2D mesh is also employed on three-dimensional mesh generation, for 3D fluid flow problems, which is one of the simulation models of this project.

The simulations results are obtained basically from the solution of the momentum conservation equations for  $x$  and  $y$  directions and the continuity equation, utilizing the Finite Element Method (FEM) for the mathematical modeling.

One of the input data is the terrain information, and this project has among its objectives to give a major flexibility on the terrain representation options, offering conversion techniques. Objects and methods related to ground representation belong to the Terrain module, which prepares the data for the mesh generation, in the Finite Elements Mesh (FEMesh) module.

This structure is part of the software engineering environment of this work, and it is based on the object oriented concept. MATLAB was chosen for the development work because it is a high-level language directed to mathematical applications that provides all the features of a traditional programming language, including OOP.

## 2. Simulation environment

The simulation environment is composed of several modules that encapsulate specific tasks in the simulation process. A brief explanation over each module is shown below.

### 2.1. Terrain

Here the terrain data sets are handled. This is where the user defines the domain of simulation.

### 2.2. FEMesh2d

This module is responsible for 2D and 3D mesh assembling.

### 2.3. ShapePhyto

The vegetation map of the terrain in study is manipulated in this module, as well as the phyto-physiognomy data.

### 2.4. Decay

This module is responsible for the biomass decay calculations. It quantifies the organic mass transferred to the water, which impacts on the chemical elements percentages present in the reservoir water.

### 2.5. Simulator

In this module all the information handled in the other modules is collected for the simulation itself. This module also has a strong coupling with the Decay module: on one hand, the Decay provides biomass quantities present in the water for the transport equations in the Simulator module, and, on the other hand, the decay coefficients are function of transport parameters, like temperature, pressure, oxygen concentration etc.

The focus of this paper is the Terrain and FEMesh modules.

The modules can be accessed using a high level script language, based on MATLAB, which gives great power and flexibility to the user. Alternatively, the user can employ a simple Graphic User Interface (GUI) to perform common tasks. Figure 1 shows a simplified scheme of the whole system of simulation, where the data flux between the modules is represented by the arrows.

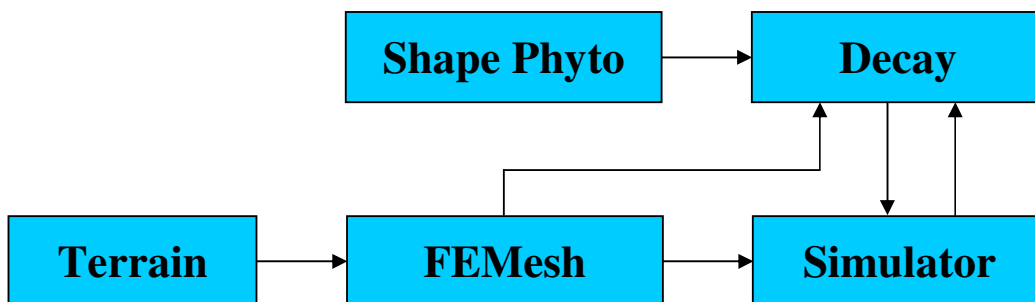


Figure 1. Data flux between the system's modules.

The FEMesh generator module is closely related to the terrain modeling module. In fact, the triangular mesh is built using topological data provided by the Terrain module. The terrain can be represented by a set of points (named CloudTerrain), a cartesian mesh of points (RasterTerrain), a set of level-curves (ShapeTerrain), or a triangular mesh (MeshTerrain). The mesh triangulation is made from the CloudTerrain representation, using Delaunay triangulation. After this stage, the mesh can be improved employing a refining algorithm, and information regarding boundary

conditions is included in the data structure, to prepare for the application of the mesh in the Finite Elements Method, in the Numerical Simulations Module.

### 3. Governing equations

In this work the two-dimensional mesh generation is analyzed for the case of shallow water hydrodynamic simulations. In this kind of problem it is convenient to consider hydrostatic pressure, integrated on the vertical direction ( $z$  axis, in a cartesian coordinates system), and average velocities only in the horizontal plane ( $x$  and  $y$  directions), which is the 2DH modeling method.

The governing equations for the hydrodynamic circulation were described by Rosman (2001), in which the 2DH technique is employed to solve the kind of problem this work is focused on. More details about 2DH hydrodynamic modeling can be found in this reference. Here, the main results will be briefly presented.

