

三次元領域のポロノイ多面体要素への自動要素分割*

小林 睦夫*¹, 前川 博*¹, 近藤 泰平*²

Automatic Discretizing of a Three-Dimensional Domain into Voronoi Polyhedron Elements

Mutsuo KOBAYASHI, Hiroshi MAEKAWA and Yasuhira KONDOU

When a partial differential equation is solved numerically, it is usual to discretize the calculating domain into many elements by elaborate handwork. For this discretizing, we devised a new method with which the three-dimensional domain can be divided automatically into Voronoi polyhedron elements around arbitrary nuclear points. This method is described in detail giving an example. When Voronoi polyhedron elements are used, the various differential equations can be easily transformed into simultaneous algebraic equations for each of the nuclear points, for which parallel processing can be done more speedily and accurately. Moreover, the Voronoi polyhedron element table has possibilities of future growth; it can be used directly as an analytical method of solving a partial differential equation and it can be transformed into an ordinary tetrahedron element table.

Key Words: Voronoi, Finite Element, Automatic Discretizing, Three-Dimensional Domain, Parallel Process, Numerical Analysis

1. ま え が き

コンピュータを使って偏微分方程式を数値的に解くとき、計算領域を小さな領域(要素)に分割し、離散化された方程式系を構成するのが一般的である。従来、計算領域の要素分割は手作業により行われ、計算処理時間の大部分はこの作業に費やされてきた。尾田・山崎⁽¹⁾の解説にも詳しいように、Green-Sibson⁽²⁾が示したポロノイ多角形を利用した手法など⁽³⁾により、二次元領域の自動分割にはある程度の進歩が見られる。しかし、三次元領域の自動要素分割については、Bowyer⁽⁴⁾や Watson⁽⁵⁾によるポロノイ多面体とデローニ四面体を利用する手法以外には、具体的な手法は著者らの知る限りでは存在しない。すでに著者ら⁽⁶⁾は、ランダムに位置を指定した要素点を含む要素形(ポロノイ多角形)に二次元領域を自動分割する手法を開発し、その分割要素を用いるとラプラス方程式は連立代数方程式に簡単かつ高精度に変換できることを示した具体的例題について報告した。このような要素を流れ場の解析に用いた例が、最近、谷口ら⁽⁷⁾によって

も報告されている。本報では既報⁽⁶⁾の手法を三次元にそのまま拡張した自動要素分割の手法を詳しく述べる。

2. ポロノイ多面体

丸善の科学大辞典⁽⁸⁾によれば、ポロノイ多面体とは空間分割によって粒子配置の特徴を表す概念で、空間の各点を最も近い粒子に振り分けることにより得られる多面体をいう。具体的には、核となる点(要素点と呼ぶ)をいくつか与え、隣接する要素点どうしを結んだ線分の垂直二等分面で空間を分けてポロノイ多面体(図1)が得られる。それは次のような性質を持つ。

- (1) 要素境界面は隣接要素点を結ぶ線分の垂直二等分面である。
- (2) 多面体要素のりょう(辺)はその要素点とそれに隣接する2要素点の3点から等距離にある線分である。
- (3) 多面体要素の頂点は、その要素点とそれに隣接する3要素点の4点から等距離にある点である。
- (4) ある多面体要素内の任意の点に最も近い要素点は、その多面体要素の要素点である。
- (5) 要素点位置を任意に与えたとき、多面体は凸多面体であるがその形状は要素点の配置によって決ま

* 原稿受付 平成2年9月3日。

¹ 正員、新潟大学工学部(〒950-21 新潟市五十嵐2の丁8050)。² 学生員、新潟大学大学院。

り、一定ではない。

3. 自動要素分割法

著者らの開発した手法は、Bowyer⁽⁴⁾やWatson⁽⁵⁾の手法とは異なり、与えられた要素点に対するポロノイ多面体自身を直接に表す表(要素表)を用いるものである。図2に示す分割手順の流れ図を参照すれば、要素点の任意な位置座標のみを追加データとして与えるだけで分割が自動的に行われることがわかる。この流れを具体的な例を示しながら以下に詳しく説明する。

3.1 要素表 要素表を用いるのがこの分割プログラムの特徴の一つである。要素名(要素生成順の番号) $I=7$ のポロノイ多面体とそれを表す要素表の一例を図1と表1に示す。分割されたおのこのポロノイ多面体を表すのに必要十分な要素点座標、面数、面名、各面の辺数(角数)、辺名が要素表中に記憶される。

面名は、その面で接する相手の要素名とする。ポロノイ多面体の性質(2)で述べたように、面を構成する辺は、考えている要素の要素点、面名の要素点ならば、もう一つの要素点から等距離の線分だから、もう一つの要素点の要素名をその辺名とする。したがって、

