Computers and Geosciences 110 (2018) 73–80

Contents lists available at ScienceDirect

Computers and Geosciences

journal homepage: www.elsevier.com/locate/cageo

Research paper

Local PEBI grid generation method for reverse faults

Xianhai Meng [a,b,*], Zhongxiang Duan [a,b], Qin Yang [a,b], Xing Liang [c]

[a] School of Computer Science and Engineering, Beihang University (BUAA), Beijing 100191, PR China
[b] State Key Laboratory of Software Development Environment, Beijing 100191, PR China
[c] Lithe IT Ltd., 69 Buchanan Street, Glasgow G1 3HL, UK

ARTICLE INFO

Keywords:
PEBI grid generation
Reverse faults
Voronoi neighbors

ABSTRACT

The 2.5D PEBI (PErpendicular BIsector) grid, which is the projection or extrusion of the 2D PEBI gird, has advantages on practical reservoir modeling. However, to appropriately handle the geological features, especially the reverse faults in reservoir, remains a difficult problem. To address this issue, we propose a local PEBI grid generation method in this paper. By constructing the Voronoi cell of a seed based on the search of its neighboring seeds in a background grid, our method is demonstrated to be efficient and adaptable to reverse fault constraints. In addition, the vertical and horizontal well constraints are also tackled and the cell quality is improved through the Centroidal Voronoi Tessellations (CVT) principle. The results demonstrated that our method enables the formation of high-quality grids and guarantees the conformity to the geological features in reservoirs.

1. Introduction

The computing accuracy, speed and convergence of the reservoir simulation are largely dependent on the grids. Compared to the Cartesian and Corner Point grids, which are commonly utilized in industry, the PEBI grid, also known as the constrained Voronoi Tessellation, commands much attention as it can reduce the orientation effect and adapt to complex structures. After reviewing early studies on reservoir simulation, Heinemann et al. (1991) claimed that the performance of PEBI grids on overcoming the grid-orientation effect is generally as good as the nine-point Cartesian grids and better than the five-point scheme. Palagi and Aziz (1994) presented the use of Voronoi grids for field scale simulations in combination with pre-defined geometrical modules that can be located, scaled and rotated in the domain, allowing a good representation of the major geological features in reservoirs.

A Voronoi cell is, by definition, always associated with a certain point, also known as the seed of the cell (Bertin et al., 1994). Since the aspect ratio of the horizontal scale to vertical in the reservoir field is often several orders of magnitude, the 2.5D Voronoi grids are usually used in reservoir simulation (Branets et al., 2009). These grids are constructed by projecting or extruding the 2D Voronoi grids in the vertical or nearly vertical directions (Gunasekera et al., 1997). In contrast to the direct generation of the Voronoi grids, such as the divide-and-conquer method (Shamos and Hoey, 1975) and plane sweep algorithm (Fortune, 1987), indirect schemes derived from the dual of a Delaunay mesh (Verma, 1996; Verma et al., 1997), are better appreciated owing to the gradual progress of the Delaunay triangulations. However, one of the key challenges is that the generated 2D grids are required to conform to some geological features, including boundaries, faults, vertical and horizontal wells, and pinch-outs. These structural constraints pose inconveniences for the PEBI grid generation. To resolve the faults with arbitrary size and orientation through Voronoi faces, in particular, becomes a more daunting task.

In the scheme that handles the faults proposed by Gunasekera et al. (1997), Voronoi seeds were set symmetrically on both sides of the faults so that the path of the faults would be part of the Voronoi cell edges. To resolve more complex structures in reservoir, Branets et al. (2009) suggested defining circular disks surrounding the constraints, where both the inside and outside of these protection areas can be split by Delaunay triangulations. In this way, a consistent dual constrained Voronoi grid is obtained. In addition, an approach to generate 3D PEBI grids was also introduced by Merland et al. (2014). They optimized the positions of the seeds by minimizing an objective function designed to meet the 3D structural features. The cells were strictly Voronoi yet the constraints were not exactly recovered.

To the best of our knowledge, most of the 2D PEBI grid generation algorithms tend to conduct a global tessellation according to the dual relationship between the Voronoi diagram and the Delaunay triangulation and improve the mesh quality through the Centroidal Voronoi Tessellations (CVT) concept (Du et al., 1999, 2010; Merland et al., 2011). However, the complexities of the faults sometimes render the global tessellation quite cumbersome to express in terms of constraints. This is especially true if the

* Corresponding author. School of Computer Science and Engineering, Beihang University, Beijing 100191, PR China.
E-mail address: x_h_meng@buaa.edu.cn (X. Meng).

https://doi.org/10.1016/j.cageo.2017.09.012
Received 9 March 2017; Received in revised form 9 July 2017; Accepted 4 September 2017
Available online 28 September 2017
0098-3004/

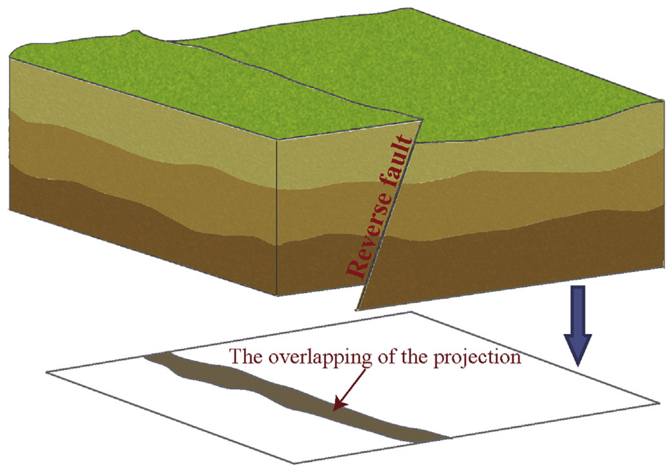

Fig. 1. Reverse fault and the 2D projection.

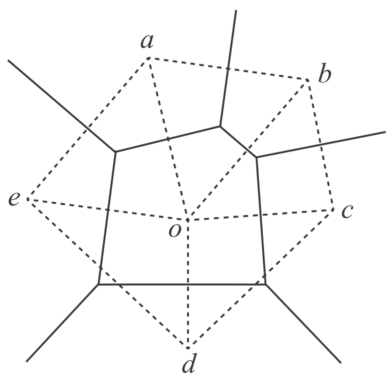

Fig. 2. Voronoi cell of o and the related *MVN*.

reservoir contains the reverse faults (Fig. 1), which it is difficult to avoid the overlapping after the projection or other simple mapping. The generation of the PEBI grids adapting to these faults becomes a tough problem due to the interference of the Voronoi seeds in the upper and lower parts of the fault area. As far as we know, none of these previous algorithms has addressed the reverse fault constraints in the PEBI grid construction.

