

An Automatic Method for Complete Triangular Mesh Conversion into Quadrilateral Mesh for Multiple Domain Geometry

Lu Sun^{1,2}, Gour-Tsyh Yeh^{1,*}, Fang-Pang Lin³, and Guoqun Zhao²

¹*Institute of Applied Geology, National Central University, Taoyuan, Taiwan, R.O.C.*

²*Engineering Research Center for Mould and Die Technologies, Shandong University, Shandong Province, P. R. China*

³*Data Computing Division, National Center for High-Performance Computing, Hsinchu, Taiwan, R.O.C.*

Received 16 January 2015, revised 24 August 2015, accepted 14 September 2015

ABSTRACT

This research developed an automatic two-dimensional finite element meshing system to resolve practical engineering problems in the fields of geology, hydrology, and water resources. This system first used the Delaunay triangulation method to create reasonable-density triangular mesh and then converted it into quadrilateral mesh by combining proper pairs of adjacent triangles. A series of combination patterns aiming at three cases were established. The effect of the number of boundary edges on the subsequent meshing procedures were studied and summarized. For the geometry with multiple domains an adjustment method is proposed to completely eliminate the residual triangles during quadrilateral meshing through adjusting the number of boundary edges in each loop to be even. A special boundary loop identification method is proposed for priority treatment. Corresponding treatment methods aimed at three different situations are established for common boundary loops. For a certain boundary loop with an odd number of boundary edges, the appropriate edge for new point insertion is determined by the position properties and relative density errors. Practical applications confirm that the method proposed in this paper could successfully implement the full conversion from the triangular mesh to the quadrilateral mesh.

Key words: Delaunay, Triangular mesh, Quadrilateral mesh, Conversion, B-spline, *Jacobian*

Citation: Sun, L., G. T. Yeh, F. P. Lin, and G. Zhao, 2016: An automatic method for complete triangular mesh conversion into quadrilateral mesh for multiple domain geometry. *Terr. Atmos. Ocean. Sci.*, 27, 139-152, doi: 10.3319/TAO.2015.09.14.01(Hy)

1. INTRODUCTION

To study complex problems in practical applications of water flow, thermal transport, salinity, sediment, and pollutant transport, numerical simulations have been employed for several decades. The finite element method was introduced into engineering applications and gained popularity and effectiveness in dealing with various types of troublesome issues. The finite element method can approximate irregular geometry more accurately because of its flexibility in choosing computational grid sizes.

The whole finite element simulation and numerical analysis process includes three major stages: pre-process, computation, and post-process. Finite element mesh generation is the essential aspect of preprocessing, which is still sometimes a bottleneck in the overall simulations.

Because of the explicit definition requirements for

each nodal point and element connectivity, the finite element meshing output preparation becomes very tedious as the computational geometry of a physical problem becomes very large and complicated. In practical applications, such as computational hydraulics, the numerical results are converged in order to insure the computational grids should be refined globally or locally. An automatic mesh generator is required to ease the burden in the finite element computations. This generator for finite element simulations should generate node coordinates and the connectivity of the elements with given specifications. The interaction and mapping relationships between one-dimensional river/stream, junctions with or without storage and two-dimensional overland flows (or surface runoff) are required as output data according to the users' requirements. In view of these demands, key finite element meshing techniques are researched in detail in this paper to develop an effective generator to solve practical engineering applications in

* Corresponding author
E-mail: gyeh@ncu.edu.tw

geology, hydraulics, hydrology, water resources, fluid mechanics, and mechanics, etc.

The finite element mesh is divided into two categories, two-dimensional mesh, and three-dimensional mesh, according to the solution space. The two-dimensional mesh further contains triangular mesh, quadrilateral mesh and hybrid mesh (Ito 2013).

Among the numerous kinds of triangular mesh generation methods, the Delaunay triangulation method is the most widely used method. It has two crucial criteria: maximum-minimum angle criterion and in-circle criterion. Sibson (1978) proved that the Delaunay triangulation result for an arbitrary set of given points in a common plane possessed the optimal advantage. The sum of minimal internal angles of all Delaunay triangles obtains the maximal value and the maximal internal angles will produce the minimal value. The maximum-minimum angle criterion is able to avoid producing long and thin elements with small internal angles automatically. The in-circle criterion signifies that the circum-circle of each Delaunay triangle will not contain any other points in the given set. Bowyer-Watson algorithm (Bowyer 1981; Watson 1981) just takes advantage of the in-circle criterion. In recent years, Delaunay triangulation algorithm has attracted many research scholars because of its excellent flexibility and extensibility (Žalik 2005; Ebeida et al. 2011). Therefore, this paper adopts Delaunay triangulation method to generate triangular meshes, which are used as the background grid to create quadrilateral meshes.

The most extensively applied quadrilateral meshing methods are grouped into two main categories: direct method and indirect method (Ho-Le 1988). The direct method generates quadrilateral meshes directly based on the geometric features of the analyzed geometry. Direct methods include the medial-axis method (Tam and Armstrong 1991), quad tree method (Yerry and Shephard 1983), and advancing front method (Blacker and Stephenson 1991), etc. The indirect method is also known as the triangle-to-quadrilateral conversion method. That is, triangular mesh is generated first and then converted into quadrilateral mesh (Shimada and Itoh 1995; Borouchaki et al. 1996).

The indirect method takes advantage of the benefits of triangle meshing, thus has more advantages than the direct method in controlling mesh density and element size, meshing efficiency and generation versatility, etc (Itoh and Shimada 2002). There are two implementation styles for the indirect method: the first is the division method; the second is the combination method. The division method subdivides a single triangle into three quadrilaterals by adding three new points at the midpoints of the three edges and one point in the triangle centroid. This method is simple, easy to implement and very versatile. Because a large number of irregular nodes are not connected to four quadrilaterals, the element quality cannot be guaranteed. The addition of the new points and elements has a serious effect on the mesh density dis-

tribution, possibly leading to a generated mesh that cannot satisfy the meshing density requirements specified by the users. In addition, the insertion of the new points on curved boundaries might cause curvature errors and area loss.

The combination method generates quadrilateral meshes through combining each appropriate pair of adjacent triangles into a single quadrilateral. This method is relatively difficult in algorithms because the choice of triangle pairs has a great impact on the conversion effect and mesh quality. However, the combination method is able to produce reasonable quality meshes that can accurately capture the geometric features of the input geometry. Since there is no new point to be inserted, the density control of users for mesh elements and nodes can be fully satisfied. As a result this paper uses the combination method to produce quadrilateral meshes.

The implementation procedures for quadrilateral mesh generation using the combination method are often troublesome and time-consuming for complex geometries, especially multiple-domain geometries. The complete conversion from triangles into quadrilaterals for each sub-domain has become a challenging research issue. The conformity of mesh elements and nodes between two adjacent sub-domains must be perfectly ensured. Until now, most literatures focused mainly on research on conversion strategies for triangles and resulting mesh quality optimization. Few papers have been published that pay attention to the number control of mesh elements and nodes for complicated geometries with multiple sub-domains.

This research conducted an intense examination of the key techniques and treatment strategies in the triangle and quadrilateral meshing procedures, containing Voronoi diagram construction, boundary point generation, the insertion of new points, the conversion from triangles into quadrilaterals and the treatment of residual triangles. The odd and even parity properties and changing rules between the numbers of boundary edges and mesh elements in each sub-domain were studied and summarized through numerous practical examples. An adjustment method is proposed for controlling the number of mesh elements and triangle conversion pairs by controlling the number of boundary edges in each loop. Corresponding treatment methods for adjusting the number of boundary edges for special and common loops are established. The relative density error concept is proposed to determine the appropriate boundary edge where the new point is inserted. The reliability and effectiveness of the methods presented in this paper are demonstrated using several practical examples.

