INNOVATIVE TOOLS FOR CONTINUUM DISCRETIZATION, BETTER MANAGEMENT OF TOUGH2 INPUT DATA AND ANALYSIS OF THE NUMERICAL SIMULATION RESULTS

Cormio C., Berry P., Bonduà S., Bortolotti V.

Department of Civil, Environmental and Materials Engineering, University of Bologna
Via Terracini, 28
40131, Bologna, Italy
e-mail: carlo.cormio3@unibo.it, paolo.berry@unibo.it, stefano.bondua@unibo.it, villiam.bortolotti@unibo.it

ABSTRACT

The natural state model and full-field exploitation simulation of geothermal reservoirs usually requires grids with an enormous number of gridblocks, in order to obtain local refinement and to investigate wider domains. This means greater computational and operational costs, in particular when using unstructured (Voronoi) grids to enhance spatial discretization. As well known, TOUGH2 does not have a native pre-processing module for unstructured grid generation and input data management—neither a post-processor nor a graphical visualization tool for output data analysis.

In order to provide effective support to the simulation of high enthalpy geothermal reservoirs, we have developed a pre-processor based upon Geographical Information Systems (GIS) called TOUGH2GIS, as well as a 3D visualization software tool called TOUGH2Viewer. TOUGH2GIS lets the user create locally refined unstructured grids, automatically assign rock types to gridblocks, and create a suitable TOUGH2 MESH file. TOUGH2Viewer enables the display of 3D grid and simulation output, a 3D flow vector of mass and heat, isosurfaces, contour maps, and scatter plots of the simulated thermodynamic variables, providing a simple and intuitive graphical interface. We have tested these two programs, which significantly reduce the time required in pre- and post-processing activities, and reduce user errors, by conducting a history-matching simulation of high enthalpy geothermal reservoirs in Tuscany (Italy).

INTRODUCTION

Attempts at numerically representing geothermal reservoirs in their entirety (full field simulation), unavoidably create huge amounts of data (due to the size of the investigated domain, usually hundreds of square kilometers and a few kilometers in depth). Typically, tens to hundreds of thousands of elementary volume elements can be required, and several thermodynamic variables and petrophysical properties have to be managed for each element and each flow interface. Discretizing the reservoir with a large number of small elements greatly improves the accuracy of the simulation outcomes, but computational time and user errors invariably grow, due to the increase in data volume.

It should also be recognized that any numerical model is a simplified, dynamic representation of the physical system that grows as the reservoir is exploited. In other words, as soon as new data are available for the system being investigated, these should be included in the model.

Spatial discretization

As already mentioned, it is quite obvious that the finer the discretization of the investigated domain, the smaller the approximation error introduced in the transition from the conceptual model to the numerical model. On the other hand, discretization improvement substantially increases the calculation time, also depending on the number of the mathematical model equations to be solved.

The compromise between accuracy and acceptable overall simulation task time usually takes the form of constructing smaller grid elements only in the areas of interest (or where measured values are available) and greater

where there are, for instance, small gradients, and there is no interest to investigate.

Local refinement within structured (telescopic, Townley and Wilson, 1980) grids is quite easy to create and manage (Figure 1) but generates many blocks that are not useful for the aims of the investigation, thus creating an undesirable increase in computational and data management time. In response to this problem, the Integral Finite Differences Method (IFDM, Narasimhan and Witherspoon, 1976) was developed, enabling the modeler to use unstructured grids with local refinement. However, this tool requires that the line joining the nodes of two adjacent blocks of the grid is orthogonal to the interface between blocks.

Creating an efficient spatial discretization can be achieved through Voronoi tessellation (Aurenhammer, 1991), which also satisfies the geometric conditions required by the IFDM and allows the user to generate locally refined grids with irregular polygons (Figure 2). Once the grid has been created, another operation to be performed, which may be a source of errors and time consuming, is properties assignation to gridblocks according to the conceptual model. Making use of maps representing, for instance, the horizons between different hydrogeological layers, the modeler can associate the proper rock type to each node of the grid.

