IAMG 2011
SALZBURG
September 5 - 9 | University of Salzburg | Austria
Mathematical Geosciences at the Crossroads of Theory and Practice

Constructing Geometric Models of the Subsurface for Finite Element Simulation

Björn ZEHNER[1]

[1] Department for Environmental Informatics, UFZ – Helmholtz Centre for Environmental Research, Permoserstrasse 15, 04318 Leipzig, Germany,
bjoern.zehner@ufz.de, bzehner@gmx.de

Peer-reviewed IAMG 2011 publication

doi:10.5242/iamg.2011.0069

Abstract

Finite element based simulation is an important tool to assess processes in the subsurface, such as groundwater flow or CO2 storage in reservoirs. The simulation requires the generation of suitable input models which are usually generated using specialized software packages for geometrical modelling such as GOCAD. Generally this software supports two types of 3D grids: structured grids that consist of hexahedra and unstructured grids that use tetrahedra. To represent complicated geological structures, tetrahedral grids would be preferable but unfortunately their construction is less well supported. In order to generate tetrahedral grids with sufficient quality, the quality of the triangular meshes that define the boundary representation of the model must be higher than is normally required for illustration and communication purposes and they must fulfil certain constraints. This contribution will explain which constraints these are and how the construction of complicated geological structures, such as faulted systems, can be done in order to meet these requirements. The software GOCAD can be extended using a plugin mechanism and so we have written several plugins that support simple interaction or modelling functionality that are not yet present in GOCAD and which in addition provide an interface from the modelling software to external meshing tools that are free for non-commercial use. Further plugins have been implemented that provide interfaces with the finite element simulation software OpenGeoSys that is developed at the UFZ and to the software Paraview for the scientific visualization of the results. While we use GOCAD in particular, the problems described and the solutions and workflows suggested are general and should be of interest to users of other software packages as well.

1 Introduction

In order to do finite element based simulation of processes in the subsurface, such as groundwater flow in the Thuringian Basin in eastern Germany or CO2 storage in reservoirs, suitable geometrical models and 3D-grids must be generated. To construct our own models and exchange models with our project partners, we use the commercial software GOCAD that is very commonly used by federal agencies, research institutions and universities. We have integrated GOCAD into our overall workflow, from the generation of the geometrical model via the simulation to the visualization in our virtual environment by interfacing it to the different software packages and libraries that we use (see Figure 1). In order to simulate the models, it is necessary to turn the geometrical models which provide a boundary representation of the subsurface into 3D grids and this requires the generation of suitable input models for the gridding process. In general GOCAD supports two types of 3D grids, stratigraphic grids that consist of hexahedra and unstructured grids that use tetrahedra as basic elements and are called Solids. For reservoir simulation purposes the use of hexahedral grids and finite difference simulation methods (e.g. using Schlumberger's Eclipse) is more common and the construction of these grids works quite well. However, if the reservoir is faulted, the stratigraphic grids become disconnected across the faults (so called "Faulted SGrids"). While some reservoir-simulators, e.g. Schlumberger's Eclipse, can handle this, many others, such as the finite element software we develop and use (OpenGeoSys, Wang 2009) can not so far. For this reason, and because unstructured grids can be better adapted to different geometries, the representation of complicated geological structures with tetrahedral grids and the use of finite element simulation would be preferable. Unfortunately the construction of these grids is less well supported in GOCAD.

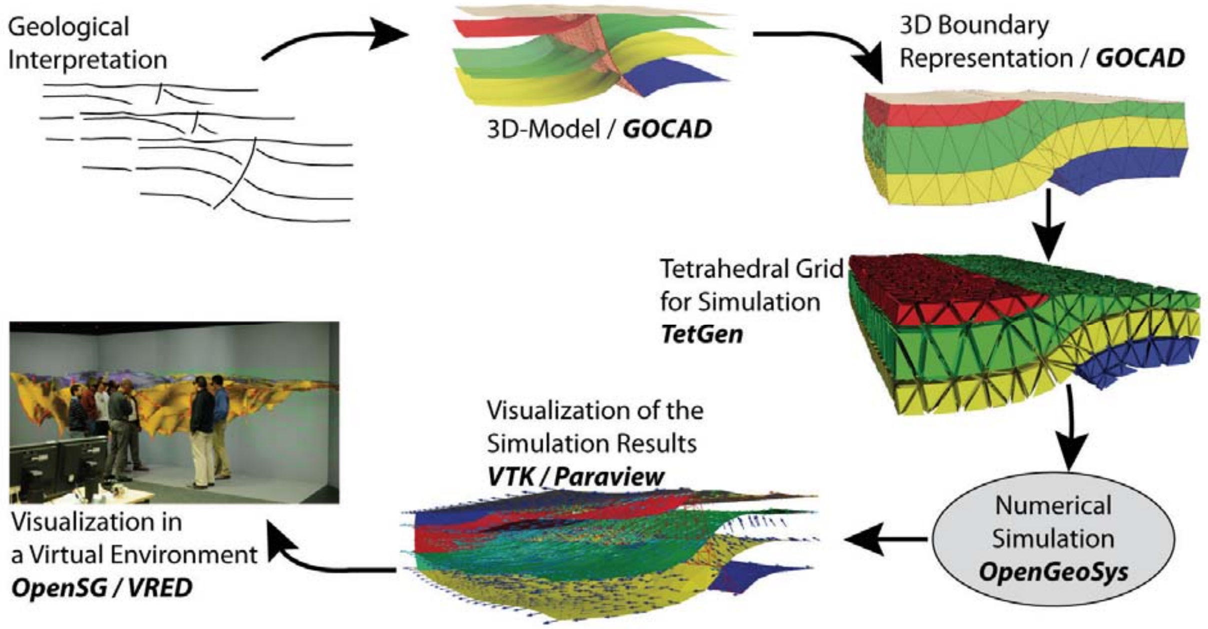
Geological Interpretation
3D-Model / GOCAD
3D Boundary Representation / GOCAD
Tetrahedral Grid for Simulation TetGen
Numerical Simulation OpenGeoSys
Visualization of the Simulation Results VTK / Paraview
Visualization in a Virtual Environment OpenSG / VRED

Figure 1: General overview of the workflow from geological interpretation via geometrical modelling and simulation to the visualization of the model and simulation results. Names in bold-italic are the names of the software packages and libraries used for this task.

