들로네이-보로노이 삼각요소생성기법에 있어서 인접성유지와 요소재생성과정을 위한 단순알고리즘 연구

A SIMPLE ALGORITHM FOR MAINTAINING ADJACENCY AND REMESHING PROCESS IN DELAUNAY-VORONOII TRIANGULATION

송 영 준*
Song, Young Joon

요 약

유한요소생성기법중 삼각요소를 생성하는 들로네이-보로노이 기법은 반복되는 국지적 요소재편을 통하여 요소망을 완성하는 기법으로 적응유한요소법의 적용에 이점이 되고 있다. 이 방법의 요체는 재편대상이 되는 요소군의 형성과 이를 대체하는 요소생산의 과정이다. 이를 간편하게 해결하는 방법으로 요소의 인접성을 나타내는 행렬을 새로이 도입하고 이에 따르는 단순 알고리즘을 제시하여 일반 PC급 이용자들도 본 요소생성기법을 이용한 적응유한요소해석의 실질적인 적용가능성을 제고하였다.

Abstract

One of the characterisrics of Delaunay-Voronoii methods of mesh generation is local remeshing ability in comparison with other methods, which is very useful in adaptive finite lement applications. Main part of the process is to construct remeshing element group out of the whole elements and to remesh it. Adjacent element array, accompanied with an additional algorithm of several lines, is introduced to make the process simple so that implementation of the concept is possible at the level of general PC users.

1. INTRODUCTION

In most cases of mesh generation schemes like Algebraic Serendipity methods[1,2], Elliptic differential equation methods[3-6], Algebraic integer net methods[7-10] and Advancing front methods[11-15], there is one common observation that once an element is generated, its connectivity and the coordinates of nodes almost become final data. Also the numbers of

* Dept. of Graphics, Korea Military Academy

이 논문에 대한 토론을 1994년 3월 30일까지 본학회에 보내주시면 1994년 9월호에 그 결과를 게재하겠습니다.

nodes and elements are determined by users a priori except for the cases of Advancing front methods. At the very end of those mesh generations smoothing scheme is carried out to regularize the shape of each individual element over the whole domain by adjusting coordinates of the nodes within limited range. But the connectivity itself never changes. In the case of Delaunay-Voronoii methods[16-19], mesh generation is performed locally and iteratively until reaching the final grids. Here the mesh generation is involved in two parts, the first part generating a coarse initial triangulation and the second one refining the triangulation as necessary to ensure acceptable solutions in finite element analysis. Beginning from the initial triangulation which usually do not satisfy certain criteria for shape and size of elements, a new node is inserted to the existing net of triangles and some of elements are regenerated by taking completely different connectivities so as to improve shape regularity locally in new triangulation. Thus a newly improved set of elements is generated with an additional node after eliminating a certain part of old meshes. As the process is repeated, the numbers of nodes and elements are increasing and shape regularities are improving locally as well as globally. The iterative process is terminated when the required conditions for size and shape are met for the global triangulation. Theoretically the method provides optimal triangulation from a given set of nodal points. Though Delaunay-Voronoii methods deal with simplex elements only, the beauty of the method lies on its abilities of local remeshing and graded mesh generation. These features are very essential parts of adaptive finite element applications. In practical implementation of code, in order to select candidate region of remeshing it is desirable to establish heap list[20] of element identifications according to severity of irregularity. Also when a new node is inserted to the existing meshes it is necessary to prepare effective way of finding neighboring element group around the node(e.g. alternate digital tree algorithms[21]), which is subject to mesh revision.

In this study a new algorithm is introduced to construct the list of neighboring elements around an arbitrary element instantaneously by maintaining adjacent element array and to provide a simple method of constructing block contour of subregion under revision, of which procedures take most of CPU time in generation process otherwise. The discussion is limited to the generation of two-dimensional simplex elements(triangular elements), which is applicable in 3-dimensional space as well.

2. DELAUNAY TRIANGULATION

There are many ways of constructing triangular meshes from a given set of points. Among them Delaunay triangulation provides best shape regularity, and it is one of the more important reasons than its local remeshing ability why Delaunay triagulation is favored in mesh generation for adaptive finite element applications.

One recalls Delaunay triangulation as follow. Let a set of points $\{P_n : n=1, 2, \cdots, N\}$ be N distinct points on two-dimensional Euclidean space. Voronoii polygon D_i with respect to point P_i is defined as follow.

$$D_i=\{x : \|x-P_i\|<\|x-P_j\|, \forall j\neq i\} \qquad (1)$$

where $\|\cdot\|$ is two-dimensional Euclidean norm. In other words Voronoii polygon D_i corresponding point P_i is intersection of half planes between points P_i and P_j for all j, j being other than i.

The union D_N of Voronoii polygons defined by equation (1) is called Dirichlet tessellation[22](dotted lines in Fig.1).

$$D_N = \bigcup_{i=1}^{N} D_i \qquad (2)$$

In Dirichlet tessellation each interior vertex is shared by three Voronoii polygons. Then a triangle is constructed by connecting lines between the three points corresponding those polygons.