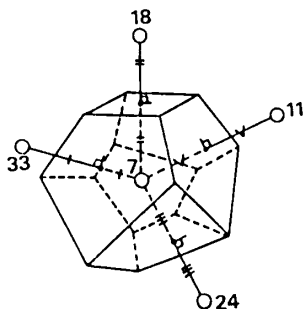


図1 ポロノイ多面体要素

辺名はその辺によりつなぎ止められている面の面名を表すことになる。面名と辺名はそれぞれ要素1(要素名1の要素)からの通し番号である面番地と辺番地で管理された配列変数に記憶する。また、辺番地は要素点

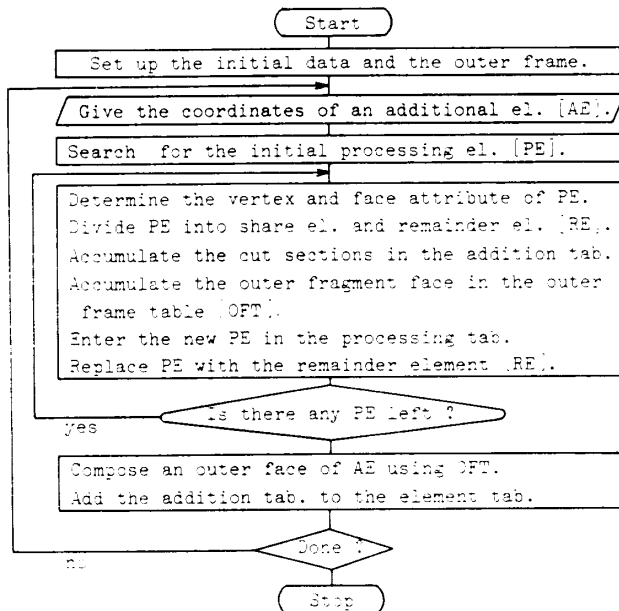


図2 分割手順流れ図

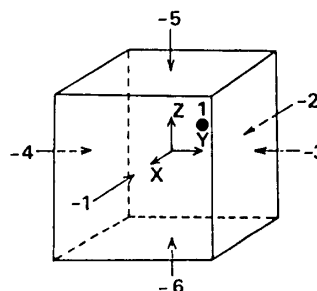


図3 座標系と外枠面

表1 ポロノイ多面体要素表

| Element Name I=7 | Element Coordinates X(7),Y(7),Z(7) | | | | Number of Faces FC(7)=10 | | | | |
|------------------------------------|-------------------------------------|-------------------------|--------------------|-------|--------------------------|--|--|--|--|
| Starting Face Address FI(7)=84 + J | Starting Side Address EI(7)=645 + K | | | | | | | | |
| Face Address J | 84 | | 85 | | 46 | | | | |
| Face Name F(J) | 24 | | 11 | | 18 | | | | |
| Number of Sides EC(J) | 4 | | 5 | | 4 | | | | |
| Side Address K | 645, 646, 647, 648 | 649, 650, 651, 652, 653 | 654, 655, 656, 657 | - , - | | | | | |
| Side Name E(K) | - , - , 11, 33 | 24, - , - , 18, 33 | - , 33, 11, - | - , - | | | | | |

表2 初期要素表(要素1)

| Element Name I=1 | Element Coordinates X(1),Y(1),Z(1) | | | | | | Number of Faces FC(1)=6 | |
|-----------------------------------|------------------------------------|-------------|-------------|-------------|-------------|-------------|-------------------------|--|
| Starting Face Address FI(1)=1 + J | Starting Side Address EI(1)=1 + K | | | | | | | |
| Face Address J | 1 | 2 | 3 | 4 | 5 | 6 | | |
| Face Name F(J) | -1 | -2 | -3 | -4 | -5 | -6 | | |
| Number of Sides EC(J) | 4 | 4 | 4 | 4 | 4 | 4 | | |
| Side Address K | 1, 2, 3, 4 | 5, 6, 7, 8 | 9,10,11,12 | 13,14,15,16 | 17,18,19,20 | 21,22,23,24 | | |
| Side Name E(K) | -3,-5,-4,-6 | -3,-6,-4,-5 | -1,-6,-2,-5 | -1,-5,-2,-6 | -1,-3,-2,-4 | -1,-4,-2,-3 | | |

からその面を見て時計回りに番地を付ける。

3・2 初期設定 図3のように、直角座標系(X, Y, Z)をとり、その原点を中心とする立方体の内部を分割領域として定義する。立方体の6面(外枠面と呼ぶ)の各面には、外枠面であることを区別させるために、図3のように-1から-6の負の番号を面名として与える。分割領域内の任意な位置に要素点1を発生させる。このとき要素1は立方体自身であり、要素表(初期要素表)は表2ようになる。

図4に示すように、分割要素数が2のときに、要素点3(追加要素点)が追加された実例を取り上げる。

3・3 母体要素の検索 その要素内に追加要素点が存在する要素を母体要素とよび、この例では要素2が母体要素である。母体要素の検索の際、ポロノイ多面体の性質(4)を利用するが、すべての要素点と追加要素点の距離を比較しなくとも、要素表を用いて効率よく母体要素をみつける次の方法が使える。はじめに、仮の母体要素を任意に選択する。仮の母体要素に隣接する要素のうちで、仮の母体要素よりも追加要素点に近い要素があれば、その隣接要素を次の仮の母体要素とする。それがなければ、その仮の母体要素が真の母体要素である。この母体要素が次に述べる最初の処理要素となる。