In order to generate tetrahedral grids, the quality of the triangular meshes for the boundary representation of the subsurface must be higher than is normally required for illustration, communication and discussion purposes. The mesh should consist of triangles with not too large an aspect ratio (longest side length divided by shortest side length). The intersections where one surface intersects another one or is connected to it, for example at the contact of a stratigraphic layer and a fault, are also critical. It is essential that both surfaces share the same vertices and segments as shown in Figure 2. Further the whole model should be represented by a boundary representation that has no holes and divides the space into volumes fully enclosed by surfaces. If all these preconditions are met, the boundary representation can easily be gridded, using open source gridding software, such as TetGen (Si, 2011) to generate a tetrahedral grid.

The aim of this work was to provide the structural geologists with a set of tools and suggest a workflow, so that they can set up the structural model in such a way that it meets all the aforementioned requirements. While this requires individual work and a geologist who has the necessary knowledge and experience with meshing, it has the advantage that the geologist can, for example, decide to do some simplifications that can be justified if some geometrical configuration occurs that will be very hard to mesh or which would require an excessive amount of triangles. This contrasts with the approach which involves constructing the model while ignoring the subsequent meshing problems, and then later improving the mesh of the model as it is e.g. shown in Pellerin et al. (2011) or as an initial stage of their work in Blöcher et al. (2010). Here, configurations can appear that are very hard to mesh. However, the geological model has often already been created in the past or by somebody else, so that the automatic approach for improving the triangulation might be favourable.

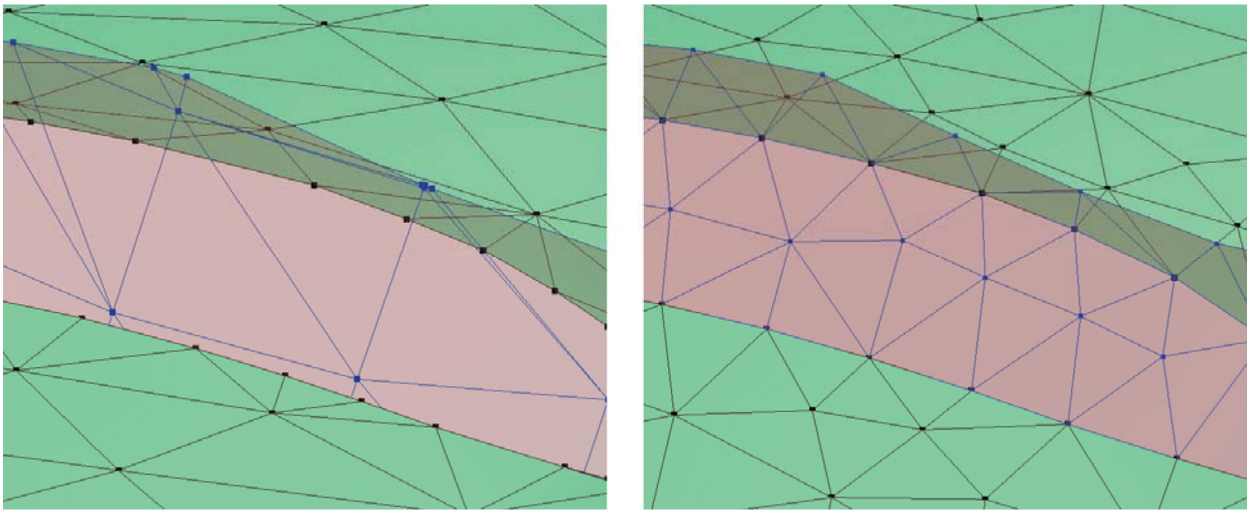

Figure 2: The same part of a model of two surfaces (green) connected to a fault (transparent reddish) shown twice. On the left side the three meshes are completely independent as they share neither vertices nor segments. The fault also shows some triangles with very bad aspect ratios. It would not be possible to construct a volume model based on tetrahedra with this boundary representation. On the right side the same surfaces and the fault are meshed in a way that would make tetrahedralization possible.

The motivation for this work came from the INFLUINS project, which deals with fluid dynamics in sedimentary basins and uses the Thuringian Basin in Germany as an example. This basin extends approximately 160 km E-W and 100 km N-S and features several complicated fault zones. The question was if it is feasible, also in terms of the work required, to set up a tetrahedron-based finite element model for the whole basin that, at least to some extent, reflects the structural complexity and provides the necessary mesh quality to perform fluid flow simulations. It is clear that large-scale modelling of groundwater flow has been done before, but the models are often oversimplified from a geological point of view.

2 3D modelling

In order to set up geometrical models, two different options have been established regarding the workflow. The first one targets constructing large-scale regional models that are structurally relatively simple and can be treated by looking at the projection of different features onto the x-y plane. The incorporation of faults is only possible if the contact line of the stratigraphic layers on the faults is already known and relatively simple. The second workflow targets the modelling of complicated fault systems but requires more individual work. In both cases functionality has been added to GOCAD using its plugin mechanism. This functionality comprises importers/exporters and algorithms from computational geometry. The algorithms and the tree structures that are used for enhancement are described in Laszlo (1996) and DeBerg et al. (1997).

In both of these workflows an important role is played by the generation of finite element conforming meshes with good triangle aspect ratios using constrained Delaunay triangulation. This is done externally using the software GMSH (Geuzaine & Remacle, 2009) which can be downloaded from the internet and is under the GNU General Public License (GPL). An exporter has been added to GOCAD that is given an enclosing polygon and the different lines and points that are to be used as constraints, and that generates an input file for GMSH. GMSH adds additional points - so called Steiner points - to the triangulation so that the mesh can be gradually refined down towards regions where original points are very dense. This is steered by a parameter that is given for each point in addition to the coordinates. We found it useful to let the user specify this parameter as a fraction or multiple of the distance to the nearest point. By using a multiple of the distance to the next point and by ensuring that the line segments are relatively short in comparison to the distance to the neighbouring lines, it is possible to suppress the insertion of Steiner points on the line segments. Further the user needs to specify a distance below which two points are assumed to be the same. The triangulated mesh that is generated by GMSH can then be read back as surface.