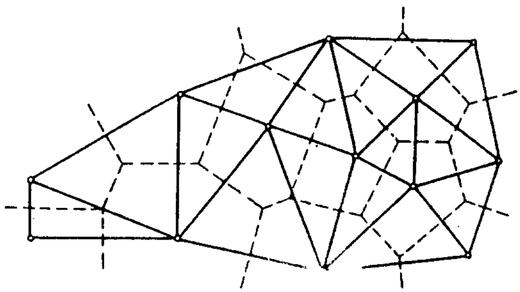

Fig.1 Voronoii polygons and Delaunay triangulation

Collection of such triangles is called Delaunay triangulation(solid lines in Fig.1). In n-dimensional space, Delaunay triagulation produces n-dimensional simplex which has n+1 vertices. One of the properties of Delaunay triangulation is that hypersphere of a simplex does not contain any points other than its vertices. This feature tends to improve shape regularity of the simplex. One notes that among triangles of equal area, the diameter of a circumcircle of an equilateral triangle is the smallest of all therefore allowing least chances to contain points exterior of the triangle, which eventually has effects on enhancing shape regularity of triangles.

3. MESH GENERATION ALGORITHM

Although the definition of Delaunay triagulation is simple but generating a set of triangles which satisfy Delaunay characteristics requires careful and judicious preparation. The procedure of utilizing Delaunay properties in mesh generation is involved in three major steps.

i) Generation of boundary(initial) nodes.

ii) Generation of boundary(initial) triangulation with boundary nodes.

iii) Development to the final triangulation by iterative refinement of local remeshing with improved regularity.

Considering the above procedures, the regularity of the initial triangulation is not so important. Main concerns in the steps i) and ii) are put on the completeness of domain coverage by the initial triangulation. The regularity is improved as the triangulation is refined in the step iii). Each iteration in the step iii) performs in two parts, creating and inserting a new node to the existing meshes, and regenerating triangles around the node in such a way to meet Delaunay properties.

One first investigates chatacteristics of the locations of prospective nodes. Let a set of points $\{x_1, x_2, \cdots, x_n\}$ constitute Delaunay trianglation as shown in Fig.2.

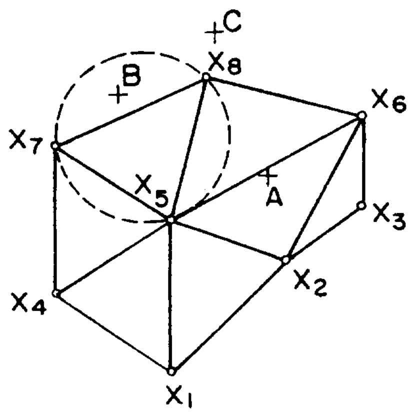

Fig.2 Candidate locations for prospective nodes

The characteristics are summarized as follow[23).

Node A : located interior of the triangulation.

Node B : located exterior of the triangulation but contained in one of circumcirles of the triangles.

Node C : located exterior of the union of their circumcircles.

It is not difficult to reconstruct triangles satisfying Delaunay charactersсs for nodes B and C but the nodes are located exterior of the existing meshes. When these cases are permitted in the course of creating a prospestive node, sometimes a new node might be located outside of problem domain, which is not acceptable. Hence one's interest is limited to the case of node A for prospective location of a new node.

In Fig.2 node A is found to be contained in circumcirlces of three triangles simutaneously. Then those three triangles are selected to form a block for remeshing. On the block of selected triangles interior sides are eliminated, then node A and boundary contour sides of the block are left so as to generate a set of new triangles as shown in Fig.3, which satisfies Delaunay properties.

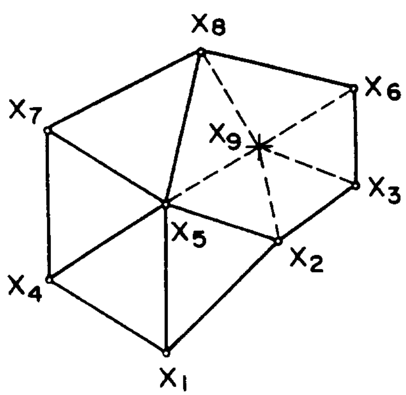

Fig.3 Local revision of elements

To summarize the process, let $x_1, x_2, \cdots, x_N$ be a set of N nodes and let T_i be triangulation out of subset $\{x_k\}$ up to k-th node, $k<N$. Initial triangulation is T_{i-k}. After inserting (K+1)-th node, let T_{i+1} be triangulation out of subset $\{x_{k+1}\}$. Let define block B_{k+1} as the union of triangles whose circumcircles contain node x_{k+1} and let denote M sequential boundary contour line segments of block B_{k+1} by S_j. Then new triangulation T_{i+1} is generated by the following logic[23).

$$T_{i+1}=(T_i-B_{k+1}) \cup \{x_{k+1}, S_j\}_j$$
$$\{x_{k+1}, S_j\}_j \equiv \{[x_{k+1}, S_j], 1 \le j \le M\} \qquad (3)$$

Here a triangle made of node x_{k+1} and line segment S_j is denoted by $[x_{k+1}, S_j]$. The block B_{k+1} is called remeshing block or revision block due to insertion of node x_{k+1}. Generation logic (3) will be used in constructing initial triangulation as well as refining triangulations later.

