



**HAL**  
open science

## Singular Structured Hexahedral Grid

Wan-Chiu Li, Cédric Borgese, Alexandre Benedicto, Nicolas Ray, Dmitry Sokolov

► **To cite this version:**

Wan-Chiu Li, Cédric Borgese, Alexandre Benedicto, Nicolas Ray, Dmitry Sokolov. Singular Structured Hexahedral Grid. RING Meeting 2021, Sep 2021, Vandoeuvre-lès-Nancy, France. hal-03375671

**HAL Id: hal-03375671**

**<https://hal.science/hal-03375671>**

Submitted on 13 Oct 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Singular Structured Hexahedral Grid

Wan-Chiu Li<sup>1</sup>, Cédric Borgese<sup>1</sup>, Alexandre Benedicto<sup>1</sup>, Nicolas Ray<sup>2</sup>, and Dmitry Sokolov<sup>2</sup>

<sup>1</sup>*Tessael, 615 rue du jardin botanique, F-54600 Villers-Lès-Nancy, France*

<sup>2</sup>*Pixel, Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France*

September 2021

## Abstract

One of the types of hexahedral grids that is the most commonly used for the subsurface is 3D structured grids. It is computational efficient, honoring the geometry of the faults and horizons of the geological model and the standard format of most of the current standard flow simulators. To build the grid, it requires that the faults to be distinguishable in two principal directions U and V.

When it is not the case, the current technologies require that the fault network be either simplified structure-wise or approximated geometry-wise by a stair-step representation, both at the expense of honoring the complexity and precision of the input geological model.

We propose a new method of building structured hexahedral grids without compromising the complexity of the structural model. Unlike current methods that generate the gridding based on a 3D parameterization UVW in which the U and V coordinates are constrained by separate families of surfaces respectively, our method does not have the limitation of classifying faults in U and V families. Our method uses a pair of parameterizations for the UV coordinates that can have local frame transformations (rotation+translation). Thus offering additional degrees of freedom during the gridding process to honor the geometry of faults on a larger family of complex geological models. These transformations are defined by transition functions that “glue” the local parameterizations back into a global one in a non-trivial way, called Global Parameterization. A hexahedral grid built this way might have singular vertices, which have valence not equal to eight. It can be used directly in next-generation unstructured flow simulators. We show through a simple treatment how to convert efficiently such a grid into a structured representation that is compatible with standard flow simulators. We call such grid Singular Structured Hexahedral (SSHex) Grid.

We present the results of simulations on SShex grids using standard flow and geomechanical simulation software. Then, we discuss the interesting potential of this new grid in the development of coupled flow-geomechanical simulation techniques.