3・4 処理要素の頂点属性の決定 処理要素とは要素点を追加した結果、最終的に追加要素と接触する

要素であり、それは分断要素(追加要素の一部になる部分)と更新要素(処理要素から分断要素が切り取られて残った部分)に分けられる。処理要素はその要素点と追加要素点を結ぶ線分の垂直二等分面によって分断される。そこでまず、処理要素点および追加要素点と処理要素の頂点の間の距離を比較して各頂点の属性を決定する。この際、ポロノイ多面体の性質(3)により、処理要素名と追加要素名および頂点を挟む辺名の四つの要素点から等距離の点として頂点座標を求めることができる。辺名が負の場合には、頂点は他の要素点から等距離にある外枠面または外枠辺上の点であることを考慮して求められる。

最初の処理要素である母体要素(要素2)の場合を図5に示す。処理要素(要素2)の要素表が表3の上部分とすると、その要素表の最初の辺名番地EI(I)を頂点属性配列変数G(L)の最初の番地(L=1)に対応させ、以下順に番地を対応させる。頂点名はそれらで構成する多角形の辺名でも代用されるが、図5ではポロノイ多面体の頂点名A, B, C, D, E, F, G, Hを用いて表示した。

- (1) 頂点が分断要素側(黒丸頂点A, B, E, F)のとき、G(L)の値は1とする。
- (2) 頂点が更新要素側(白丸頂点C, D, H, G)のとき、G(L)の値は-1とする。
- (3) 頂点が分断要素と更新要素の境界面上(すな

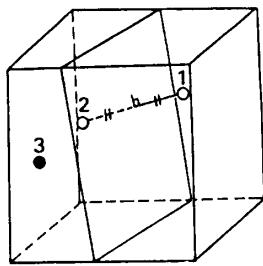


図4 要素点の追加

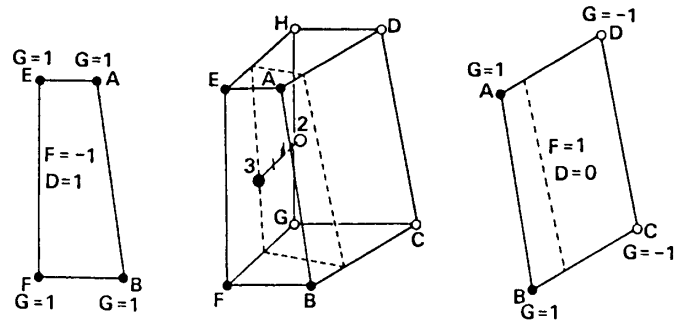


図5 処理要素(要素2)

表3 処理要素(要素2)と属性の決定

| Element Name I=2 | Element Coordinates X(2),Y(2),Z(2) | | | | | | Number of Faces FC(2)=6 |
|------------------------------------|------------------------------------|-------------|-------------|-------------|-------------|-------------|-------------------------|
| Starting Face Address FI(2)=14 + J | Starting Side Address EI(2)=43 + K | | | | | | |
| Face Address J | 14 | 15 | 16 | 17 | 18 | 19 | |
| Face Name F(J) | 1 | -1 | -2 | -4 | -5 | -6 | |
| Number of Sides EC(J) | 4 | 4 | 4 | 4 | 4 | 4 | |
| Side Address K | 43,44,45,46 | 47,48,49,50 | 51,52,53,54 | 55,56,57,58 | 59,60,61,62 | 63,64,65,66 | |
| Side Name E(K) | -6,-2,-5,-1 | 1,-5,-4,-6 | 1,-6,-4,-5 | -1,-5,-2,-6 | 1,-2,-4,-1 | 1,-1,-4,-2 | |
| Vertex Name (Fig.5) | B, C, D, A | B, A, E, F | C, G, H, D | F, E, H, G | A, D, H, E | C, B, F, G | |
| Vertex Address L | 1, 2, 3, 4 | 5, 6, 7, 8 | 9,10,11,12 | 13,14,15,16 | 17,18,19,20 | 21,22,23,24 | |
| Vertex Attribute G(L) | 1,-1,-1, 1 | 1, 1, 1, 1 | -1,-1,-1,-1 | 1, 1,-1,-1 | 1,-1,-1, 1 | -1, 1, 1,-1 | |
| Reference Address M | 1 | 2 | 3 | 4 | 5 | 6 | |
| Face Attribute D(M) | 0 | 1 | -1 | 0 | 0 | 0 | |

わち、頂点が追加要素点と処理要素点の2点から等距離にあるとき、 $G(L)$ の値は零とする。ただし、この際に計算機の丸め計算誤差を考慮して等距離の判定を下す必要がある。