PEBI grids have a local property, i.e., the shape and size of a PEBI cell are merely correlational with points neighboring the seed of the cell. Based on this characteristic, we present in this paper a novel method to build the PEBI grid that can conform to the reverse fault features. Every PEBI cell is constructed after the neighbors of its seed are searched with the help of background grid. The search strategy overcomes the interference of the seeds in the upper and lower parts of the overlapping in the reverse fault areas. Incorporating the CVT principle, we devise a strategy to generate the PEBI grids with both high quality and the conformity to the complex structures.

2. Local generation of PEBI grids

In this section, we present our local approach to the Voronoi grid generation with the basic idea of constructing the Voronoi cell of a seed according to its neighboring seeds. For a Voronoi cell, the neighbors of its seed are corresponding to its adjacent cells, which we define as the *Minimum Voronoi Neighbors* of the seed.

Definition 1. Let S and M be the set of the seeds and $M \subseteq S$. As to a single point $o \in S$, M is said to be the *Minimum Voronoi Neighbors* (*MVN*) of o if and only if $o \notin M$ and M contains $\forall p \in S$ that satisfies $Vor(p) \cap Vor(o) \neq \emptyset$.

Under the definition above, $Vor(*)$ means the Voronoi cell related to the seed and S is assumed to be in general position (Guibas and Mitchell, 1992). As is shown in Fig. 2, the *Minimum Voronoi Neighbors* of o is $\{a, b, c, d, e\}$.

According to the dual relationship between the Voronoi diagram and the Delaunay triangulation, the triangles, which are formed by connecting seeds related to adjacent Voronoi cells and sharing the same vertex o, are part of a Delaunay triangulation (Cheng et al., 2012). We call the set of these triangles the local Delaunay triangle set of o, denoted by *LDTSet* (illustrated with the dashed lines in Fig. 2). No seed, as the Delaunay triangulation is defined, falls strictly inside the circumcircle of any triangle in the *LDTSet*. In light of this fact, we design an incremental algorithm that successively adds the other seeds to the plane and replaces the elements in the *MVN* and *LDTSet* to guarantee that the circumcircles of the triangles in the *LDTSet* contain no seed.

As we show the instance in Fig. 3, the added seeds are $p_i (0 \le i \le 6)$ and before the *LDTSet* is closed (the triangles in *LDTSet* fully cover the neighborhood of point o), there are two seeds p^l and p^r where the ray op^l and op^r witness the triangles on only one side. The two rays divide the space into sector A and sector B while A is the one that contains the triangles in *LDTSet*. The signed areas $S_{\Delta(p_i,o,p^l)}$ and $S_{\Delta(p^r,o,p_i)}$ are calculated to determine which sector p_i is located in. It is noted that if any of the signed areas is positive, p_i will fall into sector B. We assume that p^l is identical to p^r and sector A covers the entire region after the *LDTSet* is closed (after Fig. 3(c)).

If the new seed p_i lies in sector B, we will connect it with the point o and p^l or p^r, with one or two new triangles brought in the *LDTSet* (Fig. 3(a) and (c)). Yet, if the seed is located in the circumcircles of the triangles in sector A, it will witness the replacement of old triangles with the new ones (Fig. 3(b) and (d)). Besides, as shown in Fig. 3(e), a few flips may also be executed to remove illegal edges for every added triangle in the *LDTSet* to maintain a Delaunay triangulation (Guibas et al., 1992). The final *MVN* set is the vertices of the triangles in the *LDTSet* except o, which is $\{p_0, p_2, p_3, p_4, p_6\}$ in Fig. 3(f). Afterwards, the Voronoi cell related to o can be constructed by collecting the perpendicular bisectors of the connected lines between o and its neighbors.

Finally, the procedure to adjust the *MVN* and *LDTSet* according to the added seed, which is depicted as the algorithm MVNTestForp and edgeLegalization, is explicated as follows.

Algorithm MVNTestForp($o, p, MVN, LDTSet$)

Input the seed o for MVN search;
the added seed p for test;
current MVN and $LDTSet$;

Output updated MVN and $LDTSet$;

1. **If** $MVN = \emptyset$, **then** mark p to be p^l and add it into MVN, **return**.
2. **If** MVN contains only one seed p^l.
   (a) **If** p lies on the segment op^l, then replace p^l with p, **return**.
   (b) **Otherwise**, add p into the MVN and update p^l and p^r. Add $\Delta(p^l, o, p^r)$ into the $LDTSet$, **return**.
3. **If** p lies in sector B, then add p into the MVN. For any of the two triangle $\Delta(p, o, p^l)$ and $\Delta(p^r, o, p)$ that has a positive signed area, add it into the $LDTSet$. Call edgeLegalization($op^l, MVN, LDTSet$) and edgeLegalization($op^r, MVN, LDTSet$) to maintain the $LDTSet$ to be a Delaunay triangularization.
4. **Otherwise**, p lies in sector A.
   (a) **If** p lies on the edge op_i, then replace the seed and vertex p_i with p for the MVN and triangles in the $LDTSet$. For any edge op_j opposite p, call edgeLegalization($op_j, MVN, LDTSet$).
   (b) **Otherwise**, let p_ip_j be the edge that intersects with the ray op. If p lies in the circumcircle of $\Delta(p_i, o, p_j)$, then add p into the MVN. Use $\Delta(p, o, p_j)$ and $\Delta(p_i, o, p)$ to replace $\Delta(p_i, o, p_j)$ in the $LDTSet$. Call edgeLegalization($op_i, MVN, LDTSet$) and edgeLegalization($op_j, MVN, LDTSet$).
5. Update p^l and p^r, **return**.