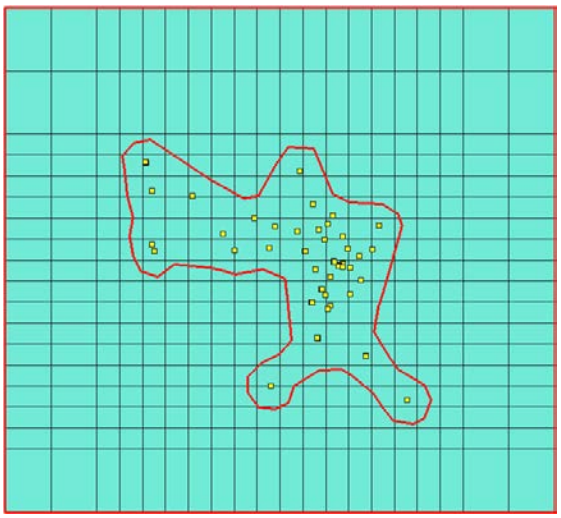

Figure 1. Structured grid with local (telescopic) refinement including an area of inter-est around geothermal wells (yellow boxes within the red boundary line).

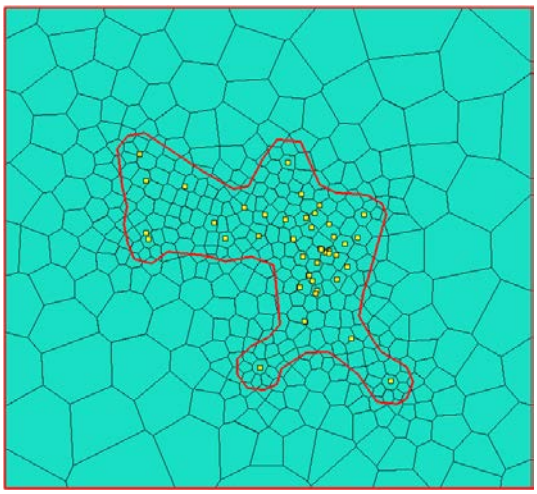

Figure 2. Unstructured grid with local refinement applied within an area of interest (red boundary line) around geothermal wells (yellow boxes).

TOUGH2 pre and post-processing

TOUGH2 does not have a native Graphical User Interface (GUI) tool to easily manage its input and output text files. This is a severe limitation, especially when creating and managing numerical models with locally refined unstructured spatial discretization—and when visualizing and analyzing the large amount of data contained in the output text file coming from a TOUGH2 simulation run. Several software programs have been developed for TOUGH2 input data pre-processing and output data post-processing and visualization, both by software houses and by scientific research groups. Most of them require the grid nodes to be provided as input, usually realized through a different software, but one of the major tasks in 3D numerical-model pre-processing is actually the automatic node generation of a locally refined, unstructured grid.

The need to improve the natural state model and exploitation simulation of high enthalpy geothermal reservoirs has led to the development of some dedicated software tools for TOUGH2 pre- and post-processing. They are: (i) TOUGH2GIS, a suite of BASH scripts that, by means of a GUI, allows a more effective and efficient creation and maintenance of 3D numerical models; (ii) TOUGH2Viewer (Bonduà et al, 2012), a home-made 3D viewer that allows a complete exploration of the

numerical model and simulation results. These new tools take into account the need to minimize operational errors and numerical model preparation time.

We have applied TOUGH2GIS and TOUGH2Viewer in modeling the large high-enthalpy geothermal fields of Tuscany, showing how these software programs significantly reduce pre- and post-processing time and user errors, in particular for full-field reservoir simulations.

TOUGH2GIS

Regardless of its source format, the information used in studying a geothermal reservoir is usually suitable to be used with a Geographical Information System (GIS), making possible a faster formulation of the conceptual model, based on the analysis and processing of a uniform dataset. Once the conceptual model is created, the numerical model can be quickly realized without further file format conversions, using raster and vector maps and database management GIS tools.

The passage from the 3D conceptual model to the corresponding 3D numerical model has been implemented in two main phases (Figure 3). The first phase produces a structured or unstructured 3D grid vertically refined but with default rock types assigned (same rock type for the nodes belonging to the same layer, not necessarily corresponding to hydrogeological rock-type distribution in the conceptual model). The second phase lets the user associate the proper material type to each gridblock using elevation raster maps (representing the bottom surface of each hydrogeological layer defined in the conceptual model).

GRASS GIS (Neteler and Mitasova, 2007) was adopted as the main development environment, in order to: (i) create a homogeneous geodatabase with all the available and required reservoir information; (ii) create exhaustive conceptual models; (iii) generate numerical models discretized through 3D locally refined structured or unstructured grids complying with the constraints imposed by the IFDM; (iii) automatically assign rock types to grid nodes.