2.1 Modelling of large-scale regional models

In order to construct large-scale regional models, a workflow has been established that allows the construction of a boundary representation for continuous stratigraphy and that is based on methods which use projection onto the xy plane. The example used here is the construction of a model of the Thuringian Basin in Germany. As a prerequisite, a triangulation is generated in the xy plane that accepts the borders of the stratigraphic units and the paths of the rivers along which later certain boundary conditions for the simulation are to be set. To complete this task, an algorithm first checks that the segments from all the lines to be written to the file as constraints do not intersect. Then, they are written to a GMSH input file, together with the points that must be incorporated into this triangulation. The user can select a minimum distance between the points and the line segments. This is necessary because GMSH can not find a good triangulation if one of the points is on a line segment or very near to it. The GMSH input file is then used externally to generate a constrained Delaunay triangulation that obeys all lines and points as constraints but adds additional Steiner points to ensure the good quality of the mesh. Subsequently the mesh is read back as triangular surface into GOCAD. To ensure that further processing can take place without problems, a triangle is not allowed to have two segments on the same stratigraphic border or to connect two borders of the same stratigraphic unit because this would result in artefacts, such as a thickness of the stratigraphic unit of zero. An algorithm has been implemented that checks this precondition and, if necessary, locally refines the mesh.

Further functionality has been added to GOCAD to remove the triangles inside or outside of closed curves so that the different surfaces can later be generated by using this initial triangulation as a template. This is shown by the top two images of Figure 3. On the left side the initial triangulation of the whole basin is shown - on the right side all triangles that do not belong to the stratigraphic unit under consideration (here the Keuper) are deleted. In this way initial surfaces for the different stratigraphic units and for the terrain that is split up into different regions depending on the outcrop of the different stratigraphic units are generated. All these surfaces are then subjected to the same process:

- The lines that describe the boundaries of the stratigraphic units at the surface are projected onto the digital terrain model. Then, using a function that has been added to GOCAD, all the vertices of the triangulation that have equal xy coordinates as the points that belong to the lines, are moved onto these border lines and are set as control nodes, so that they do not move when the surface is interpolated using GOCAD's DSI algorithm which is described in Mallet, 2002 (Figure 3c).
- If the aim is to generate a grid made of triangular prisms, a constraint can be set for the points such that they are only allowed to move vertically. If a tetrahedral grid is generated from the boundary representation, this is not necessary. In order to allow the generation of tetrahedra we must ensure that the stratigraphic unit has at least a minimum thickness and a range-thickness-constraint is applied that uses this thickness as a minimum and the thickness of the stratigraphic unit from literature as maximum. Further, points which indicate the stratigraphic interface in boreholes or on interpreted seismic sections and suchlike are set as control points (Figure 3d).
- Applying the DSI Interpolation generates a smooth surface, the boundary of which exactly coincides with the lines representing the original boundary at the surface (Figure 3e).

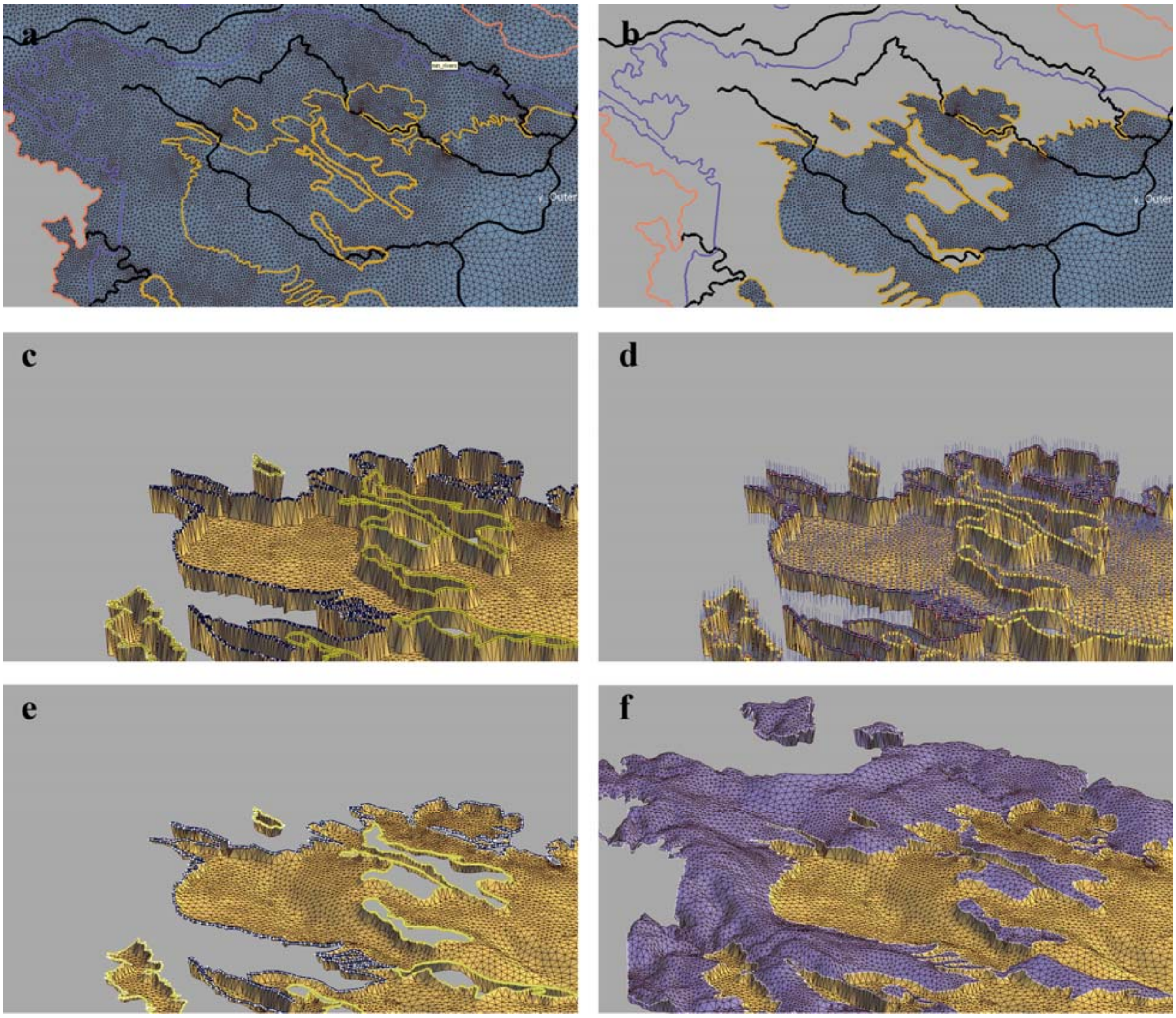

Figure 3: Construction of a boundary representation of a stratigraphic unit for large scale regional models. The visualization in c-f is 10 times superelevated. As the stratigraphic unit must reach the minimum thickness with the outer row of triangles, the dip along the outcrop line can become high. See section 2.1 for a more detailed explanation.

