Photos: DOI: 10.1051/ogst/2013204, IFPEN, X.

Dossier

This paper is a part of the hereunder thematic dossier published in OGST Journal, Vol. 69, No. 4, pp. 507-766 and available online here

Cet article fait partie du dossier thématique ci-dessous publié dans la revue OGST, Vol. 69, n°4 pp. 507-766 et téléchargeable ici

DOSSIER Edited by/Sous la direction de : **Z. Benjelloun-Touimi**

Geosciences Numerical Methods
Modélisation numérique en géosciences

Oil & Gas Science and Technology – Rev. IFP Energies nouvelles, Vol. 69 (2014), No. 4, pp. 507-766

IFP Energies nouvelles

Oil & Gas Science and Technology – Rev. IFP Energies nouvelles, Vol. 69 (2014), No. 4, pp. 565-572

DOI: 10.2516/ogst/2014020

Dossier

Geosciences Numerical Methods
Modélisation numérique en géosciences

Hex-Dominant Mesh Improving Quality to Tracking Hydrocarbons in Dynamic Basins

B. Yahiaoui[1]*, H. Borouchaki[2] and A. Benali[1]

[1] *IFP Energies nouvelles, 1-4 avenue de Bois-Préau, 92500 Rueil-Malmaison Cedex - France*
[2] *University of Technology of Troyes, 12 rue Marie Curie, BP 2060, 10010 Troyes - France*
e-mail: brahim.yahiaoui@live.com - houman.borouchaki@utt.fr - abdallah.benali@ifpen.fr

* Corresponding author

Résumé — Amélioration de la qualité d'un maillage hexa-dominant pour la simulation de l'écoulement des hydrocarbures — On propose dans cet article une méthode pour régulariser au mieux les éléments d'un maillage hexa-dominant ; cette régularisation permet l'amélioration des calculs utilisés pour la simulation de l'écoulement des hydrocarbures dans des bassins. Ces bassins contiennent des failles qui rendent la géométrie du modèle assez complexe.
La méthode d'optimisation est adaptée à une méthode de construction de maillages appelée "Méthode des Grilles Contraintes" (CGM, *Contrainded Grid Method*). Les données sont sous forme de surfaces modélisant les horizons et contenant des tags pour reconstruire les failles. Ces surfaces failles sont considérées comme des contraintes pour pouvoir générer des discontinuités dans le maillage hexa-dominant. Afin de rendre les éléments du maillage hexa-dominant réguliers, on procède comme suit : on commence par régulariser les bords des horizons pour les rendre alignés le mieux possible dans la direction des z, ce qui signifie que les noeuds de bord ont approximativement les mêmes coordonnées xy. On résout ensuite une équation de Dirichlet pour appliquer un Laplacien 2D sur les horizons dans la direction des coordonnées xy. Comme les noeuds au bord ont approximativement les mêmes coordonnées xy, les transformations obtenues grâce à l'équation de Dirichlet sont approximativement les mêmes et les positions en (x, y) sont presque pareilles sur chaque horizon. On obtient ainsi un maillage vérifiant le critère d'orthogonalité en xy et ayant des connections en z proche d'une verticale.

Abstract* — *Hex-Dominant Mesh Improving Quality to Tracking Hydrocarbons in Dynamic Basins
— The proposed method regularizes as far as possible some verticals of a hex-dominant basin mesh; this regularization is used to optimize the numerical simulation of hydrocarbons flow in basins. The studied basins contain faults and constitute complex geometries.
This mesh optimization is adapted for a new way to mesh basins called the "Constrained Grid Method" (CGM). Data constitutes some horizon surfaces; the surfaces contain tags to reconstruct the fault surfaces which represent in our case some constraints to generate a hex-dominant mesh. To make a regular hex-dominant mesh, the following steps are applied: first, the borders of horizon surfaces are extracted and optimized to get the connections between the borders as vertical as possible; this means that the border nodes have approximately the same xy *directions. Next, a Dirichlet equation is solved to apply a Laplacian smoothing 2D on* xy *directions for each horizon surface. And as the connected* (x,y) *between the borders are approximately the same, as well as the implied the harmonic solutions, then the positions on* (x,y) *are minimized.*

INTRODUCTION

The current oil production increases year after year and the explored reservoirs are deeper. The exploration must use more methods to estimate the deeper oil fields. One of these methods is the reconstruction of basin evolution. To study this evolution some mathematics models are used to simulate the basins deformation and the flow of hydrocarbons into its basins, more details are available in references [12] and [13]. The current aim of work in this context is to find a method that can give a good approximation of the simulated hydrocarbons flow. The mesh can be adapted to the method used to improve the solution. In this work, an optimization method is proposed to improve the shape of the mesh elements. The optimal mesh in this work is a mesh that verifies the 2D orthogonal principle for the *xy* directions and contains connections as vertical as possible in the *z* direction.