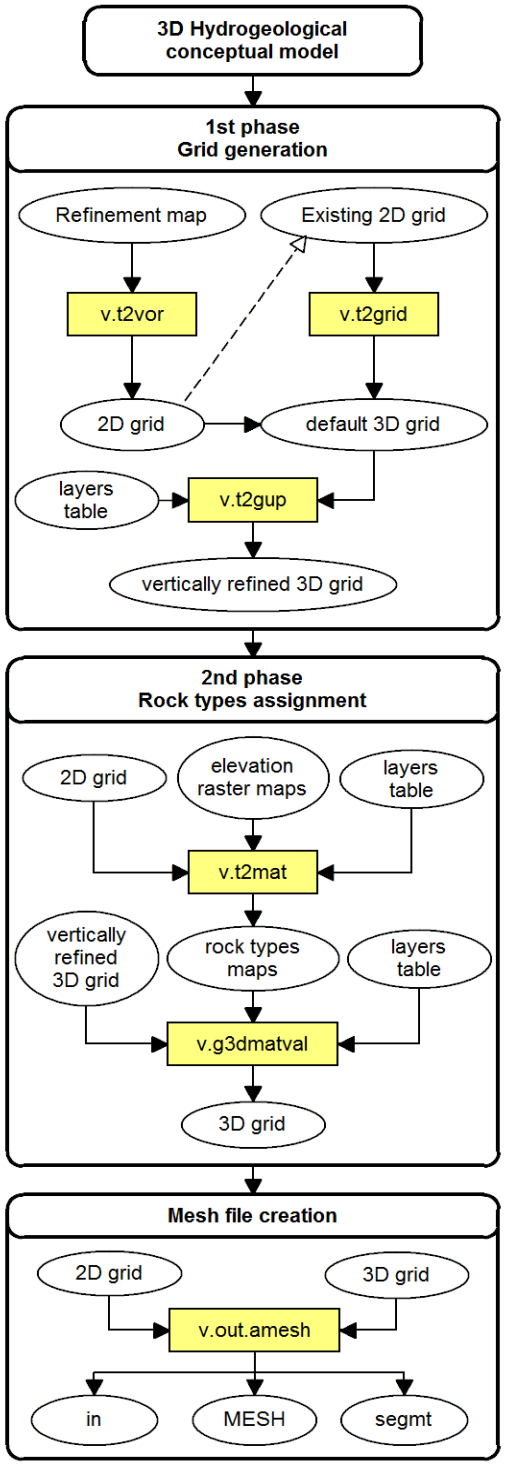

Figure 3. Procedure flow to realize the mesh file from the conceptual model with TOUGH2GIS.

Table 1. TOUGH2GIS tools overview

Tool name	Description
v.g3dmatval	Updates 3D grid attribute table with actual rock types
v.out.amesh	Creates a MESH file ready for TOUGH2
v.t2grid	Creates a “default” 3D grid from an existing 2D grid
v.t2gup	Turns the default 3D grid into a “vertically refined” grid
v.t2mat	Elaborates raster elevation maps over 2D grid producing “rock types raster maps”
v.t2vor	Creates a “default” unstructured 3D grid (all layers with the same thickness)

GRASS GIS, owing to the add-on module “v.trimesh” (Carrera-Hernandez, 2008) which implements the 2D Delaunay triangulation and mesh generator code “Triangle” (Shewchuk, 1996), is able to discretize the spatial domain generating locally refined, conforming Delaunay grids (Bern and Plassman, 2000). Through an appropriate BASH script specifically developed (v.t2vor) for this purpose, it is possible to obtain a 3D numerical model discretized with Voronoi tessellation, made up of a 3D point vector map linked to a set of database tables (specifically created to store model properties, such as geometrical information, boundary conditions, rock types assigned to grid nodes, etc.).

TOUGH2GIS pre-processing tools (Table 1 and Figure 3), enable the user to: (i) efficiently generate a locally refined unstructured 3D grid; (ii) efficiently generate a 3D grid from an existing 2D grid (structured or unstructured); (iii) automatically assign material types to each gridblock from elevation raster maps; (iv) generate a TOUGH2 MESH file; (v) export the grids generated in shapefile format; and (vi) import the simulation results and link them, through appropriate attribute tables, to the numerical model previously created, in order to be used for further model analysis and developments.