For each time step the 2DH simulator determines the vertically averaged velocity, on the  $x$  and  $y$  directions,  $U(x,y,t)$  and  $V(x,y,t)$  respectively, and the elevations on the free surface,  $z=\zeta(x,y,t)$ . The three necessary equations to determine three unknowns of hydrodynamic circulation in 2DH flow, integrated on the vertical direction,  $\zeta$ ,  $U$  and  $V$ , are represented below:

2DH equation of momentum conservation for a flow integrated on the vertical, on the  $x$  direction:

$$\frac{\partial U}{\partial t} + U \frac{\partial U}{\partial x} + V \frac{\partial U}{\partial y} = -g \frac{\partial \zeta}{\partial x} + \frac{1}{\rho_0 H} \left( \frac{\partial(H \tau_{xx}^-)}{\partial x} + \frac{\partial(H \tau_{xy}^-)}{\partial y} \right) + \frac{1}{\rho_0 H} (\tau_x^s - \tau_x^B) + 2\phi \sin \theta V \quad (1)$$

Where  $H$  is the distance from the bottom to the surface at each point, and  $\rho_0$  is the water density. 2DH equation of momentum conservation for a flow integrated on the vertical, on the  $y$  direction:

$$\frac{\partial V}{\partial t} + U \frac{\partial V}{\partial x} + V \frac{\partial V}{\partial y} = -g \frac{\partial \zeta}{\partial x} + \frac{1}{\rho_0 H} \left( \frac{\partial(H \tau_{xy}^-)}{\partial x} + \frac{\partial(H \tau_{yy}^-)}{\partial y} \right) + \frac{1}{\rho_0 H} (\tau_y^s - \tau_y^B) - 2\phi \sin \theta U \quad (2)$$

Continuity equation integrated along the vertical direction:

$$\frac{\partial \zeta}{\partial t} + \frac{\partial UH}{\partial x} + \frac{\partial VH}{\partial y} = 0 \quad (3)$$

The terms of equation (1) can be explained as follows:

$\frac{\partial U}{\partial t} \rightarrow$  represents the local acceleration of the flow, at a given position, or, in other words, the temporal variation rate of the 2DH momentum flux per mass unit. In permanent flows, this term is zero.

$U \frac{\partial U}{\partial x} + V \frac{\partial U}{\partial y} \rightarrow$  represents the advective acceleration of the flow, on a given instant. These terms represents the spatial variation rate of the 2DH momentum flux on the  $x$  direction per mass unit. In uniform flows, these terms are zero.

$-g \frac{\partial \zeta}{\partial x} \rightarrow$  represents the resultant hydrostatic pressure on the  $x$  direction (pressure gradient). Due to the water surface declivity on the  $x$  direction, as indicated by the negative sign, the flow tends to move from places where the water level is higher to places where the water level is lower.

$\frac{1}{\rho_0 H} \left( \frac{\partial(H \tau_{xx}^-)}{\partial x} + \frac{\partial(H \tau_{xy}^-)}{\partial y} \right) \rightarrow$  represents the resultant of the 2DH turbulent dynamic tensions in the flow. For example, these terms are caused by horizontal vortices in recirculation zones.

$2\phi \text{sen}\theta V \rightarrow$  represents the coriolis force. This term is present because the referential is moving with the Earth surface.

$\frac{1}{\rho_0 H} (\tau_x^s) \rightarrow$  represents the wind tension on the free surface per mass unit. If the wind is on the same direction of the flow, this term will speed up the flow; otherwise, if the wind is on the opposite direction, it will slow down the flow.

$\frac{1}{\rho_0 H} (-\tau_x^B) \rightarrow$  represents the friction tension on the bottom of the flow per mass unit. As indicated by the negative sign, this term always tends to deaccelerate the flow. It is always opposite to the flow.

Evidently, the meaning of the similar terms in the equation (2), of the momentum conservation on the y direction is analogous.

For the 2DH continuity equation (3), we have:

$\frac{\partial(\zeta + h)}{\partial t} = -\frac{\partial HU}{\partial x} - \frac{\partial HV}{\partial y} \rightarrow$  represents the variation of the water column height ( $\zeta+h$ ) in time as result of the effective fluxes through the water column on the x and y directions, respectively  $\frac{\partial HU}{\partial x}$  and  $\frac{\partial HV}{\partial y}$ .

The water source is given only by affluents, where the boundary conditions are prescribed velocity. The points of outflow are treated with prescribed height  $\zeta$  boundary conditions, and on the reservoir margins the velocity is equal to zero. The water source due to precipitations is accounted for in the affluents inflow velocities.