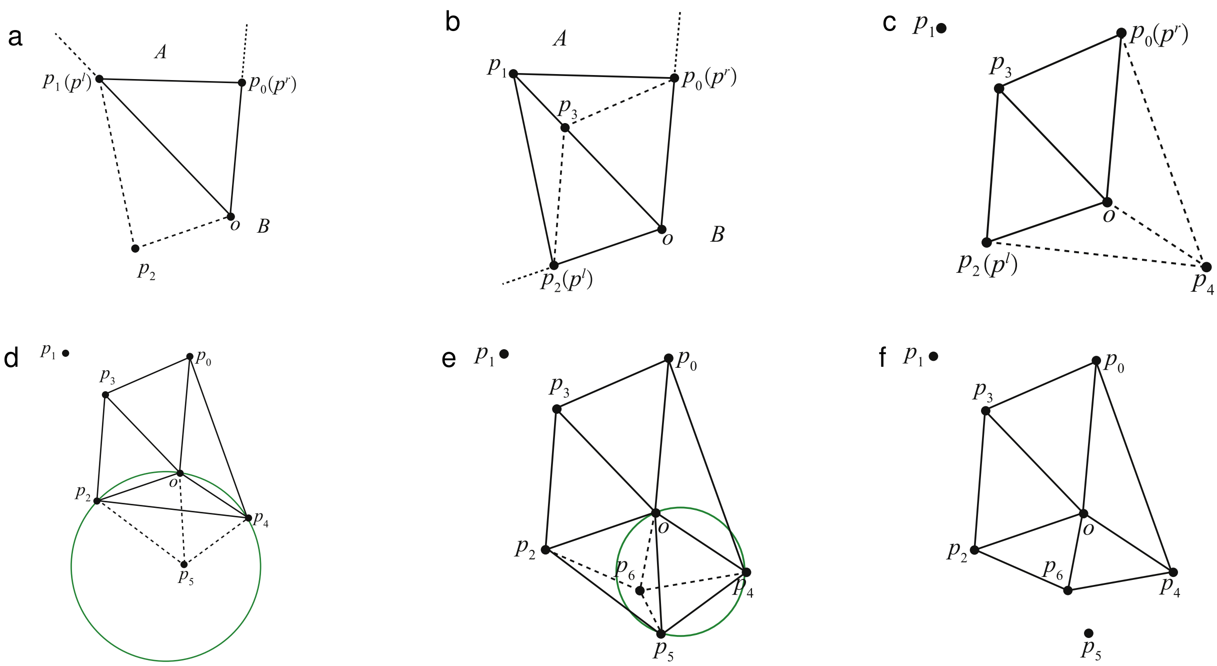

Fig. 3. Procedure for searching the *Minimum Voronoi Neighbors*.

Algorithm edgeLegalization(op_i, MVN, $LDTSet$)

Input the edge op_i that may be illegal; current MVN and $LDTSet$;

Output updated MVN and $LDTSet$;

---

1. Let $\Delta(p_k, o, p_i)$ and $\Delta(p_i, o, p_j)$ be the two triangles that share the edge op_i.
2. **If** op_i is illegal.
   (a) Remove p_i from the MVN. Replace $\Delta(p_k, o, p_i)$ and $\Delta(p_i, o, p_j)$ with $\Delta(p_k, o, p_j)$ in the $LDTSet$.
   (b) Call edgeLegalization(op_j, MVN, $LDTSet$) and edgeLegalization(op_k, MVN, $LDTSet$).
3. **return**.

The key of our *MVN* search strategy for seed *o* is to adjust the local triangulation based on the theory of the Delaunay triangulation, yet it has to consider various cases when a new seed is added. Moreover, the computational complexity to calculate the *MVN* for every seed can be $O(n^2)$ if the algorithm MVNTestForp is conducted for all the other points. To accelerate the search process, we propose using background grids in this paper to add other seeds from near to far and the *MVN* search can be terminated earlier when it is certain that no more seeds appear as element of the *MVN* set.

As shown in Fig. 4, the background grid used is a triangular mesh and every seed must be located in one of the background cells. In our method, a linear queue *L* is set to record the background cells that have been searched and the one that contains *o* is first added into the queue. For a cell in *L*, if a seed falls into its edge-adjacent cells, the MVNTestForp algorithm is conducted to test if it is a member of the *Minimum Voronoi Neighbors* and adjust the current *MVN* and *LDTSet* if needed. If those edge-adjacent cells outside the queue contain seeds in the *MVN* set or intersect with the circumcircles of the triangles in the *LDTSet*, they will

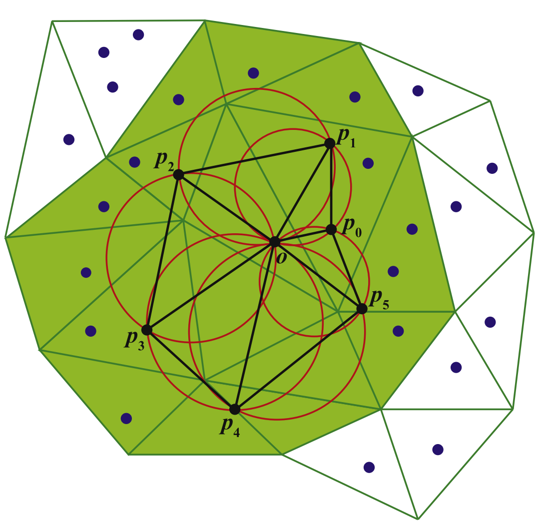

Fig. 4. *MVN* search with the background grids. The triangles in the *LDTSet* are rendered with black edges and the black dots are the seeds in the final *MVN* set. The background cells are rendered with deep green edges, including those in the search queue *L* that are filled in green. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

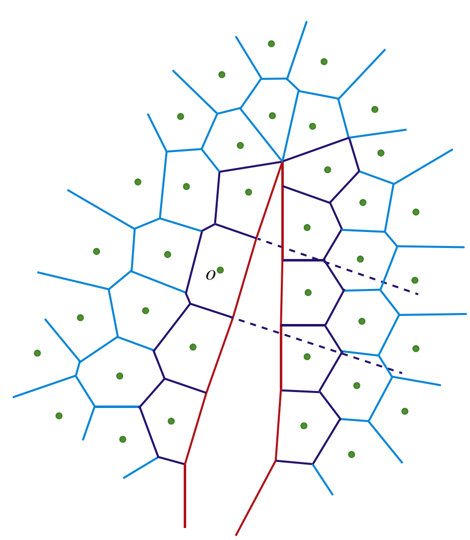

Fig. 5. Edge-cutting for the cells around the border features.

also be added into the search queue *L*. The search will terminate if the queue *L* stops increasing. In other words, there are no cells outside *L* that intersect with the circumcircles of the triangles in the *LDTSet* after the finite search around *o*. The detailed procedure for the *MVN* search based on the background grids is depicted as the algorithm BMVNSearch.

Algorithm BMVNSearch(o, S, tm)

Input o: the seed for MVN search;
S: the set of the seeds;
tm: the background grids;

Output MVN: the set of *Minimum Voronoi Neighbors*
$LDTSet$: the local Delaunay triangle set

1. $MVN \leftarrow \{\}, LDTSet \leftarrow \{\}, L \leftarrow \{\}$.
2. For the cell $t \in tm$ that contains o, L.add(t). For any other seed p in t, call MVNTestForp(o, p, MVN, $LDTSet$).
3. $i = 0$.
4. **While** $i < L.size$
   (a) Let t_l be any edge-adjacent cell of $L(i)$
   (b) $\forall p$ in t_l, call MVNTestForp(o, p, MVN, $LDTSet$).
   (c) **If** t_l contains seeds in the MVN or intersects with the circumcircles of the triangles in the $LDTSet$, then L.add(t_l).
   (d) $i++$.
   End while.
5. **return** MVN and $LDTSet$.