2. KEY TECHNIQUES IN TWO-DIMENSIONAL MESH GENERATION

The two-dimensional finite element meshing system developed in this paper adopts the Delaunay triangulation method to generate density-controlled triangular meshes.

Based on the Delaunay meshes an improved conversion method is applied to further generate quadrilateral meshes by combining appropriate pairs of adjacent triangles.

2.1 Theoretical Foundation

The Delaunay triangle meshing program was developed based on the Voronoi diagram. The curved input geometry boundaries are constructed using B-splines according to the point spacing specified by the users.

2.1.1 The Voronoi Diagram

Suppose that a set of points S in a two-dimensional plane are given, with n distinct points in all. For each point P_i , its Voronoi territory, labeled as $V(P_i)$, is defined as the locus of points in the common plane whose distance is closer to P_i than to any other points of S . $V(P_i)$ is expressed as (Bowyer 1981):

$$V(P_i) = \{P / d(P, P_i) \leq d(P, P_j)\}, P_i, P_j \in S, j \neq i \quad (1)$$

The Voronoi cell of each point in set S can be expressed as a convex polygon using Eq. (1). As a result, the whole plane where S is located is divided into n Voronoi polygons, each of which is associated with a unique point. The n convex polygons form the Voronoi diagram of the point set S . The straight line duality of the Voronoi diagram creates Delaunay triangles. Each Voronoi vertex corresponds to a unique Delaunay triangle and vice versa.

2.1.2 B-Spline

B-spline is short for basis spline, first proposed by Schoenberg (1946). It possesses strong adaptability and flexibility, and is widely used in the fields of computer-aided design and manufacturing.

Assuming that $\mathbf{P}(u)$ represents the position vector of an arbitrary point on a curved boundary, the k -th order B-spline along variable u is defined by:

$$\mathbf{P}(u) = \sum_{i=0}^n \mathbf{B}_i N_{i,k}(u) \quad (2)$$

where \mathbf{B}_i indicates the position vectors of the control points. $N_{i,k}(u)$ denotes the normalized B-spline blending functions. While constructing a k -th order B-spline, the Cox-de Boor formula (Cox 1972; De Boor 1972) of the i -th blending function is defined as:

$$N_{i,1}(u) = \begin{cases} 1 & u \in [u_i, u_{i+1}) \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

$$N_{i,k}(u) = \frac{u - u_i}{u_{i+k-1} - u_i} N_{i,k-1}(u) + \frac{u_{i+k} - u}{u_{i+k} - u_{i+1}} N_{i+1,k-1}(u) \quad (4)$$

where u_i represents the knots of the B-splines, with $u_i \leq u_{i+1}$. Note that the entire value of any term in Eq. (4) is assigned zero when the denominator is zero.

2.2 The Geometric Features of the Input Model

2.2.1 Input Data

The meshing system developed in this paper requires two input data aspects, the formed points, and the boundary edges of the geometry.

2.2.1.1 The Formed Points

Assume there are np points in total in the set S to describe the boundary features of the input geometry. First of all the position coordinates in the x - and y - axes of the Cartesian coordinate system for each point are required to be given, as $P_i(x_i, y_i) \in S, i = 1, 2, 3, \dots, np$.

In addition, users usually put forward different requirements for the density distribution of mesh elements. Some geometric features that need particular analysis and simulation require a large density to ensure computation accuracy, such as the large-curvature boundaries, slight-size features, and the domains of particular interest to users. The geometric features that do not need focus require a relatively small density to save computation time and storage space, such as the boundaries with a smooth curvature or the domains of no interest to users. The density of mesh elements and points should therefore be specified previously by users. To facilitate computation the mesh density is controlled by the point spacing. The point spacing in this paper is defined as the lengths of the edges connected to the point.

2.2.1.2 Boundary Edges

The data information stored for each boundary edge is composed of the serial numbers of its two endpoints as well as its curved or straight status. Suppose that there are nb boundary edges, then each edge is denoted as $b_i(e_0, e_1), i = 1, 2, 3, \dots, nb$, where e_0 and e_1 are the two endpoints.

It should be noted that in order to ensure successful Delaunay triangulation performance the input boundary edges for each meshed region must follow such a rule: the outer edges are arranged in counterclockwise order and the inner edges are arranged in clockwise order. Therefore, the boundary edges on the intersecting lines between two connected geometric sub-domains are overlapped. If a boundary edge in a sub-domain is $b_i(e_0, e_1)$, then its corresponding overlapped edge in the neighboring sub-domain is $b_j(e_1, e_0), j \neq i$.

Figure 1 shows a geometric model that requires meshing, containing two domains with different types of

materials, the outer domain A and the inner domain B. These two domains should be meshed separately and compatibly. P_1, P_2, \dots, P_{12} represent the data points of the geometry, with $np = 12$. These 12 data points create 16 boundary edges, $nb = 16$. Since the input geometry includes two connected sub-domains, four pairs of overlapped edges are present on the intersecting boundaries, as $P_9 - P_{10}, P_{10} - P_{11}, P_{11} - P_{12}, P_{12} - P_9$ in sub-domain A and $P_{10} - P_9, P_{11} - P_{10}, P_{12} - P_{11}, P_9 - P_{12}$ in sub-domain B. The decimal in the parentheses after the label for each point indicates its required point spacing. If the specified spacing of a point is less than the actual length of the boundary edge connected to it, new points should be generated on this edge.

2.2.2 The Identification of the Geometric Features

In order to achieve effective finite element mesh generation, identifying the boundary loops, characteristic points and boundary segments of the geometric model must be accomplished first.

The boundary loop is defined as the closed ring made up with end-to-end boundary edges. Each geometric model sub-domain is bounded by one or more boundary loops. For example, the geometric model as shown in Fig. 1 has two boundary loops, the outer loop L_1 and the inner loop L_2 . The outer sub-domain A is bounded by both of these two loops and the inner sub-domain B is formed only by the inner loop L_2 .

The characteristic point of the geometry is defined in this paper as the point that is shared by three or more boundary edges. In general, the characteristic point is connected to more than two sub-domains.

Based on the characteristic point concept this paper establishes a segmentation rule for the geometric boundaries as follows: (1) The straight edge is considered a single segment itself; (2) The end-to-end curved edges are integrated to form a whole segment; (3) If a characteristic point is encountered, its connected edges are regarded as belonging to different segments. For the geometric model in Fig. 1, all of the edges on loop L_1 are curved and form a whole segment, while the edges on loop L_2 are straight and each is treated as an individual segment.

2.3 Key Techniques in Delaunay Triangular Mesh Generation

Figure 2 shows the flow chart for describing the whole Delaunay triangulation process, including a total of nine steps. The techniques for generating boundary points and inserting new nodes are two key research aspects.

2.3.1 The Generation of Boundary Points

The boundary points are generated according to the required point spacing specified by users. For an arbitrary

edge b_i ($i = 1, 2, \dots, nb$), suppose that its two endpoints are P_a and P_b with the required point spacing as $ps(P_a)$ and $ps(P_b)$.

First, find the midpoint of b_i , denoted as P_m . If b_i is a straight edge, P_m is just the midpoint of $\overline{P_a P_b}$. If b_i belongs to a curved segment, P_m is obtained by interpolation at $u_m = (u_a + u_b)/2$ using Eqs. (2), (3), and (4).

Second, compute the practical point spacing at P_m using:

$$ps'(P_m) = \frac{1}{2}(|P_a P_m| + |P_m P_b|) \quad (5)$$

Third, determine the lower value between $ps(P_a)$ and $ps(P_b)$, denoted as $ps(\min)$.

