直線辺で構成される任意二次元領域へのデラウニー三角分割の適用

谷口健男*・太田　親**

二次元有限要素解析の対象は一般に幾何学的に複雑な形状をしているため，そのモデル生成には多大の労力が必要となる．本研究では任意二次元領域に対して有効な要素自動生成法を提案する．その基本はデラウニー三角分割であり，得られた三角形の形状は有限要素解析にとって好ましいものとなる．なお，入力データは境界上の節点だけでよく，また複数個の境界をもつ場合でも扱えるようになっている．

Keywords：*Delaunay triangulation, complex boundary geometry, 2D FEM*

1. はじめに

複雑な境界形状をもった大規模系の有限要素解析において入力データ量は膨大となり，もしこのデータ作成および入力を手動で行えば時間および経費の増加のみならず入力されたデータの信頼性の低下をも引き起こす．これらの弊害を避けるためにはプレプロセッサーの充実，なかでも要素分割の自動化は不可欠となる．

取り扱う系を二次元に限定するとその境界は線の集合で表現され，個々の線は2個の点とそれらを結ぶ線の種類で規定できる．いかに複雑な（言い換えれば入り組んだ）境界線であってもその上に節点を適切に置きさえすればそれら節点を連ねて境界線を近似して表現できる．このようにして設定された境界に関する情報を用いてその領域の認識と領域内部の要素分割の方法を考えると，四分木法とデラウニー三角分割法の2種類が考えられる[1)~3)]．

四分木法は領域すべてを包含し得るように正方形を設定し，ついでその境界を適切に表現し得るまで格子状に節点を配置し，最終的には三角形分割を行う方法である．この手法の利点は節点配置を格子状に設定することにより節点間の関係が明確に把握できること，一方欠点は境界を構成する線が入力されているもののそれを直接利用できず，境界線を明確に表現するために必要な格子状節点配置を繰り返さなければならない点である．

デラウニー法は節点が配置されていることが前提条件であり，この与えられた節点集合を対象として正三角形に最も近い三角形集合に領域を分割する．したがって，入力データは節点の情報（座標位置）だけである．しかしながら，この方法を用いると設定された節点の占有する凸部分空間がすべて三角形に分割され，境界線の局所的凹部あるいは領域内部に位置する穴（以降これを内部境界と呼ぶ）の内側といった要素分割の不必要な部分に対しても三角形要素を生成してしまう．また，実際の境界線が再現できず間違った境界線を生成してしまうといった事態も生じる危険性がある．

対象とする二次元領域の境界形状が非常に複雑な場合を考えるとそれを規定する線の設定自体困難となる場合が多いことより，ここでは入力データとして節点情報だけに限定し，隣合う2個の節点間を直線で結ぶことを考える．この入力データの条件に合った要素分割法としては上記考察よりデラウニー法が挙げられる．しかしながら，すでに述べたようにこの手法をそのままの形で複雑な境界形状をもった領域に適用することはできない．

一方，境界形状の複雑な系を対象とした要素分割法としてブロッキング法の有効性が一般に認められている[4)]．この方法では系を複数個の三角形あるいは四角形といった単純な形状の部分系に分割し，それぞれを独立に要素分割し，その結果を再結合することによって全体の有限要素モデルを作成する．しかしながら系の形状が複雑となればなるほど部分系の個数が増加し，その結果入力データ量の増加は避けられなくなる．したがって，ブロッキング法のプロセスにおいて要素分割のみならずその前段階のブロック分割も含めて自動化できれば複雑な境界形状をもつ系に対して有効な要素分割法となることが期待できる．

本研究においては直線境界で囲まれた任意二次元領域に対する要素自動分割法を提案する．その手法はデラウニー法をもとにしたものであり，まず対象領域の境界線上に認定された節点を用いて領域を粗い三角形集合に分割し，ついで内部に接点を配置し三角形要素に細分割を行うという2段階よりなる．したがって，全体の流れはブロッキング法のそれに一致する．一般にブロッキング

* 正会員　工博　岡山大学助教授　工学部共通講座
（〒700　岡山市津島中3-1-1）
** 学生会員　岡山大学大学院生

法の利点の一つとして，幾何学的に複雑な系の扱いが容易となること以外に，より柔軟な要素分割が可能となること，複合構造系といった材料定数が異なる系の要素分割が容易となることが挙げられる．以上のブロッキング法の利点を検証するために，ここではこの汎用的手法を利用して，浅水長波流れ解析用と一般二次元構造解析用の2種類の要素自動分割法を提案する．

2.　二次元非凸領域の三角分割

（1）　境界の認識

デラウニー三角分割は二次元あるいは三次元空間内に任意に設定された節点群の占有する凸部分空間を正三角形あるいは正四面体に最も近い要素群で埋める方法である．このデラウニー三角分割の相対問題としてボロノイ分割があり，これにより空間を節点数に等しい凸多角形あるいは凸多面体に分割でき，その結果よりデラウニー分割を得ることも可能である[2),3)]．デラウニー分割で求められた三角形あるいは四面体が最も正三角形あるいは正四面体に近いという保証はそれらの外接円あるいは外接球が他の節点をその内部に含まない，逆に他の節点を含まないように三角形，四面体を決定していることによる．