The size of the *MVN* set is often limited and the expectation is 6 as it is the degree of the point after the Delaunay triangulation for the region (de Berg et al., 2008). With the use of the background grids as the search index, our *MVN* search process, more often than not, involves only a small part of the other seeds nearby. Compared with the other generation methods of Voronoi grids, which the computational complexities are often $O(n\log n)$, our Voronoi tessellation strategy using the algorithm BMVNSearch exhibits better adaptability and we design to address the complex constraints in reservoirs in the later section. We estimate that the computational complexity of our method is mostly approximate to $O(n\log n)$ depending on the seed distribution.

3. PEBI tessellation on geological surface

In the field of geology, a fine grid is often employed to capture more details of the model (Khvoenkova and Delorme, 2011). This fine grid could not be directly used as it will be difficult to conduct the simulation within a reasonable period of time (Mlacnik et al., 2004, 2006). This provides a scenario for the application of our method since the original fine grid can be chosen as the background mesh to generate coarse PEBI grids. However, the geological constraints, which result from tectonic movements and reservoir engineering, merit serious considerations.

The border features in reservoir, are brought about by internal faults or reservoir bound. As shown in Fig. 5, the PEBI cells around the border features usually stretch to infinity in their directions while the remaining finite parts often maintain an acceptable shape after the cut with the border edges (red edges in the figure). On this account, we adopt direct edge-cutting means to handle the border constraints in the reservoir field, as it is a simple yet efficient approach in dealing with complex geometric structures.

The main problem with the edge cut is the serious concave cells. In fact, these concave cells can be prevented by making the sharp points of the borders the constrained vertexes of the generated PEBI grids. To achieve this effect, we intend to place the stationary seeds on the protected circles whose centers are the sharp points (Branets et al., 2009). The defined protected circles contain no seeds inside. As shown in Fig. 6(b), several PEBI cells share the same sharp point *m* and the concave cells like ***a*** in Fig. 6(a) are avoided.

In addition, the border features also contribute to a revision to our original *MVN* search algorithm. Once the background cells with the

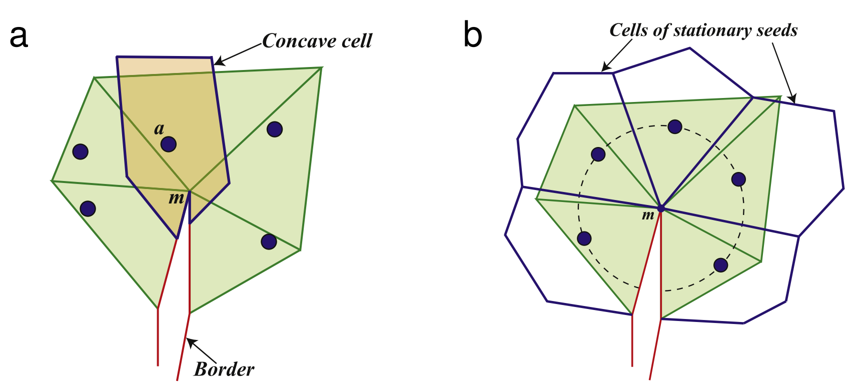

Fig. 6. Treatment for the concave cells.

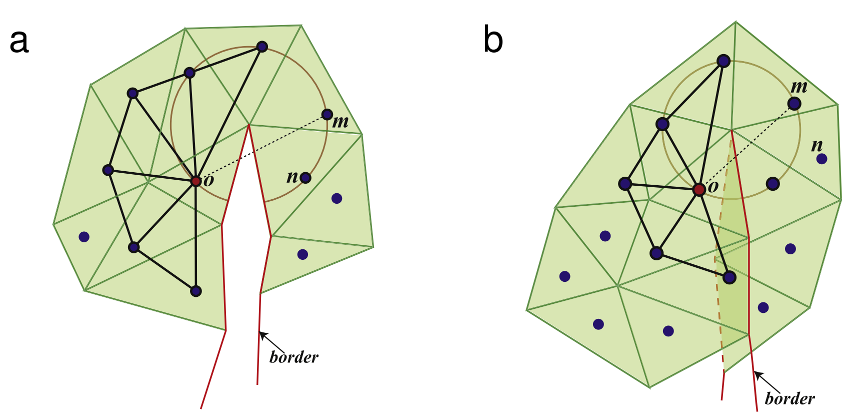

Fig. 7. Revision of the *MVN* search in view of the border-edge obstruction.

border edges are pushed into the search queue L, a barrier test should be performed for the MVN candidate seeds. If the seeds are separated from o by these border edges, they will be excluded from the MVN set. Fig. 7 shows the normal and reverse fault cases for the new revised MVN strategy. The border edges are displayed in red and seed m could not be in the MVN of o due to the obstruction. Then the background cell that contains m will not be pushed into the search queue L and the MVN search stops in this direction. Therefore, n is also excluded from the MVN of o.

This revision solves the problem of the PEBI tessellation for the reverse fault region. If the border is caused by a reverse fault, the background mesh will also overlap in the vicinity. The seeds in the upper part, in contrast to those in the lower parts, will be associated with different background cells. Therefore, it is unlikely for the seeds in different parts to be mutual MVN members due to the border partition.

Well constraints are another issue of grid generation for the reservoir simulation (Noetinger, 2016). To reflect these physical features, Palagi and Aziz (1994) proposed combining several pre-defined modules in

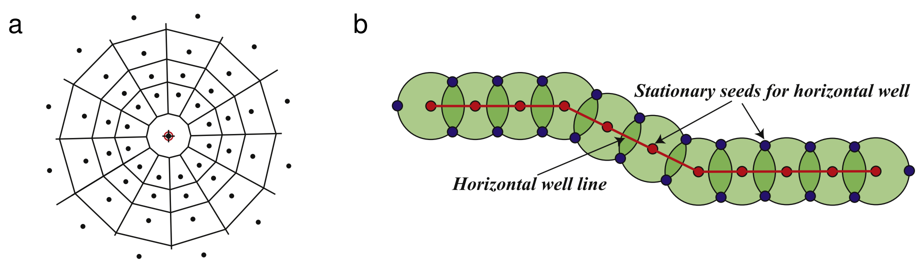

Fig. 8. Stationary seeds for well constraints. (a) The seeds and the radial grids around the vertical wells. (b) The stationary seeds for the horizontal well.