処理要素のすべての面の頂点属性を決定し、この頂点属性を用いて次に述べる面属性の決定を行う。

3.5 処理要素の面属性の決定 面属性 $D(M)$ は、面が分断要素の面になるか、更新要素の面になるか、あるいは分断されるかを表す。処理要素の各面についての $D(M)$ の値は頂点属性 $G(L)$ を用いて次のように決定される。

(1) 面のすべての頂点について $G(L)=1$ のとき、面はそのまま分断要素の面となるので、その面の $D(M)$ に1を与える。

(2) 面のすべての頂点について $G(L) \neq 1$ のとき、面はそのまま更新要素の面となるので、その面の $D(M)$ に-1を与える。

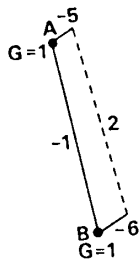


図6 分断断片面

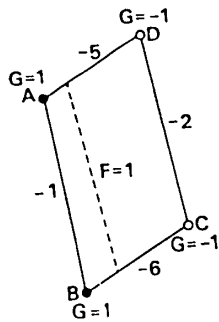


図7 分断面

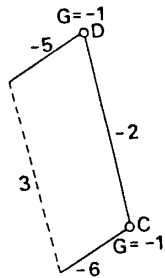


図8 更新断片面

(3) 上記以外の場合、面は分断されるので、その面の $D(M)$ に零を与える。

したがって、実例の場合、面属性は図5と表3に示すように決定される。

3.6 分断処理 処理要素の各面を分断した後でまたはそのままの形で更新要素と分断要素に分配し、さらに新しく発生した処理要素の切口断面(境界面)を生成する三つの処理を分断処理と呼ぶ。

3.6.1 面を分断しそれらを分配 面属性が零の面(分断面と呼ぶ)を、更新要素側と分断要素側に分断し、それぞれ、更新表(更新要素のデータを集積する表)と分断表(分断要素のデータを集積する表)に面名や辺名などを集積記憶する。

実例の場合、処理要素の記憶番地順に調べると、1番目の面が最初の分断面(図7、表5に再掲)であり、それは次のようにして分断される。

(1) 分断表(表4)に、分断面の面名と同じ面名1の項を設け、その面の最初の辺名を処理要素名の2とする。

(2) 更新表(表6)に、分断面の面名と同じ面名1の項を設け、その面の最初の辺名を追加要素名の3と

表4 分断表

| | |
|---------------------|------------------|
| Number of Faces | CFN=1 |
| Face Address | L 1 |
| Shared Element Name | CK(L) 2 |
| Face Name | CF(L) 1 |
| Number of Sides | CEC(L) 4 |
| Side Address | M 1, 2, 3, 4 |
| Side Name | CE(M) 2,-5,-1,-6 |

表5 分断面1の要素表と属性部分

| | | |
|---------------------|-------|-------------|
| Face Address | J | 14 |
| Face Name | F(J) | 1 |
| Number of Sides | EC(J) | 4 |
| Side Address | K | 43,44,45,46 |
| Side Name | E(K) | -6,-2,-5,-1 |
| Dividing Point | : | : |
| Vertex Name (Fig.7) | : | B, C, D, A |
| Vertex Address | L | 1, 2, 3, 4 |
| Vertex Attribute | G(L) | 1,-1,-1, 1 |
| Dividing Point | : | : |

表6 更新表

| | |
|-----------------|------------------|
| Number of Faces | TFN=1 |
| Face Address | L 1 |
| Face Name | TF(L) 3 |
| Number of Sides | TEC(L) 4 |
| Side Address | M 1, 2, 3, 4 |
| Side Name | TE(M) 3,-6,-2,-5 |

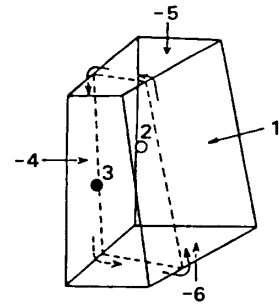


図9 分断面処理順序

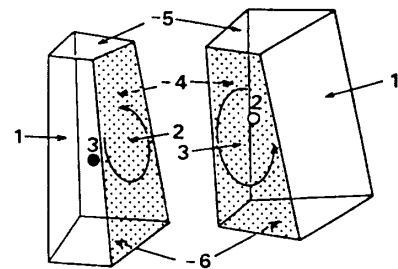


図10 境界面の作成

する。

(3) 分断面の頂点属性 G が $G \neq 1$ から $G=1$ に変わる直前の辺を、分断要素側の面の 2 番めの辺名(第 2 辺名)とし、ひき続いてその辺から、 $G \neq 1$ に変わる直前までの辺名を、処理要素の要素表より分断表に転記集積する(表 4)。

(4) 分断面の頂点属性 G が $G=1$ から $G \neq 1$ に変わる直前の辺名を更新要素側の面の第 2 辺名とし、ひき続いてその辺から、 $G=1$ に変わる直前の辺名までを、処理要素の要素表より更新表に転記集積する(表 6)。