The solution technique chosen is the Finite Element Method with Galerkin variational formulation. The method is applied on a mesh of Delaunay triangles, with the domain represented by mini-elements – pressure on the vertices and velocity on the vertices and centroid. For the convective term, the semi-lagrangian formulation was used, and the projection method was chosen to decouple the velocity and pressure calculations.

The discretization produces two algebraic linear equation systems in each time iteration. To solve these systems, the preconditioned conjugated gradient method is employed, using an incomplete Cholesky preconditioner.

#### 4. Terrain module

This is the module that deals with the terrain topological data, which can be manipulated by four alternative basic representations (Raster Terrain, Cloud Terrain, Shape Terrain and Mesh Terrain representations). The Terrain module has a class which is responsible for each representation type, described as follows:

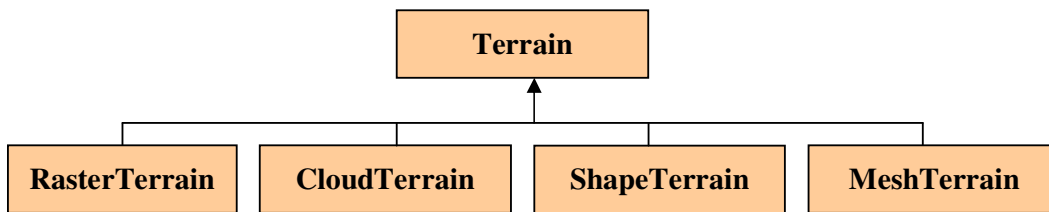


Figure 2. Terrain module classes.

##### 4.1. RasterTerrain

The terrain is represented by a set of cartesian coordinates  $(x,y,z)$ , where each one is a surface point of the terrain. The representation is based on structured quadrilateral grids. It can be constructed, for instance, using a disordered set of

points, defining the intervals between each coordinate, and interpolating the z values. This class handles Digital Terrain Model (DTM) and Digital Elevation Model (DEM) raster files, and converts MeshTerrain models to RasterTerrain.

#### 4.2. ShapeTerrain

This class represents terrain models by a set of level curves. It handles Shapefile struct files. Figures 7 and 8 show a top view visualization of a Shape Terrain. Visualizations can be also performed employing different perspective views.

#### 4.3. CloudTerrain

The terrain is represented by a “cloud” of disordered cartesian points, with position given by x, y and z coordinates. This class converts RasterTerrain and ShapeTerrain into CloudTerrain.

#### 4.4. MeshTerrain

The terrain is represented by a two-dimensional triangular mesh, obtained by a Delaunay triangulation algorithm (Shojaee, 2006; Hao,1997). Figure 9 shows a Mesh Terrain representation. The triangular grid can be generated on the terrain from a set of disordered cartesian points. It converts CloudTerrain into MeshTerrain.

Figure 3 shows the relation between the constructor classes of the terrain models and the FEMesh2D class, which belongs to the FEMesh module.

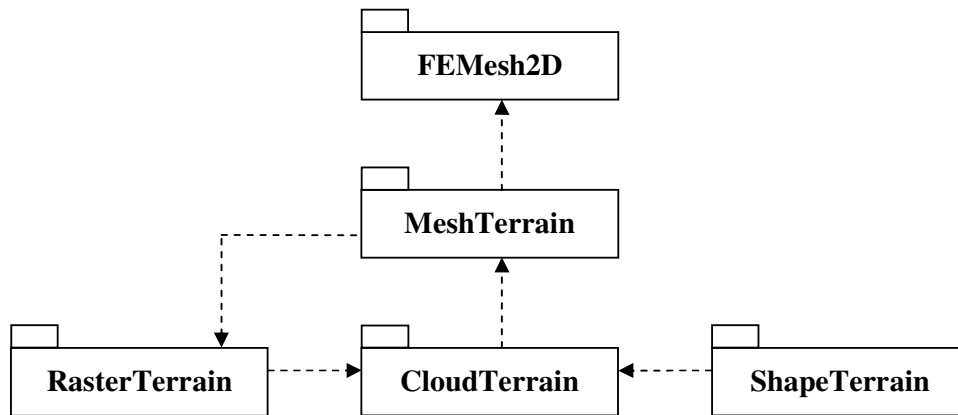


Figure 3. Package diagram showing the relation between Terrain classes and FEMesh2D class.

The availability of four possible terrain representations is a way of increasing the user’s flexibility in the data input for the simulation. However, the basic terrain type used by the simulation is the MeshTerrain, because it is this the terrain model employed by the FEMesh module to generate the mesh that will be used by the Simulator module. In fact, the class that builds 2D meshes is the MeshTerrain.