Classical hexahedral meshing methodologies as describe for example in [1] and [2] cannot be applied in dynamic basin modeling. The optimization method proposed in this paper complements an approach in [3]. With this approach it is possible to obtain a hybrid mesh[1] which can be identified with a structural grid. As shown in this paper, the advantage on the approach is the improving of the mesh to have lesser non-hexahedral elements.

In the first part of this paper, a transformation to improve the mesh in the *xy* direction called Laplacian smoothing is described. This transformation is usually used to deform a 3D object or to smoothen some noisy surfaces. In this paper, the Laplacian smoothing is applied to improve the *xy* direction quality. After this optimization a relaxation method is proposed to align the connections in each layer to verify the condition for the *z* direction.

1 OPTIMIZATION FOR THE HEX-DOMINANT MESH

1.1 Constrained Grid Approach

The hydrocarbon flow simulation depends on the mathematics, the finite volume scheme and the mesh [14]. Indeed the mesh can be adapted to get more precisions in the calculation. The most frequently occurring type of mesh for the basins simulation is the hexahedral mesh. The calculation is precise enough but we can not create this kind of mesh for complex geometries (containing a non-negligible number of faults) and the deformation caused by the basin mechanics may easily make some hexahedra convex. The second most important type is the tetrahedral mesh which contains tetrahedral elements only, all shapes can be covered with this kind of mesh but the finite volume methods aren't sufficiently exact for geological purposes. In this work, another type of mesh is used, the mesh must be precise for the calculation and compatible with any basin shape. So the basin is filling as well as hexahedra and when it is difficult to put a hexahedron it is possible to fill the area with tetrahedra. To have the conformity in the mesh some pyramidal and prismatic elements are used to connect the hexahedra and tetrahedra. This kind of mesh are called hex-dominant mesh. To generate the hex-dominant mesh for the basin a new approach in [3] can be applied. The advantage of this approach is the minimization of non-hexahedral elements (tetrahedra, pyramids and prisms) and the compatibility with the geologic models. In our case, the given data are usually horizons (limit surfaces between the sedimentary layers) and fault surfaces (surface modeling the fractures on the model). To generate the hex-dominant mesh, the approach in [3] starts by the creation of unfolded horizons. Here the unfolding seams every fault in the horizon like in Figure 1 to be able to extract the border and the fault paths (intersection between the horizon and the faults). As we can see in the same figure: with the horizon borders some 2D grids are generated with Coons Method [4]. The grids are created with the same number of nodes and hence it is possible to associate the nodes between the horizons. The structured grid *ijk* can be imagined and we can try to cut it, but the advantage of this approach in [3] is a step called the matched node method. This method takes the grid and the fault paths generated by the unfolding and relocates every node into the nearest fault path. However some rules must be followed to apply this step:

1. the deformed quads must be not degenerated;
2. if the local number of node connections is not sufficient to relocate the node (great density of faults near the node) a local refinement is applied for all the horizon grids.

With this step it is possible to minimize the number of non-hexahedral elements after having cut the *ijk* grid. This improves the precision of the simulation result by using the constructed hex-mesh.

In the approach cited on the previous paragraph, the quality shape elements is not considered. The proceeding steps take place in the unfolded space. As consequence, it is necessary to report the generated mesh in the 3D space, by using a reverse transformation.

[1] Mesh contains elements with different shapes, in this work only the following element types are tolerated: hexahedra, tetrahedra, prisms and pyramids.