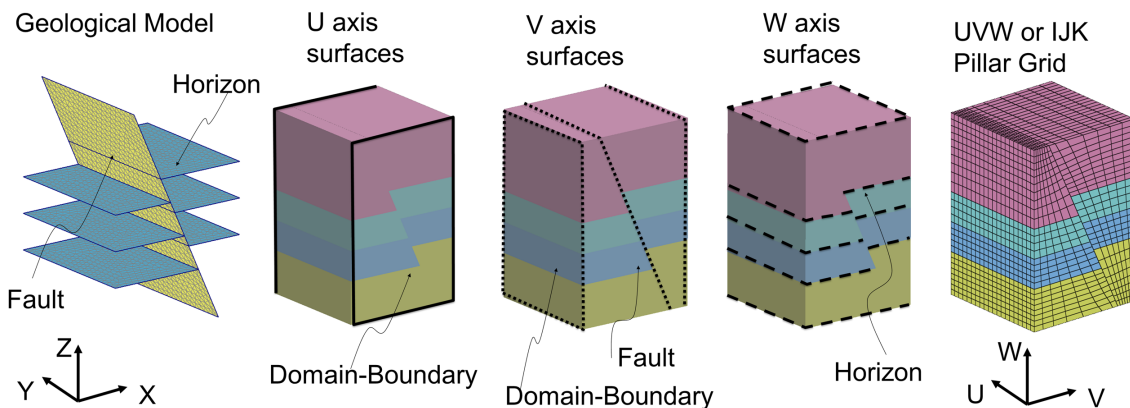


Figure 1: *Classic pillar structured grid with corner point geometry.*

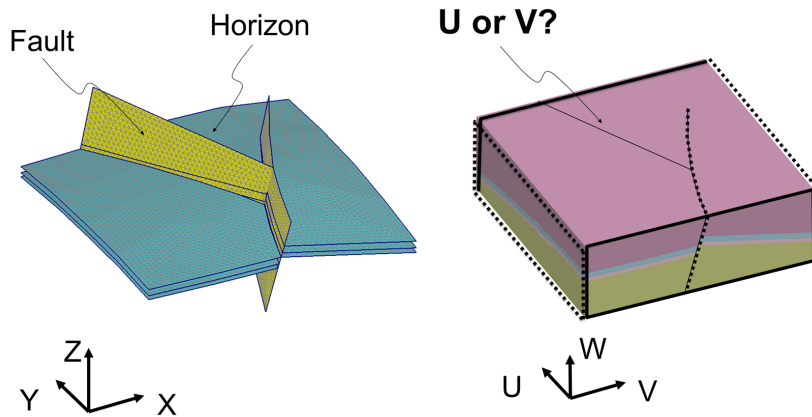


Figure 2: A "complex" fault network that cannot be meshed as a classic pillar grid.

## Introduction

In order to run reservoir simulations, one needs a volume grid on which mathematical equations describing physics can apply. Grids comprising hexahedral elements are often preferred thanks to the three axes being aligned naturally with the stratigraphic column and two principal directions of the fault network, and it gives a good calculation of the transmissibility between cells if provided good orthogonality and low distortion of the cells.

One of the types of hexahedral grids that is the most used in the industry are 3D structured grids with corner point geometry (commonly known as UVW, IJK or Pillar Grids), i.e. each hexahedral cell is indexed by a integer tuple  $[U, V, W]$ . This kind of grids are straight-forward to build. They are of low memory footprint and computational efficient to use. Most of the standard flow simulators (ECLIPSE, TEMPEST, PUMA, IMEX, tNavigator and MRST etc.) support it.

When the fault network of the model is simple, one trivial way of building it is by distinguishing the faults and the domain-boundaries in two U and V families and using them as geometric constraints to guide the gridding of intermediate layers in the respective direction. Typically, the U-family (V-family) constraints are used to build the U (V) parameterization defined on the domain volume mesh. The level-set of the U (V) parameterization is then used to define the gridding of the U (V) axis. The gridding in the W direction is computed similarly using the horizons in stratigraphic order as constraints.

This trivial way of structured grid building is interesting since one can use the same grid for coupled flow-geomechanical simulations using standard proprietary/open-source mechanical solvers such as VISAGE, ABAQUS (MONACO et al., 2011) and CODE-ASTER etc. One of the main advantages of using a single grid is that upscaling can be avoided.

## The problem we solve

When the fault network becomes more complex with intersections between the faults and domain-boundaries, it is sometimes impossible to separate the faults and domain-boundary surfaces into two U and V families. This makes it impossible to build a classic structured grid in the trivial way described above and thus not automatically benefiting from the simple flow-geomechanics coupling that such a grid offers.

Previous methods, described in the following, are proposed to solve the structured grid building problem on complex 2.5D fault networks. Although they all present interesting ideas, none of them offers a full hexahedral grid that is suitable to be used with standard off-the-shelf flow and geomechanical simulators at the same time.

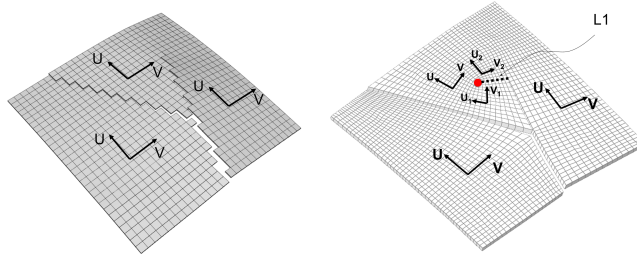


Figure 3: Our solution (right) uses rotation invariant  $UV$  basis to smoothly honor faults.

## Previous works

- *Structured:*

Methods like (GRINGARTEN et al., 2009) approximates faults using stair-step approximations to represent faults with a mix of  $U$  and  $V$  faces. The built grid is compatible with most of the current common flow simulators that substantially only accept structured hexahedral cells.

Drawbacks: When one wants to use the same grid to couple flow with mechanical simulations, the stair-step approximation of the faults is not suitable for applying mechanical boundary conditions that requires smooth approximation of the geometry. Typically, this means that two grids (or more) would have to be built. One stair-step grid for the flow and one tetrahedral mesh (or other unstructured grids) with smooth approximation of the geological features' geometry for the mechanical simulation. The coupling requires the transfer of petrophysical and mechanical properties between the two grids, which is an error-prone and time-consuming process. It also almost doubles the memory storage because there are two grids.