### 5. FEMesh module

This is the module that creates the data structures required for the Finite Element Mesh representation. This module includes three basic classes, as shown in figure 4, described as follows:

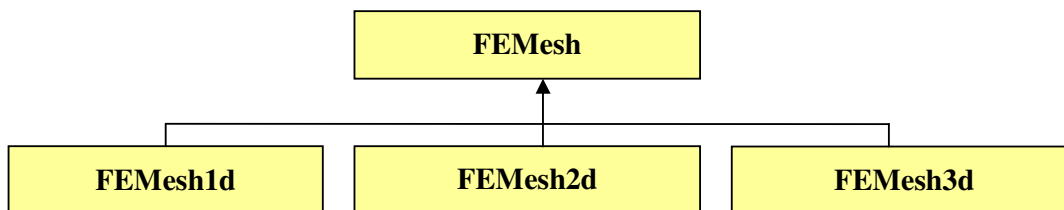


Figure 4. FEMesh module classes.

### 5.1. FEMesh1d

Represents 1d FEM Meshes. It can be employed for 1d simulations, or to collect information on 1d geometric features, such as reservoir boundaries in 2d simulations.

### 5.2. FEMesh2d

Represents 2d FEM Meshes. It can be employed for 2d simulations, or to collect information on 2d geometric features, such as reservoir boundaries in 3d simulations.

### 5.3. FEMesh3d

Represents 3d FEM Meshes. It can be employed for 3d simulations.

## 6. Mesh generation process

The mesh utilized for the terrain representation is a mesh of triangular elements. A justification for this choice is the fact that triangular grid models allow a more reliable representation of the superficial distribution in regions with regions of large altitude gradients. According to Trisiripisal (2003), since each triangular patch contains redundancy in information, it can be used to indicate the characteristics of a region. This advantage is very useful in computer vision and pattern recognition applications to describe the information present in the image.

Another advantage of using triangular meshes is a large reduction in data for processing, and this is very useful in many applications. For example, instead of transforming the entire data, only the representative vertex data in the triangular mesh is required in the calculation. This can significantly speed-up the computation. Delaunay triangulation is a good technique, from the point of view of its approximation properties in combination with finite elements, because it maximizes the minimum angle compared to any other triangulation of the same set of points.

### 6.1. Delaunay triangulation

In a few words, given a set of data points, the Delaunay triangulation is a set of lines connecting each point to its natural neighbors. The Delaunay triangulation is related to the Voronoi diagram – the circle circumscribed about a Delaunay triangle has its center at the vertex of a Voronoi polygon. A Voronoi polygon about point  $P$  is the region that is closer to this point, or as close to, than to any other point. The Voronoi or Dirichlet tessellation is made up from these Voronoi polygons and the Delaunay triangulation is the dual of the Dirichlet tessellation. The next figure shows a set of Delaunay triangles and Voronoi polygons, both made up from a set of disordered points.

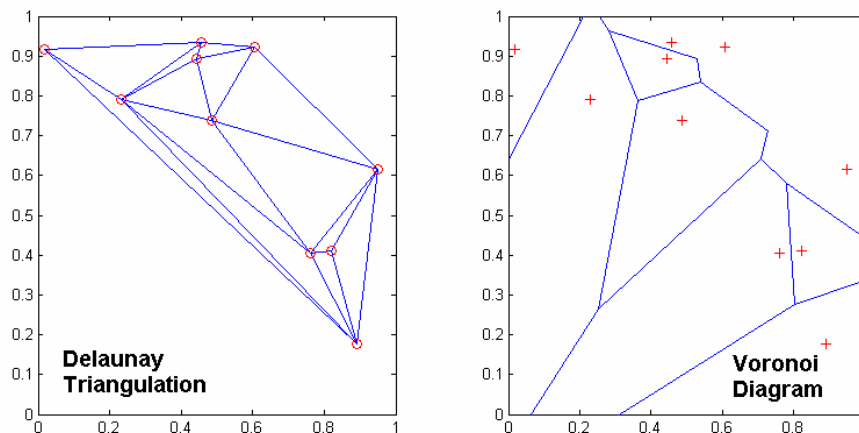


Figure 5. Mesh of Delaunay triangles (left box) and Voronoi polygons diagram (right box).

### 6.2. Mesh generation routine

The first step on the mesh assembling is to convert the terrain format into a CloudTerrain representation. We need this conversion because the Delaunay triangulation is made on a set of disordered cartesian points. After the triangulation, the mesh is assembled, and we can proceed to the next step, in the FEMesh module. Now there are some

adjustments to improve the grid. The first step is the elimination of the elements above the maximum height, which is an essential routine. The other one is a refining option, consisting in the insertion of a point in the middle of edges larger than an arbitrary size limit. All These tasks are implemented by the **FEMesh2D** class.