4. CONSTRUCTION OF BOUNDARY TRIANGULATION

After geometric modeling of the problem domain, boundary nodes are produced by discretizing the geometric contour lines, with graded spacing if necessary. Let the contour lines are divided by nodes $\{x_1, x_2, \cdots, x_P\}$ at desired spacing as shown in Fig.4. These are called boundary(initial) nodes and used in constructing boundary(initial) triangulation which should cover the whole problem domain completely.

As one recalls from section 3 that the logic (3) is stated under the implication of a new node being located interior of the existing meshes.

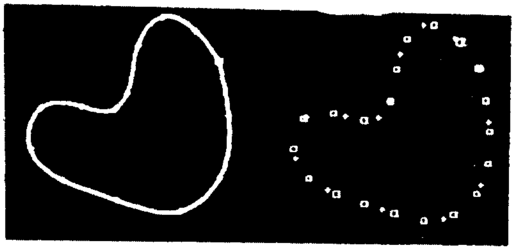

Fig.4 Boundary nodes

In order to use logic (3) in constructing the boundary triangulation, it is necessary to introduce a virtual domain to make nodes $\{x_1, x_2, \cdots, x_P\}$ interior nodes. The simplest way of doing so is to construct a rectangle big enough to contain the problem domain by introducing fictitious nodes[23,24]. Let points X_1, X_2, X_3, X_4 make a rectangle R big enough to contain points $x_1, x_2, \cdots, x_P$ which delineate a given domain Ω, and let T_4 be consisted of two triangles. (X_1, X_2, X_3) and (X_1, X_3, X_4). Taking T_4 as the initial triangulation in the course of constructing boundary triangulation, node x_1 is inserted into T_4 and triangulation T_5 is obtained according to the logic (3). When the process has been repeated for x_i, i from 1 to P, the final triangulation T_{4+P} is obtained as shwon in Fig.5. By eliminating triangles which use any of points X_1, X_2, X_3 and X_4 as their vertices, the boundary Delaunay triangulation is done. For the case of a nonconvex domain triangles which lie exterior of the domain should also be removed, hence the boundary triangulation only covers interior of the original domain but any part of the exterior as seen in Fig.6. Then the boundary elements are ready to be used as the initial elements for iterative refinement. Each refinement can be accomplished by creating an interior node at approriate location and regenerating elements around it accordingly.

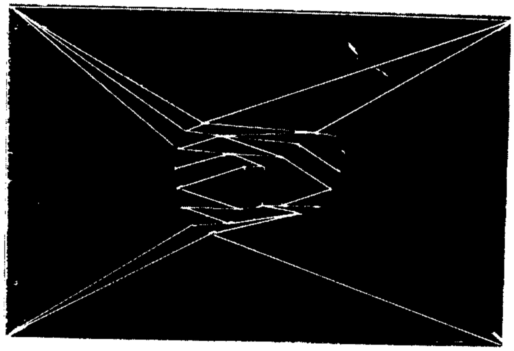

Fig.5 Boundary triangulation with fictitious nodes

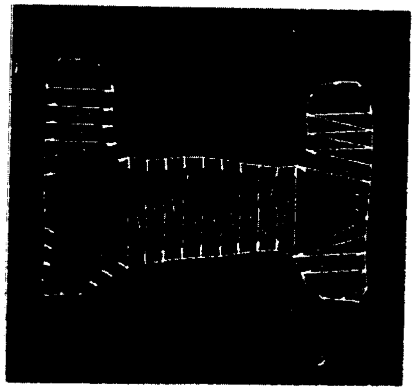

Fig.6 Initial triangulation

It is noted that in the process of constructing the boundary triangulation inserting nodes x_i to T_4 are prepared in advance and those are taken from the boundary nodes as i=1 to P. During the iterative refinement from the boundary triangulation through the final one, the inserting nodes should be created at every iteration. Density of elements in the final triangulation depends on spacings of the boundary(initial) nodes x_i.

5. REFINEMENT OF TRIANGULATION

Basic criteria for the necessity of mesh regeneration at a certain location are concerned with size and shape regularity of elements in the region. If some of triangles are too big in size or have obtuse angles, they are subject to mesh revision by inserting an additional node. Refinement of triangulation begins with searching a triangle which is too big and irregular than expected. After selecting a triangle, a prospective node is inserted in a appropriate way. This temporary node is not accepted as a regular node unless it is guaranteed that all of the regenerated triangles due to node insertion satisfy the size and shape requirements for legitimate elements. In the early part of iterations the size constraint plays a major role, and the shape constraint is also equally significant in the later part. The tests of the conditions for creating an interior node to an individual triangle are called location test, spacing test and shape test.