(5) 分断要素側の面の第 2 辺名(実例の場合 - 5) を面名とする面は、分断面である。したがって、それを次に分断し、(1)から(4)の処理を同様に行う。分断要素側の面の第 2 辺名をたどって、図 9 のように最初の分断面に戻るまで(1)から(4)の処理を繰返す。この結果として、分断表には表 7 の面番地 4 までが集積され、更新表には表 8 の面番地 4 までが集積される。

3.6.2 分断されない面を分配 例えば図 5 左に示す分断されない面は次のように処理する。分断要素の面となる面(面属性 D が 1 の面)の内容は分断表へ(表 7 面番地 5)、更新要素の面になる面(面属性 D が -1 の面)の内容は更新表へ(表 8 面番地 5)、処理

要素の要素表(表 3 面番地 15 と 16)よりそれぞれ転記集積する。

3.6.3 境界面を作成 図 10 に示すように、処理要素の分断により新しい切口断面(境界面)が発生したので、分断要素側には、面名が処理要素名 2 である面を設け、3.6.1 項で処理した面の逆順にそれらの面名を辺名として集積記憶する(表 7 面番地 6)。一方、更新要素側には、面名が追加要素名の 3 である面を設け、3.6.1 項で処理した面の順にそれらの面名を辺名として集積記憶する(表 8 面番地 6)。

3.7 断片面の処理 3.6 節で得られた分断要素の各面(断片面と呼ぶ)について、次の三つのうちのどれかの処理が行われる。

3.7.1 境界面の処理 3.6.2 項で集積された断片面の中で面名が処理要素名である断片面(表 7 面番地 6)は追加要素の面になるので、表 9 のように追

表 9 追加表

| Number of Faces AFN=1 | | |
|------------------------|-------|---------------|
| Face Address | L | 1 |
| Face Name | AF(L) | 2 |
| Number of Sides AEC(L) | | |
| Side Address | M | 1, 2, 3, 4 |
| Side Name | AE(M) | -6, -4, -5, 1 |

表 7 分断表

| Processing Element Name=2 → CK(L) | | Number of Faces CFN=6 | | | | | |
|-----------------------------------|--------|-----------------------|------------|------------|-------------|-------------|-------------|
| Face Address | L | 1 | 2 | 3 | 4 | 5 | 6 |
| Shared El. Name | CK(L) | 2 | 2 | 2 | 2 | 2 | 2 |
| Face Name | CF(L) | 1 | -5 | -4 | -6 | -1 | 2 |
| Number of Sides | CEC(L) | 4 | 4 | 4 | 4 | 4 | 4 |
| Side Address | M | 1, 2, 3, 4 | 5, 6, 7, 8 | 9,10,11,12 | 13,14,15,16 | 17,18,19,20 | 21,22,23,24 |
| Side Name | CE(M) | 2,-5,-1,-6 | 2,-4,-1, 1 | 2,-6,-1,-5 | 2, 1,-1,-4 | 1,-5,-4,-6 | -6,-4,-5, 1 |

表 8 更新表

| Element Name I=2 | | Element Coordinates X(2),Y(2),Z(2) | | | Number of Faces TFN=6 | | |
|------------------|--------|------------------------------------|------------|------------|-----------------------|-------------|-------------|
| Face Address | L | 1 | 2 | 3 | 4 | 5 | 6 |
| Face Name | TF(L) | 1 | -5 | -4 | -6 | -2 | 3 |
| Number of Sides | TEC(L) | 4 | 4 | 4 | 4 | 4 | 4 |
| Side Address | M | 1, 2, 3, 4 | 5, 6, 7, 8 | 9,10,11,12 | 13,14,15,16 | 17,18,19,20 | 21,22,23,24 |
| Side Name | TE(M) | 3,-6,-2,-5 | 3, 1,-2,-4 | 3,-5,-2,-6 | 3,-4,-2, 1 | 1,-6,-4,-5 | 1,-5,-4,-6 |

表 10 外枠表

| | | | | | | | |
|--------------------|----------|------------|----|----|------------|------------|------------|
| Outer Face Name | -I | -1 | -2 | -3 | -4 | -5 | -6 |
| Number of Fragment | BC(I) | 1 | 0 | 0 | 1 | 1 | 1 |
| Face Address | J | 1 | - | - | 1 | 1 | 1 |
| Face Fragment Name | BK(I,J) | 2 | - | - | 2 | 2 | 2 |
| Number of Sides | BEC(I,J) | 4 | - | - | 4 | 4 | 4 |
| First Side Address | BEI(I,J) | 1 | - | - | 1 | 1 | 1 |
| Side Address | M | 1, 2, 3, 4 | - | - | 1, 2, 3, 4 | 1, 2, 3, 4 | 1, 2, 3, 4 |
| Side Name | BE(I,M) | 1,-5,-4,-6 | - | - | 2,-6,-1,-5 | 2,-4,-1, 1 | 2, 1,-1,-4 |