Doing this for the top and bottom of the stratigraphic unit, and for the terrain where the unit outcrops, generates a watertight boundary representation of the unit where all surfaces meet at the original lines. The image in Figure 3f, for example, shows the interface between Muschelkalk and Keuper coloured in brown. The blue surface represents the terrain where the Muschelkalk outcrops and the base of the Muschelkalk. All three surfaces together define a closed volume that represents the Muschelkalk. Using this method, we have set up an initial model as proof of concept that is based on the data described in Kober (2008). It represents the Thuringian Basin subdivided into Bunter (Buntsandstein), Muschelkalk and Keuper as stratigraphic units and is represented as a consistent and watertight boundary representation.

2.2 Modelling of faulted systems

When modelling a faulted system, the course of the fault must be constructed as a first step. If this is done from geological cross-sections, these are often constructed in specialized software such as Geosec or 2D Move. These programs are designed more for the construction of the sections but usually do not obey the necessary constraints that must be complied with in order to be able later to generate a mesh where the triangulation reflects the geological boundaries. In order to use GOCAD to process cross-sections, we have extended it to include the following tools:

- Finding the intersection of the line representing the horizon with the line that represents the fault and inserting a vertex at this point on both lines. A tolerance can be given if the two lines do not meet exactly.
- Extending lines towards another line (the fault). A new point is inserted on the fault and the endpoint of the line is moved onto this point.
- Connecting the points that represent the contacts of a horizon with a fault on different sections and thereby constructing the contact line.
- Calculating the intersection, if some contact lines cross each other, and inserting a vertex at this position on both lines.

After the contact has been modelled consistently on the different sections, the fault can be constructed by connecting the contacts on each section which generates two contact lines, one for the hanging wall block and one for the foot wall block (Figure 4 top left). In order to make the contact line smoother, additional points can be inserted by defining a cubic spline function that retains the original points (Figure 4 top right). Alternatively, if the course of the fault is known or has been already constructed, the present GOCAD functions can be used to cut the stratigraphic layers by the fault and move them on the fault. Subsequently the contact lines can be generated by setting suitable border stones and using GOCAD’s line-from-border command. If the points on this line are very irregularly spaced, the function to resample the points on the line using a cubic spline has been implemented. These contact lines are defined for all stratigraphic interfaces or for other faults that contact or intersect the fault and define, together with the fault lines, a framework that represents the fault and is shown on the left of the second row of Figure 4.

The best-fitting plane through all points is calculated for this framework, the points and lines are projected at right angles onto this plane and the result is exported to GMSH where the constrained Delaunay algorithm is used to create a triangulation that incorporates the lines and points of the framework. Subsequently the resulting triangulation is read back as surface and the points are transformed back to their original position and set as control nodes. GOCAD’s interpolation algorithm (DSI) is then used to smooth the mesh. As a result we get a fault that has all segments and vertices in common with the surfaces that are in contact with it, as shown in Figure 4 on the right side of the second row.