Unstructured 3D grid generation procedure

As a first step in the first phase, the user needs to produce a vector map (named “refinement map”) which individuates domain boundary and refinement areas (Figure 4). This map can be realized in several ways in the GRASS GIS environment: importing a geo-referenced Computer-Aided Design (CAD) map, manually digitizing areas, automatically creating “buffers” at a given distance from points, lines, or areas of interest)—and each area has to be defined through a boundary and a centroid (a topological object which identifies the interior of the area). The same category value (an integer number that identifies, in the database table linked to the refinement map, the attribute to be associated to map objects) has to be assigned to the centroids of all refinement areas, and a different one to the not refined areas.

The spatial discretization (Figure 5) of the investigated domain is performed using v.t2vor and v.t2gup scripts to create a locally refined unstructured 2D grid and a 3D grid with vertical local refinement. This refinement is created through the parallel replication of 2D grid nodes along the z axis for such times as the number of layers defined in the vertical discretization. The 3D grid is then linked to an attribute table (automatically generated and populated with proper default values through the tools developed) containing all the information related to each gridblock, needed to produce the MESH file. The default data can be updated by the user with standard Structured Query Language (SQL) queries and GIS update tools.

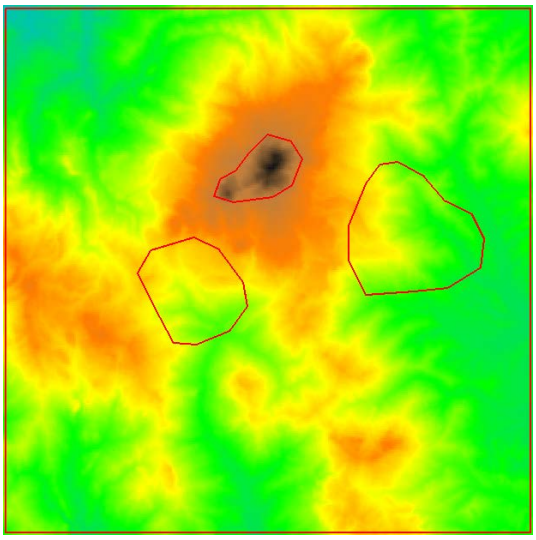

Figure 4. Refinement map overlapped to topography raster map. Red irregular boundary lines individuate refinement areas.

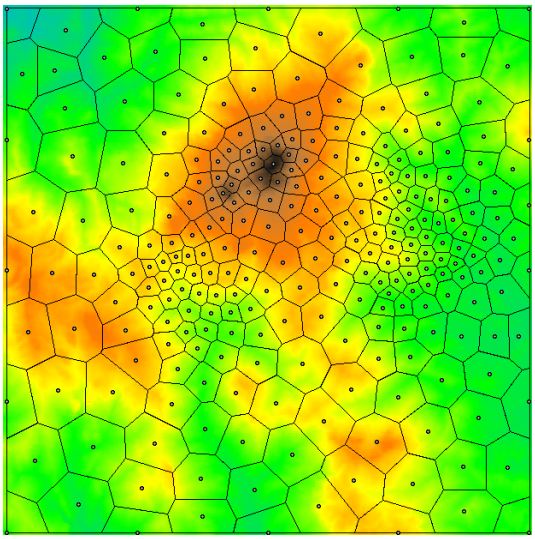

Figure 5. Locally refined unstructured 2D grid used to create the 3D grid.

Rock type assignment

The second phase involves the processing of elevation raster maps—for example, representing the horizons between rocks characterized by different permeability values—in order to automatically assign the appropriate rock material to each volume element of the 3D grid. Each elevation raster map used (topography, hydrogeological layer top/bottom surface, isothermal surface, etc.) is transformed by v.t2mat script into a "mean elevation values" raster map (m.e.v. map). The mean values are calculated within each 2D gridblock. These "graded" surfaces are then analyzed together by v.g3dmatval script, and for each vertical layer, the proper rock types are assigned, comparing node elevation value and m.e.v. maps. The result is a "rock type map" (Figure 6) which reproduces the original conceptual model discretization over the unstructured 3D grid.