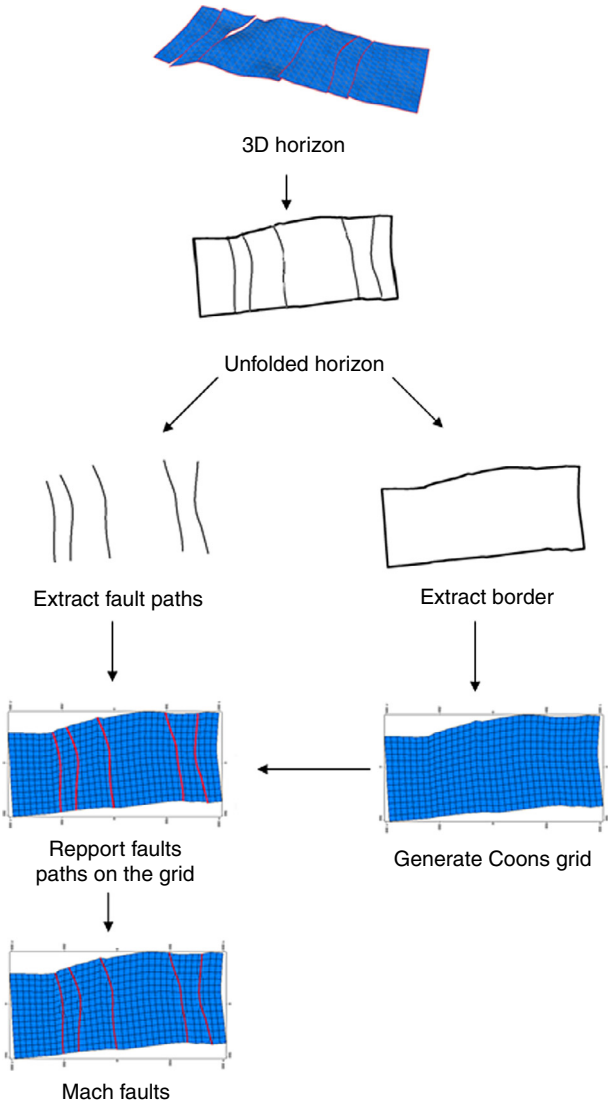

Figure 1
Constrained grid approach.

The quality of the 3D mesh thus depends on this transformation, which is imposed to us. The aim of this paper is to create an optimization method adapted for this approach. This optimization is then called when the horizons are reported in 3D and before having cut the 3D structured mesh. The proposed optimization uses the specificity *ijk* localization in the structured grid and let the nodes move[(2)] to converge towards a mesh with optimal angles. In this work, the optimal solution is a mesh

[2] The nodes are moved with respect to the original surface: the border nodes circulate on the border and the other nodes are moved and projected on the original surface corresponding.

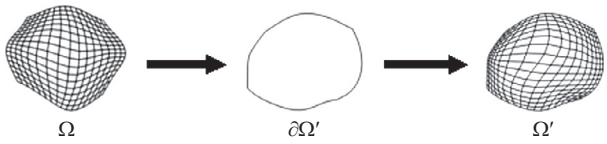

Figure 2
Laplacian smoothing principle.

which verifies the orthogonal principle in *xy* directions and the *ij* lines in *z* direction must be as vertical as possible. Therefore the following sections speak firstly about the optimization in *xy* directions only and after the optimization in *z* direction is explained to relate the two parts.

1.2 Optimization Based on the Diffusivity Principle

1.2.1 Dirichlet Principle

Let S be a 2D domain and R be its boundary. Let us consider a triangulation of S, denoted as the reference triangulation. Assume that S is deformed from a rigid displacement field associated to the boundary vertices of the reference mesh resulting to a deformed reference triangulation. The problem that we face is to modify the position of internal vertices of the deformed reference triangulation in order to minimize the deformation of the deformed reference mesh with respect to the initial reference mesh. Let us denote by $U := (u_1(x,y), u_2(x,y))$, the displacement field associated to the vertices. The minimization problem can be written as:

$$\min \|\nabla u_i\|_{L^2(\Omega)} \quad \text{for } i = 1,2$$

where Ω is the interior of S. It is expressed by:

$$\min_{u_i \in L^2(\Omega)} \left(\iint_{\Omega} \nabla u_i{}^t \nabla u_i dxdy \right) \quad i = 1,2 \qquad (1)$$

which is a particular case of Dirichlet principle minimizing the energy:

$$D_{\Omega,f}(u) = \iint_{\Omega} \|\nabla u\|^2 - uf \, dxdy \qquad (2)$$

where the source term $f \equiv 0$, which means that the solution belongs to the space of harmonic functions. The Jacobian of U is never null, therefore the displacement field is conformal. We obtain thus a Laplace problem with the displacement field $U_{|R}$ as boundary conditions (*Fig. 2*).