5.1 Location Test

Once the initial triangulation is obtained, one needs to find a triangle to place a new node inside and regenerates triangles around it. The process is repeated until reaching final triangulation. Candidate triangles are screened by checking their sizes, e.g. the diameters of their circumcirlces. When the size of a triangle far exceeds the recommended size, an appropriate location is sought to position a new node. Areacentre and circumentre of a triangle are first hand candidates for the location. The natural choice is circumentre since it is located at equidistance from each vertex. But it should be checked whether the circumcentre lies interior of the triangle or not since generation logic (3) is based on the assumption that a new node x_{k+1} is an interior point. Hence a triangle whose circumcentre lies exterior of itself is not considered as a candidate element for node insertion. This is called location test. Meanwhile since incentre of inscribed circle always lies inside of a triangle, the use of weighted average of coordinates of circumcentre and incentre for prospective location is suggested to guarantee a new node remain inside for any triangles[25]. Let R_C. R_I be circumradius and inradius and let x_C and x_I be coordinates of circumcircle and incircle of a triangle respectively as shwon in Fig.7.

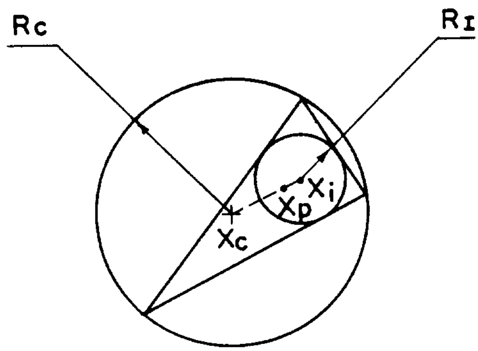

Fig.7 Positioning of a new node

In that regard the coordinates x_j of a prospective node j by weighted average scheme and weighting ratio ω are given as follow.

$$x_j=(1-\omega)\ x_I+\omega\ x_C$$
$$\omega=2\ R_I\ /R_C \qquad (4)$$

Small value of factor $R_I\ /R_C$ implies strong irregularity. Range of the factor $R_I\ /R_C$ is between 0 and 1/2, 1/2 being the case of an equilateral triangle. The factor is naturalized in order to be used as weighting ratio ω. As irregularity becomes strong, so does dependency of x_j on incentre x_I in equation (4). It provides well balanced value of weighting ratio in calculating the coordinates of a prospective node insuring inside location. For an equilateral triangle x_I, x_C and x_j coincide.

5.2 Spacing Test

Most of triangles in the initial triangulation are slender and much larger than the size limit defined implicitly by spacings of boundary nodes. In general the size of the regenerated triangles due to node insertion becomes smaller than that of the old triangles. In some triangles, node insertion does shorten the length of lateral sides well below the size constraints. Then the node insertion is not accepted. It is called spacing test. Here one needs a certain measure to indicate size or density of elements anywhere in the domain. For the purpose spacing function is introduced. When a spacing function describing element density is defined explicitly over the whole domain, locating the position of a new node is simple matter. When it is not defined, the only control over element density is discretization of boundary contour lines in constructing the initial nodes, which will provide inplicit spacing function.

Let N be the number of sides connected to a node P and let their lengths be denoted by E_i, $i=1,\cdots,N$. Then an implicit spacing function d_P of the node P under current mesh is defined as follow.

$$d_P=\min\{E_i\}, \qquad i=1,\cdots,N \tag{5}$$

When a prospective node j is inserted into a triangle consisted of nodes P,Q,R, the value of spacing function at the node j is interpolated using spacing function value d_k of vertex nodes k, k=P,Q,R, and area coordinates of node j on the triangle. Let x_P, x_Q, x_R and x_j be coordinates of nodes P,Q,R and j respectively and let area coordinates of node j be denoted by $L_k(x_P, x_Q, x_R, x_j)$. Then the interpolated spacing function D_j at node j is defined as follow.

$$D_j=d_k\, L_k(x_P, x_R, x_j),\ k=P,Q,R \tag{6}$$

Here summation conventions are assumed over k, and d_k are obtained by the equation (5) from back ground elements, not recognizing node j as a legitimate node yet. Then spacing test proceeds as follow.

