Abstract

An original technique for building optimal streamline-pressure-potential (SPP) unstructured grids is presented. This method entails the generation of Voronoi Diagrams of n-dimensional objects for partitioning a given volume of interest (basically a reservoir in the geological sense) into a set of 3D streamtubes. Unlike other methods, this process is fully 3D and does not involve any extrusion. Moreover, no restrictions are imposed on the complexity of the flow pattern and all the generated grid cells are such that their faces are either locally perpendicular or collinear to the flow direction, which has been proven to be profitable when using Finite Volume discretization schemes [1]. The originality of this method lies in the fact that the polyhedral grid is represented by a 3D raster image. This raster image can either be used directly for feeding the flow simulator or converted first to a combinatorial representation through a vectorization process.

Introduction

While handling complex reservoirs, the choice of the computational grid and of the associated discretization scheme may significantly affect the reservoir simulation. Accuracy of the solution is achieved by minimizing numerical errors and honouring the reservoir features that are likely to impact the fluid flow (layer boundaries and faults). Moreover, the grid density should be adapted to the flow rate. Computational efficiency is mainly affected by the scheme used for discretizing the flow equations, the number of cells and the number of faces per cell. The construction and updating of a faulted grid honouring complex structural geologic models can be a difficult and time consuming task. The efficiency of the gridding process should therefore be taken into account when deciding which kind of grid will be used as a support for flow simulations.

Traditional corner point grids, composed of six-sided blocks aligned along three curvilinear axes, are not flexible enough for modeling complex structures, such as bifurcating faults, highly correlated channels and deviated wells. Furthermore, these grids are not strictly orthogonal and cannot be aligned everywhere with the main flow directions or the permeability tensor, which often leads to a lack of precision and efficiency of the simulations. More flexible structured and unstructured grids have therefore been proposed to overcome these problems.

Whereas 3D unstructured PEBI grids, introduced by Heinemann [2] provide better flexibility, they present a large number of faces per grid block and cannot be handled efficiently by conventional flow simulators. 2.5 dimensional PEBI grids and hybrid grids attempt to mitigate drawbacks of purely structured and unstructured grids [3]. Another kind of simulation grids, based on a curvilinear coordinate system, defined by streamlines and equipotential lines, have been introduced by Edwards et al. in [1].

This paper presents a new method for creating such Streamline Pressure Potential based 3D grids (SPP grids). Flow equations will first be briefly reminded, and existing SPP grid construction processes will be discussed. Then, an original method based on image processing techniques will be presented.

1 Background

1.1 Flow Equations

Let $V$ an arbitrary control volume with external area $A$ and $\vec{n}$ the outward normal to its boundary. The mass conservation equation states that the flux through the volume boundary is equal to the mass of fluid that is accumulated inside it, considering one component per phase and $n_p$ immiscible phases, for each component, it can then be written:

$$- \int_A \sum_{p=0}^{p=n_p-1} \rho_p \vec{v}_p. \vec{n} dA = \int_V \left[ \frac{\partial}{\partial t} \sum_{p=0}^{p=n_p-1} \rho_p S_p \right] dV$$

(1)
The left term of the equation is the flow term, the right term is the accumulation term: $\rho_p$ is the density of phase $p$, $S_p$ is its saturation and $\phi$ is the porosity of the medium. The velocity term $\vec{v}$ is given by the Darcy’s Law:

$$v_p^i = -\frac{k_{rp}}{\mu_p} \vec{F} \nabla \Phi_p$$

(2)

where $k_{rp}$ is the relative permeability of phase $p$, $\mu_p$ is its viscosity, $\vec{F}$ is the permeability tensor and $\nabla \Phi_p$ is the potential gradient of phase $p$.