In other terms, if the boundary R of S is deformed, the internal vertices of the deformed reference triangulation of S can be relocated in order to obtain a unique valid triangulation with a uniform dispersion of vertices. Furthermore, this principle can be applied to solve a thermal diffusivity problem, thus this method has an effect of diffusion for the element shape quality. The next section, we recalls the algorithmic point of view to solve the optimization problem (see [5] and [6] for more details).

1.2.2 Algorithm to Solve the Optimization Procedure

Let D be the discrete operator $D_{S\cap R^c,0}$. We assume that U is a piecewise linear function over the reference triangulation. If $T = ABC$ is a triangle of the reference triangulation, we have:

$$D(\mathrm{U}_{|T}) = \iint_T \nabla \mathrm{U}^t \nabla \mathrm{U}\, dxdy$$

$$\simeq s_T \nabla \mathrm{U}_{|T}{}^t \nabla \mathrm{U}_{|T} = s_T \|\mathrm{U}_{|T}\|^2 \tag{3}$$

where s_T is the area of T.

Let V be a point belonging into T. Point V can be written as:

$$\exists \alpha, \beta, \gamma | V = \alpha A + \beta B + \gamma C \tag{4}$$

As U is linear on T, we have:

$$\mathrm{U}(V) = \alpha u_A + \beta u_B + \gamma u_C \tag{5}$$

where α, β and γ are the barycentric coordinates of V in T:

$$\alpha = \frac{\| \overrightarrow{VB} \wedge \overrightarrow{VC} \|}{\| \overrightarrow{AB} \wedge \overrightarrow{AC} \|},\ \beta = \frac{\| \overrightarrow{VC} \wedge \overrightarrow{VA} \|}{\| \overrightarrow{BC} \wedge \overrightarrow{BA} \|},\ \gamma = \frac{\| \overrightarrow{VA} \wedge \overrightarrow{VB} \|}{\| \overrightarrow{CA} \wedge \overrightarrow{CB} \|} \tag{6}$$

The gradient of each barycentric coordinate with $V = (x, y)$ is:

$$\nabla \alpha = \frac{1}{2s_T} \begin{pmatrix} y_B - y_C \\ x_C - x_B \end{pmatrix} \tag{7}$$

$$\nabla \beta = \frac{1}{2s_T} \begin{pmatrix} y_C - y_A \\ x_A - x_C \end{pmatrix} \tag{8}$$

$$\nabla \gamma = \frac{1}{2s_T} \begin{pmatrix} y_A - y_B \\ x_B - x_A \end{pmatrix} \tag{9}$$

and we obtain:

$$\nabla \mathrm{U}_{|T} = \frac{1}{2s_T} \times \begin{pmatrix} (y_B - y_C)u_A + (y_C - y_A)u_B + (y_A - y_B)u_C \\ (x_C - x_B)u_A + (x_A - x_C)u_B + (x_B - x_A)u_C \end{pmatrix} \tag{10}$$

which yields:

$$D(\mathrm{U}_{|T}) = \frac{1}{4s_T} \times \Big(\big((y_B - y_C)u_A + (y_C - y_A)u_B + (y_A - y_B)u_C \big)^2 + \big((x_C - x_B)u_A + (x_A - x_C)u_B + (x_B - x_A)u_C \big)^2 \Big) \tag{11}$$

This equation is obviously a quadratic form and therefore there exist a double sequence $(a_{ij})_{i,j\in\{1..3\}}$ such that we can write (11) as:

$$D(\mathrm{U}_{|T}) = \sum_i \sum_j a_{ij} u_i u_j \tag{12}$$

where $u_1 = u_A$, $u_2 = u_B$ and $u_3 = u_C$

As the quadratic form is semi-definite positive and for $u_i = u_j$, $\forall i,j$ we have $D(u_i) = 0$, by applying a result in [7], we obtain:

$$\exists (g_{ij}) | D(\mathrm{U}_{|T}) = \sum_j \sum_{i<j} g_{ij} (u_i - u_j)^2 \tag{13}$$

We calculate now g_A, g_B and g_C such that:

$$D(\mathrm{U}_{|T}) = g_A (u_B - u_C)^2 + g_B (u_C - u_A)^2 + g_C (u_A - u_B)^2 \tag{14}$$