There's also another alternative way for generating the mesh of Delaunay triangles. Instead of sending a cloud of cartesian points for the Delaunay triangulation, this routine starts from a given domain boundary definition, and selects the interior points, necessary to build the triangular elements. The advantage is that these points disposition seeks for the most well shaped triangles, which assures a more stable implementation of the FEM. The drawback is a major computational cost.

The following image shows a two-dimensional triangular mesh representing a terrain.

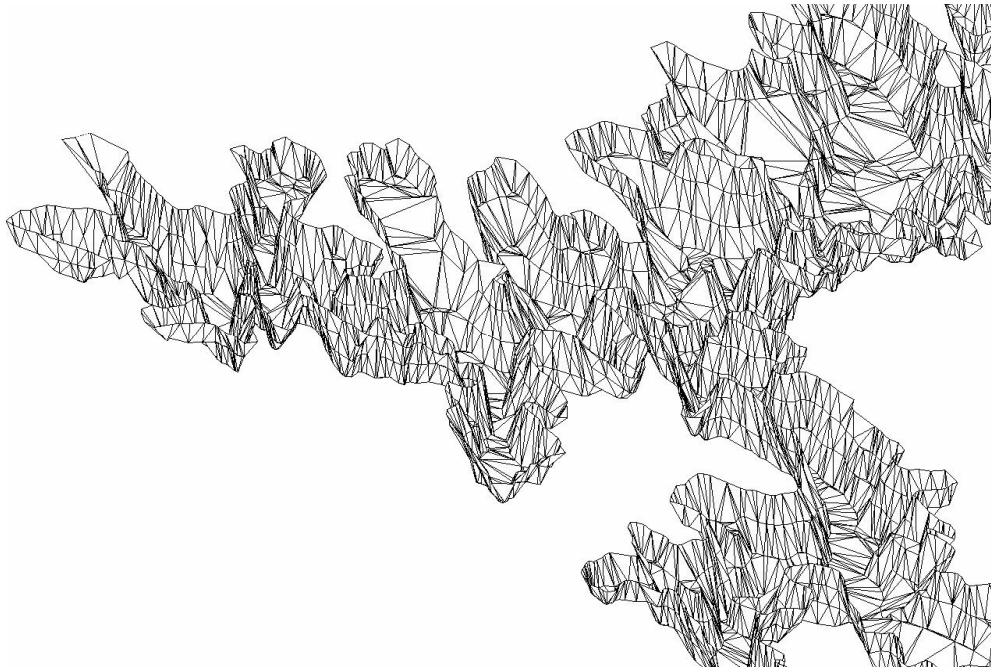


Figure 5. Example of a Terrain represented by a Delaunay triangular mesh.

When this stage is over, the mesh is ready to use in the simulation by the Transport module.

Now an example will be shown in order to illustrate the mesh generation process. Figure 6 shows a shell script sample carrying out the steps required for the generation of a FEM mesh.

```
% FEMesh2d Sample script

% Read shapefile using ShapeTerrain
filename=[filedir,'theme1'];
t1=ShapeTerrain(filename);

% Select a region
t2=trim(t1,7.365e5,7.455e5,75.23e5,75.31e5,0,1120);

% Convert to CloudTerrain
t3=CloudTerrain(t2);

% Convert to MeshTerrain
t4=MeshTerrain(t3);

% Show resulting MeshTerrain
surf(t4);

% Generate FEMesh2d
m=FEMesh2d(t4);

% Show resulting FEMesh2d
figure,surf(m);
```

Figure 6. Example of a shell script to generate a FEMesh2d object.

The script starts reading data from a Shapefile struct file of a region in the city of Nova Friburgo, in the state of Rio de Janeiro, named Caledonia. Since we start with a Shapefile struct file, we employ the ShapeTerrain constructor. Following convenient criteria, an appropriate place for a hydroelectric reservoir must be chosen. The ShapeTerrain class provides convenient clipping resources, employing the *trim* function. We will assume, for the sake of this example, that the region shown in figures 7 and 8 is a good choice.