Next, check whether $ps'(P_m)$ is larger than $ps(\min)$. If yes, accept P_m . If no, reject it. If P_m is accepted, a required

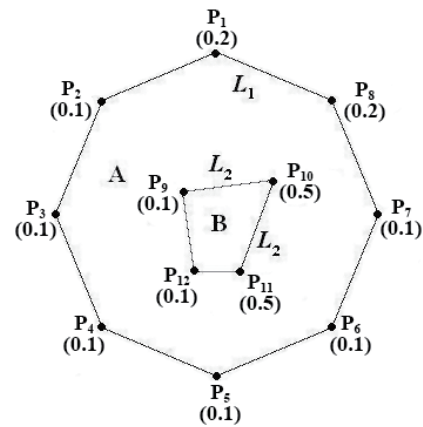


Fig. 1. The boundary features of the input geometry.

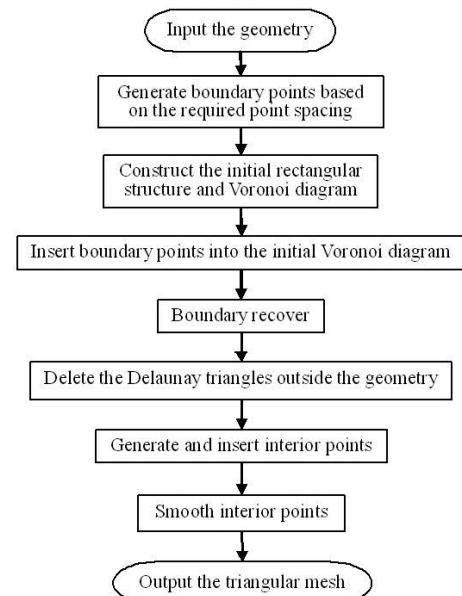


Fig. 2. The procedures of Delaunay triangular mesh generation.

point spacing, $ps(P_m)$, needs to be assigned to it using the following interpolation equation:

$$ps(P_m) = \alpha \frac{ps(P_a)|P_m P_b| + ps(P_b)|P_a P_m|}{|P_a P_m| + |P_m P_b|} \quad (6)$$

The method stated above is used to generate boundary points for the geometry shown in Fig. 1 (Fig. 3a). The number of the boundary points is increased to 67 and the number of boundary edges is increased to 78. Figure 3b provides the current boundary contour of the geometry, where the black points represent the boundary points on loop L_2 .

2.3.2 The Insertion of New Points

Since each Voronoi vertex corresponds to a Delaunay triangle, a new point insertion can be realized through the destruction and reconstruction of the Voronoi structures. The whole process is divided into the following four steps (Bowyer 1981):

- (1) Identify the Delaunay triangle containing the new point using the right-hand rule. For an arbitrary Delaunay triangle the introduced point forms three new triangles with its three edges. Next, check whether these three new triangles have the same orientations. If yes, the introduced point is inside the selected triangle. If no, the introduced point is outside the selected triangle and located on the side with the different orientation.
- (2) Determine the Voronoi vertices to be deleted. According to the in-circle criterion the Voronoi vertices whose circum-circles contain the newly inserted point should be deleted.
- (3) Construct a local Voronoi diagram and Delaunay triangles. When the Voronoi vertices identified in step (2) are deleted, an empty convex polygon appears in the deleted region. The edges forming the empty polygon are the residual edges of deleted triangles in practice. Every residual edge gives rise to a new Voronoi vertex with the introduced point.
- (4) Merge the local Voronoi diagram into the global diagram.

Figure 3c shows the resulting mesh after inserting all of the boundary points into the Voronoi diagram. Figure 3d shows the final mesh after generating and smoothing the interior points, which is used as the background grid to produce quadrilateral mesh in the subsequent procedures.

2.4 Key Techniques in Quadrilateral Mesh Generation

The quadrilateral mesh is generated indirectly by combining adjacent pairs of triangles. Figure 4 shows the whole quadrilateral mesh generation process, consisting mainly of four aspects: the initial combination of triangles, the treat-

ment of residual triangles, topological optimization, and node smooth.

2.4.1 The Initial Combination of Triangles

In order to ensure that there are as many triangles combined into quadrilaterals, this paper employed the searching advancing-front edges method to create quadrilaterals layer by layer (Owen et al. 1998, 1999).

The inherent boundaries of the input geometry are regarded as the first layer of advancing-front edges for quadrilateral creation. Each appropriate pair of adjacent triangles is combined into a quadrilateral by taking the advancing-front edge as the base. The possible combination patterns based on each advancing-front edge provide two options: between the two connected triangles sharing the other two edges of the base triangle, select one to form a quadrilateral.

There are many factors affecting the selection of combination patterns, such as the priority levels of the elements, the number and positions of advancing-front edges, etc. This paper made an intensive research on the various types of connection relationships among connected triangles. Five combination patterns aimed at three cases were established (Fig. 5) (Sun et al. 2015). The first case (Fig. 5a) is suitable for the triangle with two advancing-front edges. Since the base triangle fit for this case has only one combination candidate, it possesses the highest priority level. This case should be identified and converted first so as to avoid omission during combination. The second case (Fig. 5b) aims at the situation where two adjacent triangles contain two connected advancing-front edges. The third case (Fig. 5c) is suitable for the common base triangle with only one advancing-front edge.

2.4.2 The Movement of Residual Triangles

In the resulting mesh after initial combination, some residual triangles scattered in the quadrilaterals always remain. These residual triangles are treated and eliminated through movement and recombination. For an arbitrary residual triangle, the movement process includes the following three steps (Sun et al. 2015):

- (1) Find the moving target of the residual triangle, that is, the residual triangle whose position is closest to it. Compute the center of the target triangle, labeled as P_m .
- (2) Determine the movement path of the residual triangle. This paper introduced the determinant value of the normalized *Jacobian* matrix (the scaled *Jacobian*) (Knupp 2003; Garimella et al. 2004) to determine the movement path of the residual triangle.

Three new triangles with rotation directions are created first by the three edges of the residual triangle and the center of the target triangle, P_m .

Then, compute the scaled *Jacobians* of these three new

triangles at point P_m by:

$$A = \begin{pmatrix} x_p - x_1 & y_p - y_1 \\ x_0 - x_p & y_0 - y_p \end{pmatrix} \quad (7)$$

$$Jacobian = \det(A) \quad (8)$$

where (x_p, y_p) represent the coordinates of P_m , (x_0, y_0) and (x_1, y_1) represent the starting and ending points of the three edges of the residual triangle.

Compare the three values of the scaled *Jacobians* computed using Eqs. (7) and (8). The edge that forms the triangle with the smallest negative value is specified as the movement path.

(3) Move and recombine the residual triangles. This operation is carried out through the destruction and reconstruction of adjacent quadrilaterals (Heighway 1983).

First, find the connected triangle that shares the movement path determined in step (2).

Second, identify the quadrilateral in which the connected triangle belongs. In this case, there are two ways to

combine the residual triangle: one is its connected triangle sharing the movement path, the other is the new triangle derived from the common-edge exchanging of the identified quadrilateral. Suppose that these two ways are both applied and compute the distances from the new residual triangle to P_m respectively. Accept the way that yields the smaller value.

The movement process is repeated until two residual triangles are connected with a common edge and thus combined into a quadrilateral.

3. THE ODD AND EVEN PARITY CONTROL ON THE NUMBER OF RESIDUAL TRIANGLES

In theory, every two residual triangles in the same meshed domain are able to be recombined into a quadrilateral through moving to each other along an appropriate path. However, in practical applications, the specific problems are more complicated. The number of residual triangles and the boundary features of the geometric domains both play an important role in the feasibility and integrity of quadrilateral conversion.