- *Partially unstructured:*

Some use mixed strategies: a majority of structured hexahedral elements in the non-faulted zones combined with the faulted zones modeled by arbitrary polyhedrons (MALLISON et al., 2013) or other predefined non-hexahedral elements like tetrahedrons and pyramids, prisms (GRINGARTEN et al., 2017; KLEMETSDAL et al., 2017) near faults.

Drawbacks: Not compatible with standard flow simulators that accept only structured hexahedral grids. A new generation of flow simulators (INTERSECT, GEOSX) that accepts unstructured grids has to be used.

- *Fully unstructured:*

For simulators using unstructured grids, one can also use tetrahedral mesh or fully unstructured topology such as Voronoi or PEBI grids (KLEMETSDAL et al., 2017; MERLAND, LEVY, & CAUMON, 2011; PALAGI & AZIZ, 1994).

Drawbacks: In addition to those of partially unstructured grids, fully unstructured grids are even more memory consuming and computationally inefficient since the topology and geometry of the cells can be very different and not predefined and need explicit costly storage.

## Our contributions

Current structured and partially structured methods mentioned above generate the gridding based on a 3D parameterization  $UVW$  in which the  $U$  and  $V$  coordinates (or axes) are following two different sets of surfaces as constraints, i.e. the  $UV$  basis is fixed (Figure 3-left).

In order to deal with models with complex 2.5D fault network, we introduce the idea of using a pair of parameterizations for the  $UV$  coordinates that allows local frame transformations (rotation+translation), which offers additional degrees of freedom during the gridding process to honor the

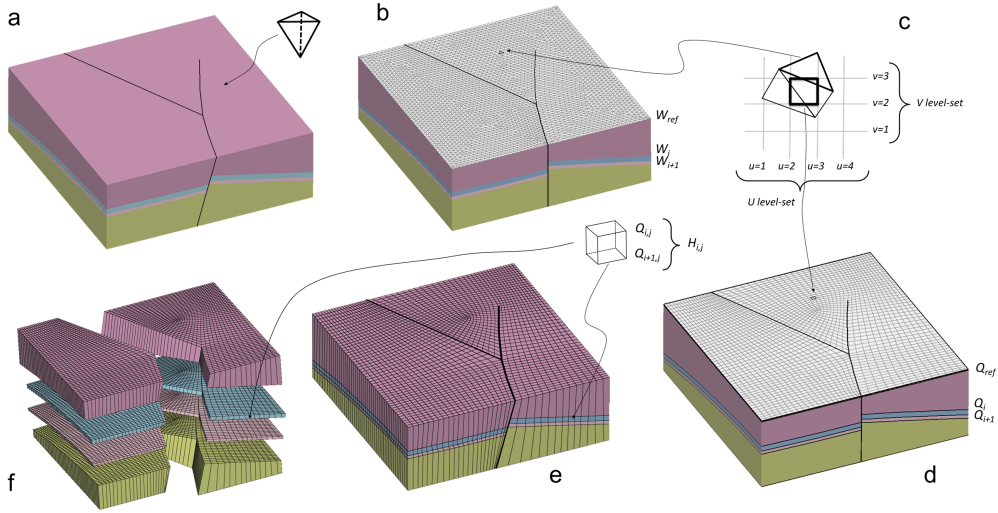


Figure 4: Main steps of generating a SSHex grid.

geometry of faults and horizons and domain-boundaries on a larger family of complex geological models. These transformations are defined by transition functions that “glue” the local parameterizations back a continuous global one in a non-trivial way. For instance, along the line  $L1$  in (Figure 3-right),  $(U_2, V_2) = rot_{\pi/2}(U_1, V_1)$ . This kind of parameterizations is called Global Parameterization (RAY et al., 2006).

Based on this concept of Global Parameterization, we propose a method of building full hexahedral grids from geological models with complex 2.5D fault network like the one shown in Figure 2. A hexahedral grid built this way might have singular vertices, which have valence not equal to eight. It can be used directly in next-generation unstructured flow simulators. We show through a simple treatment to convert efficiently such a grid into a structured representation that is compatible with standard flow simulators. We call such a grid Singular Structured Hexahedral (SSHex) Grid.

The rest of the paper is organized as follows: we start with presenting the main method in Section 1 to build SSHex grids from geological models with complex 2.5D fault network. Then, we discuss some results in Section 2, followed by conclusions.

## 1 Method of generating a singular structured hexahedral grid

In this section, we explain the major steps (Figure 4).