1) calculate the coordinates x_j of a prospective node j by the equation (4).
2) calculate interpolated space function value D_j at node j by the equation (6).
3) calculate space function value d_j of node j by the equation (5) where sides E_i are the lines made by connecting x_j with x_P, x_Q and x_R respectively.
4) admit node j as a regular node if the following condition is satisfied.

$$d_j \ge D_j \tag{7}$$

If d_j is smaller than D_j, node j will not be accepted as a regular node and mesh regeneration does not occur. When the condition (7) is forced too strict, resulting final triangulation has tendancy of having elements rather big than expected. Numerical experiments show that the following adjustment gives better results[25].

$$d_j \ge D_j/\sqrt{2} \tag{8}$$

5.3 Shape Test

The ideal shape of a triagular finite element is an equilateral triangle. As node insertion continues sides of triangles approach to equilateral ones while enhancing shape regularity. Exact equilateral triangles may never be obtained. But it does not cause any problem at all. As a matter of fact if graded elements are wanted exact quilateral triangles should be avoided at certain region of the domain. When the condition (8) is met for a prospective node, local revision of current triangulation around the node is one step closer. Before doing so, one examines the patterns of element revision by the logic (3) and these are shown in Fig.8.

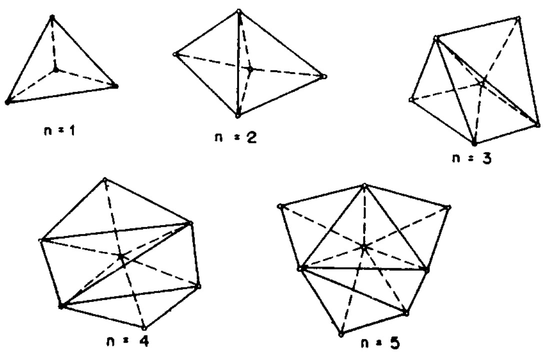

Fig.8 Patterns of element revision

Let n be the number of triangles in a remeshing block B_j as defined in the section 3 for a prospective node j. As shown in Fig.8, one notes that n+2 new triangles are generated after eliminating n triangles, which in turn leads to the fact that n+2 triangles share node j as a common vertex. In other words n+2 is the number of triangles connected to node j after local revision. Then average vertex angle of triangles at the node is 360°/(n+2).

Let shape indicator $\rho(j)$ of the inserted node j be defined as the number of triangles connected to node j, that is $\rho=n+2$. Since the ideal mesh of triangles is union of equilateral triangles, ideal value for ρ is 6 which gives average vertex angle of 60° same as the one of equilateral triangular meshes. At an erarly stage of iterative refinements the value of ρ well exceeds 6 because there should be many slender triangles in the initial traingulation which have so large a circumcircle that they would contain many points. As Delaunay triangulation is repeated values of ρ over the whole nodes tend to approach 6. In practice without imposing upper limit for the shape indicator ρ, max{ρ} seldom exceeds 8 at the end of refinements. In fact for severely graded meshes $\rho=6$ can not be forced strongly over the whole domain. Here one sees that lower limit of ρ is rather important than upper limit is. Indeed for the cases of ρ less than 5 some of regenerated triangles have deteriorated shape regularity as seen in Fig.8. Those situations are likely happened in a later stage of the iterations. In practice sole imposition of lower limit gives accetable results. Evaluation of ρ is possible only after constructing remeshing block B_k for given prospective node k. If $\rho(k)$ is smaller than a lower limit, node k will not be accepted as a regular node. Otherwise local revision will be carried on and around the triangle containing node k. This is called shape test.

5.4 Adjacent Element Array and Construction of Remeshing Block

Construction of remeshing block is one of the most important module in the process of Delaunay triangulation. It is not only necessary to evaluate the value of shape indicator ρ for the shape test but also for actual element regeneration itself. In the process a new node is inserted into an eligible triangle, then part of neighboring triangles are eliminated and new triangles are generated in that region.

Those changes are repeatedly occurring at every new insertion of node. If one has to screen all the triangles to find members of remeshing block for a newly inserted node, it might cost unacceptable computational expenses especially as the number of background elements increases. Since the process uses iterative methods, the importance is doubled. When one has informations of every element on its proximity to a given element, it will be very helpful. For information on the proximity among nodes binary tree data structure[21] may be used to save the coordinates of nodes, but it is not enough and those should be converted to proximity among elements. Hence very flexible and versatile data management is required. To do so adjacent element array NEXT is intrduced in this study along with conventional element connectivity array KLM. Array value of NEXT is defined as follow.

NEXT(i,j) ≡ adjacent element number facing i-th side of element(triangle) j.

For the background elements given in Fig.9, element connectivity array KLM and adjacent element array NEXT are given in Table 1. One notes that array values of the sides facing outward of the boundary are assigned null values.