対象を二次元領域としたデラウニー三角分割法は文献2）と3）に，そしてより高速化されたアルゴリズムがSloanによって提案されている[5)]．しかしながらそれらはいずれも節点群の占有する凸部分空間のデラウニー三角分割であり，それらを利用した場合①局所的に凹部を有するような二次元領域に対しては凹部内部に対しても三角形分割を行う．②真の境界を再現できない場合がある，③領域内部に内部境界をもつ場合内部境界の内部に対しても要素分割を行う，といった不都合さが生じる．有限要素解析の対象領域の形状は必ずしも凸とは限らず，局所的に凹部が存在する場合が多い．また，内部境界を有していることも多く，有限要素解析法のための要素自動分割法としてデラウニー法をそのまま利用することはできないことは明らかである．以降において局所的に凹部を有する場合および内部境界を有する領域の三角形分割を考える．

そこで，n個の節点が二次元空間に設定されているものとし，これら節点群に対してデラウニー三角分割を適用する．いま，一点Pを考えるとPはそれに最も近接した節点Qと線で結ばれることは明らかである．なぜなら，もし結ばれないならば，Pは他の2節点（それらはいずれもQより遠方に位置する）とで三角形を構成し，その内部にQが必ず位置することになり，デラウニー分割に反することになる．以上のことより，2点PとQを結ぶ線を生成したいとき，それら2点間の距離をその近傍におけるどの2点間距離よりも小さくすればよい．

上に述べた方法はデラウニー法を直接利用して境界を再現する一つの方法であり，それは著者らによりすでに亀裂進展解析のための要素自動分割法に利用されている[6)]．なお，文献6）においては節点配置はすべて自動化されていて，亀裂線上に設定される節点を領域内部の節点密度に比べて密に設定することにより亀裂線の生成を行っている．一方，本研究での対象系のように境界上の節点配置がユーザーに任されている場合，節点配置に関する上記制約条件を満足しない形で節点が設定されることも予想され，この場合結果として境界を再現できない危険性が生じる．

任意に設定された節点集合を対象としてデラウニー法を利用した場合，どのように三角形が生成されるかは節点の位置関係によって一義的に決まる．なお，節点の位置関係によって2種類以上の分割法の発生も考えられる．これはデジェネラシー（degenracy）と呼ばれ，その扱いについては後に述べる[2),3)]．したがって，節点配置に関してなんら制約を与えず意図しているように境界線を発生させようとすれば，節点に関して付加的な情報が必要である．いま，この付加的情報として境界上に位置する節点順序を考える．ここで，境界上の点は

- 外部境界については時計回りの方向
- 内部境界については反時計回りの方向

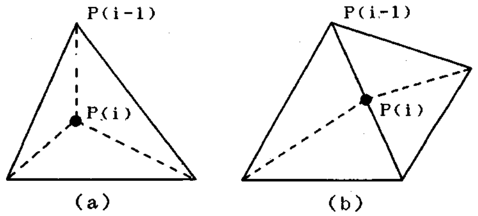
にそれぞれ点順序が認識されているものとする．なお，時計・反時計回り順序という区別は，2.（5）において有効に利用される．

（2）　境界の生成と粗い三角分割

一つの閉じた境界を構成するn個の節点，たとえば$\{P(i)\,;\,i=1,\ n\}$の順番が認識できていると仮定する．簡単のため節点$P(i)$が上記の順に境界上に位置しているものとする．ここで，$(i-1)$個の節点がすでに入力されて三角分割が行われていると仮定し，第i番目の節点$P(i)$を新たに設定する．なお，ここで利用するデラウニー三角分割は文献5）をもとに行うことにする．この方法では，すべての節点を包含できるスーパートライアングルとよばれる三角形が節点配置以前に設定されている．

節点$P(i)$はすでに三角分割されたいずれかの三角形内部もしくは分割線上に位置することになる．もし，$P(i)$を含む三角形（通常1個であるが，もしその節点が分割線上にあれば分割線を共有する2個の三角形を考える）の頂点の一つが$P(i-1)$であれば**図–1**（a），（b）に示すように三角形をさらに小三角形に細分割することで$P(i-1)$と$P(i)$間に位置する境界線を発生させ得ると同時に三角形に分割できる．

次に節点$P(i)$を含む三角形のどの頂点も$P(i-1)$ではない場合を考える．このとき節点$P(i)$を含む三



図—1　メッシュ細分割

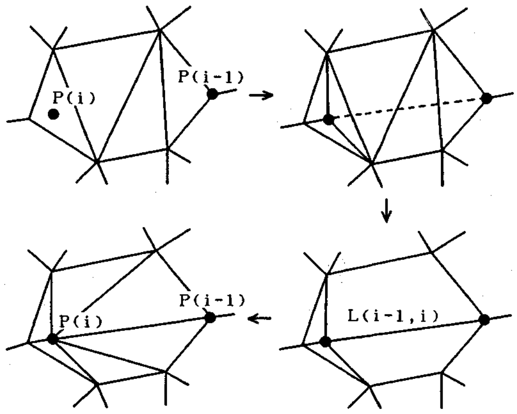
角形と P $(i-1)$ を頂点とする三角形との間に何個かの三角形が存在することになる. これら2節点間を線分(以降この線分を L $(i-1,\ i)$ で示す) で結ぶためにはこの線分で横切られるすべての線を排除しなければならない. L $(i-1,\ i)$ で横切られるこれらの線は境界辺として設定された辺でないことは明らかである. なぜなら, もしそれらが境界線であれば閉じた境界線はその部分で交差していることになるからである. 以上のことより L $(i-1,\ i)$ によって横切られる線はすべて境界辺ではなく, P $(i-1)$ 番目までの三角分割を行ううえで作成された境界辺以外の線でしかないことがわかる. したがって, それらは L $(i-1,\ i)$ を生成する段階で取り除いても良い.