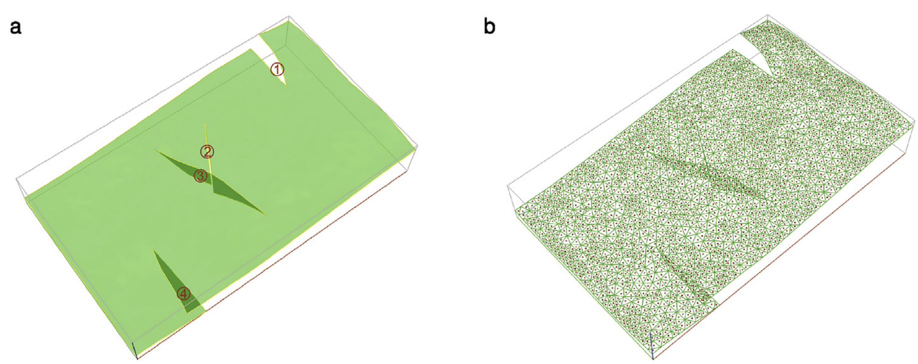

Fig. 9. Synthetic layer for the PEBI tessellation. (a) The panoramic view of the geological domain. (b) The background triangular mesh and the chosen seeds.

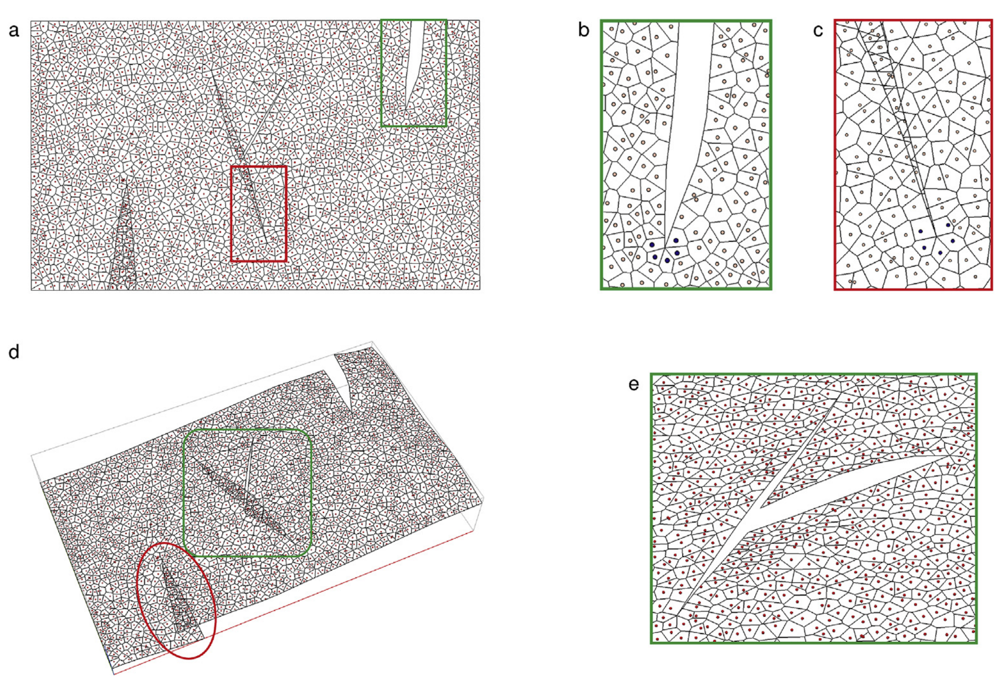

Fig. 10. Generated PEBI grids using the BMVNSearch algorithm without the CVT optimization. (a) The 2D PEBI grids. (b) The cells around normal fault 1, enclosed by the green rectangle in (a). (c) A few of the cells around reverse fault 3, enclosed by the red rectangle in (a). (d) The PEBI grids mapped on the layer. (e) The cells around fault 2 and 3, displayed in the 3D scene and marked with the green rounded rectangle in (d). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

reservoir domain. In their work, radial grids were set around the vertical wells and hexagonal or Cartesian grids were located and rotated to conform to the longitudinal direction of the horizontal wells. In our method, the radial grids can be achieved by simply putting stationary seeds on a series of concentric circles with the well position as the center (Fig. 8(a)). For the horizontal well, we set a series of protection circles and choose some points on the well trace and the circles as the stationary seeds (Fig. 8(b)). This enables the generation of local hexagonal grids, which adapt to the flow information around the horizontal well.

4. Centroidal Voronoi optimization

The cut with the border edges discussed in the last section may lead to inconsistency between the PEBI elements and the definition of the Voronoi diagram. This inconsistency exerts little impact as the border areas are set with the boundary conditions and do not strictly require the local orthogonality. However, the initial seeds with non-optimized coordinates often lead to cells with too small a size or bad shapes, which adversely affect the accuracy and convergence of the reservoir simulation.

Following the line of thought that to meliorate the PEBI grids consists in optimizing the positions of the seeds, the methodology of Centroidal Voronoi Tessellations (Du et al., 1999, 2010), is adopted in this paper. As some stationary seeds are placed to avoid the concave cells or resolve the vertical and horizontal well constraints, we continuously move the other movable seeds to the centroids of the related cells and re-conduct the PEBI tessellation with our PEBI grid generation strategy. The iteration of the seed movement and grid reconstruction are continually executed until the distances between the centroids and the seeds comply with the approximated requirements.

Based on the aforesaid details, our main procedure to generate the PEBI grids for the reservoir is specified as follows:

1. Select the initial set of seeds *S* on the background mesh while setting the stationary seeds for the sharp points of the faults as well as the vertical and horizontal wells.
2. Search the *MVN* and *LDTSet* for every single seed *o* via the algorithm BMVNSearch and construct the corresponding PEBI cells.
3. Calculate the centroids of the constructed PEBI cells. If the centroids and the seeds meet the convergence criterion, terminate and record the seeds and the PEBI cells related to them; otherwise, let the centroids and the stationary seeds form the new set *S* and go to step 2.