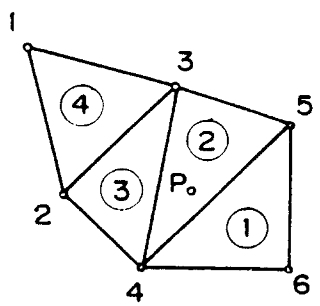

Fig.9 Insertion of a new node P

Table 1. Connectivity and adjacency of Fig.9

KLM

Element ID	Connectivity		
1	4	6	5
2	4	5	3
3	2	4	3
4	1	2	3

NEXT

Element ID	Connectivity		
1	0	0	2
2	1	0	3
3	0	2	4
4	0	3	0

Intrduction of the adjacent element array is for dual purposes.

i) Control of candidate triangles in searching members of remeshing block.

ii) Construction of boundary segment chain of remeshing block

Let the triangle in which a prospective node is created be called core triangle or core element. In order to construct remeshing block due to a newly inserted node each element listed in the adjacent element array NEXT of the core element is examined if its circumcircle contains the inserted node. If so, the element is listed as a member of remeshing block. Such a screening process is carried out levelwise. When no more member is found at a certain level, screening process ends and construction of remeshing block is completed. Shape indicator ρ is then evaluated adding the number of members in remeshing block by 2.

Let a triangular element T be the core triangle of a prospective node, then the number of candidate elements for screening at level 1 is 1 and its member is T. Let the number of candidate elements of h-th level be NELh as shown in Fig.10, then NELh are counted as follow.

NEL2=4, Members={T, T_k=NEXT(k,T}, k=1,2,3

* NEL3=10, Members={T, T_k, T_{km}=NEXT(m, T_k)}, m=1,2,3

* NEL3 = 19, Members = {T, T_k, T_{km}, T_{kmn} = NEXT(n, T_{km})}, n = 1,2,3

* (double counted elements are excluded)

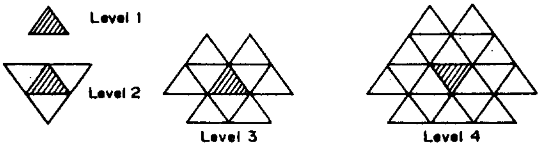

Fig.10 Level of adjacency

In earlier triangulations screening level may need to go beyond level 5, but it does not cause any concern since the number of back ground elements is relatively small. As the refinement continues the number of elements is increasing, shape regularity enhancing, and the number of big elements decreasing. Then level 3 or 4 is enough for screening. Thus controlling candidate elements by the array NEXT is very effective method compare to screening all the elements especially when there are hundreds and thousands of elements.

In order to use generation logic (3) it is necessary to construct chain of boundary contour line segments S_j out of remeshing block B_{k+1}, Again adjacent element array NEXT is used effeciently. Let S be an one-dimensional identification array of lateral sides which lie on the boundary of the remeshing block B. Side identification is given by the starting node number of each side of triangles in the sense of counterclockwise. Then construction of array S is equivalent to that of S_j in (3).Now one of triangles, say element IB, is picked from block B and each of its sides is checked if the respective adjacent triangle, say IC = NEXT(j, IB), is a member of block B or not. If not, j-th side of element IB becomes a member of array S and its starting node number is recorded and the inspection is carried on (j+1)-th side. If it is, the preceeding process is moved to the adjacent element IC and the inspection is carried on each of its sides again. The process ends when it comes across the starting element IB. The algorithm is summarized as follow.

```
1. (Initialize) Pick IB from B
   IBO = IB : j = 1
2. IN = NEXT(j, IB)
   if(IN ∈ B) then
     goto 3
   else
     S ∋ KLM(j, IB)
     j = j + 1
     if(j > 3) j = j - 3
     goto 2
   endif
3. do k = 1, 3
     if(NEXT(k, IN) = IB) then
       j = k + 1
       if(j > 3) j = j - 3
       if(j = 1 and IB = IBO) stop
       goto 2
     endif
   enddo
```

The procedure of the algorithm is shown graphically in Fig.11. This simple algorithm excludes all the interior sides of the block B and saves the boundary sides in S sequentially along the arrow-headed path without any kind of compli-

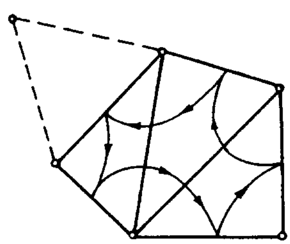

Fig.11 Constructing boundary chain of remeshing block

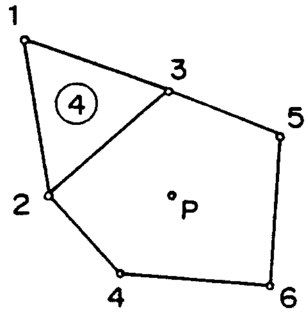

Fig.12 Boundary of remeshing block

cated calculation to find such as corner angles at nodes. As the result of the operation remeshing block B is shown in Fig.12.