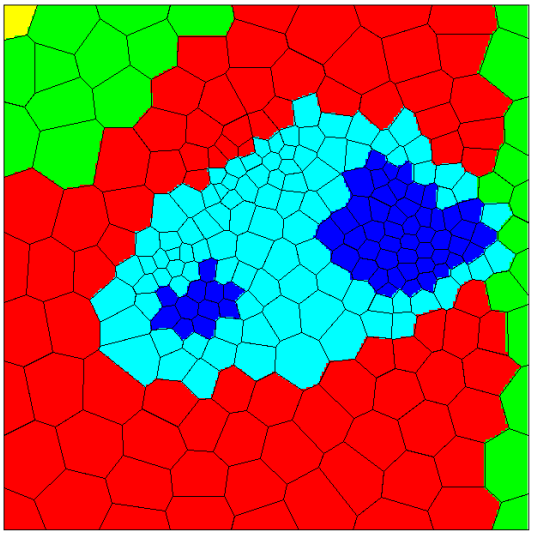

Figure 6. Example of rock types map referred to a specific grid layer. Each color identifies a different rock type.

MESH file creation

Once the attribute table linked to the 3D grid (our numerical model) has been updated with all the values required by the user (rock type, layer thickness, initial and boundary conditions, etc.), according to the conceptual model, the MESH file can be automatically generated, using the v.out.amesh script providing as input the 2D and 3D grid previously generated and updated.

TOUGH2VIEWER

This software can manage 3D locally refined structured grids, such as telescopic refinement grids, cylindrical grids, and unstructured (Voronoi complying) grids. At present, it fully manages output data files coming from the EOS EWASG (Battistelli et al., 1997) and EOS1 (Pruess et al., 1999) modules, and is easily adaptable for other EOS. It implements standard CAD commands such as zoom, rotation, and pan.

Software implementation

The software is written in Java and uses mainly Java3D libraries to display the 3D block model and Scientific Graphics Toolkit (SGT, Denbo, 1999) for contour maps visualization. GeoTools library (http://geotools.org) is used for vector maps reading (ESRI shapefile format), while JFreeChart (Gilbert, 2007) has been adopted for the construction of graphs like histograms and scatter plots and Visad (Hibbard, 1998) for isosurfaces rendering.

Input files

In order to visualize a simulated numerical model discretized with a structured grid, TOUGH2Viewer requires as input the MESH file and the simulation output file. To visualize an unstructured 3D grid, realized, for instance, with TOUGH2GIS, TOUGH2Viewer requires: (i) the "in" file, containing the x, y, z coordinates of the grid nodes, layers thickness and the rock types assigned to gridblocks; (ii) the "segmt" file, containing the coordinates of the segments constituting the boundary of the blocks base; (iii) the simulation output file. The MESH file can be read instead of the "in" file, but the "segmt" file has to be provided as input.

When using TOUGH2GIS, rock-types assignation is carried out using the elevation raster maps describing the horizons between different hydrogeological layers. TOUGH2Viewer allows the display of such surfaces as shape files together with the 3D grid, so as to increase the model visualization details and to enable the user to better check model suitability. A specific dialog box enables reading and importing shape files (which must contain polyline objects representing contour lines of a hydrogeological layer or thermophysical data, with elevation information).

Display features

3D Block Model view

The three-dimensional visualization takes place within a dialog box (Figure 7) made up of a frame window containing a visualization window object and a set of controls. The control elements are pull-down menus and slide bars that allow the user to: (i) select the type of variable to display (i.e., temperature, pressure, etc.); (ii) switch between the simulated time steps; (iii) switch to the visualization of the rock types associated to gridblocks; (iv) access the different visualization modalities. At the bottom of the visualization dialog box, all the information about the selected block (i.e. block name, x, y, z node coordinates, index, material type, displayed variable value) is given by means of the "Block selection" or "Quick Selection" functionalities. Note that it is also possible to produce plots of the variables along both Cartesian directions, usually z, and versus time.