The derivative of Equation (11) with respect to u_A is given by:

$$\frac{\partial D}{\partial u_A} = \frac{(y_B - y_C)}{2s_T} ((y_B - y_C)u_A + (y_C - y_A)u_B + (y_A - y_B)u_C) + \frac{(x_C - x_B)}{2s_T} ((x_C - x_B)u_A + (x_A - x_C)u_B + (x_B - x_A)u_C) \tag{15}$$

and that of equation (14) by:

$$\frac{\partial D}{\partial u_A} = -2g_B(u_C - u_A) + 2g_C(u_A - u_B) \tag{16}$$

Let $u_A = u_B = 0$ in Equations (15) and (16) and by comparing the two above Equations, we obtain:

$$g_B = \frac{1}{4s_T} ((y_C - y_B)(y_A - y_B) + (x_C - x_B)(x_A - x_B)) \tag{17}$$

Equation (17) can be simplified as:

$$g_B = \frac{\overrightarrow{CB} . \overrightarrow{AB}}{2\| \overrightarrow{CB} \wedge \overrightarrow{AB} \|} \tag{18}$$

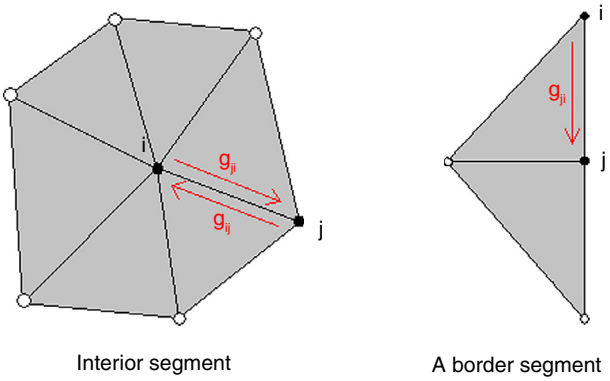

Figure 3

Disposition of angles intervening on the calculation.

By applying the same procedure for $u_C = u_B = 0$ and $u_A = u_C = 0$, we obtain:

$$g_A = \frac{\overrightarrow{BA} \,.\, \overrightarrow{CA}}{2\| \overrightarrow{BA} \wedge \overrightarrow{CA} \|} \tag{19}$$

$$g_C = \frac{\overrightarrow{AC} \,.\, \overrightarrow{BC}}{2\| \overrightarrow{AC} \wedge \overrightarrow{BC} \|} \tag{20}$$

Now considering the whole triangulation, the weight $(w_{ij})_{i,j\in 1..N}$ for each $(u_j - u_i)^2$, where N is the number of nodes, must be determined. Let us consider two vertices i and j, we denote by $j \leftrightarrow i$ if ij is an oriented edge (see *Fig. 3* which illustrate this notation).

The weight $w_{i,j}$ is defined by:

$$\begin{cases} w_{ij} = g_{ij} + g_{ji} & \text{if } j \leftrightarrow i \text{ and } i \leftrightarrow j \\ w_{ij} = g_{ji} & \text{if } i \leftrightarrow j \\ w_{ij} = 0 & \text{else} \end{cases} \tag{21}$$

where $g_{ij} = \frac{\overrightarrow{V_iV_k} \,.\, \overrightarrow{V_jV_k}}{2\|\overrightarrow{V_iV_k} \wedge \overrightarrow{V_jV_k}\|}$ and V_i, V_j and V_k are the vertices of triangle containing the oriented edge ij. In [8], another expression of the weight according to the 3D Laplacian Beltrami operator can be found. The weight w_{ij} is also used for surface flattening [9], surface deforming [10] or 2D transformations [15]. The approximation of the weight by $w_{ij} = \frac{1}{N_i}$, where N_i is the number of neighboring nodes of i, is also used for surface morphing purpose [11].