The finite volume method is a direct transcription of this equation using the grid cells as control volumes. The surface integral over the volume boundary is thus converted to a sum over the cell faces:

$$\sum_{\text{fac} = 0}^{i} \vec{F} \Phi_i A_i = 0$$

(3)

Let us consider a cell face separating two cells $i$ and $j$, and an orthonormal basis $(x', y', z')$ such that $x'$ is perpendicular to the face. The components of $\vec{v}$ that are coplanar with the face do not contribute to the mass transfer between $i$ and $j$. The component $v_{x'}^i$ that is perpendicular to the face can be expressed as follows:

$$v_{x'}^i = -\lambda \vec{F} \Phi_i$$

(4)

Depending on the form of the permeability tensor $\vec{F}$, and on the geometry of the grid, this term can be computed either using only the value of the pressure for both cells contiguous to the considered face (Two Point Flux Approximation, TPFA) or the value of the pressure for all the cells surrounding the face (Multiple Point Flux Approximation, MPFA). Since TPFA is more stable and computationally efficient than MPFA, it is desirable to build grids such that TPFA can be applied. Considering a face between two cells $i$ and $j$, and $\vec{s}_{ij}$ the vector linking both cell centers, the following condition must be satisfied:

$$\vec{k} \nabla \Phi i \vec{s}_{ij} = k \frac{\Phi_j - \Phi_i}{|\vec{s}_{ij}|^2} \lambda \vec{s}_{ij}$$

(5)

If the permeability is isotropic, any grid such that $\vec{s}_{ij} \parallel \vec{n}_{ij}$ satisfies this condition (operator $\parallel$ denotes colinearity); this is the case in PEBI grids. When the permeability tensor is not isotropic, TPFA can be applied if $\vec{s}_{ij} \parallel \vec{k} \nabla \Phi$. If it is too anisotropic, the only grid geometry that guarantees the applicability of the TPFA is such that the grid faces are either perpendicular or colinear to the flow (SPP grids, see Figure 1).

1.2 Related Works

The methods that are described below assume that a set of streamlines has been traced over the volume of interest prior to the grid construction. These streamlines can be deduced from a simple single phase flow simulation on a fine cartesian grid.
Edwards proposed in [1] a method for building 2D SPP grids. This method consists in selecting arbitrarily a set of streamlines, interpolating the pressure field over these streamlines and distributing grid nodes at fixed pressure values. Grid nodes belonging to neighboring streamlines are then linked together in order to build iso-potential lines. In [5], Castellini et al extended this technique to 3D. In their method, several strata of streamlines are drawn simultaneously and then linked together. In order to build the grid edges corresponding to iso-potential lines, the neighborhood relationship between streamlines have to be known a priori: the grid is actually extruded along the streamlines. As a result, this technique is limited to simple flow patterns and boundary conditions, and cannot be applied alone to grid a volume containing several wells.

Mlacnik introduced in [4] a new technique for building 2.5D unstructured SPP grids. This method consists in computing a 2D Voronoi diagram of points that have previously been distributed on streamlines, then extruding the resulting grid along coordinate lines. Mlacnik proposes to use a Delaunay triangulation of the grid points followed by a dual operation to create the Voronoi diagram (see Figure 1). Using this technique, it is possible to build 2D grids that can take into account complex flow patterns. These grids are strictly PEBI but not strictly SPP. Actually, since the nodes of the Delaunay triangulation are not 4 to 4 cocircular, there are some “oblique faces” in the dual mesh (faces that are neither perpendicular nor parallel to the flow). Further more, this technique cannot be applied in 3D since it would lead to the creation of more oblique faces and of flat tetraedra.

2 Proposed Approach

2.1 Overview

Our method consists in building first two independant partitions of space:

- A set of streamtubes. The geometry of the streamtubes are approximated by the Voronoi regions associated with the input streamlines.
- A set of pressure regions; each region corresponding to a known pressure range.

Then, the grid is obtained by co-refining the streamtubes with the iso-pressure bounded regions. The originality of this method lies in the fact that all these operations are performed on a 3D structured regular grid (a 3D digital image composed of voxels). The resulting SPP-grid is therefore such that its cells are actually represented by a set of contiguous voxels. The raster grid can then be converted to a combinatorial representation (vectorization step).