P $(i-1)$ と P (i) との間に位置する三角形集合で共通辺（2個の三角形に共通する辺）を取り除くと多角形ができる. この多角形において L $(i-1,\ i)$ を設定し, 2個の部分（一般には多角形）に分割する. ついで, それぞれの部分を三角形に分割すると2点 P $(i-1)$ と P (i) 間の境界辺を発生させ, 同時に三角形分割できることになる（図—2参照).

上に述べた操作を繰り返すと最終的には P (n) を設定した段階で L $(n-1,\ n)$ が生成されるが, L $(n,\ 1)$ は作成できていない. この境界辺の作成には上で述べたのと同様にして P (n) と P (1) 間に位置する三角形の集合を探しだし, L $(n,\ 1)$ を作成した後三角形分割を行えばよい.

次に内部境界の生成法を考える. いま一つの閉じた内部境界上の節点を Q $(i : i=1,\ m)$ で表し, この順序に並んでいるとする. この順序で節点を設定し常に Q $(j-1)$ と Q (j) 間に位置する境界辺を上の外部境界辺を発生させるとき用いたのと同様にして作成すれば, 内部境界も必ず生成できることは明らかである.

上に述べた手法により外部境界であれ内部境界であれ境界辺を生成するのと同時に境界辺を有する三角形も作成されるが, この三角形作成時にデラウニー三角分割の考え方を導入する. いま, L $(i-1,\ i)$ を作成できた時点でこの辺を共有する2個の三角形が作られる. この2個の三角形をそれぞれ △ $(i-1,\ i,\ j)$ と △ $(i,\ i-1,\ k)$ で示す. この2個の三角形の外接円を求め, おのおのの内部に他の点を含むかどうか調べる. いま, △ $(i-1,\ i,\ j)$ の外接円が節点 m を含んだと仮定しよう. デラウニー三角分割では4個の節点, $i-1,\ i,\ j,\ m$ で作られる四辺形の対角線を入れ換えて新しい2個の三角形を作るがここでは“もしこの対角線の入れ替えで境界線を壊さない限り”という条件を付けて三角形の修正を行う. この操作により境界辺を作成しながら同時にデラウニー三角分割と同じ幾何学条件を備えた三角形を生成することができる.



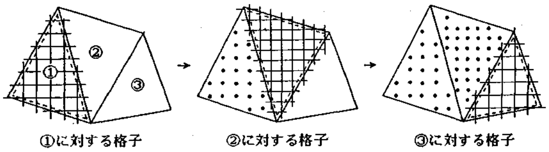
図—2　境界線生成と細分割

(3) 領域内部の節点の設定法

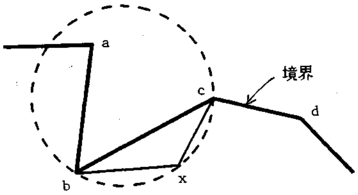
デラウニー法の特長として節点の配置は任意であることが挙げられ, 節点配置はその手法を利用する利用者に任されている. 本研究では次に示す方法で節点配置を行うことにする. なお, すでに述べた手法を用いることにより対象とする二次元領域は三角形状の小領域（これをブロック法と呼ぶことにする）に分割されており, ここに示す節点配置法はそれぞれのブロック内への節点の自動発生法である.

個々のブロック内において設定したい節点間距離によって格子間隔を決定し, その格子で対象領域全体を覆い, いま考えているブロック内に位置する格子点だけを残す. 以降この方法を格子状節点配置と呼ぶことにする. この手法を用いると隣接する2つのブロック間の共有辺近傍において節点が密に配置されるおそれがあり, それを防止するために辺からの距離が格子間隔の $1/a$ 以内のゾーンには格子点を設置しない（図—3参照). なお, 本研究では $a=3$ としている.

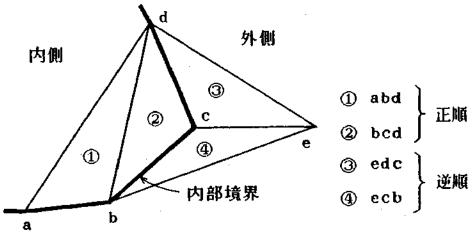
もし三角形のブロックの形状がゆがんでいる場合鋭角部には節点を設定できなくなることが生じ, その部分で節点間隔が疎らとなり三角分割後得られた要素自体もゆがんでしまう. これを防止するために2.(2) ですでに述べたようにデラウニー三角分割を用いたブロック作成を行う.