### 1.1 Compute 3D tetrahedral tessellation

A 3D tetrahedral mesh is computed from the input B-Rep (Boundary Representation) structural model using Constrained Delaunay Tessellation (SI, 2015) using the model’s faults, horizons and domain-boundaries as surface constraints. These surfaces are typically represented as triangulated meshes.

### 1.2 Compute UV gridding

In order to build the quad gridding for the UV axes, a UV global parameterization is first computed on the tetrahedral mesh using the faults and domain-boundaries as geometric constraints. Although one can compute the UV global parameterization using a 3D formulation (SOKOLOV et al., 2016), for the sake of minimizing the number of singularities, better controlling the placement of the singularities and achieving better computational efficiency and memory footprint, we propose an algorithm that solves the problem in a 2.5D approach. Namely, we reduce the original 3D problem to a 2D one (Section

1.2.1), solve the 2D Global Parameterization (Section 1.2.2) and compute the UV gridding (Section 1.2.3) in 2D. This 2D solution is then extruded in the third  $W$  dimension (Sections 1.3 and 1.4).

### 1.2.1 Verticalize constraint surfaces

The reduction of the computation of the UV global parameterization from a 3D problem to a 2D problem is through a process of verticalization of the surface constraints, i.e. fault and domain-boundary surfaces. This way, the UV global parameterization can be solved merely on a 2D section of the volume, for instance, a horizon surface of the model. The verticalization is achieved by computing a new deformed geometry of the original tetrahedral mesh in a consistent way using the verticalized version of the fault and domain-boundary surfaces as geometry constraints. The following two-step algorithm is applied to compute the deformed model  $\Omega(T)$ :

- a) For each fault and domain-boundary surfaces  $S_i$ , which is a triangulated mesh comprising a subset of triangles that are faces of the underlying tetrahedral mesh  $T$ , a new geometry  $(X_\Omega(S_i), Y_\Omega(S_i), Z_\Omega(S_i))$  of each of the vertices of the triangulated mesh  $S_i$  is computed in such a way that  $S_i$  is verticalized. This is achieved by a numerical optimization process minimizing an energy defined using the distance between the original and new geometry of  $S_i$ , subject to the verticality constraint requiring that the  $z$ -component of the normal of all the triangles in the resulting geometry to be zero.
- b) The vertical geometry of the surfaces is then used to compute the deformed geometry on the vertices of the tetrahedral mesh  $(X_\Omega(T), Y_\Omega(T), Z_\Omega(T))$  by means of interpolation. This is done by a numerical optimization process minimizing the distance between the original and deformed geometry of  $T$ , subject to the verticality constraints defined on the vertices of the surfaces  $S_i$ .

### 1.2.2 Compute Global Parameterization

This step computes the UV global parameterization on a chosen horizon in the deformed model. We illustrate this by identifying as the reference surface the top horizon  $W_{ref}$  that has a flat geometry in this case but it could be of any curved horizon in the model, such as the one in Figure 3. The reference surface is a triangulated mesh comprising a subset of triangles that are faces of the underlying tetrahedral mesh  $\Omega(T)$ . To compute the global parameterization, we adapt a classic two-step frame field approach (BOMMES, ZIMMER, & KOBELT, 2009; KÄLBERER, NIESER, & POLTHIER, 2007; RAY et al., 2006).

- a) Interpolate frame field

A smooth frame field (field of sets of 4 unit vectors invariant by a  $\pi/2$  rotation) is computed on  $W_{ref}$ . In order that the UV gridding follows precisely the input geometry of faults and domain-boundary surfaces, this field is constrained using the curve given by  $\delta W_{ref}$  (see RAY et al., 2006). Such strong geometric constraint can generate sometimes singularities in the frame field and consequently in the global parameterization (red dot in Figure 3). These singularities correspond to the extraordinary vertices (i.e. valence not equal to 4) in the UV quad gridding extracted from the global parameterization.

- b) Integrate frame field

The global parameterization is obtained by integrating the computed frame field, while imposing integer constraints at  $\delta W_{ref}$  and singularities (see BOMMES, ZIMMER, & KOBELT, 2009). The global parameterization  $\Phi$  is represented on  $W_{ref}$  as a pair of real number coordinates  $(U, V)$  stored on each of the three corners of each triangle of  $W_{ref}$ . The coordinates are defined on the whole triangle through a piece-wise linear interpolation scheme. By integration, we mean finding two scalar functions  $U, V$  whose gradients are as close as possible to the input frame field under the integer constraints. Due to the 4-symmetry characteristic of the input frame field, non-identity transfer function between pair of adjacent triangles is allowed:  $(U_2, V_2) = rot_{\pi/2}^i(U_1, V_1) + (j, k)$ , where  $i, j, k$  can be non-zero integers.

