RESEARCH ON CONSTRUCTING ISOLINES AND STREAMLINES IN SEEPAGE FLOW FIELD

Haisheng Li[1], Qiang Cai[1], Yuewu Liu[2], Yi Chen[1]

1 Beijing Technology and Business University, Beijing 102488, China
2 Institute of Mechanics, Chinese Academy of Sciences, China
lihsh@th.btbu.edu.cn

Keywords: Visualization, Delaunay Triangulation, Isoline, Streamline, Seepage flow field.

Abstract

Isolines and streamlines are two important tools used to analyze and visualize the flow through porous media, such as oil flow and underwater flow. Algorithms of generating isolines and streamlines based on constrained Delaunay triangulation are presented in this paper. A novel gradients calculation method is also put forward. In addition, "compass" algorithm based on "wing-edge" data structure is adopted to fasten the streamline integration process. Examples show the algorithms generate satisfied isolines and streamlines in domain.

1 Introduction

Isolines and streamlines are two important tools used to analyze and visualize the flow of oil and underwater. Many reservoir engineers have paid close attention to the streamline study [1]. Streamlines can clearly describe the flow movement in the underground, and it is of great significance in oil reservoir exploration etc.

Streamline construction is one of the most fundamental techniques for visualizing flow fields. A streamline is a curve everywhere tangent to the field. In practice, a streamline is often represented as a polyline (series of points) iteratively elongated by bidirectional numerical integration started from a seed point, until it hits the domain boundary, reaches a critical point or generates a closed path.

Given a fluid flow with velocity field $\vec{u}(\vec{x}(t))$, a streamline is an integral curve of $\vec{u}$. That is, a streamline can be calculated by solving the following equation: $\frac{d\vec{x}(t)}{dt} = \vec{u}(\vec{x}(t))$ where t is a parameter along the streamline and is not to be confused with time. A path line is the path of a particle in the fluid. A streak line arises when dye is injected in the flow from a fixed position. In steady flows, streak lines and path lines are identical to streamlines [2, 3].

Isoline is the curve whose points have the same attribute value. The gradient of the attribute value indicates the fastest value changed direction. Take oil flow in underground for example, the reservoir pressure make the oil move. The oil flows along the direction of reservoir pressure gradient. That is to say, the isolines of reservoir pressure and streamline should be perpendicular theoretically.

This paper is organized as follows. In Section 2, algorithm of adaptive constrained Delaunay triangulation in domains is discussed. In Section 3, a novel gradients calculation method is put forward. Interpolation in vector field is discussed in Section 4. In Section 5, the numerical integration method and the selecting strategy of integration step size are described. Point location problem is discussed and the efficiency of "compass" algorithm is analyzed in Sector 6. Algorithms of generating isolines and streamlines are presented in Section 7 and two examples are shown at last.

2 Adaptive constrained Delaunay triangulation in domains

Triangles are the simple complex. They are convex and planar, which facilitates containment tests and face intersection tests. In addition, the triangle cells can conform domain boundary precisely.

A triangulation of a set S of points is a partition of the convex hull of S into triangles. A triangulation is called Delaunay if the interior of the circumcircle of any triangle in the triangulation contains no points of S.

Delaunay triangulation is the dual of the Voronoi diagram and has a number of nice properties. A Delaunay triangulation maximizes the smallest interior angle of any triangle. Delaunay triangulations tend to "rounder" triangles than other triangulations. This property is desirable for numerical applications.

The constrained Delaunay Refinement algorithm [4] is used in this paper to obtain the triangle mesh. The radius of circumcircle and minimal edge Ratio of triangle (Radius-Edge Ratio) is used to character the quality of triangle mesh. The less of Radius-Edge Ratio is, the more fat the triangle is. For Equilateral triangle, the Radius-Edge Ratio is smallest, it is $1/\sqrt{3}$.

Constrained Delaunay Refinement Algorithm (CDR Algorithm):

Input: Constrained Segments Set. (Include interior boundary, exterior boundary and constrained edge in domain. All of them can be represented as any polyline set).

Output: A triangle set TS that has Delaunay properties.