5.5 Generation of Triangles and Element Numbering

Consider the insertion of a new node P in Fig. 10 again. Node P and array S defined in section 5.4 correspond with x_{k+1} and S_j in generation logic (3) respectively, i.e. $x_{k+1}=x_P$ and $\{S_j\}=S$. New set of triangular elements are generated by $\{x_{k+1}, S_j\}_j$ counterclockwise with their sides radiating from node P. Regenerated elements are temporarily assigned element identification numbers in such a way that the first of regenerated element bears the element number following immediately after the last element number of trianguraltion T_i and the second one incremented by 1 and so on. Thus new elements always take the last portion of sequential numbering. Then current numbering system represents triangulation $T_i\{x_{k+1}, S_j\}_j$ in which no elements are eliminated yet and new elements are overlaped over the existing ones in remeshing block B. The old elements in remeshing block B are now eliminated and the remaining ones are stack-downed without allowing any vacancy in sequential element numbering. Upon dliminating all of the old elements in block B_{k+1}, element connectivity array KLM and adjacentr element array NEXT are updated accordingly. Generation of new triangulation $T_{i+1}=(T_i-B_{k+1})\cup\{x_{k+1}, S_j\}_j$ is then finalized. The result of the example problem of Fig.9 is shown in Fig.13.

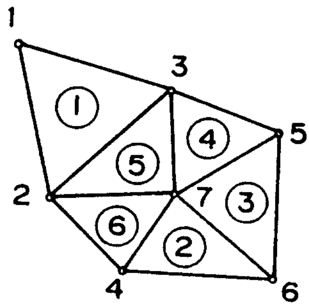

Fig.13 Regeneration of elements of Fig.9

The present method of element numbering provides a convenient way of controlling the number of iterations to obtain the final triangulation. It gives informations where to start location test for next iteration as well as when to stop iteration and to finish refinements.

Let E_{k+1} be the lowest element number among the eliminated ones, i.e.

$$E_{k+1}=\min\{\text{the element numbers in block } B_{k+1} \text{ of } T_i\}$$

In new triangulation T_{i+1} the number $E_{k+1}$is assigned to the element whose element number is the lowest among the higher than E_{k+1} in the remaining elements of triangulation $T_i\cup\{x_{k+1}, S_j\}_j$. Therefore elements which have not taken location test yet, including regenerated elements, always bear the element numbers equal to or higher than E_{k+1}. In the next refinement(for triangulation T_{i+2}) the element bearing the number E_{k+1} in T_{i+1} is the first one to take location test.

As the number E_{k+1} approaches to the highest element number of current traingulation, one can see that refinement is near to end. When no element is picked for location test, spacing test and shape test from element E_{k+1} through the last element(the element with the highest element number), it is the end of refinement and the final triangulation is obtained. Fig.14 shows a sequence of the procedures described in section 5. Thus the present numbering system provides very practical means of controlling not only where to begin the next iteration but also when to end refinements.

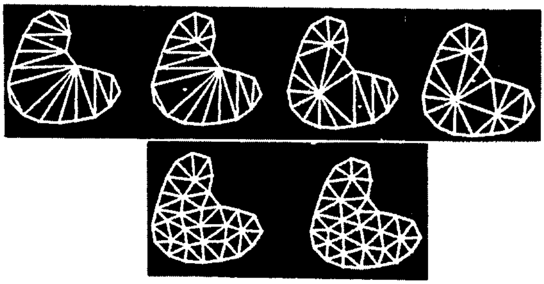

Fig.14 Refinement steps to the final triangulation

6. REGULARIZATION

The last step of triangulation is regularization of global meshes. In each revision of triangles shape regularity is honoured locally as much as possible due to properties of Delaunay triangulation. But for the whole domain it is necessary to enhance global smoothness. Especially when element generation is done locally like in advancing front method and in Delaunay triangulation method, global regularization is unavoidable. For the purpose concept of barycenter is used at each node finding its new position at areacenter of a polygon which surrounds the node. When relocation of the nodes has been done by the method at each and every node over the whole domain iteratively, it enhances regularity of individual element as well as global meshes. Relocation of individual node is done as follow.

Let denote the nodes connected to a given node P on triangular meshes of triangles by P_i, $i=1,\cdots,N$, and the triangles which have node P as the common vertex by ${}_P\triangle^i$ respectively. Node P is relocated at barycenter of a polygon which is union of ${}_P\triangle^i$, and its new coordinstes x_P are calculated as follow.

$$x_P=\frac{1}{N}\frac{a_i}{A}\,{}_PX^i \qquad (9)$$

where summation convention is assumed over i, and

a_i=area of triangle ${}_P\triangle^i$

$A=\sum^{i} a_i,\ i=1,\cdots,N$

${}_PX^i$=coordinates of areacenter of triangle ${}_P\triangle^i$

When relaxation factor γ is introduced for rapid convergence, new position of node P at (m+1)th iteration is given as follow.

$$x_P^{(m+1)}=(1-\gamma)\,x_P^{(m)}+\frac{\gamma}{N}\frac{a_i^{(m)}}{A}\,{}_PX^i\,(m) \qquad (10)$$

Here summation convention is not applied on iteration indices. Iteration (10) is carried out on every node as node P runs through the whole nodes except those on the boundary. In general after 3 or 4 global iterations there are no noticable coordinate changes. The last picture of Fig.14 is the result of regularization. In the case of nonconvex domain special attention is called for x_P not to be displaced to the exterior of the problem domain.