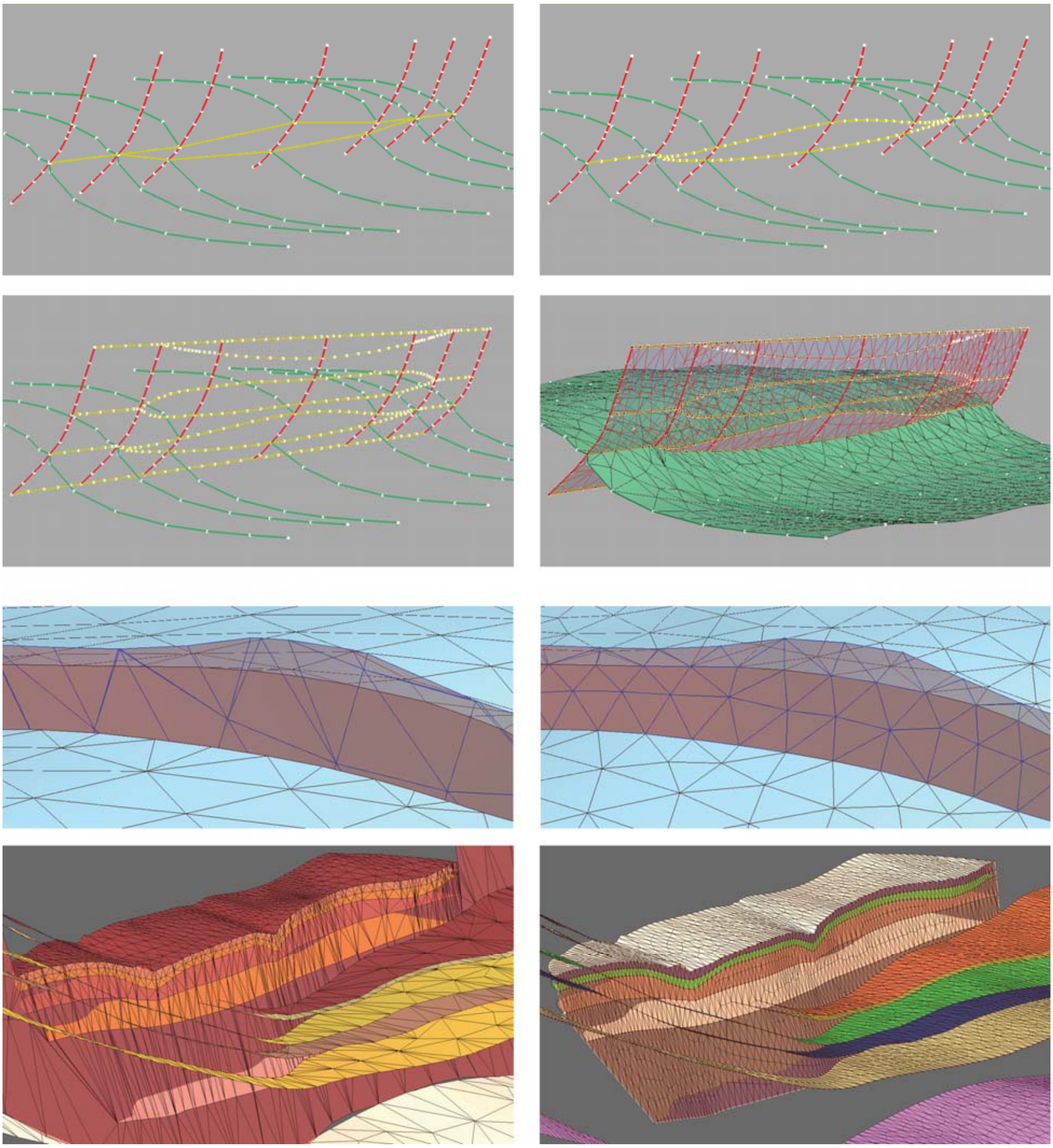

Figure 4: Top four images: Generating a fault with a mesh that shares segments and vertices with the stratigraphic surfaces that are in contact with it. See text for detailed explanation. The yellow lines are the contact lines of the surface, the red lines represent the fault and the green lines indicate one of the stratigraphic interfaces. Bottom four images: Examples where the methods have been used to improve the triangulation of the original model. On the left is the original we have generated using standard GOCAD commands. On the right we have generated new meshes after resampling the contact lines using spline functions. The bottom row shows part of the Finne fault zone in the Thuringian Basin for which the original model has been set up by Alexander Malz, University of Jena.

We have used the tools to work on sections to generate a 2D finite element grid for a section through the Thuringian Basin that has been constructed by Stephan Donndorf from the University of Jena using the software 2D Move. We have also already tested the feasibility of constructing or remeshing more complicated faulted zones by working over a model of parts of the Finne fault zone in the Thuringian Basin that has been constructed by Alexander Malz from the University of Jena, using a set of balanced cross sections. At the bottom of Figure 4 we can see the model of the fault that has been constructed by him using GOCAD on the left side and the version that has been remeshed using the methods described here on the right.

2.3 Outlook on the integration of the first two workflows

The geological interpretation and the construction of the different fault zones in the Thuringian basin is still an ongoing project with our partners at the University of Jena. For this reason we have not yet been able to test the combination of the two methods described. The idea is to model fault zones and their near surroundings in detail but coarsen the meshes towards distant borders where the triangulations are prescribed. At these distant borders the same triangulation is prescribed for the large-scale regional model that has been constructed using the first workflow. The different fault zones and the regional model are then tetrahedralized independently and are subsequently stitched together, which can be easily done because the same triangulation has been used on the common borders and so the tetrahedra match along them.

3 Tetrahedralization, simulation and visualization

A boundary representation model that has been constructed as described before can be tetrahedralized by the software TetGen (Si, 2011). To accomplish this, the different surfaces that describe the model are exported into a TetGen input file. In this file the vertices have to be globalized, which means that the different vertices that belong to different surfaces but describe points at the same position must become one global point. The necessary sorting is accomplished by using an Octree and the user must specify at which distance two points are treated as the same. If the volume of the model is partitioned by internal surfaces, TetGen can recognise this and each tetrahedron is assigned a number that indicates to which partition it belongs. In the simulation software this number can be later used to assign different materials to different partitions. The output of TetGen, containing all the tetrahedral elements, is read back into GOCAD as Solid where the different partitions are assumed to be different parts. Figure 4 shows at the top left the boundary representation (the surfaces towards the viewer are not rendered in this case) and on the right the corresponding tetrahedralization with the cells shrunk to provide a better overview.

In order to carry out finite element simulations, we use the software OpenGeoSys which is developed in our research group (Wang, 2009). The software is command-line-oriented, reading its configuration and the information on geometry, grids and materials from a set of files. Geometric objects can be used to set initial and boundary conditions. For example, the hydraulic head might be set on one of the boundaries of the model or the injection rate along a borehole or at a point. These geometries can be generated in GOCAD very efficiently. In order to link the simulation software closely to GOCAD, exporters have been written that can export the structured or unstructured grids directly as simulation grids for OpenGeoSys and further geometries can be exported to an XML file format that can be read by it. In the configuration files for the simulation, the user must specify which geometry has to be used for which boundary condition.

After running a simulation, OpenGeoSys outputs the results to files in the format of the Visualization Toolkit (Schroeder et al., 1996, VTK). VTK is an open-source programming library that provides a large number of different filters to generate visualizations from numerical data, such as glyph filters or streamline calculation. There is also the open-source software called Paraview that is built on top of VTK and is used by many groups from the field of simulation to do post-processing and visualization. In order to visualize the simulation results together with the geological model and the parametrization, VTK has been linked into GOCAD in order to provide an interface to export geometric objects as polydata, SGrids as structured grids and Solids as unstructured grids. In this way the geological model can be visualized together with the simulation results and the parameters - see for example bottom of Figure 5.