Procedure:

1) Set the Radius-Edge Ratio threshold value Q_T of expected triangle mesh;
2) Split the constrained edge according to interested regions. If constrained edge is located in key interested region, split the edge as small as expected according to reality;
3) Add all constrained segments into a segment set CSS;
4) Add start point and end point of all segments into a point set PS, remove duplicated point in PS;
5) Initialize triangulation TS with helper bounding triangle;
6) For a point P∈PS, add P into TS, if circumcircle of any triangle T contain P, delete T from TS, form a Delaunay cavity. Connect every vertex of Delaunay cavity and P form new triangle set, update TS according;
7) Repeat steps 6 until all points have been added to TS.
8) For segment CS∈CSS, if CS is not an edge of a triangle, add the midpoint of CS into TS, update CSS according;
9) Repeat step 8 until all segments have been occurred in TS.
10) For any triangle T in TS, if Radius-Ratio Ratio is greater than Q_T, compute the circumcircle center CP of triangle T.
11) For any constrained segment CS∈CSS,
    a) if the circumcircle of CS contains CP, add the midpoint of CS into TS, update CSS according;
    b) else, add CP into TS, the procedure is similar to step 6.
12) Finally, TS is the desirable conforming Delaunay triangulation. Algorithm CDR ends.

For Delaunay triangulation, two triangles in it share one common edge. The "wing edge" data structure [5] (See Figure 1) is used to fasten the triangulation process. It can fasten point location and interpolation process dramatically during streamline tracing process because it contains the topological relationship between adjacent triangles. See detailed in Section 6.

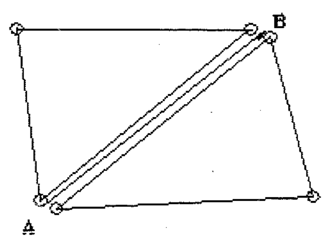

Figure 1 "Wing-Edge" data structure

3 Calculating the gradients in flow field

In Computational Fluid Dynamics (CFD), the reservoir pressure can be calculated by numerical methods. In order to obtain streamlines, the gradients of reservoir pressure must be evaluated firstly.

Take the value of reservoir pressure P as Z, and extend the triangle mesh from 2d to 3d. The three vertexes of every triangle form a plane. Without loss of generality, suppose the plane equation is:

$$AX+BY+CZ+D=0 \tag{1}$$

The gradients of reservoir pressure are as follows:

$$\begin{cases} v_x = \dfrac{\partial z}{\partial x} = -\dfrac{A}{C} \\ v_y = \dfrac{\partial z}{\partial y} = -\dfrac{B}{C} \end{cases} \tag{2}$$

The normal of the triangle plane is (A,B,C). The normal can also be expressed as $\left(\frac{A}{C},\frac{B}{C},1\right)$ using homogeneous coordinates. So we can use the reversed direction of regulated normal of triangle as the gradients of reservoir pressure according to equation (2). The three vertices of triangle can be easily obtained and the normal of triangle can be calculated by vector product easily.

For each triangle in domain, the normal of triangle is easily to be computed. In order to evaluate the gradients of triangle vertex in domain, area-weighted interpolation is used here. Suppose one vertex was shared by n triangles in 3d space, the normal of the vertex can be evaluated as follows:

$$\begin{cases} v_x = \dfrac{\sum_{i=1}^{n} v_{xi} \times A_i}{\sum_{i=1}^{n} A_i} \\ v_y = \dfrac{\sum_{i=1}^{n} v_{yi} \times A_i}{\sum_{i=1}^{n} A_i} \end{cases} \tag{3}$$

For an arbitrary triangle in 3d space, we can get its area by computing the mixed product of two vectors, as shown in equation (4).

$$S_{\Delta ABC} = \frac{1}{2}\left|AB \times AC\right| = \frac{1}{2}\begin{vmatrix} x_A & y_A & z_A \\ x_B & y_B & z_B \\ x_C & y_C & z_C \end{vmatrix} \tag{4}$$

4 Vector field interpolation

The CFD simulation result deals with discrete data which is supplied at a finite number of vertices. We must perform interpolation in order to get the velocity of any point in the filed domain. The common interpolation algorithm in data field visualization has linear interpolation, inverse distance weighted interpolation, thin spline interpolation and area-weighted interpolation et al. Due to the reason that we have got the triangle mesh topology of fluid flow in Section 2, the area-weighted interpolation method (see Figure 2) is to be adopted to compute every components of vector attributes for any points located in the fluid flow domain.