### 1.2.3 Extract quadrilateral grid

The U and V level-sets of iso-lines with integer coordinate values define a quadrilateral grid on  $W_{ref}$  such that its boundary  $\delta W_{ref}$  is entirely represented by a subset of boundary edges of the quadrilateral mesh, namely, honoring the geometry constraint given by  $\delta W_{ref}$ . These level-sets and their intersections in each triangle of  $W_{ref}$  are computed as pre-image of the regular grid defined in the local parametric space. Vertices are created for the intersections of U and V integer iso-lines. Edges are created between two vertices if they have the same coordinate on one axis and an absolute difference of one on the other axis (see LI, LÉVY, & PAUL, 2005). Quadrilaterals are then identified and extracted from these pre-images with particular care when handling neighboring triangles with non-trivial transition functions and triangles around the singularities. All quadrilaterals are stored as a single grid  $Q_{ref}$  that defines the gridding for the UV axes. The size of the quadrilaterals can be determined by the user-defined density of the iso-lines.

## 1.3 Compute W gridding

In order to build the gridding for the W axis, a W parameterization is first computed on the tetrahedral mesh from Section 1.1 using the horizons as geometric constraints. The W parameterization is a scalar function of the coordinate  $W$  defined on the vertices of the tetrahedral mesh. The horizons are first sorted in a stratigraphic order and assigned a real number in function of its stratigraphic depth. All the vertices of the same horizon are constrained to have same assigned value. The parameterization is then obtained by interpolating from the assigned values on the vertices of all the horizons by solving a system of linear equations. The W gridding is then given by the level-set of iso-surfaces  $W_i$ .

## 1.4 Build hexahedral grid

By combining the UV gridding given by the quadrilateral mesh  $Q_{ref}$  and the W gridding given by the level-sets of the W parameterization  $W_i$ , for each of the quadrilateral elements  $Q_{i,j}$  in the quadrilateral grids except the last one, a hexahedron  $H_{i,j}$  is built using it and the quadrilateral element immediately below it, i.e.  $Q_{i+1,j}$  (see Figure 4-d). The hexahedrons  $H_{i,j}$  built in this deformed space  $\Omega(T)$  are transformed back to the original space using the inverse of the deformation  $\Omega^{-1}(T)$  (see Figure 4-e). All the hexahedrons built constitute a hexahedral grid honoring all the faults, horizons and domain-boundary constraints in the input model (Figure 4-f is an exploded view of the hexahedral grid showing this).

## 1.5 Convert from unstructured to structured grid format

The built hexahedral grid can be unstructured if there are singular vertices, namely, the cells cannot be indexed in a trivial way using a tuple  $[U,V,W]$  in a single index space. Such a grid is compatible with flow simulators using unstructured grid such as INTERSECT.

We propose a method that allows such grids with singularities to be compatible with standard flow simulators accepting only structured grids.

### 1.5.1 Partition hexahedral grid

We first partition the grid into sub-grids. Each of them can be indexed locally in a trivial way using a tuple  $[U,V,W]$ . Since all the W planes of cells share the same UV gridding or topology, the partitioning only needs to be done on the quadrilateral grid  $Q_{ref}$ . First, the singularities are identified, a cut line starts from each of the edges departing from a singularity and stops when it intersects the boundary  $\delta Q_{ref}$  or another cut line (see Figure 5-left). These cut lines segment  $Q_{ref}$  into a set of sub-grids in which each quadrilateral can be indexed using a pair  $[U,V]$  in a trivial way. Through an extrusion in the W axis, these cut lines become cut planes that similarly segment the final hexahedral grid into a set of hexahedral sub-grids in which each hexahedron can be indexed using a tuple  $[U,V,W]$ .