Now we have $D(\mathrm{U}) = \sum_{j\leftrightarrow i} w_{ij}(u_j - u_i)^2$ and its minimization yields to the following linear system:

$$\sum_{j\leftrightarrow i} w_{ij}(u_j - u_i) = 0 \tag{22}$$

This system can be expressed as: $LU = b$. To solve this linear system, the Gauss-Seidel method is applied. The deformation field at vertex i is thus the following:

$$u_i = \frac{\sum_{j\leftrightarrow i} w_{ij}u_j}{\sum_{j\leftrightarrow i} w_{ij}} \tag{23}$$

The following pseudo-code allows us to calculate L:

Determination of matrix L

```
L := initialisedAllCoefTo(0);
FOR f := 0 TO nomberOfFaces() - 1 DO
  FOR iLoc := 0 TO 2 DO
    globale(iLoc) := getVertexNum(iLoc, f);
  END
  FOR iLoc := 0 TO 2 DO
    jLoc := (iLoc + 1) mod 3;
    kLoc := 3 - iLoc - jLoc;
    g(kLoc) := calculateCoefG(kLoc);
    I :=max(globale(iLoc), globale(jLoc));
    J :=min(globale(iLoc), globale(jLoc));
    L(I,J) := L(I,J) + g(kLoc);
    ProfileL := calculateNewProfileOf(L);
  END
END
```

The advantage of this computation is the explicit dependence against data and the totally determination of L after one loop over the elements. Notice that for each triangle f, $(iLoc, jLoc)$ belongs to the graph $\{(0,1);(1,2);(2,0)\}$ where triangle vertices are locally referenced by $0, 1, 2$. The expression *calculateCoefG*$(kLoc)$ computes the component of L corresponding to edge (i,j) of triangle f which is added to the global term (I,J) of L.

1.3 Optimize the 3D Mesh

As it is explained in the constrained grid approach section the aim is to obtain a mesh that verifies the 2D orthogonal principle on xy directions and the z direction must be as vertical as possible. To have the verticals the xy directions should be the same for each grid. The idea is to have approximately the same xy directions for each corresponding node on the borders (*Fig. 4*). When the nodes on the border are corresponding it is possible to obtain approximately the same solution with the Laplacian smoothing transformation. Then the proposed solution aligns as nicely as possible the nodes on the borders and afterwards aligns the interior nodes using the Laplacian smoothing.

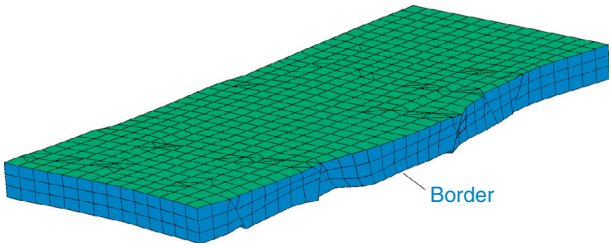

Figure 4

Area of first optimization in z direction (blue area).

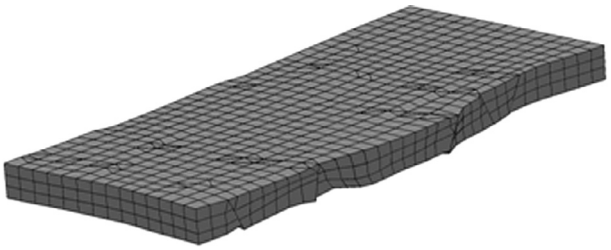

Figure 6

3D model.

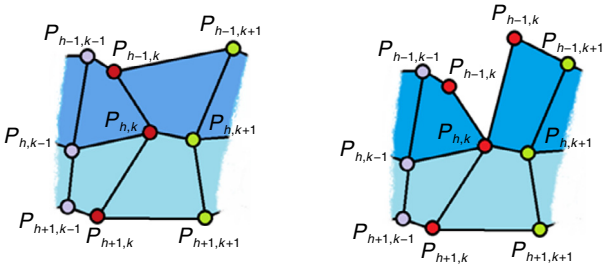

Figure 5

Illustration of hk connections ($h = i,j$).

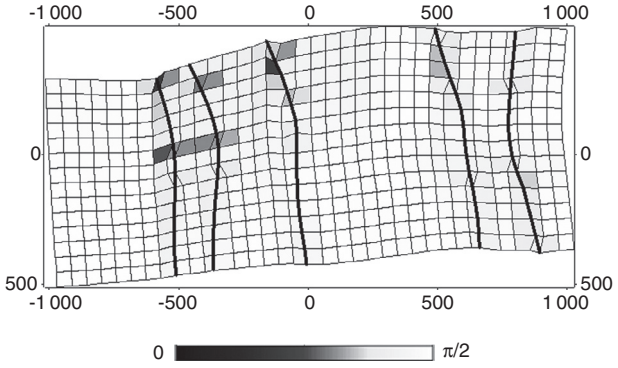

Figure 7

Optimized horizon on xy directions.