加表(追加要素の内容を集積する表)に転記集積する。

3・7・2 外枠断片面の処理 断片面のなかで、負の番号の面名を持つ断片面(外枠断片面と呼ぶ)は外枠表に集積記憶する。追加要素は分断要素を寄せ集めてできる要素だから、分断要素が外枠断片面を持つ場合は、他の分断要素の同じ面名の外枠断片面と合成して一つの面にする必要がある。したがって、外枠断片面は集積記憶しておき、後でそれらの面の合成処理を行う。これについては3・10節で説明する。要素2から切り取られる分断要素の場合、-1, -4, -5, -6の面名の外枠断片面が得られるので、表10に示すそれぞれの外枠表に、それらの断片面(表7面番地2~5)を分断表より転記集積する。このとき、その断片面を切り取られた要素名 $CK(J)$ (現在処理中の要素名)は、外枠表には外枠断片面名 $BK(I, J)$ の呼び名で転記される。

3・7・3 処理表への登録 3・7・1項と3・7・2項で処理された以外の断片面(すなわち、更新要素との境界面以外の正の面名を持つ断片面)に接している要素は、分断処理を必要とする要素(処理要素)である。したがって、これらの要素名を表11に示すような処理表の $J(JA)$ に登録記憶しておき、これを用いて未処理の処理要素を探し出して分断処理を行う。

処理表の先頭の $J(1)$ にはあらかじめ母体要素名が記憶されており、それに続けて分断処理の必要な要素名が $J(JA)$ に登録記憶される。処理済みチェック $C(JA)$ は、要素 $J(JA)$ の処理の完了を調べるための値であり、初めは登録時の処理要素名を与えておき、要素 $J(JA)$ の分断処理が済んだときに $C(JA)$ の値を零とする。次に分断処理を行う要素名は、 $C(JA)$ を整理番号順に調べて、その値が零でない $J(JA)$ とすれば

表 11 処理表

| | | |
|-------------------------|---------|---------|
| Reference Address | JA | 1, 2, 3 |
| Processing Element Name | $J(JA)$ | 2, 1, 0 |
| Finished Check | $C(JA)$ | 2, 2, 0 |

表 12 処理済みチェック後の処理表

| | | |
|-------------------------|---------|---------|
| Reference Address | JA | 1, 2, 3 |
| Processing Element Name | $J(JA)$ | 2, 1, 0 |
| Finished Check | $C(JA)$ | 0, 2, 0 |

表 13 全分断処理終了時の処理表

| | | |
|-------------------------|---------|---------|
| Reference Address | JA | 1, 2, 3 |
| Processing Element Name | $J(JA)$ | 2, 1, 0 |
| Finished Check | $C(JA)$ | 0, 0, 0 |

よい。新たに分断処理をして得られたこの節の分断要素の面名の中で、すでに $J(JA)$ に登録記憶されている面名は登録記憶しない。もし登録されたすべての $C(JA)$ が零になれば、要素点の追加による分断処理はすべて終了する。

3・8 処理要素の更新 処理要素は分断要素を切り取られて更新要素に変形したので、元の処理要素の要素表(表3)を更新表(表8)に置き換える。この際、更新表の容量が元の要素表の容量を超えた場合は、処理要素の要素番号より大きい要素の内容を記憶番地の後方へずらして、容量を増してから置き換える。

処理要素に対するここまでの処理が終了したとき、 $C(JA)$ の値を零とする。実例の場合、処理表は表12のようになる。

3・9 処理要素の探索 処理表から次の処理要素を探し、3・4節から3・8節までの処理を行う。実例の場合の処理表(表12)より、整理番地の順で $C(JA)$ が零でないものは $JA=2$ のときである。したがって、 $J(2)$ の値である1が次に分断処理を行う要素名である。

要素1について3・4節から3・6節までの処理を行うと、図11に示す分断要素が得られる。分断要素の面名1の断片面は追加要素の面になるので、その断片面のデータを追加表(表15面番地2)に集積記憶し、面名が負の番号の、-1, -5, -6の断片面を外枠表に集積記憶する。残った面名2の断片面については、更新要素との境界面ではない正の面名なので、処理表

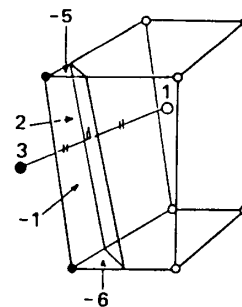


図 11 要素1の分断処理

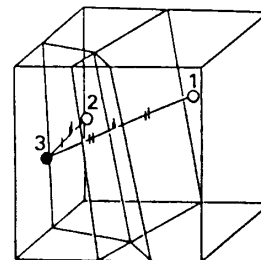


図 12 分断処理結果

に登録記憶するところだが、すでに処理表に記憶してあるから記憶しない。要素1の処理が終わったとき、表13のように処理表のすべての $C(JA)$ が零なので、分断処理はすべて終了したことになる。このときの分割領域の様子を図12に示す。