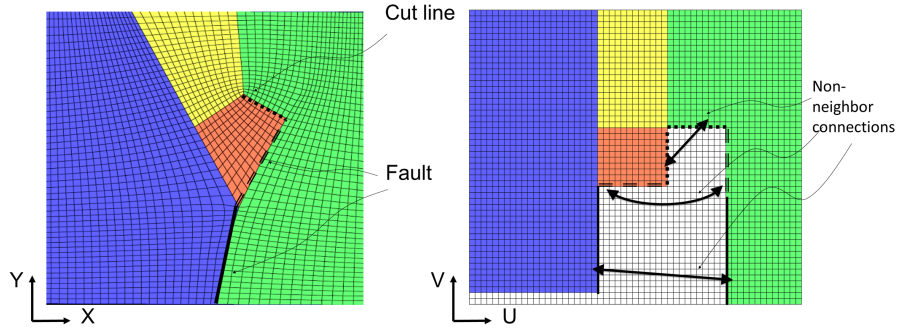


Figure 5: *Converting a singular hexahedral grid into a structured grid.*

### 1.5.2 Pack sub-grids

Each of the sub-grids has its own local index space that can potentially conflict with other sub-grids. In order to make the grid compatible with standard flow simulators using a global structured grid indexing, the individual local index spaces of the sub-grids have to be packed into a global index space. Again, we only have to do the packing on the UV index space once and propagate the result to the third dimension. The packing is a NP-complete problem. We propose a **Tetris**-inspired packing algorithm (see Figure 5-right) in order to minimize the number of unoccupied indices in the global index space.

### 1.5.3 Compute non-neighbor connections

After the packing process, along the fault surfaces and cut planes, pairs of geometrically touching or connecting hexahedrons are given  $[U, V, W]$  indices in such a way that they might not be immediate neighbors in the index space. Namely, neither of the absolute value of the difference of the U or V coordinates equals to one. For standard flow simulators, these geometric contacts have to be explicit given. To do so, we compute explicitly these connections for the packed grid, along all the faults and created cut planes (see Figure 5-right).

## 2 Results and discussions

The SSHex grid (9260 cells in Figure 4-e) generated using the proposed method is used to run tests of pressure and oil saturation using ECLIPSE-100 flow simulator. The fault transmissibility is set to zero. By comparing the flow patterns of an injection well (A) placed in the vicinity of the only singularity in the 2D topology and one (B) placed in an area with ordinary gridding, we observe that both gave similar expected results (see Figure 6-top).

Then, through a run of 3D restoration (see Figure 6-middle) using Kine3D-3, we compute the stress and strain properties (see Figure 6-bottom) of the present-day model using RUDKIEWICZ (2021). This demonstrates that thanks to the smooth approximation of the fault geometry, the SSHex grid is suitable for mechanical simulation purposes.

In Figure 7, we show another example of a similar flow simulation on a more complex model. The generated SSHex (70760 cells) has 3 singularities in the 2D topology. We place in their vicinity a production well each. A fourth production well is placed as reference in an area where the gridding is ordinary. Each such well is paired with an injection well. Expected pressure and oil saturation values are observed as shown in Figure 7. In particular, one can observe 1) the source-sink effect on the injection-production well pairs; 2) the interesting effect of the production well P1 under the influence of two neighboring injection wells; 3) the constant value in the compartments sealed off by faults containing no wells.

On the performance side, the grid generation for the two models took less than 3 minutes each on a 6-core 2.6GHz laptop with 16GB RAM. The method scales well with the complexity of the faults.



