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

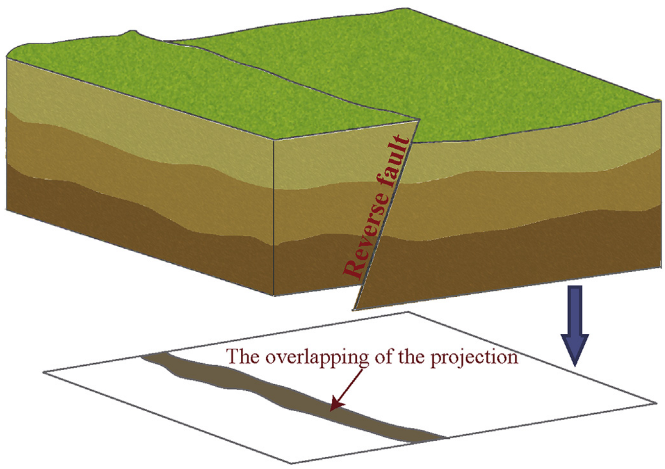

Fig. 1. Reverse fault and the 2D projection.

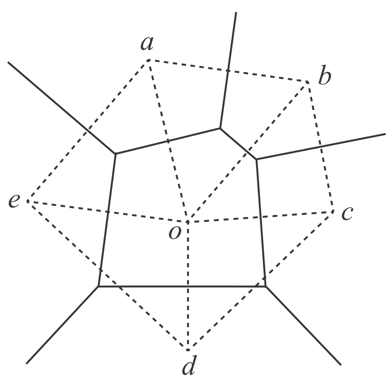

Fig. 2. Voronoi cell of *o* and the related *MVN*.

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**.