5. Applications in geology

We have implemented our method of the PEBI grid generation in the geological domain to test its adaptability to complex structural features. At first, the PEBI tessellation on a synthetic layer model is executed to

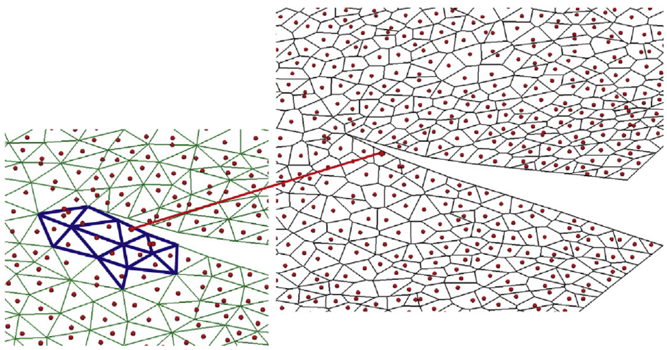

Fig. 11. Background triangles in the search queue *L* for one seed (marked in blue in the left picture) and the PEBI cells around the reverse fault 4 (right picture, enclosed by the red ellipse in Fig. 10(d)). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

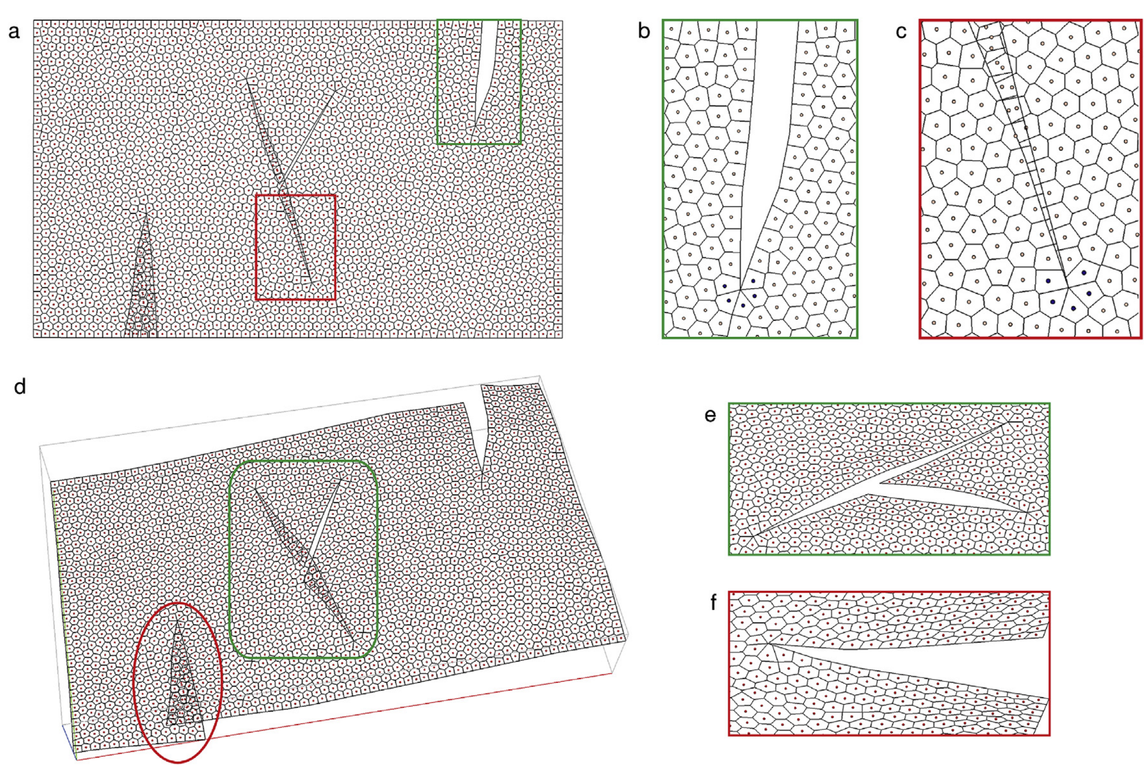

Fig. 12. Generated PEBI grids after the CVT optimization. (a) The 2D CVT optimized PEBI grids. (b) The cells around normal fault 1, enclosed by the green rectangle in (a). (c) A few of PEBI cells around fault 3, enclosed by the red rectangle in (a). (d) The CVT optimized PEBI grids mapped on the layer surface. (e) The cells around fault 2 and 3 displayed in 3D and marked with the rounded rectangle in (d). (f) The cells around reverse fault 4, marked with the red closed ellipse in (d). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

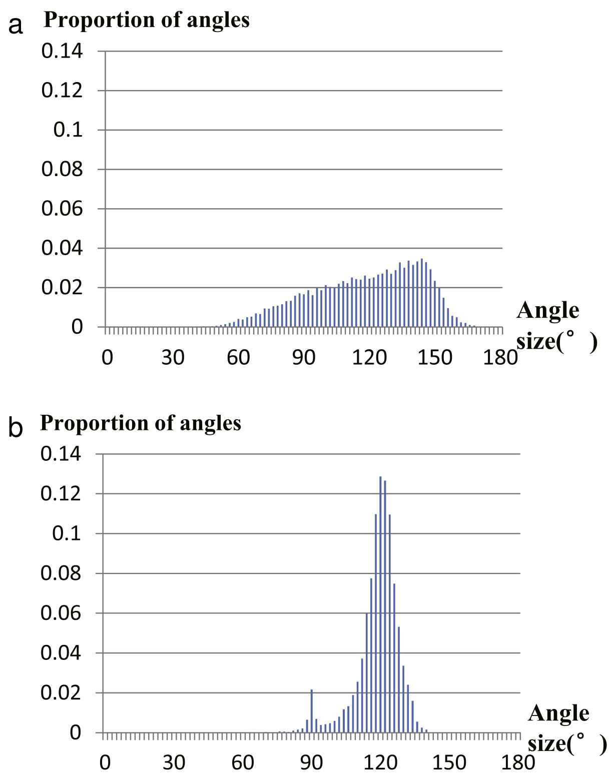

Fig. 13. Statistic angles of the generated cells. (a) Un-optimized. (b) Optimized for 180 times.

demonstrate how our algorithm handles the normal and reverse fault constraints. As shown in Fig. 9, the layer contains two normal faults (marked with 1 and 2) and two reverse faults (marked with 3 and 4). The background grid to describe this layer is a triangular mesh of 2565 triangles with 2569 seeds inside (Fig. 9(b)).

Fig. 10(a) presents the 2D generated PEBI grids using our algorithm while Fig. 10(b) and (c) present some of the 2D cells around fault constraints. These PEBI cells have not undergone the CVT optimization. Besides, as shown in Fig. 10(d), we also vertically projected the PEBI cells back to the layer surface. Owing to the projection, in Fig. 10(e), the cells around the fault 2 and 3 can be observed from a different point of view in the 3D scene compared with those in Fig. 10(d). It can be seen that the PEBI grids constructed by our method can resolve all these normal and reverse fault constraints. In addition, none of the PEBI cells around the sharp points is spotted as a concave polygon due to the placed stationary seeds.