図―3　格子状節点配置



図―4　外接円と境界点の関係



図―5　内・外三角形の認識

（4）　領域内部に設定された点を対象とした三角形分割

2.（3）において新しく領域内部に設定された点を用いた三角形分割について述べる．方法は，デラウニー三角分割を用いるが，境界上の点に対するデラウニー三角分割と同様，“境界辺を壊さない限り”という条件付きで，三角形の修正を行う．

この条件付きデラウニー三角分割を行うことより，境界近傍の三角形は，厳密な意味でのデラウニー分割ではなくなる場合がある．図―4にその例を示す．a-b-c-d は境界辺であり，いま点 x を新しく挿入したとき，△ bxc の外接円は点 a をその内部に含んでいるが，この場合，辺 ab，bc は境界辺であることより，対角線 bc を ax で入れ換えることはできない．

（5）　領域内部の三角形の認識

2.（4）で示した方法により境界上の節点および領域内に任意に設定された節点を用いて三角形分割できたとすると，領域内部のみならず外部境界の外側あるいは内部境界の内側部といった要素分割の不要な部分においても三角形が生成される．ここでは不要なこれらの三角形をどのようにすれば認識できるかを考える．

いま，外部境界上の節点順序が時計回りに，内部境界上の節点が反時計回りに順次入力され，その後上記方法で三角分割されたとする．なお，生成されたおのおのの三角形は3個の頂点番号で認識され，またそれらの頂点番号は通常の有限要素解析の要素-節点関係で用いられるように反時計回りに格納されるものとする．

節点はスーパートライアングルの構成点である3個の節点以外は境界内部に設定されていることより，領域外部に位置する三角形の構成点は境界上の節点とスーパートライアングルの構成点であることは自明である．一方，領域内部に位置する三角形は境界点もしくは内部に設定された節点をその構成点としている．

領域内部に位置し境界節点を2個以上含む三角形は明らかに入力順序とは逆順に格納され，一方外部境界の外部および内部境界の内部といった不要な三角形は入力と同順序に格納されていることになる．以上のことより作成された三角形の頂点番号の格納順序を調べることにより不用な三角形を見つけ出し排除することができる（図―5参照）．

（6）　点を含む三角形の探査法

デラウニー三角分割では，点を1個ずつ設置し，その点を含む三角形をまず探査しなければならない．対象とする点数が増加するにつれ，この探査に要する演算時間が重要な問題となることが予想され，新しく設定された節点 P を包含する三角形 △（P）を探すために次の手法を利用する[5]．

いますでに作成された三角形のうちで最も最後に作成されたものを △（a，b，c）とする．ここで △（a，b，c）は頂点番号を示し，これら節点番号は反時計回りに格納されているものとする．この △（a，b，c）の3辺の方程式を求め，それらの直線と節点 P との関係を調べる．ここでおのおのの直線は a から b へ，b から c へ，c から a へ向かうものとする．もし3辺に対して節点 P が直線の左側に位置すれば △（a，b，c）が求める三角形である．△（a，b，c）が求める三角形でない場合節点 P はこれらの3辺のうちいずれかの辺に対してその右側に位置する．この場合，その辺を共有する三角形を新しい △（a，b，c）として上に述べた操作を繰り返す．

（7）　デジェネラシー（Degeneracy）[2),3)]

デジェネラシーとはいま新しく節点を設置してデラウニー三角分割しようとしたとき外接円上にたとえば4個の節点が位置し，複数の三角形分割が存在することをいう．このような場合どの三角分割を採用してもその部分においてはデラウニー法の幾何学条件を満足していること，またその選択の結果はその他の部分に影響を及ぼさないことより，ここでは最初に発生された三角形の組合せを採用することにする．

3.　デラウニー法を基本とした二次元非凸領域の三角分割法

前節で検討した結果を用いて複雑な境界形状をもち，また場合によっては内部境界ももつような任意二次元領

域に対する三角分割法を具体化する.

三角分割法は2段階より構成される. まず第1段階は境界の生成とその内部のブロック分割であり, 第2段階は領域内部への節点配置と要素分割である. 今後前者をブロック分割, 後者を要素分割と呼ぶことにする.

第1段階のブロック分割での入力データは次のとおりである.

- 境界の数
- 各境界上の節点番号
- 境界上の節点の座標値

なお, これら境界上の節点以外に領域内部に節点を設定してもよく, その場合にはそれら節点個数と座標値の入力が必要となる. 以下ではこれらの節点番号を $Q(k)$ で示す. なお, k は節点数を表す.

第2段階の要素分割での入力データは格子状節点配置法を用いるため必要なおのおののブロックでの格子間隔である.

[第1段階：ブロック分割]

ステップ1. データの入力

ステップ2. スーパートライアングルの設定

ステップ3. 外部境界上の第1番目の節点を設定し, スーパートライアングルを3個の三角形に分割する. $(N=2)$

ステップ4. 外部境界上の第 N 番目の節点を設定し, それを含む三角形（これを $\triangle(N)$ で示す）を探す（方法については2.(6)参照). ついで, $\triangle(N)$ を3個の三角形に分割.

ステップ5. $N-1$ 番目の節点を頂点とする三角形（$\triangle(N-1)$）を探す. もしそれらの一つが節点 N を含む場合, 2.(2)で述べたデラウニー三角分割により三角形の修正を行う. その後 $N=N+1$ としてステップ4へ. その他の場合はステップ6へ.

ステップ6. もし $\triangle(N-1)$ と $\triangle(N)$ が隣接している場合, $L(N-1, N)$ を作り, 2.(2)のデラウニー分割を行い, その後 $N=N+1$ としてステップ4へ. その他の場合ステップ7へ.

ステップ7. $\triangle(N)$ と $\triangle(N-1)$ の間に位置する三角形を探し, それら三角形の共有辺を取り除いて一つの多角形を作成し, さらに節点 N と $N-1$ を線で結び, 多角形を2個の部分に分割.

ステップ8. 2個の部分（一般には多角形）をそれぞれ2.(2)のデラウニー三角で分割. $N=N+1$ としてステップ4へ.