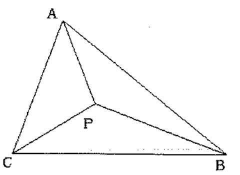

Figure 2 Area-weighted interpolation

Similar to Section 3, to get the vector attributes of a given point P in the triangle cell T, we firstly compute the barycentric coordinates ω_i of P in T. The ω_i is in proportional to the area of the subtriangle defined by point P and the edge opposite to a vertex Pi. Therefore, barycentric coordinates of a triangle are sometimes also called area coordinates. The components of area coordinates can be computed via equation (5). Let E(P) be a component of vector attributes, and E_A, E_B, E_C be the corresponding attribute of three vertices for a triangle, $\omega_A, \omega_B, \omega_C$ be the three components of area coordinates of Point P. E(P) can be computed via equation (6).

$$\begin{cases} \omega_A = \dfrac{S_{\Delta PBC}}{S_{\Delta ABC}} \\ \omega_B = \dfrac{S_{\Delta APC}}{S_{\Delta ABC}} \\ \omega_C = \dfrac{S_{\Delta ABP}}{S_{\Delta ABC}} \end{cases} \qquad (5)$$

$$E(P) = \omega_A E_A + \omega_B E_B + \omega_C E_C \qquad (6)$$

The area of triangle can be calculated according to equation (4). Attention, the three vertices of triangle are in order and can not be inversed. The triangle area calculated according to equation (4) may be minus. Only when point P lies in the triangle, the three components of area coordinates are all between 0 and 1.

5 Streamline construction and integration step

The mostly used numerical integration methods in streamline integration are Euler scheme, second order Runge-Kutta scheme and fourth order Runge-Kutta scheme [6]. The order and step of streamline integration should take both the accuracy and computational cost consider on. Euler scheme is too inaccurate although it is fast. Higher order integration techniques, such as fourth order Runge-Kutta, are more accuate, but lead to increased computational loads. The second order Runge-Kutta technique to integrate the streamline is adopted in this paper.

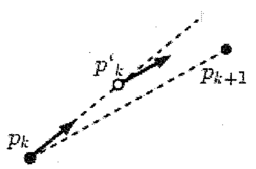

Figure 3 Runge-kutta second order integrator (The empty circle represents the intermediate point, and the gray disk represents the Euler integrated point).

As shown in Figure 3, this method introduces an intermediate point p_k' between p_k and p_{k+1} to increase the precision of the computation, where:

$$\begin{cases} p_k' = p_k + \dfrac{1}{2} h * v(p_k) \\ p_{k+1} = p_k + h * v(p_k') \end{cases} \qquad (7)$$

In [7], Buning suggested to choose the integration step size based on the cell size and the inverse of velocity magnitude. If the integration step is too small, the process of streamline computing may be long, while the integration step too big, the streamline that we got are more error-prone, and may lost the key features of flow. Considering the character of second order Runge-Kutta integration, the inscribed circle radius of triangle is used as the integration step. Due to the triangle mesh is adaptive and Delaunay, the contradiction between speed and accuracy is solved ideally. The inscribed circle radius of triangle can be calculated by the following equation:

$$R_{InscribedCircle} = \frac{2 \times S_{\Delta ABC}}{P_{\Delta ABC}} \qquad (8)$$

where $S_{\Delta ABC}$ stands for the area of triangle, $P_{\Delta ABC}$ stands for the perimeter of triangle.

6 Point location in triangle mesh

In the process of streamline integration, point location algorithm is performed for every point of streamline. So the efficiency of point location algorithm is crucial to the whole streamline integration algorithm.

"Compass" algorithm which based on the "wing-edge" data structure of triangle is adopted to speed the point location process. The "Compass" algorithm is illustrated as follows:

"Compass" algorithm:

Input: Point P in streamline that we have just gotten and triangle set TS

Output: The triangle that contains point P

Procedure:

1) Get an arbitrary triangle T from TS, as the start triangle for searching;
2) Calculate the area coordinates of P for triangle T: the location relationship of point P and triangle T by the area coordinates can be calculated:
   a) If the three components of area coordinates are all positive, then the current triangle contains point P, return triangle T and end.
   b) If there have the negative area coordinates, select the smallest area component and get the edge E that the smallest area component opposed. Then get the triangle T' that has the common edge E with triangle T by the "wing-edge" data structure. And set the triangle T' to current triangle T, goto (2).