Before adding the fault surfaces in the CGM, the mesh contains only hexahedral elements. More specifically, this mesh is a 3D grid with splitted nodes and it is possible to retrieve the ijk connections. These connections are used to optimize the mesh on x, y and z directions for each horizon.

To align the border nodes a relaxation algorithm is used, so let $(P_{h,l})_{h=1\ldots N_h, l=1\ldots N_l}$ (*Fig. 5*), where N_h and N_l are respectively the number of horizons and the number of path connections between the borders. The nodes on the borders can be split and in this case the fault passes the node and is considered as rigid. For the rest of the nodes an average node $P^{l,iter}_{mean}$ is calculated for every iteration and for each path connection l. This average is calculated as following:

$$\overrightarrow{oP^{l,iter}_{mean}} = \frac{1}{iter * N_h} \sum_{n=1}^{iter} \sum_{h=1}^{N_h} \overrightarrow{oP^{iter-1}_{h,k-1}} + 1/2 \overrightarrow{P^{iter-1}_{h,k-1} P^{iter-1}_{h,k+1}}$$

In every iteration the nodes related to the path connection l move toward $P^{l,iter}_{mean}$ with a constant step ρ. The iteration is then applied until the movement is stabilized. However it is possible to use the Laplacian smoothing to align the nodes as well as possible when the transformation is approximately the same for all the grids.

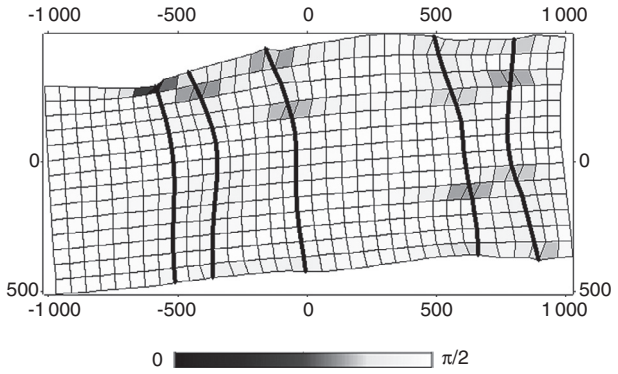

Figure 8

Initial horizon on xy directions.

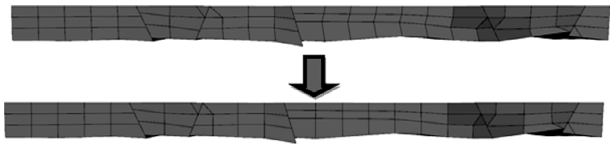

Figure 9

3D model of basin - first example.

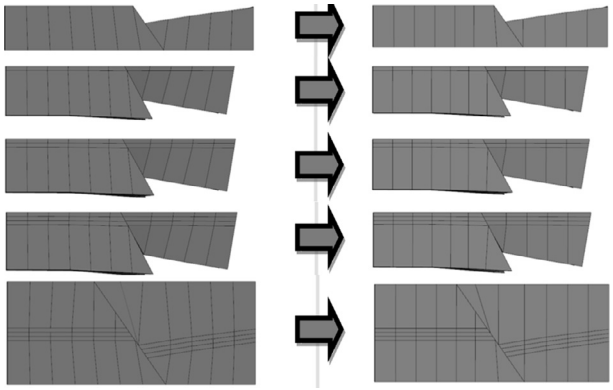

Figure 10

3D model of basin - second example.

2 RESULTS

2.1 Result for the 2D Optimization

In this section, an example is elaborated to compare the 2D grid before and after the optimization. It is important to note that the transformation of the Laplacian smoothing is defined for only boundary closed sets.

The example to solve is illustrated in Figure 6 and it contains five faults and six components. In the color scale, white is for the excellent elements, black for degenerated elements and the gray scale represents in between values. The quality calculated here represents the minimum of angles compared with $\pi/2$. The solution shown in Figure 7 is the result of optimization with the Laplacian smoothing. Then it may to see the improving of elements with clearer color compared with Figure 8.