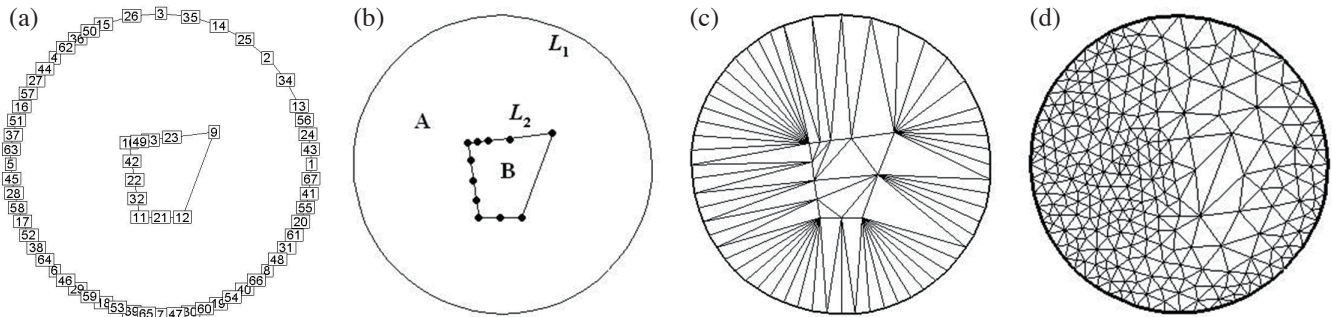


Fig. 3. Delaunay triangular mesh generation (a) all the generated boundary points, (b) the boundary contour after generating boundary points, (c) the resulting mesh after inserting boundary points, (d) the final Delaunay triangular mesh.

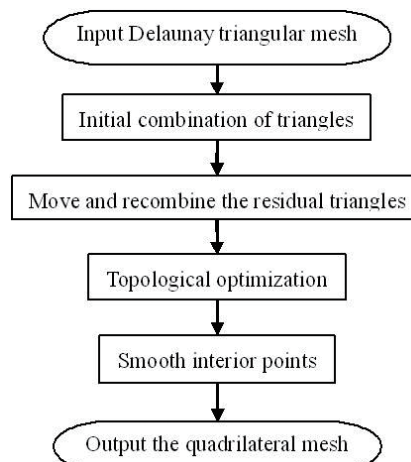


Fig. 4. The procedures for quadrilateral mesh generation.

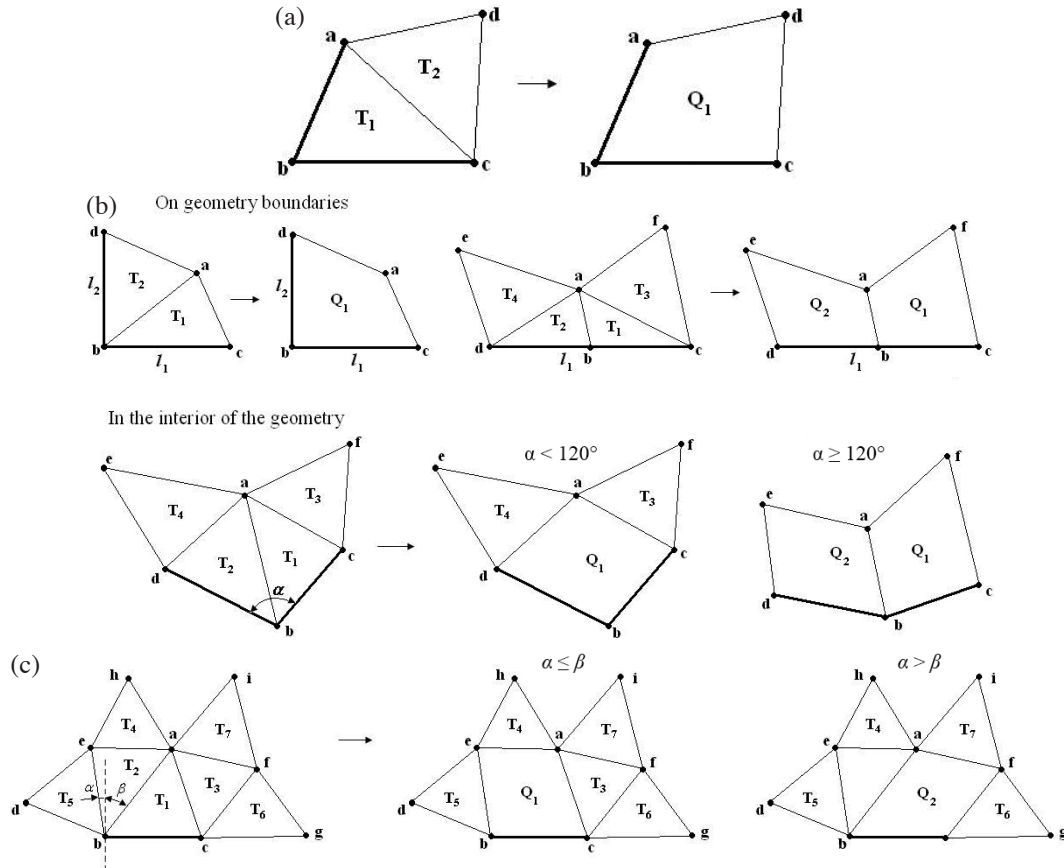


Fig. 5. Combination patterns for three cases. Thick real lines represent the advancing-front edges (a) a triangle with two advancing-front edges, (b) two triangles with two connected advancing-front edges, (c) a common base triangle with only one advancing-front edge.

3.1 The Treatment Strategies of Residual Triangles

Due to the diversity of the boundary features for the geometry, the complexity in treatment strategies of residual triangles has different degrees. In general, the number and distribution of the sub-domains have a considerable affect on the ways to handle residual triangles.

3.1.1 The Situation for the Geometric Model with One Sub-Domain

If the geometric model to be analyzed contains only one single domain, there will be no overlapped edge on the input boundaries. In this situation, a full conversion from triangles into quadrilaterals is able to be achieved regardless whether an odd or even number of triangles is generated in the Delaunay mesh.

When the Delaunay triangular mesh consists of an even number of triangles, the resulting mesh after adjacent triangle combination still contains an even number of residual triangles. After movement, all of the residual triangles are recombined in pairs and there is no residual triangle left in the final quadrilateral mesh

When the Delaunay triangular mesh contains an odd

number of triangles, the resulting mesh after initial combination still contains an odd number of residual triangles. With the movement and recombination of residual triangles, one and only one residual triangle will remain in the finally generated mesh. In this case, the only residual triangle can be treated using three stages: (1) Move it onto the nearest boundary edge of the geometry. (2) Insert a new point on the nearest boundary edge. If this edge belongs to a curved segment, B-splines should be constructed first. (3) Convert the only residual triangle into quadrilaterals according to the transformation template (see Fig. 6). In this way the complete conversion from triangles into quadrilaterals is realized.

3.1.2 The Situation for the Geometric Model with Multiple Sub-Domains

If the meshed geometry contains two or more sub-domains, then the edges lying on the intersecting boundaries between neighboring sub-domains are overlapped. As proper pairs of adjacent triangles are combined into quadrilaterals, the integrity of the combination is determined to the maximum extent by the number of the triangles in each sub-domain.

After the geometry is triangulated, some sub-domains

may contain even numbers of triangles and some may contain odd numbers. Since the triangles are combined in pairs, the number of the triangles that have been converted is certainly even. If a sub-domain contains an even number of Delaunay triangles, it will leave an even number of residual triangles in the resulting mesh after preliminary combination. All of the residual triangles in this sub-domain are able to be eliminated using pairwise movement and recombination. However, if a sub-domain contains an odd number of Delaunay triangles, there will be an odd number of residual triangles left in this meshed region after preliminary combination. It should be noted that the movement of each residual triangle is only restricted to the sub-domain that it belongs to. Thus, a single residual triangle will remain in this sub-domain after treatment. If this sub-domain is located on the boundaries of the geometry, the single residual triangle can be moved to the nearest outer edge and converted into quadrilaterals using the transformation template in Fig. 6. If this sub-domain is inside the geometric model, all of the boundary edges surrounding it are overlapped. In this case, while the single residual triangle is moved to the nearest overlapped edge and topologically handled, it will lead to an incompatible meshing result. This is because of new point insertion on the overlapped edge during applying the conversion template in Fig. 6. The newly inserted point cannot be simultaneously compatible with the elements in both connected sub-domains and will thereby produce unqualified meshes.