7. CLOSING REMARKS

Delaunay-Voronoii methods of mesh generation have shown clear advantage of local remeshing abilities in comparison with other mesh generation methods, which are essential features in implementing adaptive finite element analysis. Major modules of the method in practical codes are subroutines for constructing revision blocks and regenerating elements on them, of which procedures take most of cpu time. The use of adjacent element array can provide data for adjacency of elements anywhere in the problem domain. Then grouping of elements for revision blocks comes by immediate result. Also with the use of the array, regeneration data on the block can be prepared by simple algorithm of less than 20 lines as seen in section 5.4. The idea can make the analysis environment much closer to the level of general personal computer users in implementing adaptive methods of their own in the field of finite element analysis.

REFERENCES

1. O.C. Zienkiewicz, The Finite Element Method, McGraw-Hill, 1977.
2. A. El-Zafrany and R.A. Cookson, 'Derivation of Lagrangian and Hermitian space function for quadrilateral elements', Int.J. Numer. Meth Eng., vol.23, 1986.
3. A.M. Winslow, 'Numerical solution of the quasi-linear Poisson equation in a nonuniform triangular mesh', J.Computer Physics, 2, 1967.
4. K.N. Gia and V. Gia, 'Numerical generation of system of curvilinear coordinates for turbine cascade flow analysis', Aero. Eng. and Appl. Mech. Rept., Univ. of Cincinnati, OH, 1975.
5. C.W. Mastin and J.F. Thompson, 'Transformation of three dimensional regions onto rectangular regions by elliptic systems', Numer. Math., 29, 1978.
6. J.F. Thompson, Elliptic Grid Generation, Numerical Grid Generation, 1982.
7. W.C. Thacker, A. Gonzalez and G.E. Putland, 'A method for automating the construction of irregular computational grid for storm surge forecast models', J. Computer Physics, 37, 1980.
8. M.A. Yerri and M.S. Shephard, 'A modified quadree approach to finite element mesh generation', IEEE CG-A, 1983.
9. J.H. Cheng, P.M. Finning, A.F. Hathway, A. Kela and W.J. Schoeder, Grid Generation in Computational Fluid Mechanics 188, Miami, 1988.
10. M.S. Shephard, F. Guerinoni, J.E. Flaherty, R.A. Ludwig and P.L. Baehmann, 'Adaptive solutions of the Euler equations using finite quadree and octree grids', Computers & Structures, 30, 1988.
11. A.J. George, 'Computer implementation of the finite element method', Stan-CS, Ph.D. Diss., 1971.
12. J. Carnet, Une méthods heuristiquede maillage dans le plan pour la mise en oeuvre des elements finis, These, Paris, 1978.
13. S.H. Lo, 'A new mesh generation scheme for arbitrary planar domain', In.,J. Numer. Meth Eng., vol.21, 1985.
14. J. Peraire, M. Vahdati, K. Morgan and O.C. Zienkiewicz, 'Adaptive remeshing for compressible flow computation', J. Computer Physics, 72, 1987.
15. J. Peraire, J. Peiro, L. Formaggia, K. Morgan and O.C. Zienkiewicz, 'Finite element Euler computations in three-dimensions', Int. J. Numer. Meth Eng., 26, 1988.
16. F. Hermeline, Une méthode automatique de maillage en dimension n, These, Université Paris 6, Paris, 1980.
17. W.H. Frey and J.C. Cavendish, Fast planar mesh generation using the Delaunay triangulation, General Motors Res.Pub. GM-4555, 1983.
18. Z.J. Cendes, D.N. Shenton and H.

Shahnasser, 'Magnetic field computations using Delaunay triangulations and complimentary finite element methods', IEEE Trans., Magnetics 21, 1985.
19. S.H. Lo, 'Delaunay triangulatin of non-convex planar domain', Int. J.Numer. Meth Eng., 28, 1989.
20. H. Jin and N.-E. Wiberg, 'Two-dimensional mesh generation, adaptive remeshing and refinement', Int. J.Numer. Meth Eng., vol.29, 1990.
21. J. Bonet and J. Peraire, 'An alternating digital tree algorithm for 3D geometric searching and intersection problems', Int. J.Numer. Methods Eng., vol.31, 1991.
22. P.J. Green and R. Sibson, 'Computing Dirichlet tessellation in the plane', Comp. J., 21, 1978.
23. P.L. George, Génération automatique de maillages, Masson, 1990.
24. D.F. Watson, 'Computing n-dimensional Delaunay tessllation with applications to Voronoii polytopes', Comp. J., 24, 1981.
25. W.H. Frey, 'Selective refinement : A new strategy for automatic node placement in graded triangular meshes', Int. J.Numer. Meth Eng., vol.24, 1987.

(接受 : 1993. 1. 26)