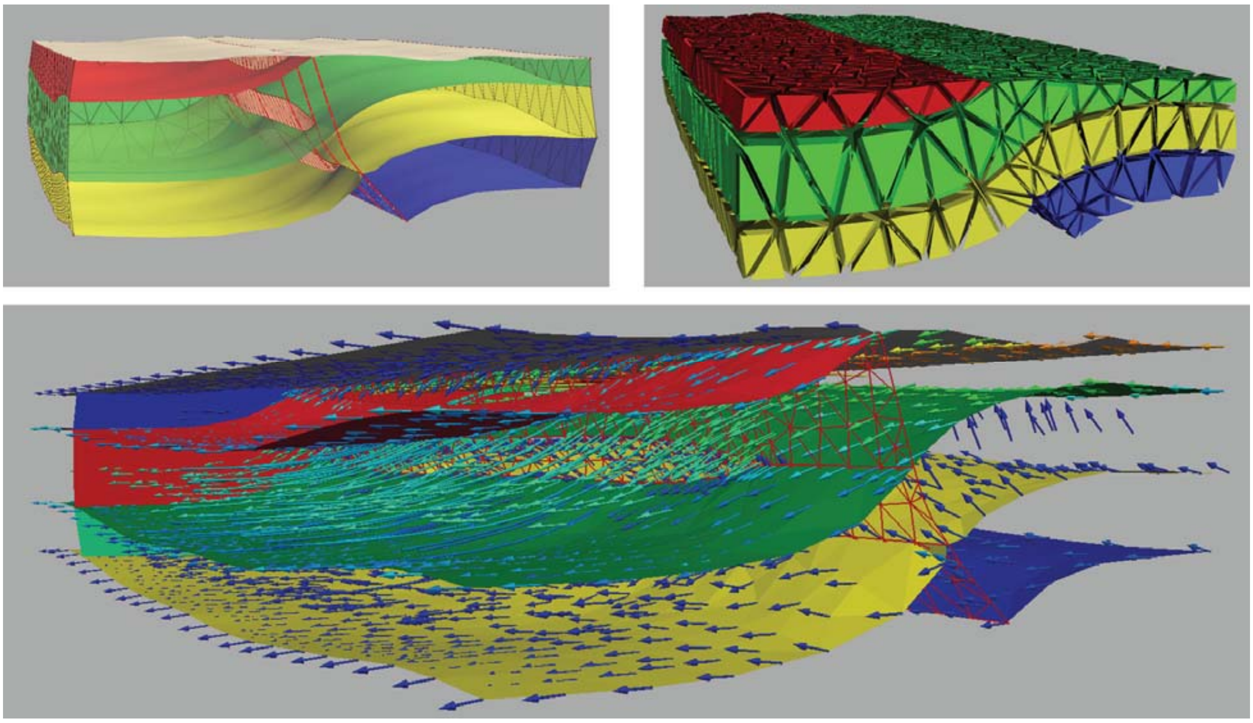

Figure 5: Top left: Boundary representation of a fault. Top right: Corresponding tetrahedralization, rendered using Paraview. Bottom: Geological model shown together with the results of a groundwater simulation. Groundwater flow is indicated by vector glyphs and streamlines. Rendering has been done using Paraview

Using this simulation workflow it was possible to generate a tetrahedral grid from the boundary representation of the Thuringian basin that has been described in section two. The corresponding solid model consists of approximately 600,000 tetrahedra and has been used to run a groundwater flow simulation. Figure 6 shows at the top the tetrahedron-based Thuringian Basin (as a Solid within GOCAD). The colours indicate the different parts that can also be used later to apply different materials in the simulation software. At the bottom the simulation results are shown, rendered in Paraview.

If the models are structurally very complicated, it is best to look at them in a large scale stereoscopic virtual reality environment. The Helmholtz Centre for Environmental Research - UFZ has such a system that can be run as a CAVE like system but also can be used to augment the interactive 3D stereoscopic visualization on the rear screen and on the floor with additional 2D visualizations that are coupled to the 3D visualization and are shown on two side-wings (Zehner, 2010). Our VR environment is run by the commercial software VRED from PI-VR GmbH that is built on top of the OpenSG scenegraph (Reiners et al., 2002) and can be extended according to our needs. In order to ensure the rapid visualization of our GOCAD models in our VR environment, we have linked the OpenSG library, which is open source, into GOCAD and implemented an exporter that can export the geometries of the whole project directly in binary OpenSG format. The VTK interface described above can be used for SGrids and Solids, as we also implemented an interface to load structured and unstructured grids in VTK format into VRED. Figure 7 shows a group of people looking at the GOCAD model of the Thuringian Basin in our VR environment.