Figure 7a shows the resulting mesh after preliminary combination for the triangular mesh in Fig. 3d. The figure shows that there are 17 residual triangles in the outer domain A and 3 residual triangles in the inner domain B. The num-

bers of residual triangles in these two sub-domains are both odd. These residual triangles are then moved and recombined within their own sub-domains. The resulting mesh is shown in Fig. 7b. It is obvious that there is only one residual triangle left in each sub-domain. The treatment method for the remaining residual triangle in domain A is as follows: (1) Move it onto the outer boundaries of the geometric model (Fig. 7c); (2) Convert it into quadrilaterals according to the conversion template (Figs. 7d, e). However, this treatment method is not suitable for the only residual triangle in domain B. This is because the conversion template cannot ensure conformity for both sub-domains.

If the single residual triangle is located in the interior domain of the geometry, it cannot be converted into quadrilaterals using any topological optimization template under the premise of ensuring mesh conformity. For this reason, this problem is resolved by controlling the number of the residual triangles in the previous steps.

3.2 The Relationships Between the Number of Boundary Edges and the Number of Residual Triangles in Each Sub-Domain

Intensive research was conducted in this work on the relationships between the number of boundary edges and the number of mesh elements for a certain sub-domain using numerous triangle and quadrilateral meshing examples. Several crucial rules are summarized as follows:

- (1) If a sub-domain consists of an odd number of boundary edges it will generate an odd number of triangles during Delaunay triangulation. Similarly, if a sub-domain consists of an even number of boundary edges, it will

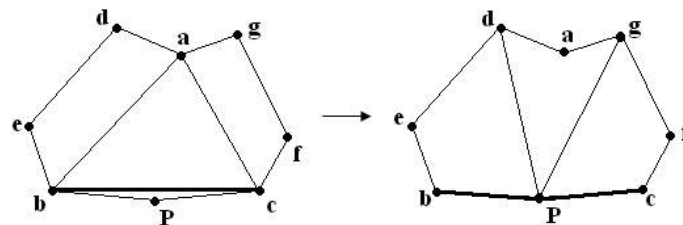


Fig. 6. The transformation template for the only residual triangle.

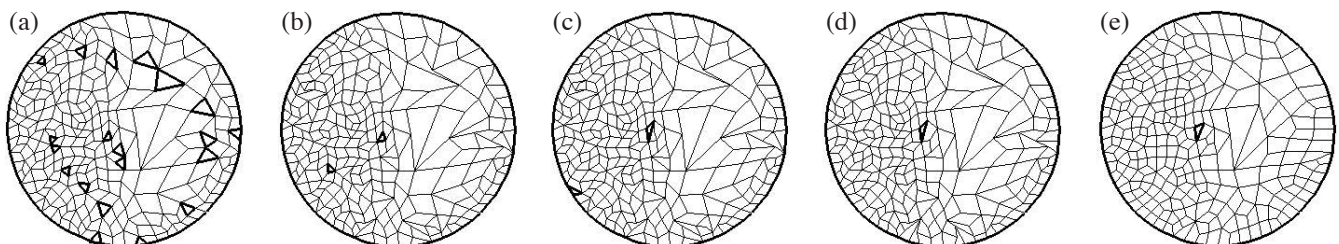


Fig. 7. Quadrilateral mesh generation (a) the initial conversion from triangles into quadrilaterals, (b) the movement and recombination of residual triangles, (c) the movement of the only remaining triangle, (d) topological optimization, (e) the final mesh after smooth.

generate an even number of Delaunay triangles.

- (2) Once a new point is inserted to the current Voronoi diagram using the method described in section 2.3.2, the number of the Delaunay triangles in this sub-domain will be changed by $2n$ (n is an integer). In other words, the odd and even parity of the number of the triangles in each domain will not be changed with the addition of new points, regardless whether boundary points or interior points are inserted.
- (3) If a sub-domain in the Delaunay triangular mesh contains an odd number of triangles, the resulting mesh after initial combination also contains an odd number of residual triangles. After movement and recombination of residual triangles, a residual triangle still remains in this sub-domain. If a sub-domain contains an even number of triangles in the Delaunay triangular mesh, the number of residual triangles in the initially combined mesh will also be even. There will be no residual triangle left in the resulting mesh after moving and recombining residual triangles.

As a result the number of boundary edges for the given geometry has a straightforward affect on a series of subsequent meshing procedures. In summary, for a certain sub-domain, an odd number of boundary edges will lead to the following results directly or indirectly: (1) This sub-domain contains an odd number of triangles in the resulting mesh after inserting boundary points; (2) This sub-domain contains an odd number of triangles after inserting interior points; (3) The number of residual triangles in the mesh after initial conversion is odd; (4) There will be one residual triangle left in this sub-domain after moving and recombining the residual triangles.

Similarly, for a certain sub-domain, an even number of the boundary edges will lead to the following results directly or indirectly: (1) This sub-domain contains an even number of triangles in the resulting mesh after inserting boundary points; (2) This sub-domain contains an even number of triangles after inserting interior points; (3) The number of residual triangles in the resulting mesh after initial conversion is even; (4) There is no residual triangle left in this sub-domain after treatment and all of the triangles are converted into quadrilaterals.

3.3 The Adjustment for the Number of Boundary Edges

Based on the rules summarized in section 3.2, this paper proposed an adjustment method to control the number of the residual triangles appearing in quadrilateral conversion through adjusting the number of boundary edges in the previous steps.

Since each sub-domain of the geometry may contain several boundary loops and each loop is probably shared by other sub-domains, the control on the number of boundary edges for different sub-domains may influence each other.

In order to avoid this cross interaction, this paper proposes adjusting the number of the boundary edges on each loop to even. In this way there will always be an even number of boundary edges in each sub-domain and thus the number of generated triangles will be even as well. If the number of boundary edges in a loop is odd an additional point is required to be inserted and an appropriate edge for the insertion of the new point should be determined at the same time.

3.3.1 The Factors Affecting the Selection of Boundary Edge

For the geometric model with multiple domains, the intersecting boundary between two adjacent sub-domains is composed of a collection of overlapped edges. The selection of the boundary edge for new point insertion is an important issue. If the boundary edge is not selected correctly, the conformity of the mesh between adjacent sub-domains will possibly be destroyed. The selection of the boundary edge is affected by two factors:

- (1) **The position properties of boundary loops:** The boundary loops for a complex geometric model surround a sub-domain formed by chains of edges lying in different positions. Some edges may be located on the outer boundaries of the geometry and only belong to this sub-domain, while other edges may lie on the intersection boundaries between two connected sub-domains and are overlapped. As the new point is inserted on the overlapped boundary edge, it must be compatible with both connected sub-domains. Thus, the new point insertion will impact the odd and even parity of the number of boundary edges in both of these sub-domains.
- (2) **The topological connections of boundary loops:** If the geometric model contains several sub-domains, different boundary loops usually contain different numbers of overlapped boundary edges and the complexity of their topological connections with adjacent sub-domains varies. When the points belonging to a boundary loop have relatively complicated topological connections the number of control methods becomes more difficult. For example, if a sub-domain is surrounded by boundary loops with one or more characteristic points, especially when they are inside the geometric model, the new point insertion process will be hindered by the mutual influence among the sub-domains connected to the characteristic points. In order to resolve this problem the boundary loop types should be identified and adjusted preferentially.

3.3.2 The Identification of Special Boundary Loops and Common Boundary Loops

Certain boundary loops with special geometric topologies need to be identified and adjusted preferentially. Only when these special loops are adjusted to have an even number

of boundary edges should other common loops be checked.