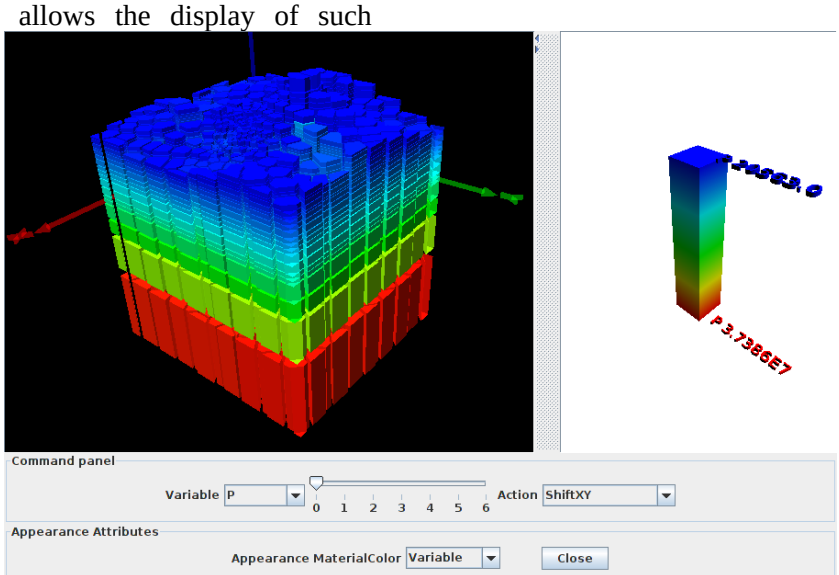

Figure 7. 3D Block model dialog box

3D Flow vector view

The “flow vector” visualization modality gives vector flows that are shown in two different ways—one in which the length of the shown vector is proportional to flow rate intensity, and one in which all vectors have unitary length, and the flow-rate intensity is represented in color scale. To increase readability, each grid node is represented by the vector resulting from the sum of all flows across the interfaces of the block.

Contour plot

This option allows the creation of 2D contour maps (vertical or horizontal sections for a given time step) of the selected variable, making an interpolation of data points. The user can interact with the interpolation process defining the resolution of the estimation grid, and the location and orientation of the sectioning plane (Figure 8). The variable analyzed is shown in false-color scale, and the style of representation can be customized. The user can also create the isolines of the selected variable, and overlap them on the contour map (see Figure 8). In order to work properly with the SGT library, which requires equally spaced data, we have implemented an interpolation processing of the original data, based on the IDW (Inverse Distance Weighting) method.

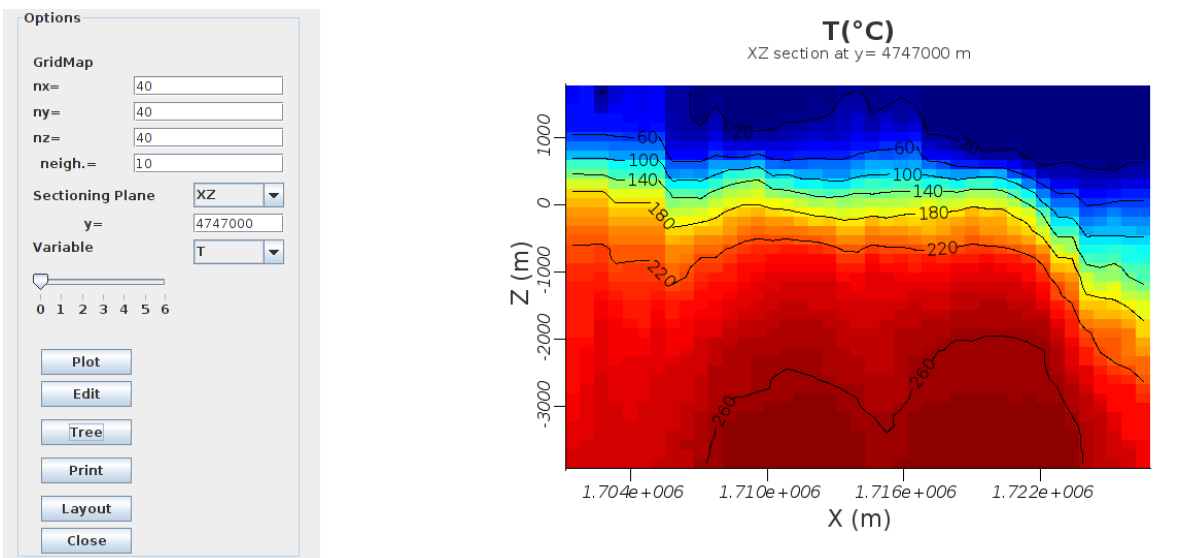

Figure 8. Pressure contour map.

3D isosurface

To better understand the behavior of thermophysical properties, is helpful a representation of a isosurface of the simulated variables (Figure 9). By means of two slide selector, it is possible to change the threshold value and the time step (the graph is updated in real time). This representation is obtained by Visad library (Hibbard, 1998), that requires equally spaced data, so it's been introduced an interpolation processing of the original data based on IDW method.