Fig. 11 illustrates an instance how our method prevents the seeds in the upper and lower parts from interfering when handling the reverse fault constraints. In the figure, the left image presents all the triangles in the background search queue *L* after the *MVN* search for one seed near the fault. These triangles are marked in blue and are observed at a different point of view together with the PEBI cells on the right compared to Fig. 10(d) and (e). The figure shows that our *MVN* search strategy is able to exclude the background cells on the other side of the borders.

Fig. 12 presents the generated PEBI grids we optimized for 180 times with the CVT principle. Similar to Fig. 10, Fig. 12(a) and (d) show the overall view of the grids and Fig. 12(b), (c), (e) and (f) display the local cells around the fault features. Moreover, the cells in Fig. 12(d), (e) and (f) are observed at different points of view in 3D scene. Like the un-optimized cells, the CVT-optimized grids can also resolve the constraints. Furthermore, the quality of the cells, including those around the borders, is significantly improved.

Fig. 13 shows the proportions of the cell angles in the neighborhood of some certain angle sizes we recorded before and after the optimization. It was observed that more angles tend to be around the peak 120° after the CVT optimization, indicating that more cells are close to a regular hexagon. As the border edges of the cells around the faults and the boundary are nearly perpendicular to the other edges, a few of the cell angles are approximately 90° as shown in Figs. 12 and 13(b). All these experimental results farther proved our method ensures the generation of correct and qualified grids for the reservoir simulations.

Another geological domain for our PEBI tessellation is a real 3D reservoir. As is shown in Fig. 14, the reservoir is described by 1617 triangles with 1677 seeds inside. The real reservoir contains not only normal and reverse faults but also vertical and horizontal well constraints. Besides, the borders of the faults are more arbitrary.

Fig. 15(a) shows our 2D CVT optimized PEBI grids for this reservoir. For the grids in the figure, the CVT optimization was conducted 100 times. Fig. 15(b) and (c) demonstrate some of the PEBI cells around the fault and well constraints. Evidently, the PEBI grids constructed by our algorithm successfully adapt to all features. Not only were the fault constraints on the reservoir appropriately handled, the radial and local hexagonal grids were also, as expected, generated around the vertical and horizontal wells. Fig. 15(d) shows the 2.5D PEBI grids constructed by extruding the 2D CVT optimized PEBI grids generated for the reservoir. This clearly demonstrates the fault and well features in different layers. More information about the construction of the 2.5D grids, which is beyond the scope of this paper, is available in the work of Gunasekera et al. (1997).

6. Conclusions

In this paper, we introduce a new approach for the PEBI grid generation as a solution to the reverse fault. The essential of our methodology

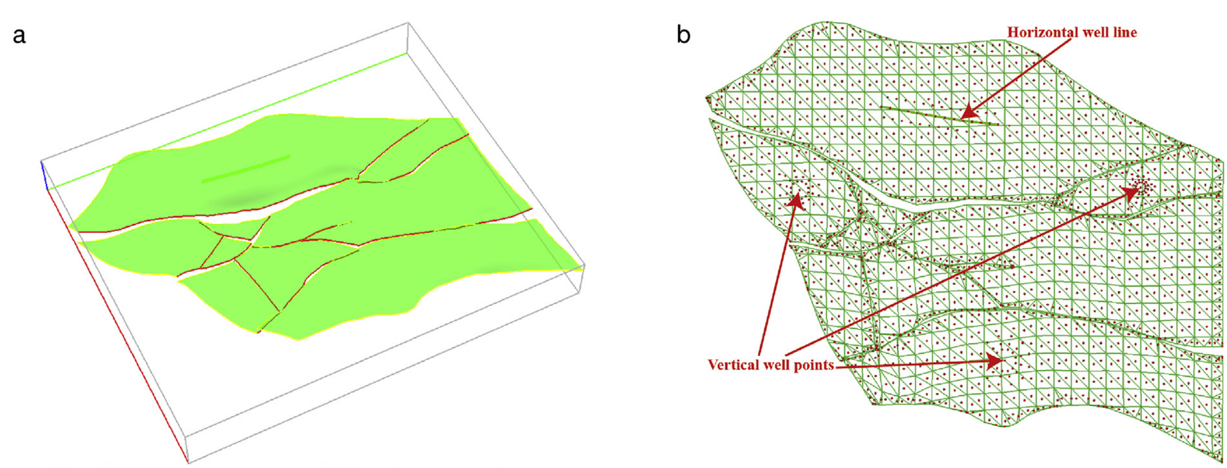

Fig. 14. Real reservoir and the distributed seeds. (a) The panoramic view of the reservoir. (b) The background triangular mesh and the picked seeds.