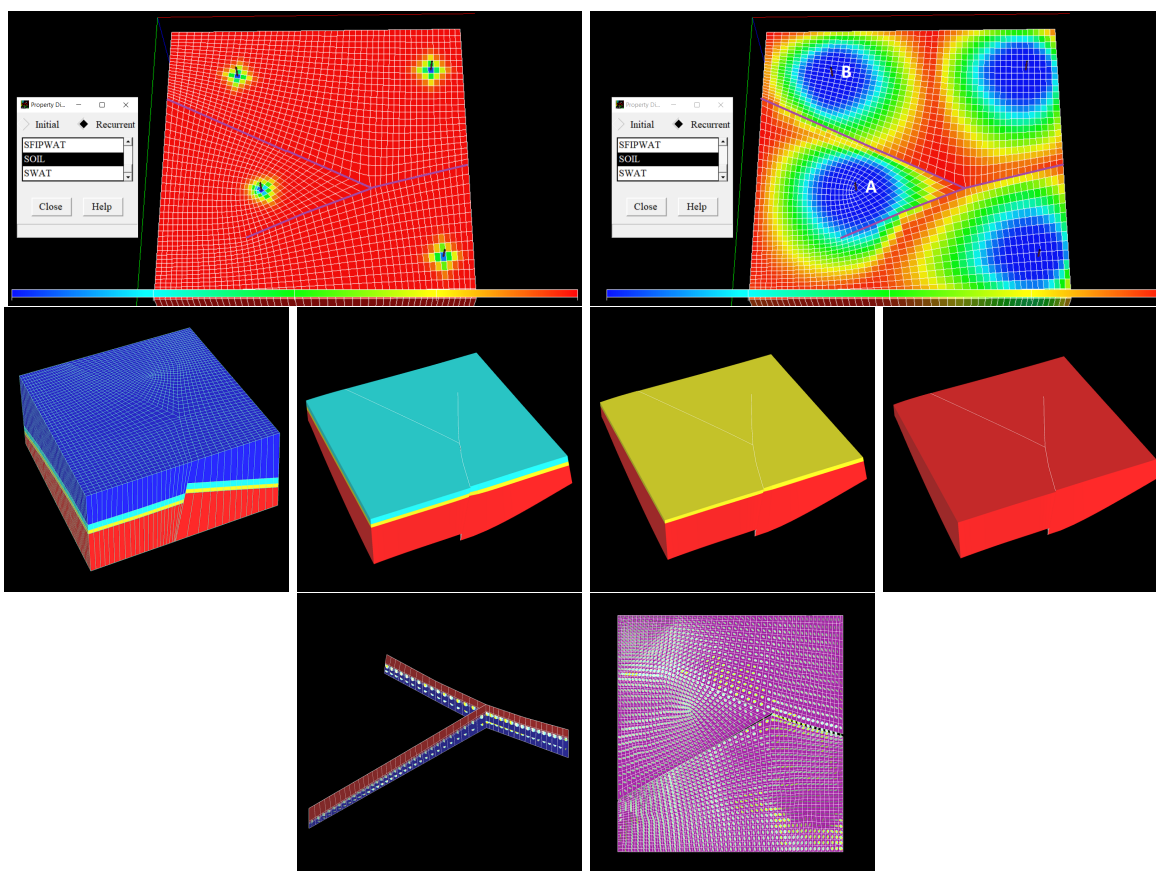


Figure 6: Pressure and oil saturation simulated in ECLIPSE (top); different stages of 3D restoration obtained in Kine3D-3 (middle); strain property derived from the 3D restoration displayed as ellipses on faults and a horizon (bottom).

## Conclusions

This paper introduces a new hexahedral grid building method for reservoirs by using the concept of Global Parameterization. A simple processing can be done to make this grid, which is unstructured by nature due to the singularities, compatible with standard flow simulators. This enlarges considerably the family of fault networks that one can handle using structured hexahedral grids without recourse to stair-step fault approximation or unstructured grids.

Encouraging results of flow and geomechanical simulations respectively show that we are on a good path towards a powerful framework of coupled flow-geomechanical simulation. The next step would be to couple flow and geomechanics to fully leverage the benefit of our single-grid approach where upscaling is not needed. Testing using commonly available tools such as ECLIPSE and ABAQUS would show the benefit of our method on treating much more complex fault network by capitalizing existing infrastructure in a cost-effective way.

## Acknowledgments

We thank RING-GOCAD Consortium for accepting our paper for presentation as non-member; Schlumberger for sponsoring the Eclipse software license during the summer internship of Alexandre Benedicto at Tessaël; and IFPEN for providing the stress and strain computation results using the Kine3D-3 software.