(1) The priority adjustment for special boundary loops:

The special boundary loop required for priority adjustment is identified as the loop that conforms to the following two conditions: (a) all of the boundary edges lying on it are overlapped; and (b) it includes at least one characteristic point. If a special boundary loop consists of an odd number of boundary edges, a new point must be preferentially inserted.

(2) The adjustment of the common boundary loops:

When the special boundary loops required for priority treatment have been identified and adjusted to contain an even number of boundary edges, the next step is to check the odd and even parity of the number of boundary edges on the common loops. This paper proposes corresponding adjustment rules aimed at three different situations for the treatment of common boundary loops.

The first rule is suitable for the case where the boundary loop has no overlapped edges and lies on the outer boundaries of the geometry. It should be noticed that, if a boundary loop contains no overlapped edge it will generally contain no characteristic point. If the boundary loop fit for this case consists of an odd number of boundary edges the new point can be inserted on any one of the edges belonging to it.

The second situation aims at the boundary loops whose boundary edges are all overlapped and have no characteristic point. Such boundary loops are located in the interior of the geometry independently. The geometric meaning is expressed as the entire intersection lines between two adjacent sub-domains. The odd and even parity control on the number of boundary edges for this type of boundary loop has an identical affect on its two connected sub-domains. If this type of boundary loop consists of an odd number of boundary edges, the new point can be inserted on any one of the edges belonging to it.

The third rule is fit for the situation where the boundary loop contains overlapped edges and also non-overlapped edges. In this case, these two types of boundary edges have different priority for new point insertion. In order to avoid bringing adverse impact on the odd and even parity control of boundary edges in the adjacent sub-domain, the non-overlapped boundary edges have a higher priority level for new point insertion.

3.3.3 The Further Determination of Boundary Edge for Inserting New Points

When a boundary loop is required for adjustment, regardless whether it is a special loop or common loop, there is more than one option for the edge that is chosen for new point insertion. It is necessary to find the most appropriate edge from the candidate options.

In order to reduce the error between the practical point spacing and the required point spacing to the minimal level,

this paper proposes a judgment method to determine the most appropriate edge for new point insertion based on the relative point spacing error. This method is realized using the following procedures:

First, find all of the candidate boundary edges that belong to the certain loop and are able to be selected for new point insertion. Those are all of the boundary edges on special loops or the first two cases of common loops, and the non-overlapped edges in the third case of common loops.

When all of the candidate boundary edges are identified in the first step, compute their relative density errors. The relative density error of each edge is defined by the point spacing error of its two endpoints. Assuming the starting and ending points of an edge are labeled as e_0 and e_1 , its relative density error is computed using the following equation:

$$error = \frac{[ps(e_0) - ps'(e_0)] + [ps(e_1) - ps'(e_1)]}{|e_0e_1|} \quad (9)$$

where $ps(e_0)$ and $ps(e_1)$ represent the required point spacing of points e_0 and e_1 , and $ps'(e_0)$ and $ps'(e_1)$ represent their practical point spacing. $|e_0e_1|$ indicates the length of the boundary edge.

Among all candidate edges sort out the one that obtains the smallest relative density error value. This edge is selected as the edge where the new point is generated.

Figure 8 shows the flow chart for describing the whole boundary edge number control process for each loop, where i represents the current loop to be adjusted and nl represents the number of boundary loops. It can be seen that boundary point generation contains two aspects: the first is based on the required point spacing, and the second is based on the number control.

For the geologic model shown in Fig. 1, the outer loop L_1 contains 8 boundary edges and the inner loop L_2 contains 4 boundary edges. After the boundary points are generated based on the required point spacing, the number of edges on loop L_1 is increased to an even number, 56, and that on loop L_2 is increased to an odd number, 11 (see Figs. 3a, b). That is to say, there are a total of 67 boundary edges in the outer sub-domain A (bounded by loops L_1 and L_2) and 11 boundary edges in the inner sub-domain B (bounded by loop L_2). Each of these two sub-domains consists of an odd number of boundary edges. In this case, a residual triangle will remain in each sub-domain of the resulting mesh with initial conversion (Fig. 7b). The adjustment method proposed in this paper is applied to resolve this problem. Since loop L_2 contains an odd number of edges, a new point P is inserted on it to change the odd and even property (Fig. 9a). The number of the edges in loop L_2 is increased to 12, an even number. A total of 68 boundary edges in sub-domain A and 12 boundary edges in sub-domain B are produced. Both of these two sub-domains contain even numbers of boundary edges. The

Delaunay triangular mesh with number adjustment is shown in Fig. 9b. The resulting mesh after quadrilateral conversion is shown in Fig. 9c, with 18 residual triangles in sub-domain A and 2 residual triangles in sub-domain B. The numbers of residual triangles in these two sub-domains are both even and can be recombined. The residual triangles are handled and the mesh quality is improved (Figs. 9d, e). Neither of the sub-domains contains a residual triangle and all of the triangles are converted into quadrilaterals.

4. APPLICATIONS

The meshing generator developed in this work is used in this section to create quadrilateral meshes for complex geometries using several practical engineering application sub-domains. The adjustment method proposed is applied to implement full quadrilateral conversion by controlling the odd and even number parity of the boundary edges on each loop.

The interaction between one-dimensional river and

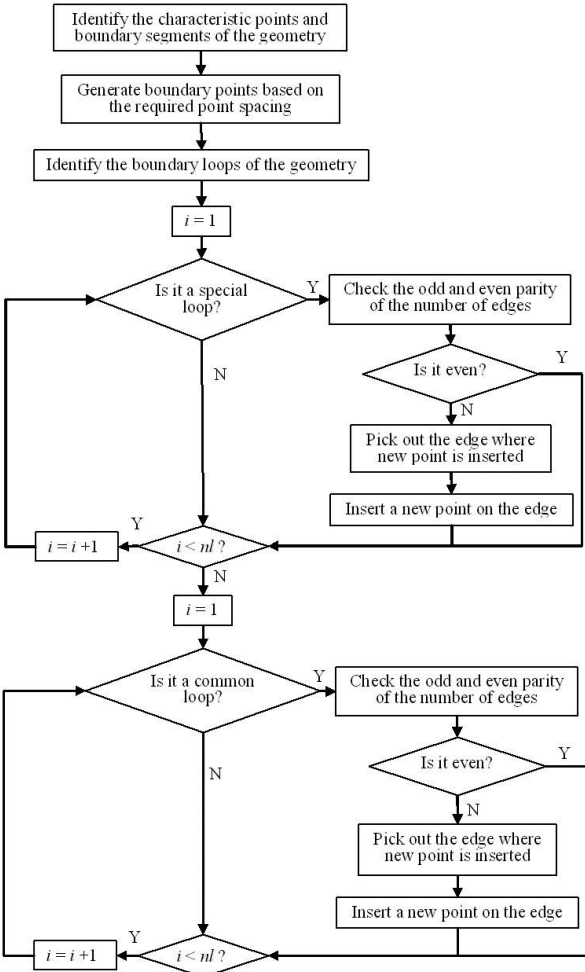


Fig. 8. The odd and even parity control on the number of boundary edges.