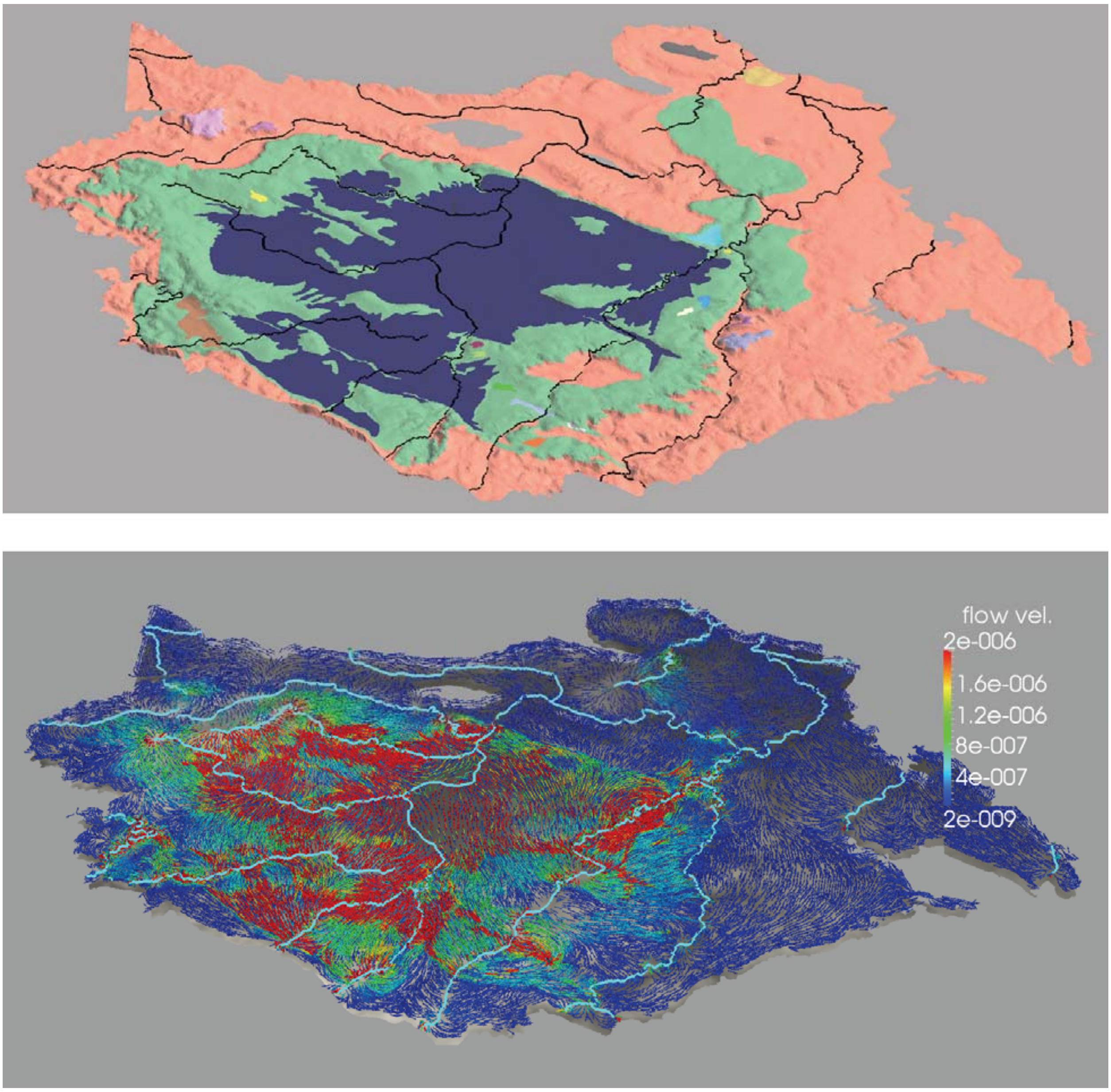

Figure 6: Top: The model of the Thuringian Basin as a tetrahedral mesh rendered in Gocad. The colours indicate the different closed volumes, using GOCAD's "Parts" command. Bottom: Result of a groundwater simulation rendered in Paraview, showing flow direction and velocity [m/sec].

Figure 7: Members of the Influins project looking at a virtual model of the Thuringian Basin.

4 Conclusions and future work

We have described two ways in which GOCAD could be used for constructing a geometric model that obeys geometric constraints, so that it can be used later to generate a solid model for finite element simulation. Currently the construction of fault zones requires a lot of manual manipulation but after gaining more experience, parts of this could be automated or done more quickly using wizards. The workflows have been used to construct a model of the Thuringian Basin in Germany and for remeshing a more complicated model of a fault zone.

A key role is played in both workflows by constrained Delaunay triangulation, for which the external software Gmsh has been used. While this worked reasonably well in practice, one problem with Gmsh was that it does not always respect the defined constraints and generates line segments that cross the line segments initially given. These errors are easy to repair with GOCAD's standard tools, such as flip triangles, but in large meshes with thousands of triangles they are hard to find. For this reason it might be interesting to try out other meshing tools that support constrained Delaunay triangulation, such as Shewchuk's Triangle package (Shewchuk, 1996) or the Computational Geometry Algorithm Library (CGAL, www.cgal.org). Further it might be an interesting option to integrate the constrained Delaunay triangulation directly into GOCAD as this would simplify and speed up the workflow and would allow us to experiment with other potential applications. The tetrahedralization with the software TetGen works well if the given triangulation/boundary representation model does not have any errors, such as hanging nodes or holes. If it has any errors, the given information on them can be very sparse, so that it would be helpful to integrate an algorithm into GOCAD that checks all necessary predicates and finds hanging nodes, holes, crossing lines or lines that intersect faces and highlights them directly within GOCAD.

In order to provide an uncomplicated seamless workflow from the geological interpretation via simulation to visualization, we have interfaced GOCAD with different tools so that it can be used in the preprocessing for our simulator and support the postprocessing and visualization of the results. The overall data flow is shown in Figure 8. We have tested this workflow successfully within the INFLUINS project.