（ステップ4から8をすべての内部境界上の節点についても繰り返す.）

ステップ9. スーパートライアングルの構成点を頂点とする三角形を排除.

ステップ10. 境界上の節点を入力順序と同順序で作られている三角形をすべて排除.

[第2段階：要素分割]

ステップ11. データの入力

ステップ12. 格子間隔の決定と格子状節点配置

ステップ13. ブロック内に位置する節点を残す.

ステップ14. デラウニー法による三角分割

ステップ15. ラプラシアン法による節点の移動

なお, ラプラシアン法とは節点をその節点を頂点とするすべての三角形の重心位置に移動させる方法であって, その移動量が十分小さくなるまで反復計算を行う. 下式における $P(i)$ は注目している節点の座標位置を, n は i 点を頂点にもつ要素の数を示す.

$$P(i)=\frac{1}{2n}\sum_{i=1}^{n}\{P(j)+P(k)\} \quad \cdots\cdots(1)$$

上記アルゴリズムのうち2.(2)で示したデラウニー三角分割は節点 N の設定位置によってステップ5, 6, 7の3種類に分類して処理している. これは一般に節点 N と $N-1$ は近傍に位置していることが多く, この性質を利用して三角分割の処理速度を速めるためである.

4. 適用例

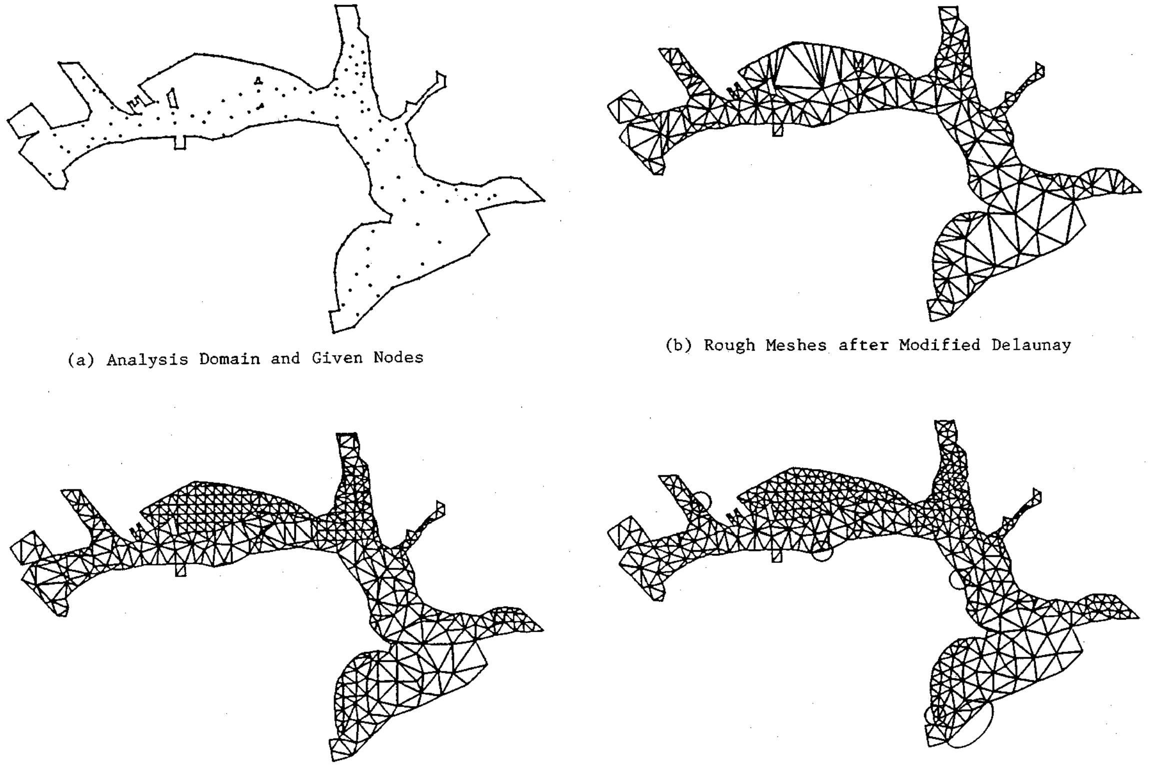
前節において提案した任意二次元領域に対する三角分割法をもとにして実際の工学分野で要求される要素自動分割法を作成する. ここで取り上げる問題は浅水長波流れ解析で必要な湾あるいは河川の有限要素モデル作成のための要素自動分割法と二次元複合構造系の有限要素モデル作成のための要素自動分割法である.

(1) 浅水長波流れ解析のための要素自動分割法

浅水長波流れ解析で扱われる対象系の幾何学的特色として二次元領域の境界形状が非常に複雑であること, また外部境界と同時に内部境界をもつことが多いこと等が挙げられる. さらに要素分割時において下式で示される Courant 数を考慮しなければならない.

$$c=\frac{\Delta t}{\Delta x}\sqrt{gh} \quad \cdots\cdots(2)$$

なお, 上式中 Δx は要素寸法, Δt は数値積分の時間幅, g は重力加速度, h は要素位置における水深である. 数値解析における計算時間を考えれば Δt を大きくできることが望ましく, また数値安定の面よりおのおのの要素について求められる c ができるだけすべての要素について一定値に近づけることが望ましい[7]. さらに一般的な要求として要素の形状ができる限り正三角形のようなゆがみの少ないものを作成できることが望ましい. ここではこれらの条件をできるだけ充足させる要素自動分割法