Supposed there is a domain which is composed of n triangle cells. Generally, the algorithm time complexity for point location is O(n). Regarding on "compass" algorithm, the worst situation is to search the diagonal of triangle in rectangular domain, and search the diameter triangle in circle domain. So the time complexity of "compass" algorithm is

$O\left(\sqrt{n}\right)$ and hence improves the streamline tracing speed.

7 Algorithm for constructing isolines, streamlines and examples

In section 2, the constrained Delaunay triangulation for domain has been obtained. The isoline generating algorithm based on triangulation is as follows:

Iso reservoir pressure line Generating Algorithm:

Input: Constrained Delaunay triangulation, reservoir pressure value that need to be isolined

Output: The iso reservoir pressure line set

Procedure:

1) For every triangle T in TS,
   a) For every reservoir pressure value V that need to be isolined, new a segment set ISS, to store the isoline segment,
      i) If all three pressure values of triangle vertex are greater or less than V, continue;
      ii) Else, compute the two points on two edge of triangle that have same pressure value V according to linear interpolation. New a segment whose two endpoints coincide the iso value on two edges and add it to the end of ISS.
2) For every pressure value, link the start point and endpoint of ISS, to get the polyline set of a certain underground pressure value.
3) In order to get smooth isolines, take the PtSet of every polyline as data points, compute the control points of cubic B-spline that can pass these data points. Use the cubic B-spline to replace original polyline as isoline. The generated isolines are satisfied because they are C^2 continous.

Regarding on the above algorithm, during the process of isoline generating, all triangles were only visited once. So the time complexity of isoline generating algorithm is O (n). It is in proportional to the number of triangles.

As for streamline constructing algorithm, the first problem is to determine when the streamline will stop. If one of the three conditions was satisfied, streamline integration will stop:

(1) The streamline hits the boundary of domain;

(2) The vector value of current integration point is zero. This means current integration point is critical point;

(3) The streamline has reached the maximum integration times that the user set, or exceed the length of streamline.

The algorithm for streamlines construction is as follows:

1) Given seed points in the physical space;
2) For each seed point, find triangle cell containing initial position;
3) While none of three stop conditions of streamline integration is satisfied:
   a) Determine velocity at current position by interpolation;
   b) Determine the integration step according to the inscribed circle radius of triangle containing point P;
   c) Calculate new position of point P;
   d) Find triangle cell containing new position of point P by "compass" method.

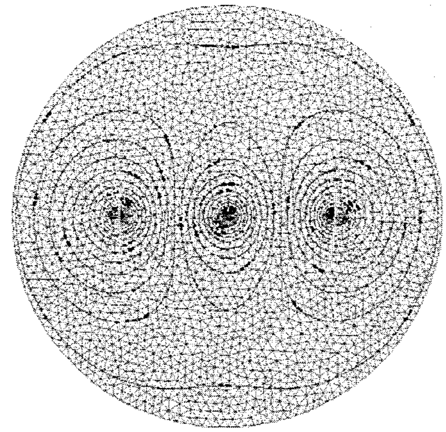

Figure 4a Isolines and streamlines in Plane XOY

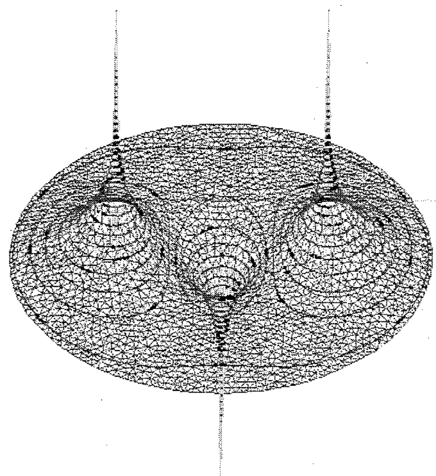

Figure 4b Reservoir pressure value magnified 1000 times as Z value, extends to 3D.