3・10 合成処理 分断処理が全部終了すれば、境界面は追加要素の面として追加表に集積記憶してある。しかし、追加要素が外枠面部分を持つとき、3・7・2項で述べたように、合成処理を行う必要がある。

外枠面名 $-I$ の断片面の数(断片面数)が外枠面の配列変数 $BC(I)$ に記憶されているので、まず $BC(I)$ が1の外枠面名を探し出し、その面名の断片面を追加表に集積記憶する。実例では $BC(4)$ が1なので、面名 -4 の外枠面に記憶されている断片面をそのまま追加表(表15面番地3)に転記集積する。

外枠面名 $-I$ の断片面数 $BC(I)$ が2以上のときは、それらの断片面を合成して一つの面(合成外枠面と呼ぶ)を完成させる。合成外枠面の辺(外周辺と呼ぶ)とな

るのは、新しい切口線として生じた辺かまたは外枠辺の一部である。新しい切口が生じるときは、3・6・1項(1)項を参照すればわかるように、分断要素の面項目の最初の番地の辺名(第1辺名)と外枠断片面名 $BK(I,J)$ は必ず一致している。したがって、辺が外周辺となる条件(外周条件)は、その辺名が負であるかまたは $BK(I,J)$ と一致する第1辺である。合成処理によりこれら以外の辺(接合辺)は他の外枠断片面の辺と互いに接合される。辺名はつなぎ止められる面名であるから、接合辺名と同じ外枠断片面名 $BK(I,K)$ を持つ面項目が外枠表中に必ず存在する。これらのことを考慮して図13の流れ図により合成処理が行われる。

実例の場合、 $BC(1),BC(5),BC(6)$ の値が2であるから、外枠面 $-1, -5, -6$ のそれぞれについて合成処理を行う必要がある。外枠面 -1 の場合を例として取り上げ、図14にその概要を示し、その流れに沿って合成処理を次に詳しく述べる。

3・10・1 先頭外周辺探索 第1辺名で外周条件を満足する辺を外枠表中から探し、それを先頭外周辺名と呼ぶ。そして、追加表に合成する外枠面名の項を設け、最初の辺名としてそれを書き込む。

実例の場合、外枠面 -1 の外枠表は表14のようになっているから、面番地が2の第1辺が外周条件を満足する。そこで、先頭外周辺名1を追加表中の面名 -1 の項(表15面番地4)の最初の辺番地($M=13$)の辺名として書き込む。

3・10・2 隣の辺の取出し 時計回りで隣の辺とは辺番地順で次の辺であり、最終隣の隣の辺は第1辺である。

実例の場合、表14の面番地2の辺名1の隣の辺

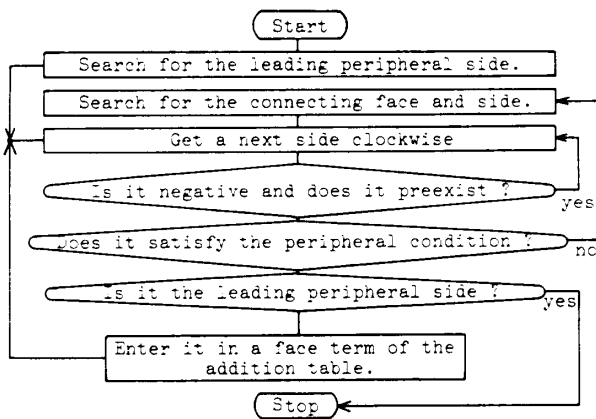


図13 合成処理流れ図

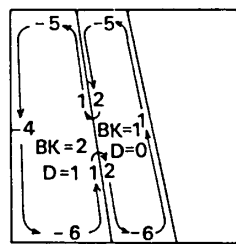


図14 合成処理(外枠面 -1)

表14 面名 -1 の外枠表

| Outer Face Name | $-I$ | -1 | |
|--------------------|------------|------------|------------|
| Number of Fragment | $BC(I)$ | 2 | |
| Face Address | J | 1 | 2 |
| Face Fragment Name | $BK(I,J)$ | 2 | 2 |
| Number of Sides | $BEC(I,J)$ | 4 | 4 |
| First Side Address | $BEI(I,J)$ | 1 | 5 |
| Side Address | M | 1, 2, 3, 4 | 5, 6, 7, 8 |
| Side Name | $BE(I,M)$ | 1,-5,-4,-6 | 1,-5, 2,-6 |

表15 追加表

| Element Name N=3 | Element Coordinates X(3),Y(3),Z(3) | | | | Number of Faces AFN=6 | | |
|------------------|------------------------------------|-------------|-------------|------------|-----------------------|-------------|-------------|
| Face Address | L | 1 | 2 | 3 | 4 | 5 | 6 |
| Face Name | AF(L) | 2 | 1 | -4 | -1 | -5 | -6 |
| Number of Sides | AEC(L) | 4 | 4 | 4 | 4 | 4 | 4 |
| Side Address | M | 1, 2, 3, 4 | 5, 6, 7, 8 | 9,10,11,12 | 13,14,15,16 | 17,18,19,20 | 21,22,23,24 |
| Side Name | AE(M) | -6,-4,-5, 1 | -6, 2,-5,-1 | 2,-6,-1,-5 | 1,-5,-4,-6 | 2,-4,-1, 1 | 2, 1,-1,-4 |