(a) Analysis Domain and Given Nodes

(b) Rough Meshes after Modified Delaunay

(c) Fine Meshes after Delaunay

(d) Meshes after laplacian

図―6　要素分割事例（岡山水道）

を考える．

上記特色のうち，境界に関する幾何学的な条件は前節で示した方法により解決できる．ゆがみの少ない三角形要素を作成するには節点の配置が重要である．一方，Courant 数および時間幅に関する条件を考慮して，いま c を一定，Δt を既知と設定し，これを式（2）に代入すると

$$\Delta x=\frac{\Delta t}{c}\sqrt{gh} \quad\cdots\cdots(3)$$

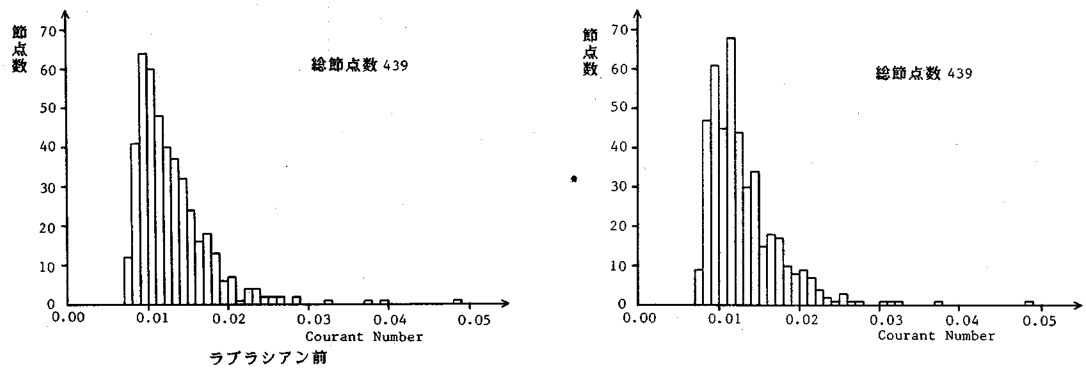
と書き改められ，個々の要素長はそれが位置する水深によって一義的に決定できることになる．これより浅水長波流れ解析では節点配置は単純に平面的な幾何学条件だけでは決定されず，三次元的な幾何学条件を満足するようにして行わなければならないことがわかる．このために本研究では次のような方法で節点配置を行う．

外部および内部境界上に位置する節点と水深既知点を入力データとして第1段階のブロック分割ができたとする．個々の三角形状のブロックの3頂点の水深は既知であり，したがってそれらの平均値をそのブロックの水深とする．この水深を式（3）に代入することにより個々のブロックでの要素長 Δx を決定できる．この値を用いて格子状節点配置法により節点を配置する．

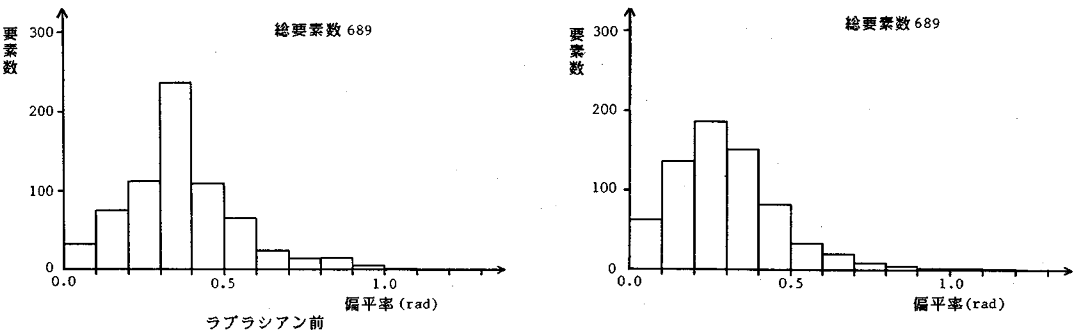
以上の考察をもととして作成された要素自動分割法は次のようになる．

ステップ1．外部境界上と内部境界上に位置する節点，および水深既知点を入力．
ステップ2．第1段階のブロック分割を行う．
ステップ3．各ブロックの平均水深を求め，要素長を決定する．
ステップ4．格子状節点配置法により節点を設定．
ステップ5．第2段階の要素分割を行う．
ステップ6．ラプラシアン法で領域内節点の移動を行う．

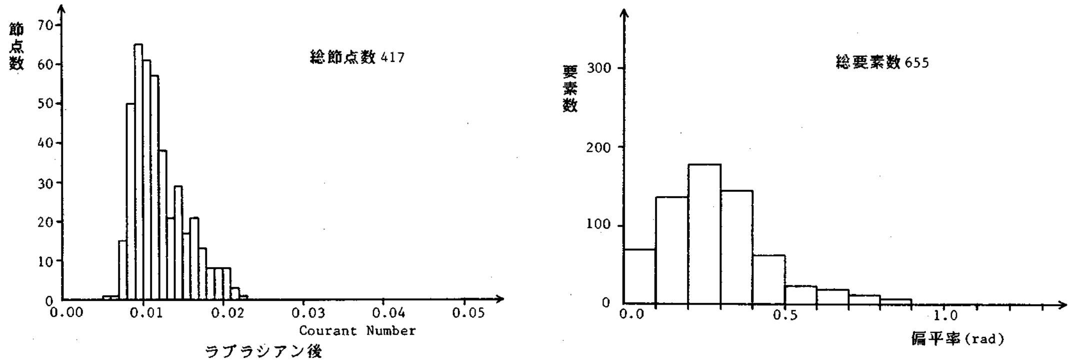
複雑な二次元領域を対象とした要素発生の流れを示したのが図―6である．この事例では外部境界のみならず内部境界も存在している．図―7と図―8は結果として得られた三角要素の Courant 数と三角要素のゆがみの分布を示す．左図はラプラシアン法を適用する以前の，そして右図はそれを適用して三角形形状を修正した後の結果の表示である．なお，Courant 数についてはこの要素分割では $c=0.01$ と仮定していることによりその分布が 0.01 近傍に集まることが望ましい．また，要素形状のゆがみの分布はおのおのの三角形要素について｜頂角 $-60°$｜の和を求めたものを示している．したがってこ