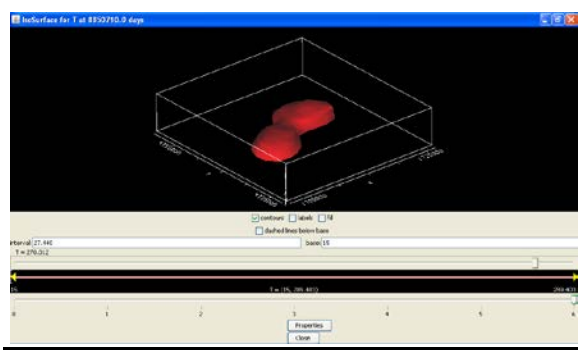

Figure 9. Example of isosurface representation.

CONCLUSIONS

The software tools described in this paper both represent improvements for the TOUGH user related to pre- and post-processing of TOUGH model simulations. The pre-processor TOUGH2GIS is a powerful tool by which the creation phase of numerical modeling for computer codes using the IFDM has been significantly improved. This improvement can be seen both in the simplification of complex 3D numerical model creation and management, and in the reduction of operational errors occurring during the handling of substantial amounts of data. Similarly, TOUGH2Viewer, with its easy-to-use GUI, provides the user with powerful tools by which to display and examine data in 3D graphic mode. It strongly reduces the tasks necessary to understand and interpret simulation outputs. The use of these two software programs greatly increases the efficiency of the overall simulation task.

REFERENCES

Aurenhammer, F., Voronoi diagrams—a survey of a fundamental geometric data structure, *ACM Computing Surveys*, New York, NY, USA, pp. 345-405, 1991.

Bern M., and Plassmann P, *Chapter 6 - Mesh Generation*, In: Sack, J.R., and Urrutia, J., Editor(s), "*Handbook of Computational Geometry*", North-Holland, Amsterdam, pp 291-332, 2000, ISBN 9780444825377, 10.1016/B978-044482537-7/50007-3.

(http://www.sciencedirect.com/science/article/pii/B9780444825377500073)

Bonduá, S., Berry, P., Bortolotti, V., and Cormio, C., TOUGH2Viewer: a post-processing tool for interactive 3D visualization of locally refined unstructured grids for TOUGH2, *Computers & Geosciences*, Available online 25 April 2012, ISSN 0098-3004, 10.1016/j.cageo.2012.04.008. (http://www.sciencedirect.com/science/article/pii/S0098300412001318?v=s5)

Carrera-Hernandez, J., University of Alberta, Edmonton Canada (2008).

http://www.valledemexico.ambitiouslemon.com/vtrimesh.html.

Denbo, D.W, The Scientific Graphics Toolkit. In: Proceedings of *OCEANS '99 MTS/IEEE. Riding the Crest into the 21st Century*, Seattle, Washington, USA, September 13-16, Vol. 1, pp.470-473, 1999.

doi: 10.1109/OCEANS.1999.799789.

Geotools, 2010. Open Source Geospatial Foundation, www.geotools.org, [Accessed November 21,2011]

Gilbert, D., 2007. *The JFreeChart Class Library, Version 1.0.4, Developer Guide*. Object Refinery Limited, Harpenden, Hertfordshire, UK, 642pp.

Hibbard, W., VisAD: Connecting people to computations and people to people, Computer Graphics 32(3), 10-12, 1998.

Narasimhan, T.N. and P.A. Witherspoon, An Integrated Finite Difference Method for Analyzing Fluid Flow in Porous Media, *Water Resour. Res.*, Vol. 12, No. 1, pp. 57 – 64, 1976.

Neteler M., and Mitasova H. *Open Source GIS: A GRASS GIS Approach. Third edition*, Springer, New York, 2007.

Pruess, K., C. Oldenburg, and G. Moridis, *TOUGH2 User's Guide, Version 2.0*, Report LBNL-43134, Lawrence Berkeley National Laboratory, Berkeley, Calif., 1999.

Shewchuk J.R., Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator, *Applied Computational Geometry towards Geometric Engineering*, Springer-Verlag, Berlin, 1996.

Townley, L.R., Wilson J.L., 1980. *Description of an user's manual for a finite element aquifer flow model*. Report 79-3, AQUIFEM-1: Ralph M. Partson Laboratory Technology Adaptation Program, Cambridge Massachusetts.