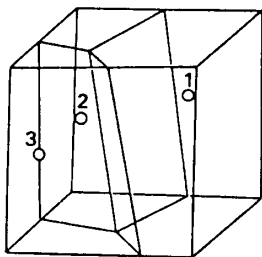


図 15 分割完了図

は-5である。

3・10・3 既設定辺判定 辺名が負で、しかも直前に追加表中に外周辺名として書き込まれているときは3・10・2項に戻る。

実例の場合、辺名-5はまだ書き込まれていない。

3・10・4 外周条件判定 辺名が $BK(I, J)$ と等しいかまたは負であるという外周条件を満足するかどうかを判定する。満足しないときは、辺は接合辺だから3・10・7項に行く。

実例の場合、辺名-5は外周条件を満足する。

3・10・5 先頭外周辺一致判定 外周条件を満足した辺名が先頭外周辺名と一致したとき合成処理は完結する。

実例の場合、辺名-5は先頭外周辺名と一致しない。

3・10・6 外周辺名書込み 追加要素の外枠面名項の辺番地を1だけ増して辺名を書き込み3・10・2項に戻る。

実例の場合、辺名-5を次の辺名として表15辺番地14に書き込む。この後、3・10・2項に戻り隣の辺名2が取り出され、3・10・3項と3・10・4項の判定がされると、この辺名は外周条件を満足しないから、処理中の外枠断片面と接合する相手の外枠断片面を探すための接合面名となる。

3・10・7 接合面と接合辺探索 まず、外枠断片面名 $BK(I, J)$ が接合面名に一致する面名項を外枠表の中からみつける。次に、この面名項の辺名の中からその辺名が接合面名(3・10・2~3・10・6項で処理した外枠断片面名)と一致するものを見つめる。

実例の場合、 $BK(I, J)$ が接合面名2に等しい面名項は表14面番地1の外枠断片面である。そして、この面名項の辺名の中から接合辺名1が辺番地1の所でみつける。

図13の流れ図にしたがって、この後3・10・2項以降から3・10・5項の判定条件による停止までの処理が

実行されて(図14)、合成処理が完了すると表15の面番地4の項目が得られる。

-5, -6の外枠面についても3・10節の合成処理を行い、追加要素に集積記憶し、追加要素3の追加表が表15のように完成する。また、分割領域の様子は図15のようになる。

3・11 追加要素の登録 この完成した追加表を、要素表の最後に付け加える。このとき、ランダムに要素点位置が与えられたときには、著者ら⁽⁹⁾の経験によれば、ポロノイ多面体の平均面数は15で、一つの面の平均辺数は5であることを考慮して面名と辺名の記憶変数配列の後ろに余分の記憶容量を確保しておくことと処理効率が増大する。

4. あとがき

この自動分割法は、要素を一つずつ追加する手法なので、すでに与えた要素点を取り除いたり位置を変更するとき処理効率が悪いとか、要素自身の形状をあらかじめ指定できないなどの軽微な欠点はあるが、次に述べるような長所と応用性がある。

(1) 分割領域に任意位置の要素点を追加するだけで、三次要素(ポロノイ多面体要素)の自動分割ができるから、分割する物体の形や分割要素の粗密を全く自由に選択できる。

(2) 自動分割で得られたポロノイ多面体要素表は簡潔であるから、記憶容量の小さな計算機でもこの自動分割法が使える。

(3) ポロノイ多面体要素の特徴から、いろいろな偏微分方程式を高精度かつ高速度な並列処理可能な連立代数方程式に容易に変換できる。

(4) 分割して得られた要素表から、通常の四面体要素への分割変更が簡単に行える。

(5) ポロノイ多面体要素表は偏微分方程式の数値計算手段そのもの自身としても使える応用性がある。

文 献

- (1) 尾田・山崎, 機械の研究, 37-6(1985), 704, 養賢堂.
- (2) Green, P. J. and Sibson, R., *Comput. J.*, 21-2(1978), 168.
- (3) 例えば, Ohya, T., ほか2名, *J. Oper. Res. Soc. Jpn.*, 27-4(1984), 306.
- (4) Bowyer, A., *Comput. J.*, 24-2(1981), 162.
- (5) Watson, D. F., *Comput. J.*, 24-2(1981), 167.
- (6) 小林・ほか2名, 第25回伝熱シンポジウム講演論文集, 1(1988), 328.
- (7) 谷口・ほか2名, 機論, 55-513, B(1989), 1324.
- (8) 科学大辞典, (1985), 1325, 丸善.
- (9) 小林・ほか3名, 機論, 57-537, B(1991), 1795.