

名古屋大学 学生員○尾崎哲明  
 名古屋大学 正 員 二宮公紀  
 名古屋大学 正 員 梶田建夫

### 1. まえがき

有限要素法による解析は今日幅広く用いられているが、適用において問題となることは入力データの作成と結果の表示である。特に問題に適した分割を効率よく行うことが非常に重要なことである。このようなことにより要素の自動分割に関する研究が多く行われているが、それらのアルゴリズムは大型計算機や高度な画像端末を対象としているものが多い。ここでは、大型計算機と専用回線で結合されたパソコンを対象として要素の自動分割プログラムのアルゴリズムについて研究を行った。

### 2. 自動分割のアルゴリズム

解析領域を三角形に要素分割する場合は、各要素がなるべく正三角形に近い形状であることが望まれる。そのため、正三角形をすきまなく配置した規定ネットの手法<sup>1)</sup>が用いられることがある。この規定ネット法は多くの研究者によって種々の方法が提案されているが、正三角形の網を作り、それを対象領域に重ね境界線での要素分割のみを平滑化する処理といえる。ここでは、この規定ネットにおける節点の配置を示すものとして六角形パターンを用いることにした。以下にその手順を示す。

1) Node positioning まず領域全体を図-1のように六角形で覆う。各六角形の中央には節点の候補点が存在し、隣り合う候補点間を $r_0$ （作成される要素の一辺の長さであり、入力データで与える）とする。ここで候補点間の位置関係を明確にするために、領域外を含めすべての六角形に一定の順で番号を与えておく。この番号は六角形の位置関係が自動的に得られる様に各段左から右へつけられる。したがって、番号と対応して節点番号を格納しておけば節点相互の平面上での位置関係は容易に知ることができる。実際の節点はまず境界線上より $r_0$ の間隔で発生させる。この時、各節点の含まれる六角形の番号と対応させてその節点番号を格納する訳であるが、境界節点の場合は、内部節点と区別するために負号をつけておく。続いてすべての六角形内の候補点について領域内であるかどうかを調べる。候補点が領域内にあれば内部節点として節点を発生させる。ここで、六角形内に既に節点が存在する場合は、内部節点は発生させない。また、領域外である場合は節点番号を0としておく。以上の操作によって図-2の様な節点配置表が出来る。

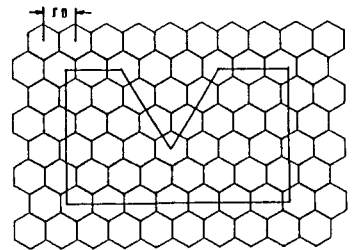


図-1 六角形パターン

|   |     |     |     |     |     |     |     |     |   |
|---|-----|-----|-----|-----|-----|-----|-----|-----|---|
| 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 |
| 0 | -1  | -2  | -3  | 0   | 0   | -4  | -5  | -6  | 0 |
| 0 | -7  | 28  | -8  | 0   | -9  | 29  | 30  | -10 | 0 |
| 0 | -11 | 31  | 32  | -12 | -13 | 33  | 34  | -14 | 0 |
| 0 | -15 | 35  | 36  | -16 | 37  | 38  | 39  | -17 | 0 |
| 0 | -18 | 40  | 41  | 42  | 43  | 44  | 45  | -19 | 0 |
| 0 | -20 | -21 | -22 | -23 | -24 | -25 | -26 | -27 | 0 |
| 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 |

図-2 節点配置表

2) Element forming 本来このschemeは非常に演算時間を要求されるところである。それは解析領域内に任意に節点を発生させ、これより要素を作成しようとする場合、どの節点を組合わせて要素を作成するかということ、すなわち節点の位置関

係を知るためにかなりの計算時間を要するからである。しかし、ここでは図-2の様に節点の位置関係が発生段階で既に決定されているため、要素の作成に要する演算ステップを減少させることができる。つまり、要素作成schemeでは六角形パターンにおける隣り合う六角形内の節点を結ぶことになる。実際には図-2の節点配置表から順次節点番号を取り出し、0でないものについては次の点と次の行のその点に近い二点の番号を節点配置表より抜出し要素を作る。解析領域の形状によっては境界付近で特異な節点の配置が生じることがあるので、あらかじめ生じ得るパターンそれぞれに対して処理するルーチンを作っておき、これとのパターンマッチングを行うことにより対処している。

### 3. 自動分割プログラム

実際に要素分割を行う場合には、部分的に要素を細分割を要求されることが普通である。これに対処するために、解析対象領域をまずsub-regionに分ける。1つのsub-region内では要素密度は一定として、sub-regionごとに要素密度を指定する。したがって、要素を細かくする必要がある部分をsub-regionとして指定しておき細分割すればよい。また、このプログラムでは均一な規定ネットの他に放射状に要素の密度が変化するもう1つの規定ネットを有している。これは六角形パターンにおける六角形の候補点を次式によって配置するものである。この式はどの位置でも要素が正三角形に近くなるような条件で誘導されたものである。このネットを図-3に示す。

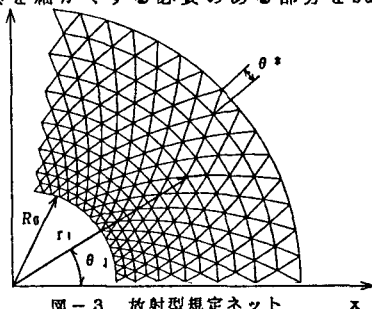
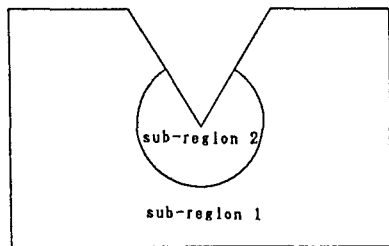


図-3 放射型規定ネット

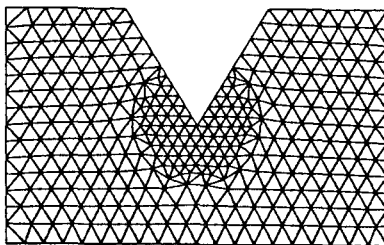
$$\begin{aligned}
 x(r, \theta) &= r_i \cos \theta_j & \text{ここで} & & r_i &= R_0(1 + 2\theta^* G^i) & (i = 1 \sim N) \\
 y(r, \theta) &= r_i \sin \theta_j & & & \theta_j &= \theta^*(2j + (1 + (-1)^i)/2) & (j = 1 \sim M) \\
 & & & & \theta^* &= 4\pi/M \\
 & & & & G &= 1 + (2\sqrt{3}\theta^*)/(4 - \sqrt{3}\theta^*)
 \end{aligned}$$

ただし、Nは半径方向、Mは円周方向の分割数である。

### 4. 適用例



sub-region 分割



要素分割

節点数 350

要素数 618

### 5. あとがき

ここでは、主として三角分割のアルゴリズムの紹介を行ったが、要素分割の過程、解析プログラムへのデータ転送等については当日発表を行う。なお、プログラムの開発はNEC PC-9801にて行った。

参考文献 1) 尾田十八、山崎光悦：有限要素自動分割の現状とその利用法(1)(2)、機械の研究、第37巻、第6、7号、1985