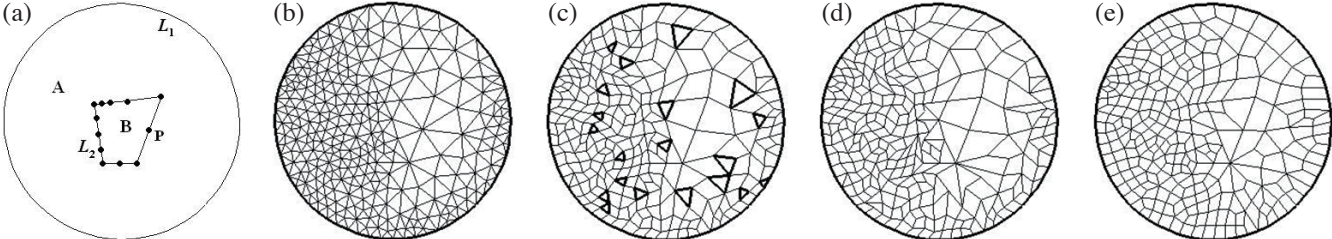


Fig. 9. The adjustment for the number of elements (a) the adjustment for the number of boundary edges, (b) Delaunay triangular mesh, (c) the initial conversion from triangles to quadrilaterals, (d) the moving and recombining of residual triangles, (e) the final mesh after quality improvement.

two-dimensional overland flows involves two cases: the first occurs between overland and river nodes, and the second between overland and junction nodes. Each river node is associated with two overland nodes. Each junction node is associated with several overland nodes. Two treatment strategies are used to handle this problem realized by establishing the mapping relationships between one-dimensional river and two-dimensional overland. The first treatment strategy (Fig. 10) locates the two mapping nodes belonging to two separate regions on the same position as the original

river node. The second treatment (Fig. 11) locates the two mapping nodes for each river point on their practical positions on the connected overland.

Figure 10 shows an actual example to resolve realistic geology and hydrology problems for the Kaoping River drainage basin in Taiwan. Figure 10a shows the boundary map of the analyzed district captured from satellite images. The meshed region consists of four domains separated by river reaches intersecting at two junctions, J_1 and J_2 (two characteristic points of the geometry). Each domain should

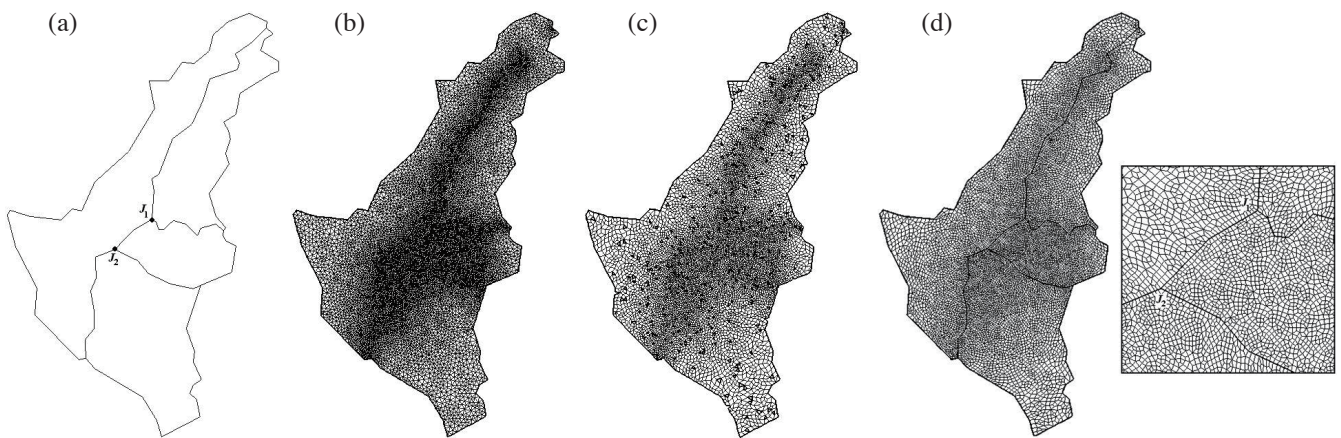


Fig. 10. Quadrilateral mesh generation for the Kaoping River of Taiwan (a) input geometric boundaries, (b) Delaunay triangular mesh, (c) resulting mesh after initial combination, (d) final quadrilateral mesh.

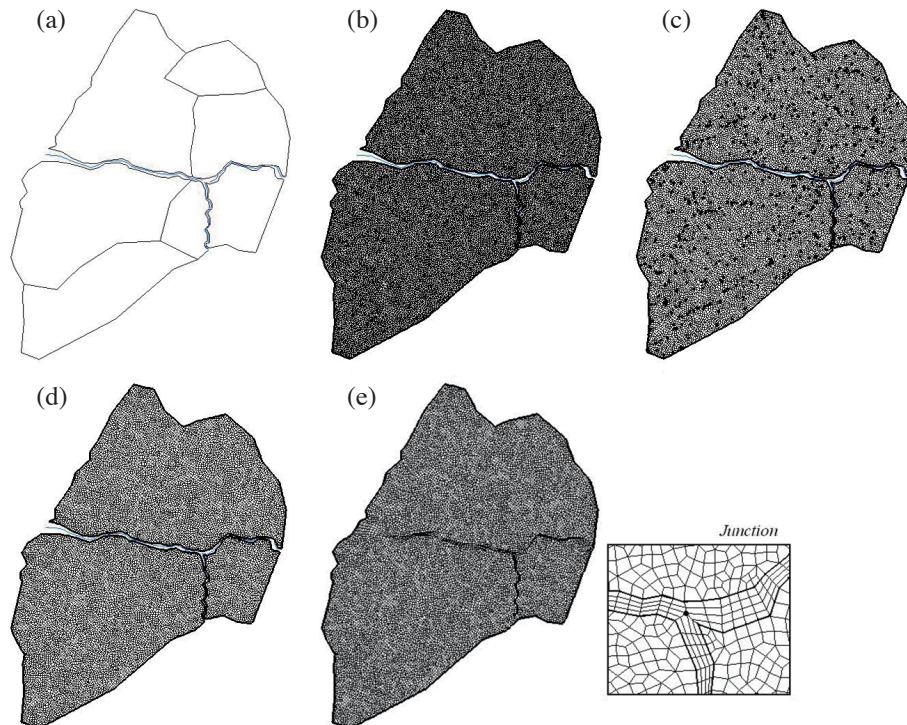


Fig. 11. Quadrilateral mesh generation for the Choshui River of Taiwan (a) input boundaries, (b) Delaunay triangular mesh, (c) resulting mesh after initial combination, (d) quadrilateral mesh after treating residual triangles and quality improvement, (e) quadrilateral mesh after filling river and junction regions.

be meshed separately and the points on the intersecting boundaries between adjacent domains must be conformal. As boundary points are generated, each loop is adjusted to have an even number of edges and the overlapped edges have the lower priority level for adding new points. Figure 10b provides the triangular mesh generated using the Delaunay triangulation method, containing 10186 nodes and 20071 triangles. Appropriate pairs of adjacent triangles are combined into quadrilaterals based on the conversion patterns in Fig. 5. Figure 10c shows the resulting mesh after initial combination. Each sub-domain contains an even number of residual triangles. In this case all of the residual triangles can be moved and recombined into quadrilaterals. Figure 10d provides the final mesh after treating residual triangles and quality improvement, consisting of 10222 nodes and 9823 quadrilaterals in all. This mesh can accurately describe the boundary features of the input geometry and achieve the full conversion of quadrilaterals. It should be noted that each of the nodes on the river reaches is associated with two over-land nodes.

Figure 11 provides a practical quadrilateral meshing problem for handling realistic geology, hydrology, and water resources for the Choshui River of Taiwan. The boundary map of the analyzed district is captured from satellite images (Fig. 11a), containing seven meshing regions with three river reaches intersecting at a junction. While generating boundary points each loop is adjusted to have an even number of edges. Figure 11b shows the triangular mesh produced by the Delaunay triangulation method based on the specified point spacing, containing a total of 14109 nodes and 27480 triangles. Note that each mesh domain consists of an even number of triangles. The quadrilateral mesh is created based on this triangular mesh by combining appropriate pairs of connected triangles in each domain (Fig. 11c). In each domain an even number of residual triangles remain. Figure 11d shows the final quadrilateral mesh after residual triangle treatment and quality improvement, consisting of 14133 nodes and 13430 quadrilaterals in all. All of the elements in the finally obtained mesh are quadrilaterals without any residual triangles left. If additional quadrilateral meshes are requested to be generated in the river and junction area, the resulting mesh as shown in Fig. 11e is achieved, where the amplified view shows the local mesh in the vicinity of the junction where three river reaches meet. This mesh includes 14283 nodes and 14070 elements, with all of the blank areas completely covered by quadrilaterals.