図—7 Courant 数比較表



図—8 三角形ゆがみ分布表

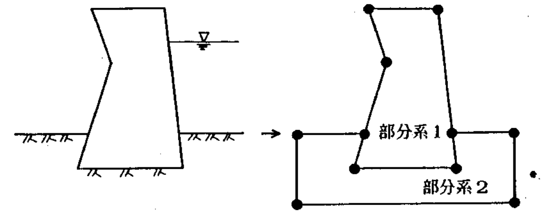


図—9 境界節点配置修正後の Courant 数，ゆがみ分布表（ラプラシアン適用後）

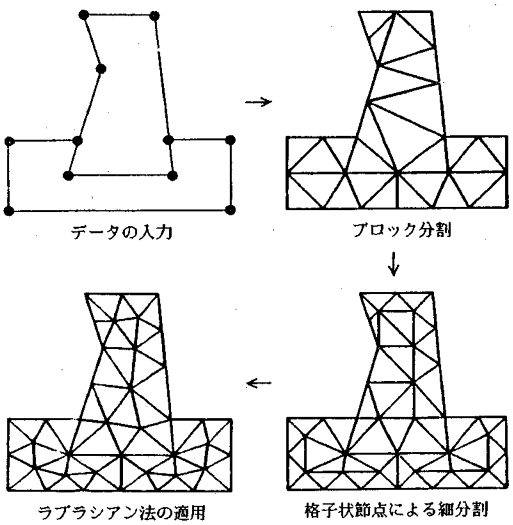
の図では0°近傍に分布することが望ましい．図—7と図—8の結果を比較するとラプラシアン法によってある程度形状の修正はなされているといえるものの，いまだ修正が不十分な三角形が見出される．これらのゆがんだ三角形の位置を調べると図—6(d)中に円で囲んだ要素であることが判明する．いずれも境界辺をもつ三角形であり，境界辺長があまりに短過ぎることより必然的にゆがんでしまったことがわかる．以上の考察より入力データの作成，特に境界辺上の節点間隔に注意を払わねばならないことが判明した．この考察結果を考慮して境界上の節点位置の修正を手動でもって作り直し，要素分割を行った結果を図—9に示す．図—7，図—8と図—9と比較すれば大幅な結果の改良がなされたことがわかる．

(2) 二次元構造解析のための要素自動分割法

二次元構造解析で扱われる系の特色として境界の形が複雑，穴のような内部境界の存在，また複合構造系（数種類の材料より構成される系）といった事柄が挙げられる．さらに，構造解析では系の一部に発生している応力集中を精度良くとらえることを目的とする場合が多く，したがって系のもつ幾何学的な複雑さ以外にそれによっ



図―10　複合構造系の例



図―11　複合構造系の要素分割例

て発生した応力集中を精度良くとらえることを目的とする場合が多く，したがって系のもつ幾何学的な複雑さ以外にそれによって発生する解の精度向上を図ることが要求される．ここで複合構造系，すなわち系が複数個の部分系より構成されている場合について考える（図―10参照）．

いま，系全体を K で示すと複合系は

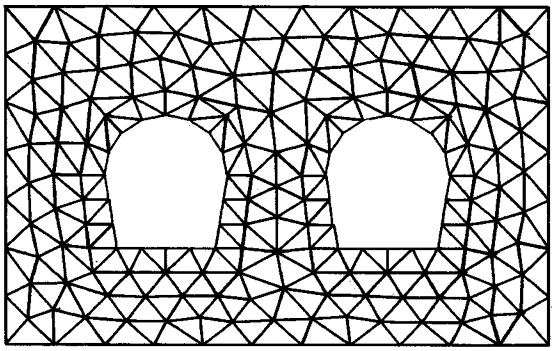
$$K=\bigcup_{i=1}^{n}\{k(i)\} \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (4)$$

で示される．ここで，$k(i)$ は第 i 番目の部分系を表す．その境界は閉じた1個の境界線 $L(i)$ で規定され，さらに，$L(i)$ は境界線上に設定された節点の集合 $\{N(i): i=1, m\}$ で示される．$L(i)$ の形状は一般に複雑であり，したがって個々の部分系の分割自体が3.ですでに述べた手法の対象となることは明らかである．以上のことより複合系の要素分割は結合された複数個の任意二次元領域の要素分割として考えられることになり，その要素分割法として

ステップ1．系を部分系の集合に置き換える．

ステップ2．個々の部分系を対象として要素分割を行う．

ステップ3．要素分割された部分系を再結合する．



図―12　適用例

という方法が考えられる．この方法を採用すれば第2ステップの個々の部分系の分割は3.の手法により可能である．したがって次に残された問題のステップ1と3について考察する．