2.2 Creation of a raster grid

2.2.1 Definitions

Discrete Voronoi Diagrams Methods based on computational geometry (e.g. divide-and-conquer, Fortune’s sweepline) aim at creating Voronoi diagrams in the form of polyhedral grids. These grids can be represented by topological data structures such as the Maps or the Generalized Maps [6].

Image-analysis derived methods do not provide such a combinatorial representation of the Voronoi diagrams. Instead, the obtained Voronoi diagram is represented by a n-dimensional image in which every pixel is assigned a label corresponding to the nearest Voronoi site. This procedure is called the Voronoi labelling, the resulting image is a Discrete Voronoi Diagram (see Figure 5-A). When the Voronoi sites are objects of dimension $d > 0$, the Voronoi regions cannot be represented by n-polygons and the diagram is said to be generalized.

Euclidean Distance Map A distance map is an image where the value assigned to each pixel is the distance from this pixel to the nearest pixel belonging to a given set or object. A distance transform (DT) is an algorithm that computes a distance map from a binary image representing this set of pixels.

2.2.2 Partitioning space in a set of streamtubes

Let us consider a set of streamlines represented by polygonal lines (see Figure 2(a). Our approach consists in computing a 3-dimensional discrete Voronoi diagrams of streamlines using a modified version of Saito-Toriwaki’s Euclidean Distance Transform (EDT) [7, 8]. Saito-Toriwaki’s EDT algorithm has been modified in such a way that each pixel is assigned not

---

1A Voronoi diagram of a set of points $P$ in $n$ dimensions is a partition of space in a set of $n$-polygons such that each polygon is constituted of all the points of space that are closer to a point $P_i$ than to the other points of $P$. 

9th European Conference on the Mathematics of Oil Recovery - Cannes, France, 30 August - 2 September 2004
only the value of the distance to the closest streamline, but also a label corresponding to the closest streamline (the streamtube index, denoted as $S_i$). It has been proven in [9] that the algorithm of Saito-Toriwaki is the fastest Euclidean Distance Transform (EDT) in three dimensions, with image size less than $256 \times 256 \times 256$.

This process yields a 3D digital image in which all voxels belonging to the same streamtube share the same “colour” (see Figure 3(a)). Notice that since the streamlines are objects of dimension $d > 0$, the boundaries of the created regions are not planar.

2.2.3 Iso-Pressure regions

This step aims at computing another 3D digital image in which the cells are colour-coded according to the pressure interval they belong to (Figure 3(b)). It is assumed that the value of the pressure is known everywhere on the background grid (see Figure 2(b)). In order to create $n_p$ different pressure regions, a pressure index $P_i$ is affected to every voxel of the grid. For a constant pressure interval, the following formula can be used, where $P$ represents the pressure value in the voxel and $(P_{\text{min}}, P_{\text{max}})$ represent the pressure bounds for the whole reservoir (int denotes the integral part):

$$P_i = \text{int}\left(\frac{P_{n_p}}{P_{\text{max}} - P_{\text{min}}}\right)$$

However, it is of primary importance to carefully select the pressure bounds. Constant pressure interval leads to very small cells close to the wells and very large cells far from it. On the opposite, the pressure intervals can be defined in such a way that all the pressure regions have a similar volume. This is performed by computing a cumulative distribution function of the pressure and splitting it in regular intervals. This method leads to thick regions around the wells and thin regions elsewhere. It is then possible to balance both solutions in order to obtain regularly spaced pressure classes.