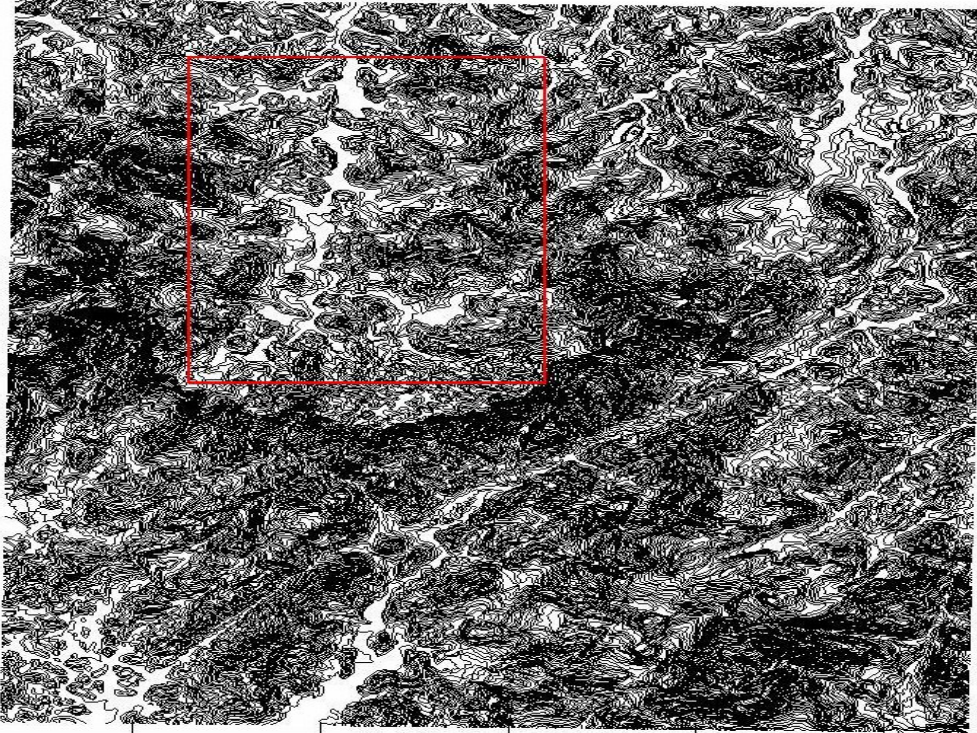


Figure 7. ShapeTerrain representation before trim.

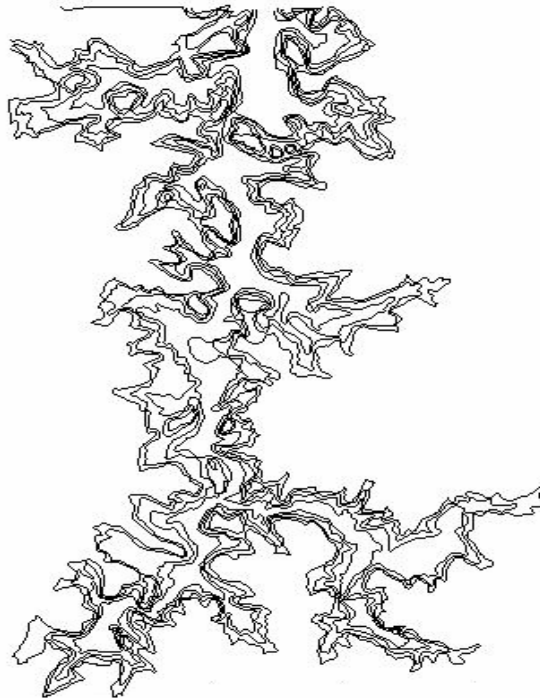


Figure 8. Selected region



The next step is to convert the terrain level curves into a CloudTerrain representation.

The MeshTerrain class can now be employed to perform the Delaunay triangulation to build the mesh (Figure 9), and send it to the FEMesh module, for the final adjustments (Figures 10 and 11).