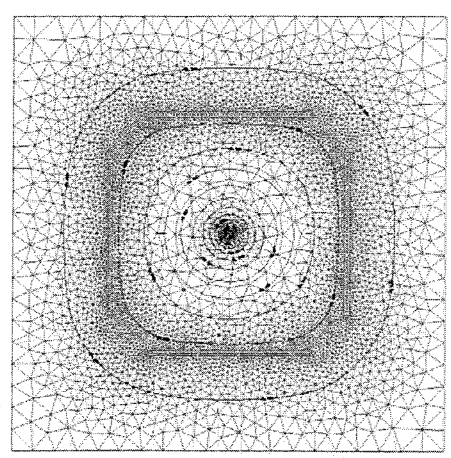

Figure 5a Isolines and streamlines in Plane XOY

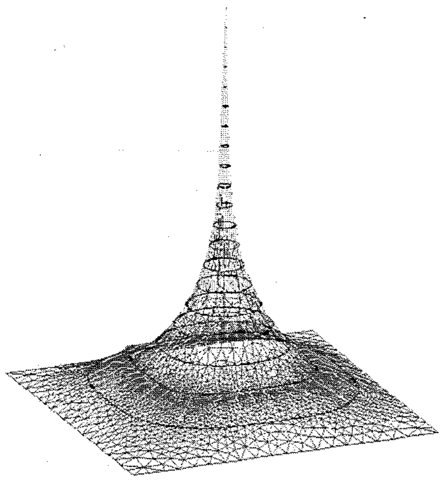

Figure 5b Reservoir pressure value magnified 500 times as Z value, extends to 3D.

Figure 4 and Figure 5 are two examples. Red lines are Delaunay triangulation. Blue lines are generated reservoir pressure isolines. Green lines are generated streamlines. Figure 4a and Figure 5a are planar visualization graph. Take reservoir pressure value magnified 1000 times as Z value, and extends to three dimensional space, the isolines and streamlines in 3D space was shown in Figure 4b. In Figure 5b, the reservoir pressure was magnified 500 times. Examples show that the reservoir pressure isolines and streamlines are perpendicular both in 2D and 3D, which coincide with theoretical situation. Streamline distributions in Figure 4 and Figure 5 can clearly visualize the reservoir fluid's movement tracks between the producers and/or injectors.

8 Conclusions and future work

Isolines and streamlines are effective way of analyzing and visualizing fluid motion in steady flow. Algorithms of generating isolines and streamlines are presented. A novel gradients calculation method and fast point location algorithm based on "compass" algorithm are also discussed in this paper. Examples show the isolines and streamlines are perpendicular both in plane and in space. The simulation result is consistent with the theoretical result. Algorithms presented in this paper have been successfully applied to oil reservoir simulation software to analyze and visualize the results of numerical computation. Future work will be focus on the construction of stream surface and stream ribbon, which are more effective way for visualize the flow through porous media.

Acknowledgements

This research is partially supported by the Natural Science Foundation of Beijing under Grant No. 4062010.

References

[1] HouJian, WangYudou, ChenYueming, Research on streamline distribution of flow through porous media with complex boundary, Chinese Journal of Computational Mechanics, Vol.20, No.3, June 2003, P335~338 (In Chinese)

[2] A. Mebarki, P. Alliez, and O. Devillers, "Farthest Point Seeding for Efficient Placement of Streamlines," Proc. IEEE Visualization '05, pp. 479–486, 2005

[3] Charles D. Hansen, Chris R. Johnson., The visualization handbook, Elsevier-Butterworth Heinemann Press, 2005

[4] Martin Isenburg, Yuanxin Liu, Jonathan Shewchuk, and Jack Snoeyink, Streaming Computation of Delaunay Triangulations, ACM Transactions on Graphics 25(3): 1049-1056, July 2006. Special issue on Proceedings of SIGGRAPH 2006.

[5] Li Haisheng, The Study on the Theory and Application for Weighted Conforming Delaunay Triangulation in 3-D Data Field Visualization[D], Ph.D. Thesis, Beihang University, July 2002 (In Chinese)

[6] Steven C. Chapra, Raymond P. Canale, Numeriacal Methods for Engineers, McGraw-Hill Companies, Inc. 1998

[7] Buning, P. Numerical algorithms in cfd post-processing. In Computer Graphics And Flow Visualization in Computational Fluid Dynamics, L. S. 1989-07, Ed. Von Karman Institute for Fluid Dynamics, 1989, pp. 1--10.