2.2.4 Co-refinement

Images of streamtubes and of iso-pressure classes are then combined together in order to create an image of the SPP-grid (Figure 4(a)). Since the number of iso-pressure regions $n_p$ is known, a unique grid index $G_i$ can be affected to every voxel:

$$G_i = P_i + n_p S_i$$

2.2.5 Limits

The use of a digital image makes grid construction much easier than with data structures commonly used for representing polyhedral grids. However, the use of a discretization of space (the 3D digital image) implies several kinds of
Figure 3: Two partitions of space represented on the raster grid

errors:
- The geometrical precision of the SPP-grid is bound to the resolution of the background image;
- If the resolution of the digital image is not sufficient, several Voronoi sites or pressure regions may be rasterized as a single pixel, thus affecting the topological consistency of the diagram;
- The topology of the background grid impacts the topology of SPP-grid. For example, in 2D, there can be at most four different regions around a grid vertex. As a result, some grid blocks, which should meet, may become disconnected.

2.3 Vectorization

This section describes the actual polyhedral cell generation. Given a discrete Voronoi diagram as a digital image, the Voronoi regions are converted to a set of polyhedra, yielding a combinatorial representation of the grid (see Figure 4(b)). In 3D, building a consistent representation of the resulting polyhedral grid needs the loci of the grid nodes, adjacency relationships between these nodes (edges) and the cell faces limited by these edges. Topological information can be handled by data structures such as the Maps or the GMaps [6].

2.3.1 Principle

Let us introduce a few definitions of topological embedding:
- A \textit{v-element} is a topological element of the digital image (edge, node, or voxel). A \textit{v-cell} is a \(N\)-dimensional \textit{v-element} in a \(N\)-dimensional image, namely a voxel, where the Voronoi labels are stored.
- A \textit{s-element} is a topological element of the SPP-grid.

In dimension \(N\), a \(p\)-dimensional \textit{v-element} is said to belong to a \(p\)-dimensional \textit{s-element}, if they both delimit the same set of \(N\)-dimensional Voronoi regions. \textit{v-elements} belonging to \textit{s-elements} will be noted \(v^*\)-\textit{element}. The proposed method consists in finding first a set of \(v^*\)-\textit{elements} and then building the \textit{s-elements} from the corresponding \(v^*\)-\textit{elements}.

After Voronoi labelling, the Voronoi region indices are known for each \textit{v-cell} of the raster. A simple method for determining whether or not a \textit{v-element} belongs to a \textit{s-element} therefore consists in counting the number of \textit{v-cells} contiguous to this element that belong to distinct Voronoi regions. In 2D, a \textit{v-node} corresponds to a \textit{s-node} if it is surrounded by voxels belonging to at least three different Voronoi regions; a \(v^*\)-\textit{edge} is always located between two different Voronoi regions.
In 3D, a $v^*$-edge is surrounded by $v$-cells belonging to at least three different Voronoi regions. However, it is not always possible to determine whether a $v$-element belongs to a $s$-element by simply counting the number of $v$-cells belonging to different Voronoi regions around it. In 3D, for example, it is not possible to determine whether a $v$-node corresponds to a $s$-node by counting the number of distinct Voronoi cells around it. This “direct” approach is actually limited to the $v^*$-elements of dimension $d \geq N - 2$. $v^*$-elements of dimension $d < N - 2$ can be deduced from the $v^*$-elements of higher dimension: a $v^*$-element of dimension $d$ is always contiguous to at least $N + 1 - d$ $v$-elements of dimension $d+1$.

These rules do not apply to the $v$-elements located on the raster image boundary. However, boundaries of a $N$-dimensional raster image actually are $N - 1$-dimensional raster images and can therefore be treated separately.

### 2.3.2 Implementation

An efficient algorithm for creating 3D polyhedral grids from discrete Voronoi diagrams has been designed following these principles.