2.2 Result for the 3D Optimization

In this section, we continue the same basin example solved in the Laplacian smoothing section, so this example contains three layers and five faults and is dynamic (at each time step, the mesh is deformed to take into account the corresponding horizon). The top of Figure 9 shows the example without optimization. It is possible to compare with the bottom of the same figure the mesh before and after optimization. For the second example a simple case is tested. The case contains six layers and one fault as we can see on the non-optimized mesh on the left the Figure 10. The comparison is shown in Figure 10 and it is possible to see in the non-optimized mesh a degenerated element at the last instant and near the fault. In the optimized mesh illustrated by the right of Figure 10, the degenerated element is optimized and the elements are better aligned.

CONCLUSIONS

As mentioned in this paper the approach in [3] creates a hex-dominant mesh with a lesser number of non-hexahedral elements. So the optimization proposed here is a method which complements this approach.

A relaxation method with fixed step is used to let the border nodes free to move for each horizon and to have almost verticals on the borders. This alignment constructs approximately the same border conditions for the Laplacian smoothing transformation and is used to have approximately the same results on *xy* directions. The resulting mesh verifies the conditions to be optimal in shape: 2D orthogonal condition for *xy* directions and verticals between the horizons.

So it is possible to control with the relaxation method the angles in a mesh. Indeed the interaction between nodes for each iteration can be used to get the required solution while imposing the angles. To improve the relaxation method it is possible to use the optimization with variable step and use a dichotomy process to have a more exact optimization.

REFERENCES

1 Remacle J.-F., Henrotte F., Carrier-Baudouin T., Béchet E., Marchandise E., Geuzaine C., Mouton T. (2013) A frontal Delaunay quad mesh generator using the L^{∞} norm, *International Journal for Numerical Methods in Engineering* **94**, 5, 494-512.

2 Owen S.-J., Shelton T.-R. (2013) Validation of Grid-Based Hex Meshes with Computational Solid Mechaniccs, *22nd International Meshing Roundtable*, Springer-Verlag, pp. 39-56.

3 Ran L., Borouchaki H., Benali A., Bennis C. (2012) Hex-dominant mesh generation for subterranean formation modeling, *Engineering with Computers* **28**, 255-268.

4 Coons S.A. (1967) *Surfaces for Computer-Aided Design of Space Forms*, MIT CSAIL Publications, Cambridge, MA.

5 Allaire G. (2007) *Numerical Analysis and Optimization. An Introduction to Mathematical Modelling and Numerical Simulation*, translated from french by Alan Craig, Oxford University Press.

6 Botsch M., Kobbelt L., Pauly M., Alliez P., Levy B. (2010) *Polygon Mesh Processing*, Peters A.K. (ed.), ISBN 978-1-56881-426-1.

7 Duffin R.J. (1959) Distributed and Lumped Networks, *Journal of Mathematics and Mechanics* **8**, 5, 793-826.

8 Kin-Chung Au O., Tai C.L., Liu L., Fu H. (2006) Dual Laplacian Editing for Meshes, *IEEE Transactions on Visualization and Computer Graphics* **12**, 3, 386-395.

9 Desbrun M., Meyer M., Alliez P. (2002) Intrinsic Parameterizations of Surface Meshes, *Eurographics 2002*, Vol. 21, No. 2, Oxford, UK.

10 Masuda H., Yoshioka Y., Furukawa Y. (2006) Interactive Mesh Deformation Using Equality-Constrained Least Squares, *Computer & Graphics* **30**, 6, 936-946.

11 Sorkine O., Lipman O., Cohen-Or D., Alexa M., Rössl C., Seidel H.P. (2004) *Laplacian surface editing*. In *SGP '04: Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pp. 175-184.

12 Hantschel T., Kauerauf A.I. (2009) *Fundamentals of Basin and Petroleum Systems Modeling*, Springer, Berlin.

13 Welte D.H., Horsfield B., Baker D.R. (1997) *Petroleum and Basin Evolution, Insights from Petroleum Geochemistry*, Geology and Basin Modeling, Springer, Berlin.

14 Versteeg H.K., Malalasekra W. (2007) *An Introduction to Computational Fluid Dynamics, the Finite Volume Method*, 2nd Edition, Prentice Hall, Essex.

15 Wang Y., Xu K., Xiong Y., Cheng Z.Q. (2008) 2D shape deformation based on rigid square matching, *Computer Animation and Virtual Worlds*, pp. 411-420, Changsha-China.

Manuscript accepted in March 2014
Published online in July 2014