いますべての部分系についてそれぞれの境界上の節点を別個の外部境界として反時計回りに順次入力し，3.の手法を部分系の数だけ繰り返せば各部分系は別個に取り扱えることになり，各部分系は小さな三角形の小ブロックの集合に分割される．この方法を採用すればステップ1の問題は解決する．このために要求される3.のアルゴリズムの変更点は外部境界を複数個に変えるだけである．

次にステップ3の問題を扱う．いま，相隣合う2つの部分系 $k(i)$ と $k(j)$ を考えると

$$L(i)\cap L(j)\neq 0 \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (5)$$

が成り立つ．第2ステップの個々の部分系の要素分割において式（5）で示される共通な節点番号をもつ節点をそのまま利用すればこの境界辺に接する三角形要素において共通の節点番号が $k(i)$ と $k(j)$ の両部分系において用いられることになる．要素自動分割法の目的は要素分割，厳密には要素-節点関係を作成することであり，上に述べた手法はこの目的を達成している．

節点配置法はすでに述べた格子状節点配置法を採用する．なお，格子間隔の設定には

（1）　手動設定：おのおのの部分系での格子間隔の直接入力する

（2）　自動設定：おのおのの部分系の境界上節点間隔の平均値を求めそれを格子間隔とする．

の2種類を準備し，ユーザーが自由に選択できるようにしている．

以上の結果をまとめて次のような要素自動分割法を得る．

ステップ1．データの入力

ステップ2．個々の部分系の外部境界上の節点情報を用いてブロック分割を行う．これを外部境界

数だけ繰り返す.

ステップ3．内部境界があればその節点情報を用いて同様に分割を行う.

ステップ4．節点配置法の選択．もし，自動設定を選択した場合ステップ6へ．格子間隔の手動設定を選択した場合ステップ5へ.

ステップ5．おのおのの部分系の格子間隔を入力する．ステップ7へ.

ステップ6．おのおのの部分系の境界上の節点間隔の平均値を求め，それをその部分系での格子間隔とする.

ステップ7．上で設定された格子間隔を用いて格子状に節点を配置し，おのおのの部分系内部に位置する節点だけを残す．この操作を部分系の数だけ繰り返す.

ステップ8．個々の部分系を対象として要素分割を行う.

ステップ9．ラプラシアン法によりそれぞれのブロック内の節点移動を行う.

ここに提案した手法の流れを図—11に，さらに適用例を図—12に示す.

5. あとがき

本研究ではデラウニー三角分割法をまず直線境界をもった任意二次元領域に適用できるように修正し，ついでその結果を利用して浅水長波流れ解析および複合構造といった複雑な二次元有限要素解析のための2種類の要素自動分割法を提案した．両者ともにデラウニー法を基本としていることより得られる要素の形状はユーザーが設定した節点分布に関して最適な幾何学形状をしている．したがって，もし得られた三角形要素の形状にゆがみが見つかればそれは節点の配置が不十分であることになり，節点配置を再度部分的に行えばより良い形状か得られることになる.

ここで提案した手法は全体としてみたときブロッキング法の流れに一致する．しかしながら，それはデラウニー法を2段階に適用したものであり，厳密にはデラウニー法の拡張ともいうべきものである．このようにデラウニー法をその基本としていることより一層良い節点配置を与えることができれば，数値解析的な意味においてより良い有限要素モデルの設定が可能となる.

参考文献

1) Verry, M. A. and Shephard, M. S. : Automatic three-dimensional mesh generation by the modified octree technique, Int. J. Num. Meth. Eng., Vol.20, pp.1965～1990, 1984.

2) Bowyer, A. : Computing Dirichlet tesselations, The Computer Journal, Vol.24, No.2, pp.162～166, 1981.

3) Watson, D. F. : Computing *n*-dimensionay Delaunay tesselation with application to Voronoi polytopes, The Computer Journal, Vol.24, No.2, pp.167～172, 1981.

4) Taniguchi, T. : An interactive automatic mesh generator for microcomputer, Computers & Structures, Vol.30, No.3, pp.715～722, 1988.

5) Sloan, S. W. : A fast algorithm for computing Delaunay triangulations in the plane, Advances in Engineering Software, Vol.9, No.1, pp.34～55, 1987.

6) Taniguchi, T. : Crack propagation analysis in civil engineering structures, CIVIL COMP 89, Vol.1, pp.371～378, 1989.

7) Kasiyama, K. and Okada, T. : Automatic mesh generation for shallow water flow, ISCFD-NAGOYA 1989

(1990. 7. 16 受付)

APPLICATION OF DELAUNAY TRIANGULATION TO ARBITRARY 2D DOMAIN WITH STRAIGHT BOUNDARIES

Takeo TANIGUCHI and Chikashi OHTA

The Delaunay triangulation is an effective tool as a mesh generation method for the finite element analysis, and it generates meshes for any convex domain which is occupied by data points. This suggests that the Delaunay triangulation must be modified when it is applied to any domain surrounded by boundaries with geometrical complexity. The main purpose of this investigation is to extend the Delaunay triangulation so that it can be used as a mesh generator for any 2-dimensional domain. Proposed method consists of two processes ; the coarse mesh generation and the fine mesh generation. Using this method, the authors show two mesh generators for engineering problems.