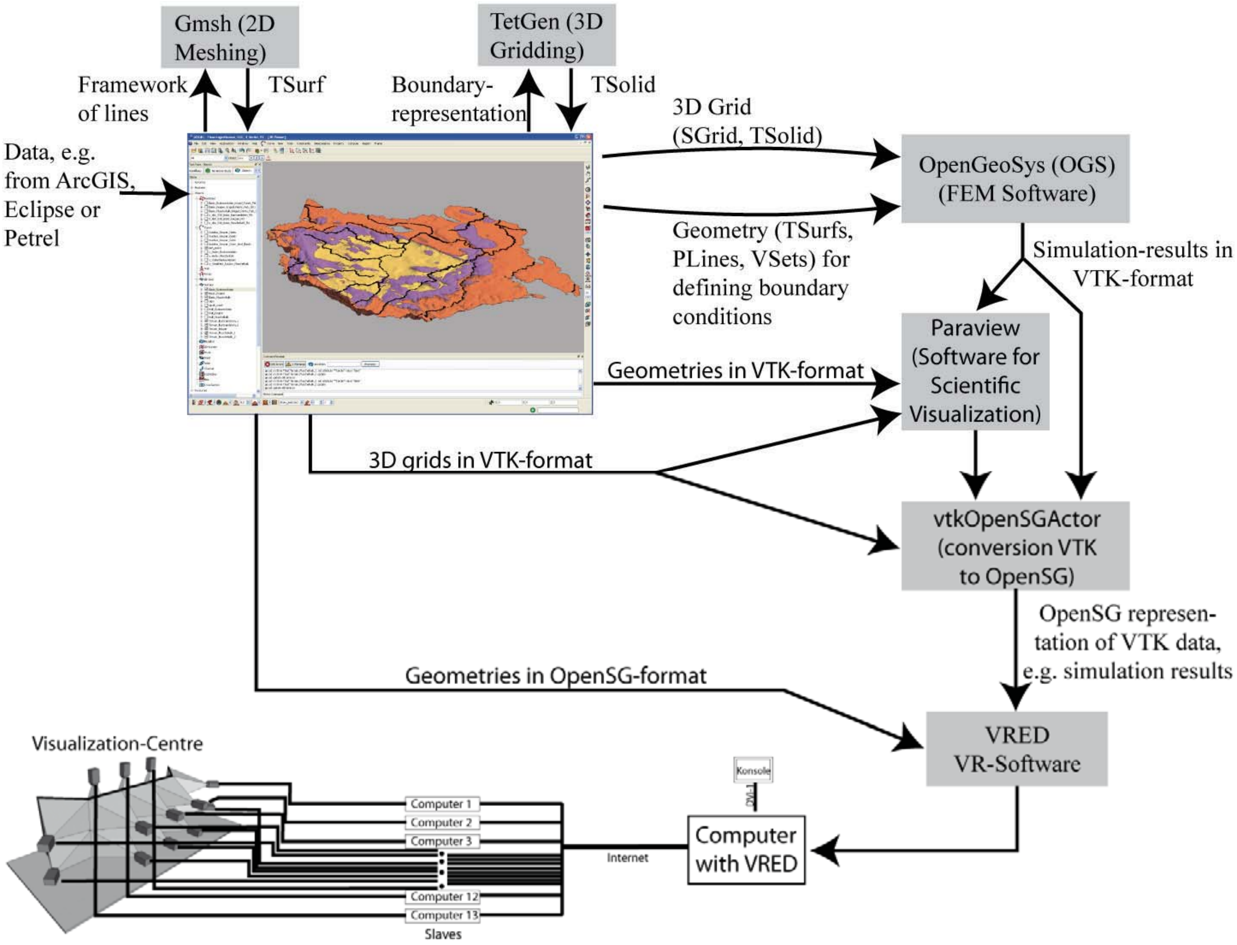
Gmsh (2D Meshing)
TetGen (3D Gridding)
Framework of lines
TSurf
Boundary-representation
TSolid
Data, e.g. from ArcGIS, Eclipse or Petrel
3D Grid (SGrid, TSolid)
OpenGeoSys (OGS) (FEM Software)
Geometry (TSurfs, PLines, VSets) for defining boundary conditions
Simulation-results in VTK-format
Paraview (Software for Scientific Visualization)
Geometries in VTK-format
3D grids in VTK-format
vtkOpenSGActor (conversion VTK to OpenSG)
OpenSG representation of VTK data, e.g. simulation results
Geometries in OpenSG-format
VRED VR-Software
Visualization-Centre
Konsole
Computer 1
Computer 2
Computer 3
Computer 12
Computer 13
Slaves
Internet
Computer with VRED

Figure 8: Overall data-flow diagram for the workflow from modelling using GOCAD via simulation, using OpenGeoSys to the visualization in UFZ's virtual reality environment.

Acknowledgements

The Author thanks Prof. Reinhard Gaupp and Martin Kober from the University of Jena for giving us access to their data from the Thuringian Basin. Prof. Jonas Kley, Alexander Malz and Stephan Donndorf from the University of Jena provided sections through the Thuringian Basin and the geometrical model of a part of the Finne fault zone that we could use as a test case. Alraune Zech and Thomas Fischer from UFZ used the generated grids from the Thuringian Basin for testing groundwater simulations using OpenGeoSys and provided valuable feedback. Prof. Helmut Schaeben and his group from the Technical University in Freiberg were very helpful during the initial phase of this project when we were starting with GOCAD-modelling and -programming. Finally thanks go to Alison E. Martin for proofreading the manuscript.

References

BLÖCHER, G., CACACE, M., LEWERENZ, B., ZIMMERMANN, G. (2010): Three dimensional modelling of fractured and faulted reservoirs: Framework and implementation. Chemie der Erde, 70, Supplement 3, Elsevier, p. 145-153.

DE BERG, M., VAN KREVELD, M., OVERMARS, M., SCHWARZKOPF, O. (1997): Computational Geometry – Algorithms and Applications, Springer, 367p.

GEUZAINE, C., REMACLE, J.-F. (2009): Gmsh: a Three-Dimensional Finite Element Mesh Generator with Built-in Pre- and Post-Processing Facilities. International Journal for Numerical Methods in Engineering 79, Wiley, p. 1309 – 1331.

KOBER, M. (2008): Erstellung eines ArcGIS- und Petrel-basierten digitalen Untergrundmodells der Thüringer Mulde. Diplomarbeit, Friedrich-Schiller-Universität Jena, Germany, 114p.

LASZLO, M. (1996): Computational Geometry and Computer Graphics in C++. Prentice Hall, Upper Saddle River, USA, 266p.

MALLET, J.-L. (2002): Geomodeling. Applied Geostatistics Series, Oxford University Press Inc. Oxford, UK, 624p.

PELLERIN, M., LÉVY, B., CAUMON, G. (2011): Topological control for isotropic remeshing of non-manifold surfaces with varying resolution: application to 3D structural models. IAMG 2011 Conference, September 2011, Salzburg, Austria, accepted.

REINERS, D., VOSS, G., BEHR, J. (2002): OpenSG: Basic Concepts. OpenSG Symposium 2002. Available from: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.16.2385, last time visited July 2011.

SCHROEDER, W., MARTIN, K., LORENSEN, B. (1996): The Visualization Toolkit, An Object-Oriented Approach to 3D Graphics. Prentice Hall, Upper Saddle River, USA, 826p.

SHEWCHUK, J.R. (1996): Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In: M. C: Lin and D. Manocha (Editors), Applied Computational Geometry: Towards Geometric Engineering. Lecture Notes in Computer Science 1148, Springer, p.203-222.

SI, H. (2011): TetGen, A Quality Tetrahedral Mesh Generator and a 3D Delaunay Triangulator, http://tetgen.berlios.de/, last time visited July 2011.

WANG, W., KOSAKOWSKI, G., KOLDITZ, O. (2009): A Parallel Finite Element Scheme for Thermo-Hydro-Mechanical (THM) Coupled Problems in Porous Media. Computers and Geosciences 35, Elsevier, p.1631-1641.

ZEHNER, B. (2010): Mixing Virtual Reality and 2D Visualization - Using Virtual Environments as Visual 3D Information Systems for Discussion of Data from Geo- and Environmental Sciences. In: Proceedings of the International Conference on Computer Graphics Theory and Applications (GRAPP2010), p. 364-369. INSTICC Press, Available from: http://www.ufz.de/index.php?en=19329, last time visited July 2011.