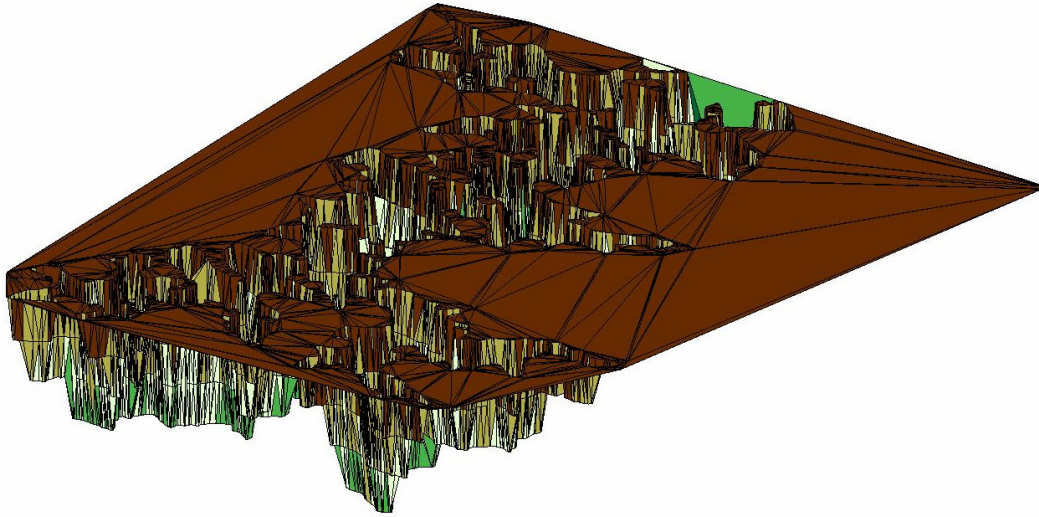


Figure 9. MeshTerrain representation. Here a primitive mesh is assembled. But note that there are many triangles on the maximum height of the selected region. These elements do not represent the terrain surface, therefore, they will be eliminated in the next stage, processed by the FEMesh module.

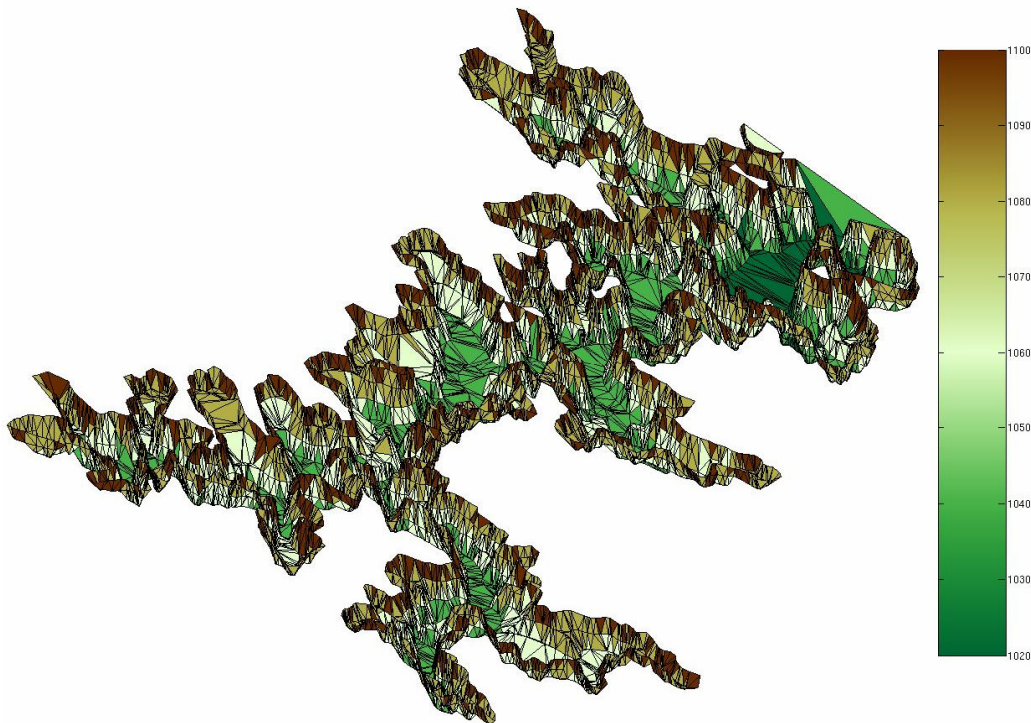


Figure 10. FEMesh2D mesh. This is the mesh after deleting the elements on the maximum height. The scale on the right side of the figure indicates the altitude in meters.