A hierarchical approach has been chosen (see Figure 5-B). First, $v^*$-nodes are located at every corner of the 3D raster. Then, edges of the 3D raster, that can be considered as 1D rasters, are processed: $v^*$-nodes are placed at each Voronoi
region boundary. These $v^*$-nodes are used as seeds for finding the $v^*$-edges that point towards both raster faces shared by the edge. Then, a recursive algorithm is used for tracking the $v^*$-edges and $v^*$-nodes which are on the raster faces. Finally, a similar recursive algorithm is used for tracking $v^*$-edges and $v^*$-nodes inside the 3D grid, using the boundary $v^*$-nodes as seeds (see Algorithm 1).

```
1: for all the $v$-edges around the initial node do
2: cur = current $v$-edges
3: if marked then
4: continue
5: end if
6: mark
7: if cur does not belong to a $s$-edge then
8: continue
9: else
10: if cur belongs to same $s$-edge than previous $v^*$-edge then
11: goto 1 with its mate extremity
12: else
13: create a new $s$-node
14: finish the current $s$-edge
15: start a new $s$-edge
16: goto 1 with its mate extremity
17: end if
18: end if
19: end for
```

Algorithm 1: Connectivity graph construction

### 2.4 Creating a graph of pipes

The raster grid can be used directly (without vectorization) for providing the whole set of parameters needed for performing flow simulations.

#### 2.4.1 Principle

When using TPFA, the flux $q_{ij}$ between two neighboring cells $i$ and $j$ only depends on the property of the considered fluid, on geometry of the grid and on the pressure difference between two neighboring cells:

$$q_{ij} = \lambda_{ij} T_{ij} (\Phi_j - \Phi_i)$$

$\lambda_{ij}$ accounts for the physical properties of the considered fluid and the transmissivity term $T_{ij}$ depends on the permeability of the media and on the geometry of the grid:

$$T_{ij} = \frac{k_{ij} A_{ij}}{d_{ij}}$$

$A_{ij}$ is the area of the considered cell face and $d_{ij}$ the distance between both cell centers.

The grid can therefore be represented by a graph of pipes linking neighboring cell centers. The cell centers hold some rock and fluid properties (e.g. porosity, pressure, saturation), while the edges linking them (the pipes) hold a transmissivity property.

#### 2.4.2 Retrieving topological and geometrical information

In order to build the graph of pipes, it is therefore necessary to know the following information:

- the topology of the graph (neighbourhood relationships between cells);
- the area of the cell faces;
- the location of the cell centers and the distance between two neighboring centers.

All these information can be retrieved using a simple sweep algorithm. In this process, the 3D digital image is swept along its three axis and information concerning the $s$-cells and their faces is recorded.
Cell volume, pore volume and graph nodes  The cell volume is computed by counting the number of voxels belonging to each s-cells and multiplying it by the volume of a voxel. The pore volume can be computed the same way (weighting the count by the porosity of each voxel). The cell center locations are found by computing the barycentre of all the voxels sharing the same grid index.

Grid topology  If two neighboring voxels hold different indices, the v-face between them correspond to a s-face. The topology of the grid can therefore be deduced from the set of v*-faces that are significantly different.

Area of the cell faces  Although the cell faces appear to be stair-stepped in the 3D raster grid, our purpose is to compute the area of the corresponding planar facet (the s-face). This process involves two steps:

1. Determining the orientation of the plane containing the considered s-face with respect to the 3D image axis;
2. Computing the area of the projection of the corresponding v*-faces onto this plane.

Both these operation can be performed by simply counting the number of v*-faces in each direction.

Conclusion

A method for building three dimensional SPP grids has been presented. Created grids are represented by 3D digital images such that the pixels belonging to the same grid block share the same colour. A vectorization method for converting these images to actual polyhedral grids has been presented. Direct use of the raster representation has also been discussed.

In the future, we would like to explore the following issues: multiresolution techniques for data resolution adaptation, decomposition of large reservoirs in a set of small 3D images, integration of horizon, faults and fractures into the grids using co-refinement operations.

Acknowledgements

Many thanks to Pascale Neff that tackled the problem of converting Voronoi diagrams from discrete combinatorial representation as a part of her Master thesis, to Francois Lepage and David Ledez for their critical comments and helpful suggestions and to Olivier Grosse for his excellent topological kernel. This research works was performed in the frame of the Gocad research project. The industrial and academic members of the Gocad consortium are hereby acknowledged.

References