5. CONCLUSIONS

This paper developed an automatic mesh generation system for numerical simulations in the fields of geology, hydrology, and water resources using C++ and Fortran programming. The developed mesh generator serves as a preprocessor for finite element models in solving two-dimensional subsurface flow and water transport problems.

It is designed to generate three-point triangular or four-point quadrilateral elements for two-dimensional domains.

The key techniques in the meshing procedures were studied in detail, such as the generation of boundary points, the insertion of new points, combination of triangles and the treatment of residual triangles, etc.

- (1) Five combination patterns aimed at three cases were proposed to convert a triangular mesh into a quadrilateral mesh. The application of these patterns realized that there were as many triangles combined into quadrilaterals during the initial conversion. The quality of the generated quadrilaterals was ensured well.
- (2) The treatment strategies for residual triangles for arbitrary types of geometric models were researched and presented. Intensive research was conducted on the effect for boundary edges surrounding each sub-domain on the subsequent meshing procedures.
- (3) For the geometry with several sub-domains, an adjustment method was proposed to control the number of boundary edges on each loop to ensure that each sub-domain contained an even number of Delaunay triangles. This method was able to implement the full conversion from triangles into quadrilaterals.
- (4) The number of boundary edges on each loop was adjusted by inserting new points. The adjustment rules for special loops and three cases of common loops were presented. The most appropriate edge for new point insertion was determined by the relative point spacing error. This treatment approach could ensure reasonable density distribution for mesh elements and nodes.
- (5) The two-dimensional triangular and quadrilateral mesh can be stretched along the z-coordinate axis into a three-dimensional triangular prism and hexahedral mesh. The proposed program was also designed to accommodate the need for aquifers with heterogeneous systems by identifying the type of material in each layer of elements.

Acknowledgements This research is supported, in part, by National Science Council under contract Nos. NSC 101-2116-M-008-002 and NSC 101-2811-M-008-067 with National Central University and, in part, by National Central University under the 5/500 Fund.

REFERENCES

- Blacker, T. D. and M. B. Stephenson, 1991: Paving: A new approach to automated quadrilateral mesh generation. *Int. J. Numer. Methods Eng.*, **32**, 811-847, doi: 10.1002/nme.1620320410. [[Link](#)]
- Borouchaki, H., P. J. Frey, and P. L. George, 1996: Unstructured triangular-quadrilateral mesh generation. Application to surface meshing. Proceedings of 5th International Meshing Roundtable, Sandia National

- Laboratories, 229-242.
- Bowyer, A., 1981: Computing Dirichlet tessellations. *Comput. J.*, **24**, 162-166, doi: 10.1093/comjnl/24.2.162. [[Link](#)]
- Cox, M. G., 1972: The numerical evaluation of B -splines. *IMA J. Appl. Math.*, **10**, 134-149, doi: 10.1093/imamat/10.2.134. [[Link](#)]
- De Boor, C., 1972: On calculating with B -splines. *J. Approx. Theory*, **6**, 50-62, doi: 10.1016/0021-9045(72)90080-9. [[Link](#)]
- Ebeida, M. S., S. A. Mitchell, A. A. Davidson, A. Patney, P. M. Knupp, and J. D. Owens, 2011: Efficient and good Delaunay meshes from random points. *Comput. Aided Design*, **43**, 1506-1515, doi: 10.1016/j.cad.2011.08.012. [[Link](#)]
- Garimella, R. V., M. J. Shashkov, and P. M. Knupp, 2004: Triangular and quadrilateral surface mesh quality optimization using local parametrization. *Comput. Meth. Appl. Mech. Eng.*, **193**, 913-928, doi: 10.1016/j.cma.2003.08.004. [[Link](#)]
- Highway, E. A., 1983: A mesh generator for automatically subdividing irregular polygons into quadrilaterals. *IEEE Trans. Magn.*, **19**, 2535-2538, doi: 10.1109/TMAG.1983.1062810. [[Link](#)]
- Ho-Le, K., 1988: Finite element mesh generation methods: A review and classification. *Comput. Aided Design*, **20**, 27-38, doi: 10.1016/0010-4485(88)90138-8. [[Link](#)]
- Ito, Y., 2013: Challenges in unstructured mesh generation for practical and efficient computational fluid dynamics simulations. *Comput. Fluids*, **85**, 47-52, doi: 10.1016/j.compfluid.2012.09.031. [[Link](#)]
- Itoh, T. and K. Shimada, 2002: Automatic conversion of triangular meshes into quadrilateral meshes with directionality. *Int. J. CAD/CAM*, **1**, 11-21.
- Knupp, P. M., 2003: A method for hexahedral mesh shape optimization. *Int. J. Numer. Methods Eng.*, **58**, 319-332, doi: 10.1002/nme.768. [[Link](#)]
- Owen, S. J., M. L. Staten, S. A. Canann, and S. Saigal, 1998: Advancing front quadrilateral meshing using triangle transformations. Proceedings of 7th International Meshing Roundtable, 409-428.
- Owen, S. J., M. L. Staten, S. A. Canann, and S. Saigal, 1999: Q-morph: An indirect approach to advancing front quad meshing. *Int. J. Numer. Methods Eng.*, **44**, 1317-1340, doi: 10.1002/(SICI)1097-0207-(19990330)44:9<1317::AID-NME532>3.0.CO;2-N. [[Link](#)]
- Schoenberg, I. J., 1946: Contributions to the problem of approximation of equidistant data by analytic functions. Part A: On the problem of smoothing of graduation, a first class of analytic approximation. *Quart. Appl. Math.*, **4**, 45-99.
- Shimada, K. and T. Itoh, 1995: Automated conversion of 2D triangular mesh into quadrilateral mesh. In: Atluri, S. N., G. Yagawa, and T. Cruse (Eds.), *Computational Mechanics '95*, Springer Berlin Heidelberg, 350-355, doi: 10.1007/978-3-642-79654-8_58. [[Link](#)]
- Sibson, R., 1978: Locally equiangular triangulations. *Comput. J.*, **21**, 243-245, doi: 10.1093/comjnl/21.3.243. [[Link](#)]
- Sun, L., G. T. Yeh, F. P. Lin, and G. Zhao, 2015: Automatic quadrilateral mesh generation and quality improvement techniques for an improved combination method. *Comput. Geosci.*, **19**, 371-388, doi: 10.1007/s10596-015-9473-z. [[Link](#)]
- Tam, T. K. H. and C. G. Armstrong, 1991: 2D finite element mesh generation by medial axis subdivision. *Adv. Eng. Softw. Workstations*, **13**, 313-324, doi: 10.1016/0961-3552(91)90035-3. [[Link](#)]
- Watson, D. F., 1981: Computing the n -dimensional Delaunay tessellation with application to Voronoi polytopes. *Comput. J.*, **24**, 167-172, doi: 10.1093/comjnl/24.2.167. [[Link](#)]
- Yerry, M. A. and M. S. Shephard, 1983: A modified quadtree approach to finite element mesh generation. *IEEE Comput. Graph. Appl.*, **3**, 39-46, doi: 10.1109/MCG.1983.262997. [[Link](#)]
- Žalik, B., 2005: An efficient sweep-line Delaunay triangulation algorithm. *Comput. Aided Design*, **37**, 1027-1038, doi: 10.1016/j.cad.2004.10.004. [[Link](#)]