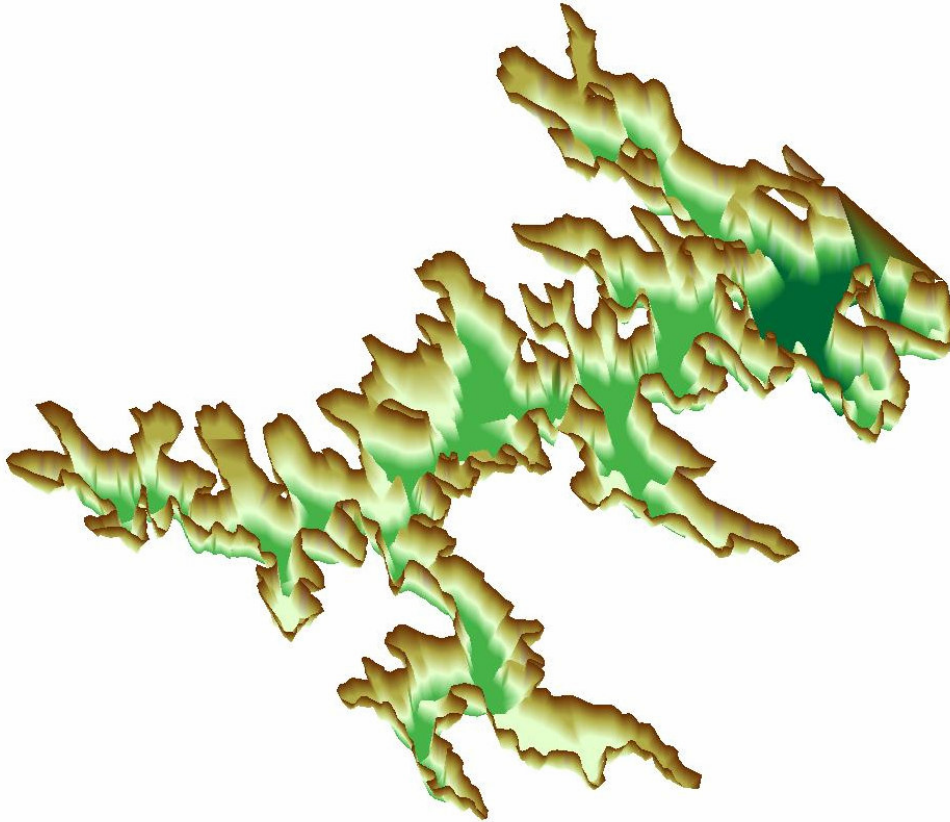


Figure 11. Same mesh of the previous figure, but with a color interpolation. The colors have the only purpose of coloring the figure and give some perspective notion.

## 7. Conclusion

This work is part of a major project, under development, which proposes the construction of a methodology to study and forecast the environmental impacts caused by a hydroelectric power plant reservoir flooding. All its elaboration has been based on software engineering concepts, aggregated to object oriented programming. The development of the Terrain and FEMesh modules, as well as the other ones, is inserted in this methodology, aiming the optimization of the structure and construction speed. This strategy allows a more flexible communication between the modules, in such a way that, for example, if a necessary correction must be done in the internal structure of the Terrain module, without any modification in its output data, the FEMesh module doesn't have to be altered.

The employed software engineering methodology allowed the implementation of simple yet powerful tools for terrain data manipulation in several representations, providing adequate visualization and simulation to the user. This last includes the mesh generation, which can start from any of these possible terrain representations.

One objective of this research is the construction of a friendly tool for simulations of the consequences of a hydroelectric reservoir filling process. The availability of different terrain representation types and the visualization resources are signs of this intent.

## 8. Acknowledgements

The authors thank FURNAS Centrais Elétricas S.A. for financial support, PROCIÊNCIA/UERJ and CNPq for the aids granted to N. Mangiavacchi.

## 9. References

- Trisiripisal, P., 2003, "Image Approximation Using Triangulation", Thesis Submitted to the Faculty of Virginia Polytechnic and State University, Blacksburg, Virginia, pp. 2-4.
- Rosman, P.C.C., 2001, "Um Sistema Computacional de Hidrodinâmica Ambiental", in Silva, R. C. V.,

- “Métodos Numéricos em Recursos Hídricos 5.1”, ed. Porto Alegre: ABRH, v. 5, pp. 1-161.
- Burgos, B. X. V., 2000, “Approximation and Geometric Processing of Digital Images With Adaptive Triangular Meshes”, Universidad Politécnica de Cataluña, pp. 72-93.
- Fowler, M., Scott, K., 2000, “UML Essencial, Um Breve Guia Para a Linguagem-Padrão de Modelagem de Objetos”, ed. Bookman, pp. 103-111.
- Paetau, P., 2005, “On the Benefits and Problems of the Object-Oriented Paradigm Including a Finnish Study”, Swedish School of Economics and Business Administration, Helsingfors, pp. 53-94.
- Shojaee, D., Helali, H., Alesheikh, AA., 2006, "Triangulation for Surface Modelling", Islamic Azad University, Maybod Branch, Yazd, Iran, pp 1-5.
- Hao, C., Jonathan, B., 1997, "Delaunay Triangulation for Curved Surfaces", in Meshing Research Corner, pp 1-13.