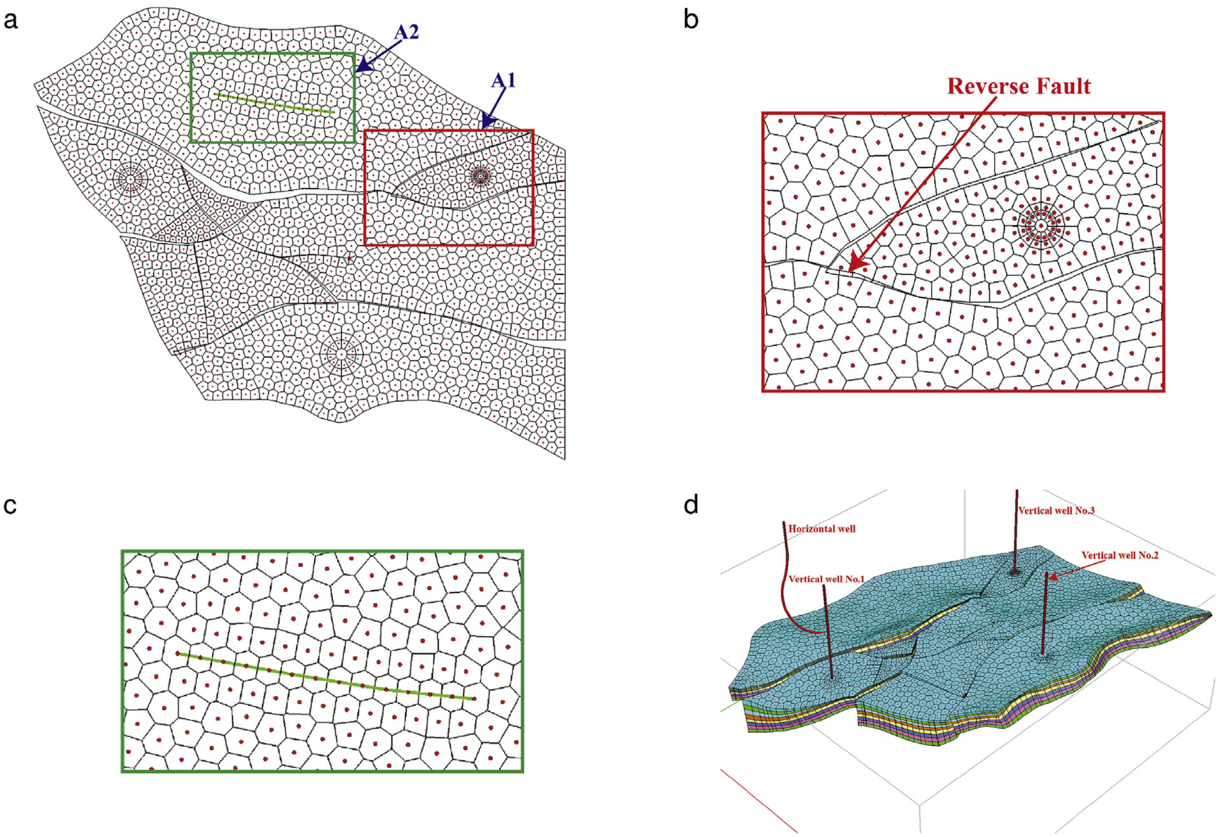

Fig. 15. CVT optimized PEBI grids for the reservoir and the generated 2.5D PEBI grids. (a) The 2D optimized PEBI grids. (b) A few of the cells around the faults and the vertical well, marked with A1 in (a). (c) A few of the cells around the horizontal well, marked with A2 in (a). (d) The generated 2.5D PEBI grids.

is to search the Voronoi neighbors of every seed efficiently and build the associated cells. During the searches, the background grids are used to involve only a small number of other seeds around and the barrier of the internal boundaries rule out the possibility for the points in the upper and lower parts of the reverse fault areas of becoming mutual neighbors. This helps construct reasonable cells around the reverse fault features. Our grid generation process is automatic after the seed set is identified, and its application in geology shows that the generated PEBI grids can effectively resolve all of the features, including the reverse and normal faults and well constraints. Meanwhile, the quality of the cells is considerably improved with the use of CVT principle. The local PEBI grid generation approach is found elastic and adaptable to the complex structural features in reservoir. Yet, the application of our method to more complex topologies and geometries remains an issue for future investigations.

Acknowledgments

National Natural Science Foundation of China (Grant 61003110) and Fund of the State Key Laboratory of Software Development Environment (Grant SKLSDE-2010ZX-10) jointly supported this work.

References

Bertin, E., Marchand-Maillet, S., Chassery, J.-M., 1994. Optimization in Voronoi diagrams. In: Serra, J., Soille, P. (Eds.), Mathematical Morphology and its Applications to Image Processing. Springer Netherlands, Dordrecht, pp. 209–216.

Branets, L.V., Ghai, S.S., Lyons, S.L., Wu, X.-H., 2009. Challenges and technologies in reservoir modeling. Commun. Comput. Phys. 6 (1), 1–23.

Cheng, S.-W., Dey, T.K., Shewchuk, J., 2012. Delaunay Mesh Generation, first ed. Chapman & Hall/CRC.

de Berg, M., Cheong, O., van Kreveld, M., Overmars, M., 2008. Computational Geometry: Algorithms and Applications, third ed. Springer-Verlag, Berlin, Heidelberg.

Du, Q., Faber, V., Gunzburger, M., 1999. Centroidal Voronoi tessellations: applications and algorithms. SIAM Rev. 41 (4), 637–676.

Du, Q., Gunzburger, M., Ju, L., 2010. Advances in studies and applications of centroidal Voronoi tessellations. Numer. Math. Theory, Methods Appl. 3 (2), 119–142.

Fortune, S., 1987. A sweepline algorithm for Voronoi diagrams. Algorithmica 2 (1–4), 153.

Guibas, L.J., Knuth, D.E., Sharir, M., 1992. Randomized incremental construction of Delaunay and Voronoi diagrams. Algorithmica 7 (1), 381–413.

Guibas, L.J., Mitchell, J.S.B., 1992. Voronoi diagrams of moving points in the plane. In: Proceedings of the 17th International Workshop. WG '91. Springer-Verlag, London, UK, pp. 113–125.

Gunasekera, D., Cox, J., Lindsey, P., Jan. 1997. The generation and application of K-Orthogonal grid systems. In: SPE Reservoir Simulation Symposium. Society of Petroleum Engineers, Dallas, Texas.

Heinemann, Z., Brand, C., Munka, M., Chen, Y., May 1991. Modeling reservoir geometry with irregular grids. SPE Reserv. Eng. 6 (2), 225–232.

Khvoenkova, N., Delorme, M., 2011. An optimal method to model transient flows in 3d discrete fracture network. In: IAMG Conference, pp. 1238–1249.

Merland, R., Caumon, G., Lévy, B., Collon-Drouaillet, P., 2014. Voronoi grids conforming to 3d structural features. Comput. Geosci. 18 (3), 373–383.

Merland, R., Lévy, B., Caumon, G., et al., Jan. 2011. Building centroidal Voronoi tessellations for flow simulation in reservoirs using flow information. In: SPE Reservoir Simulation Symposium. Society of Petroleum Engineers, The Woodlands, Texas, USA.

Mlacnik, M., Durlofsky, L., Heinemann, Z., Jan. 2004. Dynamic flow-based PEBI grids for reservoir simulation. In: SPE Annual Technical Conference and Exhibition. Society of Petroleum Engineers, Houston, Texas.

Mlacnik, M., Durlofsky, L.J., Heinemann, Z.E., Sep. 2006. Sequentially adapted flow-based PEBI grids for reservoir simulation. SPE J. 11 (03), 317–327.

Noetinger, B., 2016. About the determination of quasi steady state storativity and connectivity matrix of wells in 3d heterogeneous formations. Math. Geosci. 48 (6), 641–662.

Palagi, C.L., Aziz, K., 1994. Use of Voronoi grid in reservoir simulation. SPE Adv. Technol. Ser. 2 (02), 69–77.

Shamos, M.I., Hoey, D., 1975. Closest-point problems. In: Foundations of Computer Science, 1975., 16th Annual Symposium on. IEEE, pp. 151–162.

Verma, S., Aziz, K., others, 1997. A control volume scheme for flexible grids in reservoir simulation. In: SPE Reservoir Simulation Symposium. Society of Petroleum Engineers, pp. 215–227.

Verma, S.K., 1996. Flexible Grids for Reservoir Simulation. Ph.D. thesis. Stanford University, California.