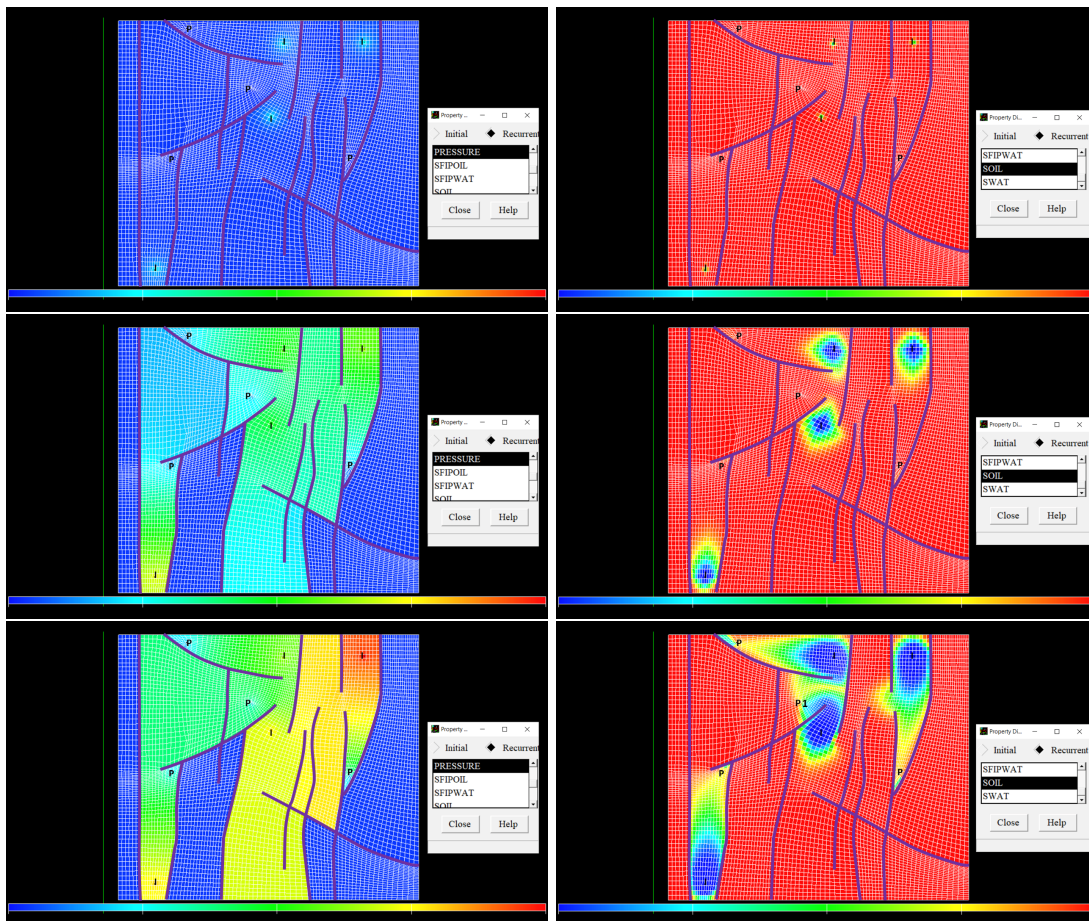


Figure 7: Pressure and oil saturation simulated in ECLIPSE on a model with 13 faults.

## References

- BOMMES D, ZIMMER H, & KOBELT L. (2009). Mixed-Integer Quadrangulation. *ACM Transactions on Graphics*, 28.
- GRINGARTEN E, HAOUESSE A, ARPAT B, & NGHIEM L. (2009). Advantages of Using Vertical Stair Step Faults in Reservoir Grids for Flow Simulation. In *Proc. SPE Reservoir Simulation Conference*.
- GRINGARTEN E, LECUYER JD, VILLARUBIAS E, COSSON C, & LI WC. (2017). Optimized Grids for Accurately Representing Geology in Geomechanical Simulations. In *Proc. SPE Annual Technical Conference and Exhibition*.
- KÄLBERER F, NIESER M, & POLTHIER K. (2007). QuadCover - Surface Parameterization using Branched Coverings. *Comput. Graph. Forum*, 26, 375-384.
- KLEMETSDAL S, BERGE RL, LIE KA, NILSEN HM, & MØYNER O. (2017). Unstructured Gridding and Consistent Discretizations for Reservoirs with Faults and Complex Wells. In *Proc. SPE Reservoir Simulation Conference*.
- LI WC, LÉVY B, & PAUL JC. (2005). Mesh Editing with an Embedded Network of Curves. In *Proc. IEEE International Conference on Shape Modeling and Applications* (p. 62-71).
- MALLISON B, SWORD C, VIARD T, MILLIKEN W, & CHENG A. (2013). Unstructured Cut-Cell Grids for Modeling Complex Reservoirs. *SPE Journal*, 19.
- MERLAND R, LEVY B, & CAUMON G. (2011). Building PEBI Grids Conforming To 3D Geological Features Using Centroidal Voronoi Tessellations. In *Proc. IAMG*.
- MONACO S, CAPASSO G, DATYE D, MANTICA S, & VITALI R. (2011). Field Scale Geomechanical Modelling Using a New Automated Workflow in Abaqus. In *Proc. SPE - International Petroleum Technology Conference*.

- PALAGI CL, & AZIZ K. (1994). Use of voronoi grid in reservoir simulation. *SPE Advanced Technology Series*, 2, 69-77.
- RAY N, LI WC, LÉVY B, SHEFFER A, & ALLIEZ P. (2006). Periodic Global Parameterization. *ACM Transactions on Graphics*.
- RUDKIEWICZ JL. (2021). Deriving a Restoration Workflow to Compute Paleo Stress. In *Proc. 2021 RING Meeting*.
- SI H. (2015). TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator. *ACM Transactions on Mathematical Software*, 41, 1-36.
- SOKOLOV D, RAY N, UNTEREINER L, & LÉVY B. (2016). Hexahedral-Dominant Meshing. *ACM Transactions on Graphics*, 35(5), 1 - 23.